ICINCO 2010

7th International Conference on
Informatics in Control, Automation and Robotics

Proceedings

Volume 3

Funchal, Madeira - Portugal · 15 - 18 June, 2010

Sponsored by:

Co-sponsored by:

In Cooperation with:

ICINCO 2010

Proceedings of the
7th International Conference on
Informatics in Control, Automation and Robotics

Volume 3

Funchal, Madeira, Portugal

June 15 - 18, 2010

Co-Sponsored by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication
IFAC – International Federation of Automatic Control

In Cooperation with
AAAI – Association for the Advancement of Artificial Intelligence
WfMC – Workflow Management Coalition
APCA – Associação Portuguesa de Controlo Automático
ACM SIGART – Association for Computing Machinery / Special Interest Group on Artificial Intelligence

Edited by Joaquim Filipe, Juan Andrade Cetto and Jean-Louis Ferrier

Printed in Portugal
ISBN: 978-989-8425-02-7
Depósito Legal: 311150/10

http://www.icinco.org
icinco.secretariat@insticc.org

BRIEF CONTENTS

INVITED SPEAKERS

José Santos-Victor

Instituto Superior Técnico

Portugal

Alicia Casals

Institute for Bioengineering of Catalonia.IBEC and Universitat Politècnica de Catalunya.UPC

Spain

Bradley Nelson

Robotics and Intelligent Systems at ETH-Zürich

Switzerland

Wissama Khalil

Ecole Centrale de Nantes, IRCCyN

France

Oleg Gusikhin

Ford Research & Adv. Engineering

U.S.A.

John Hollerbach

University of Utah

U.S.A.

ORGANIZING AND STEERING COMMITTEES

CONFERENCE CHAIR

Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

PROGRAM CO-CHAIRS

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

Jean-Louis Ferrier, University of Angers, France

PROCEEDINGS PRODUCTION

Patrícia Alves, INSTICC, Portugal

Helder Coelhas, INSTICC, Portugal

Vera Coelho, INSTICC, Portugal

Andreia Costa, INSTICC, Portugal

Patricia Duarte, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Mauro Graça, INSTICC, Portugal

Raquel Martins, INSTICC, Portugal

Liliana Medina, INSTICC, Portugal

Carla Mota, INSTICC, Portugal

Vitor Pedrosa, INSTICC, Portugal

Filipa Rosa, INSTICC, Portugal

José Varela, INSTICC, Portugal

CD-ROM PRODUCTION

Elton Mendes, INSTICC, Portugal

Pedro Varela, INSTICC, Portugal

GRAPHICS PRODUCTION AND WEBDESIGNER

Daniel Pereira, INSTICC, Portugal

SECRETARIAT

Marina Carvalho, INSTICC, Portugal

WEBMASTER

Sérgio Brissos, INSTICC, Portugal

PROGRAM COMMITTEE

Arvin Agah, The University of Kansas, U.S.A.

Alexande Poznyak, CINVESTAV-IPN, Mexico

Andrew Adamatzky, University of the West of England, U.K.

Eugenio Aguirre, University of Granada, Spain

Hesham Alfares, King Fahd University of Petroleum and Minerals, Saudi Arabia

Adel Al-Jumaily, University of Technology, Sydney, Australia

Francesco Amigoni, Politecnico di Milano, Italy

Peter Arato, Budapest University of Technology and Economics, Hungary

Alejandro Hernandez Arieta, University of Zurich, Switzerland

Marco Antonio Arteaga, Universidad Nacional Autonoma de Mexico, Mexico

Vijanth Sagayan Asirvadam, Universiti Teknologi PETRONAS, Malaysia

T. Asokan, Indian Institute of Technology Madras, India

Ruth Bars, Budapest University of Technology and Economics, Hungary

Adil Baykasoglu, University of Gaziantep, Turkey

Karsten Berns, University of Kaiserslautern, Germany

Mauro Birattari, IRIDIA-CoDE, Université Libre de Bruxelles, Belgium

Christian Blum, Universitat Politècnica de Catalunya, Spain

Patrick Boucher, Supélec, France

Bernard Brogliato, INRIA, France

Kevin Burn, University of Sunderland, U.K.

Clifford Burrows, Innovative Manufacturing Research Centre, U.K.

Dídac Busquets, Universitat de Girona, Spain

Javier Fernandez de Canete, University of Malaga, Spain

Giuseppe Carbone, LARM - Laboratorio di Robotica e Meccatronica, Italy

J. L. Martins de Carvalho, Instituto de Sistemas e Robótica - Porto, Portugal

Alessandro Casavola, University of Calabria, Italy

Riccardo Cassinis, University of Brescia, Italy

Ratchatin Chanchareon, Chulalongkorn University, Thailand

Antonio Chella, Università di Palermo, Italy

Wen-Hua Chen, Loughborough University, U.K.

Graziano Chesi, University of Hong Kong, China

Sung-Bae Cho, Yonsei University, Korea, Republic of

Ryszard S. Choras, University of Technology & Life Sciences, Poland

Carlos Coello Coello, CINVESTAV-IPN, Mexico

António Dourado Correia, University of Coimbra, Portugal

José Boaventura Cunha, University of Trás-os-montes and Alto Douro, Portugal

Mingcong Deng, Okayama University, Japan

Guilherme DeSouza, University of Missouri, U.S.A.

Denis Dochain, Université Catholique de Louvain, Belgium

Tony Dodd, The University of Sheffield, U.K.

Venky Dubey, Bournemouth University, U.K.

Frederick Ducatelle, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Switzerland

Ashish Dutta, Indian Institute of Technology Kanpur, India

Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil

Atilla Elci, Middle East Technical University, Turkey

Ali Eydgahi, University of Maryland Eastern Shore, U.S.A.

Jean-marc Faure, Ecole Normale Superieure de Cachan, France

Paolo Fiorini, Università degli Studi di Verona, Italy

PROGRAM COMMITTEE (CONT.)

Georg Frey, Saarland University, Germany

Wai-Keung Fung, University of Manitoba, Canada

Dragan Gamberger, Rudjer Boskovic Institute, Croatia

Andrea Garulli, Universita' di Siena, Italy

Ryszard Gessing, Silesian University of Technology, Poland

Lazea Gheorghe, Technical University of Cluj-Napoca, Romania

Paulo Gil, Universidade Nova de Lisboa, Portugal

Alessandro Giua, University of Cagliari, Italy

Luis Gomes, Universidade Nova de Lisboa, Portugal

Dongbing Gu, University of Essex, U.K.

Kevin Guelton, University of Reims Champagne-Ardenne, France

Maki K. Habib, The American University in Cairo, Egypt

Wolfgang Halang, Fernuniversitaet, Germany

Onur Hamsici, Qualcomm, U.S.A.

John Harris, University of Florida, U.S.A.

Inman Harvey, University of Sussex, U.K.

Dominik Henrich, University of Bayreuth, Germany

Suranga Hettiarachchi, Indiana University Southeast, U.S.A.

Victor Hinostroza, University of Ciudad Juarez, Mexico

Wladyslaw Homenda, Warsaw University of Technology, Poland

Guoqiang Hu, Kansas State University, U.S.A.

Joris Hulstijn, Vrije Universiteit, Amsterdam, The Netherlands

Fumiya Iida, Robot Locomotion Group, U.S.A.

Atsushi Imiya, IMIT Chiba University, Japan

Giovanni Indiveri, University of Salento, Italy

Mirjana Ivanovic, Faculty of Science, University of Novi Sad, Serbia

Sarangapani Jagannathan, Missouri University of Science and Technology, U.S.A.

Masoud Jamei, Simcyp Ltd, U.K.

Ping Jiang, The University of Bradford, U.K.

Graham Kendall, University of Nottingham, U.K.

DaeEun Kim, Yonsei University, Korea, Republic of

Won-jong Kim, Texas A&M University, U.S.A.

Israel Koren, University of Massachusetts, U.S.A.

Gerhard K. Kraetzschmar, Bonn-Rhein-Sieg University of Applied Sciences, Germany

Mianowski Krzysztof, Politechnika Warszawska, Poland

H. K. Lam, King's College London, U.K.

Alexander Lanzon, University of Manchester, U.K.

Kathryn J. De Laurentis, University of South Florida, U.S.A.

Kauko Leiviskä, University of Oulu, Finland

Zongli Lin, University of Virginia, U.S.A.

Guoping Liu, University of Glamorgan, U.K.

Jing-Sin Liu, Institute of Information Science, Academis Sinica, Taiwan

José Tenreiro Machado, Institute of Engineering of Porto, Portugal

Anthony Maciejewski, Colorado State University, U.S.A.

Frederic Maire, Queensland University of Technology, Australia

Om Malik, University of Calgary, Canada

Hervé Marchand, INRIA, France

Philippe Martinet, Lasmea, France

Rene V. Mayorga, University of Regina, Canada

Seán McLoone, National University of Ireland (NUI) Maynooth, Ireland

PROGRAM COMMITTEE (CONT.)

Prashant Mehta, University of Illinois at Urbana-Champaign, U.S.A.

Carlo Menon, Simon Fraser University, Canada

António Paulo Moreira, INESC Porto / FEUP, Portugal

Vladimir Mostyn, VSB - Technical University of Ostrava, Czech Republic

Rafael Muñoz-salinas, University of Cordoba, Spain

Kenneth Muske, Villanova University, U.S.A.

Andreas Nearchou, University of Patras, Greece

Luciana Nedel, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil

Sergiu Nedevschi, Technical University of Cluj-Napoca, Romania

Monica N. Nicolescu, University of Nevada, Reno, U.S.A.

Klas Nilsson, Lund University, Sweden

Urbano Nunes, University of Coimbra, Portugal

Manuel Ortigueira, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Portugal

Selahattin Ozcelik, Texas A&M University-Kingsville, U.S.A.

Igor Paromtchik, INRIA, France

Mario Pavone, University of Catania, Italy

Claudio de Persis, Sapienza University of Rome, Italy

D. T. Pham, Cardiff University, U.K.

Michael Piovoso, The Pennsylvania State University, U.S.A.

Raul Marin Prades, Jaume I University, Spain

José Ragot, Centre de Recherche en Automatique de Nancy, France

Jerzy Respondek, Silesian University of Technology, Poland

A. Fernando Ribeiro, Universidade do Minho, Portugal

Robert Richardson, University of Leeds, U.K.

Mihailo Ristic, Imperial College London, U.K.

Juha Röning, University of Oulu, Finland

Agostinho Rosa, IST, Portugal

Danilo De Rossi, University of Pisa, Italy

Fariba Sadri, Imperial College London, U.K.

Mehmet Sahinkaya, University of Bath, U.K.

Priti Srinivas Sajja, Sardar Patel University, India

Abdel-badeeh Salem, Ain Shams University, Egypt

Marcello Sanguineti, University of Genova, Italy

Jurek Sasiadek, Carleton University, Canada

Sergio M. Savaresi, Politecnico di Milano, Italy

Carla Seatzu, University of Cagliari, Italy

Michael Short, Teesside University, U.K.

Silvio Simani, University of Ferrara, Italy

Dan Simon, Cleveland State University, U.S.A.

Olivier Simonin, INRIA - LORIA, France

Joaquin Sitte, Queensland University of Technology, Australia

Adam Slowik, Koszalin University of Technology, Poland

Andrzej Sluzek, Nanyang Technological University/Nicolaus Copernicus University, Singapore

Safeeullah Soomro, Yanbu University College, Saudi Arabia

Stefano Squartini, Polytechnic University of Marche, Italy

Burkhard Stadlmann, University of Applied Sciences Wels, Austria

Chun-Yi Su, Concordia University, Canada

Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland

Kazuya Takeda, Nagoya University, Japan

PROGRAM COMMITTEE (CONT.)

Daniel Thalmann, VR Lab EPFL, Switzerland

N. G. Tsagarakis, Istituto Italiano di Tecnologia, Italy

Avgoustos Tsinakos, University of Kavala Institute of Technology, Greece

Antonios Tsourdos, Cranfield University (Cranfield Defence and Security), U.K.

Angel Valera, Universidad Politécnica de Valencia, Spain

Eloisa Vargiu, University of Cagliari, Italy

Annamaria R. Varkonyi-koczy, Obuda University, Hungary

Ramiro Velazquez, Universidad Panamericana, Mexico

Damir Vrancic, Jožef Stefan Institute, Slovenia

Bernardo Wagner, Leibniz Universität Hannover, Germany

Dianhui Wang, La Trobe University, Australia

James Whidborne, Cranfield University, U.K.

Sangchul Won, Pohang University of Science and Technology, Korea, Republic of

Qishi Wu, University of Memphis, U.S.A.

Marek Zaremba, Université du Québec (UQO), Canada

Janan Zaytoon, University of Reims Champagne Ardennes, France

Primo Zingaretti, Università Politecnica delle Marche, Italy

AUXILIARY REVIEWERS

Lounis Adouane, LASMEA, UMR CNRS 6602, France

Iman Awaad, Bonn-Rhein-Sieg University, Germany

Yao Chen, Institute of Systems Sciences, Chinese Academy of Sciences, China

Rohit Chintala, Texas A and M University, India

Jonathan Courbon, LASMEA, France

Kun Deng, University of Illinois at Urbana Champaign, U.S.A.

Anibal Ferreira, University of Porto, Portugal

Leonardo Fischer, Universidade Federal do Rio Grande do Sul - UFRGS, Brazil

Fernando Fontes, Faculty of Engineering, University of Porto, Portugal

Ronny Hartanto, Bonn-Rhein-Sieg University of Applied Sciences, Germany

Laurent Harter, -, France

Nico Hochgeschwender, ESG GmbH, Germany

Ana Lopes, Institute of Systems and Robotics - University of Coimbra, Portugal

Jan Paulus, Bonn-Rhein-Sieg University oAS, Germany

Cristiano Presmebida, Institute of System and Robotics, Portugal

Michael Reckhaus, Bonn-Rhein-Sieg University, Germany

Lynda Seddiki, Paris 8 University, France

Renato Silveira, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil

Yu Sun, University of Illinois, Urbana-Champaign, U.S.A.

C. Renato Vázquez, Universidad de Zaragoza, Spain

Huibing Yin, University of Illinois at Urbana-champaign, U.S.A.

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2010 will be published by Springer-Verlag in a LNEE Series book. This selection will be done by the Conference Chair and Program Co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICINCO 2010 Program Committee members.

FOREWORD

This book contains the proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010) which was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) and held in Funchal, Madeira - Portugal. ICINCO 2010 was co-sponsored by the International Federation for Automatic Control (IFAC) and held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI), the Workflow Management Coalition (WfMC), the Portuguese Association for Automatic Control (APCA) and the Association for Computing Machinery (ACM SIGART).

The ICINCO Conference Series has now consolidated as a major forum to debate technical and scientific advances presented by researchers and developers both from academia and industry, working in areas related to Control, Automation and Robotics that benefit from Information Technology.

In the Conference Program we have included oral presentations (full papers and short papers) and posters, organized in three simultaneous tracks: "Intelligent Control Systems and Optimization", "Robotics and Automation" and "Systems Modeling, Signal Processing and Control". We have included in the program six plenary keynote lectures, given by internationally recognized researchers, namely - José Santos-Victor (Instituto Superior Técnico, Portugal), Alícia Casals (Institute for Bioengineering of Catalonia.IBEC and Universitat Politècnica de Catalunya.UPC, Spain), Bradley Nelson (Institute of Robotics and Intelligent Systems at ETH-Zürich, Switzerland), Wisama Khalil (Ecole Centrale de Nantes, IRCCyN, France), Oleg Gusikhin (Ford Research & Adv. Engineering, U.S.A.) and John Hollerbach (University of Utah, U.S.A.).

The meeting is complemented with one satellite workshop, the International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP), and one Special Session on Intelligent Vehicle Controls & Intelligent Transportation Systems (IVC & ITS).

ICINCO received 320 paper submissions, not including those of the workshop or the special session, from 57 countries, in all continents. To evaluate each submission, a double blind paper review was performed by the Program Committee. Finally, only 142 papers are published in these proceedings and presented at the conference. Of these, 94 papers were selected for oral presentation (27 full papers and 67 short papers) and 48 papers were selected for poster presentation. The full paper acceptance ratio was 8%, and the oral acceptance ratio (including full papers and short papers) was 29%. As in previous editions of the Conference, based on the reviewer's evaluations and the presentations, a short list of authors will be invited to submit extended versions of their papers for a book that will be published by Springer with the best papers of ICINCO 2010.

Conferences are also meeting places where collaboration projects can emerge from social

contacts amongst the participants. Therefore, in order to promote the development of research and professional networks the Conference includes in its social program a Conference and Workshop Social Event & Banquet in the evening of June 17 (Thursday).

We would like to express our thanks to all participants. First of all to the authors, whose quality work is the essence of this Conference. Next, to all the members of the Program Committee and auxiliary reviewers, who helped us with their expertise and valuable time. We would also like to deeply thank the invited speakers for their excellent contribution in sharing their knowledge and vision. Finally, a word of appreciation for the hard work of the INSTICC team; organizing a conference of this level is a task that can only be achieved by the collaborative effort of a dedicated and highly capable team.

Commitment to high quality standards is a major aspect of ICINCO that we will strive to maintain and reinforce next year, including the quality of the keynote lectures, of the workshops, of the papers, of the organization and other aspects of the conference. We look forward to seeing more results of R&D work in Informatics, Control, Automation and Robotics at ICINCO 2011.

Joaquim Filipe
Polytechnic Institute of Setúbal / INSTICC, Portugal

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

Jean-Louis Ferrier
University of Angers, France

CONTENTS

SHORT PAPERS

POSTERS

INVITED SPEAKERS

KEYNOTE SPEAKERS

BIOINSPIRED ROBOTICS AND VISION WITH HUMANOID ROBOTS

José Santos-Victor
Instituto Superior Técnico, Lisboa, Portugal

Abstract: In this talk, I will describe recent results on exploring recent results from neurophysiology and developmental psychology for the design of humanoid robot technologies. The outcome of this research is twofold: (i) using biology as an inspiration for more flexible and sophisticated robotic technologies and (ii) contribute to the understanding of human cognition by developing biologically plausible (embodied) models and systems.

One application area is the domain of video surveillance and human activity recognition. We will see how recent findings in neurophysiology (the discovery of the mirror neurons) suggest that both action understanding and execution are performed by the same brain circuitry. This might explain how humans can so easily (apparently) understand the actions of other individuals, which constitutes the building block of non-verbal communication first and then, language acquisition and social learning.

The second aspect to be addressed is the use of development as a methodological approach for building complex humanoid robots. This line of research is inspired after the human cognitive and motor development, a pathway that allows newborns to progressively acquire new skills and develop new learning strategies. In engineering terms, this may be a way not only to structure the sensed data but also to master the complexity of the interaction with the physical world with a sophisticated body (sensing and actuation).

During the talk, I will provide examples with several humanoid platforms used for this research: Baltazar is a humanoid torso we developed to study sensorimotor coordination and cognition; the latest results are implemented in the iCub humanoid robot, for which we designed the head, face and body covers as well as the attention and affordance learning system.

BRIEF BIOGRAPHY

José Santos-Victor received the PhD degree in Electrical and Computer Engineering in 1995 from Instituto Superior Técnico (IST - Lisbon, Portugal), in the area of Computer Vision and Robotics. He is an Associate Professor with "Aggregation" at the Department of Electrical and Computer Engineering of IST and a researcher of the Institute of Systems and Robotics (ISR) and heads the Computer and Robot Vision Lab - VisLab.

He is the scientific responsible for the participation of IST/ISR in various European and National research projects in the areas of Computer Vision and Robotics. His research interests are in the areas of Computer and Robot Vision, particularly in the relationship between visual perception and the control of action, biologically inspired vision and robotics, cognitive vision and visual controlled (land, air and underwater) mobile robots.

Prof. Santos-Victor was an Associated Editor of the IEEE Transactions on Robotics and the Journal of Robotics and Autonomous Systems.

HUMAN
Robot Cooperation Techniques in Surgery

Alícia Casals
Institute for Bioengineering of Catalonia (IBEC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
alicia.casals@upc.edu

Keywords: Medical Robotics, Human Robot Interaction, Human Machine Interfaces, Surgical Robots.

Abstract: The growth of robotics in the surgical field is consequence of the progress in all its related areas, as: perception, instrumentation, actuators, materials, computers, and so. However, the lack of intelligence of current robots makes teleoperation an essential means for robotizing the Operating Room (OR), helping in the improvement of surgical procedures and making the best of the human-robot couple, as it already happens in other robotic application fields. The assistance a teleoperated system can provide is the result of the control strategies that can combine the high performance of computers with the surgeon knowledge, expertise and will. In this lecture, an overview of teleoperation techniques and operating modes suitable in the OR is presented, considering different cooperation levels. A special emphasis will be put on the selection of the most adequate interfaces currently available, able to operate in such quite special environments.

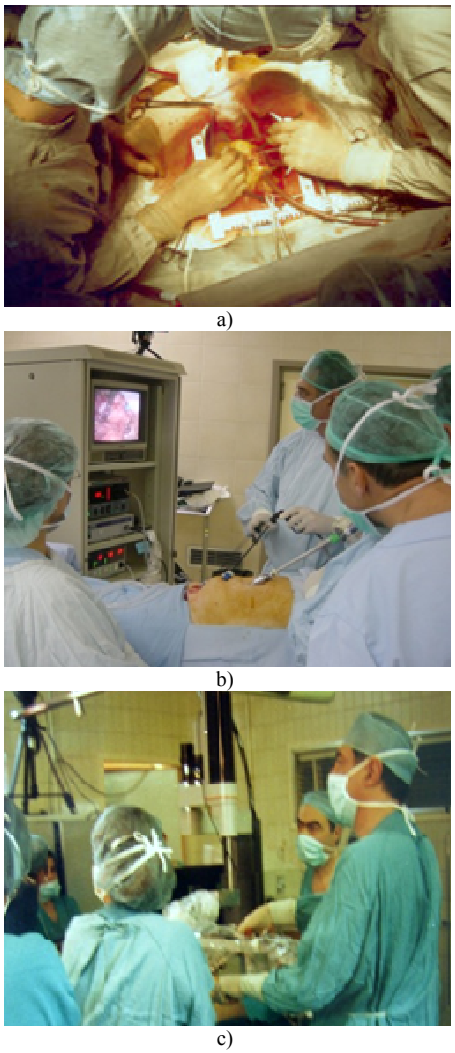
1 INTRODUCTION

Technological evolution has continuously been introducing new equipments and changes in the Operating Room. Technology does not only affect real surgical interventions, but it also has a bear on all diagnosis and planning strategies, according to the diagnosed pathology. The history of surgery has suffered a continuous evolution through which three significant phases can be identified. They can be summarized in fig.1. From the practice of open surgery, fig. 1 a), in which surgeons get in touch directly with the patient organs or corresponding body parts, as purely manual actuation, the advent of new instruments and visualization techniques opened the era of minimally invasive surgery. Instruments with long handles allow entering the body through natural holes and small incisions over the patient fig. 1.b). This image shows the common scenario in laparoscopic surgery. At this stage, surgeons rely on instruments and specific equipment to perform surgery. The era of surgical robotics emerges as these instruments acquire new performances, or others appear in the scene, fig. 1.c).

New surgical procedures, not conceivable several decades ago, are more complex and require much higher performances. To face the challenges this kind of surgery relies on robots cooperating with humans, so as to extract the best of both of them. Humans provide intelligence and decision making while robots contribute with their precision, computing capabilities and no tiredness. In this context, with human and machines sharing the working scene, and the task itself, more powerful interfaces and interaction means become necessary.

Both, cooperation and interface requirements will depend on the typology of surgery. Speaking in terms of robotics technology and considering that technological needs vary enormously with the kind of surgery, surgery can be classified in: microsurgery, neurosurgery, intracavity and orthopedic, and percutaneous and transcutaneous interventions. As significant distinction among them, some characteristics, as the kind of tissue (hard or soft) or the body parts undergoing surgery, are to be considered.

Since hard tissues are able to maintain its shape, when they can be immobilized some techniques applied in industry can be exported to surgery, otherwise, a tracking system to dynamically determine the changing reference frames is required. Soft tissues present the problem of deformability, making robot operation more complex. In this case, teleoperation is an alternative solution.



a)

b)

c)

Figure 1: Evolution of surgical procedures. a) open surgery, b) minimally invasive surgery, c) robot assisted surgery.

2 HUMAN ROBOT COOPERATION MODES

Human robot cooperation is implemented by means of teleoperation, therefore, a master device in the surgeon side controls a slave arm, a teleoperated robot, patient side. In between, a computer implements the required assistive functions that enhance human capabilities, resulting in a "super surgeon". Such assistive functions can be a change of scale, defining constrains within the working space, tremor reduction and movement compensation (breathing or heart beating) and so. The surgeon can be located in a close position, or in any other location, a few meters or some kilometers away. Fig. 2 shows a schema of such system.

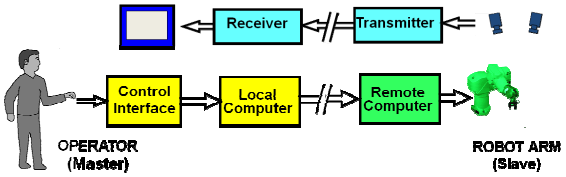


Figure 2: Schema of a teleoperation system.

However, the surgeon could also be in close contact with the robot, which guides and supervises his or her actions. In this cooperative mode, comanipulation, the surgeon hand and the robot end effector move simultaneously holding the surgical instrument. Working in these conditions, a change of scale or movement compensation is not an issue, but assisted teleoperation allows establishing constraints, virtual fixtures, operating over reference frames either fixed or floating over the patient anatomy. Comanipulation also allows directly perceiving both, images and the operating environment, what is especially useful in orthopedic surgery. The definition of virtual fixtures during the surgical planning facilitates a safer operation reducing the surgeon stress in critical interventions. Working with this configuration the robot itself behaves as a *haptic* device, Fig. 3.

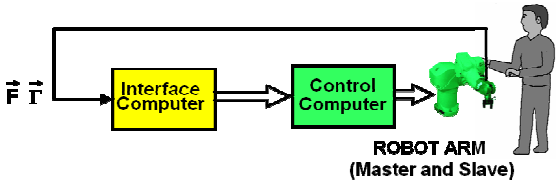


Figure 3: Schema of a comanipulation system.

The level of cooperation can also vary with the typology of the surgical interventions since the level of preplanning and programming varies depending on the predictability of the intervention. Three levels can be considered: manual guidance, supervised guidance and autonomous control.

The role of the interface in such cooperation systems is crucial as the surgeon cannot pay much attention to the robotic system, but to the patient and the own surgical procedure. A schema of the characteristics an interface should provide is shown in fig. 4.

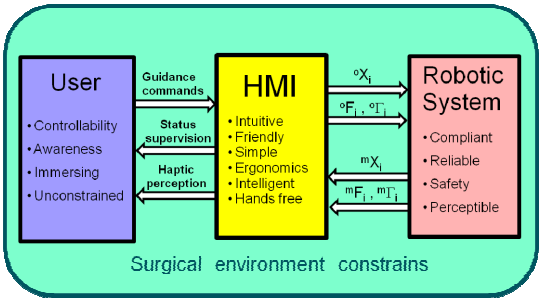


Figure 4: Schema of the characteristics of a teleoperation interface.

3 ROBOT ARCHITECTURES

Different surgical specialties work along different scale ranges and with different working requirements. Considering the variety of working conditions and from the robotics point of view, surgical procedures can be classified as follows: microsurgery, neurosurgery, transcutaneous, percutaneous, intracavities interventions and orthopedics. In what follows the main characteristics of each of them are described.

3.1 Microsurgery

Most interventions requiring microsurgery performances are related to sewing nerves in transplants or to ophthalmology. In this specialty, one or more high precision and high accessibility 6 Degrees of Freedom (DoF) arms are necessary. Additional DoF might be required to increase accessibility. As teleoperation assistance functions, a change of scale between the master and the slave increases the achievable precision as the application requires. Compared to former manipulators or holding devices used in classical manual practice, robots substitute them with advantage

3.2 Neurosurgery

Interventions in the skull also require high precision. However, its accessibility requirements are no so demanding since insertions are applied through incisions done just over the target area. In this case, movement compensation or floating reference frames are not needed as the skull can be fixed, with stereotaxis devices.

3.3 Transcutaneous

Some interventions can be performed through the skin and soft tissues using radiation focused over the target area. This radiation can be of different types: RX, Gama, or High Intensity focused US (HIFU). In such minimally invasive technique, 5DoF are enough to focus the therapeutic beam over a point in a 3D space, with any orientation.

3.4 Percutaneous

Relatively simple interventions as biopsy, aspiration, ablations or releasing therapeutic payload, are more and more used due to the few invasiveness of the technique. This technique implies inserting needles with high precision, to produce the advance and drilling movements when the needle is manually oriented into the insertion point. 5 DoF are necessary to place and orient the needle with a robot.

Taking advantage of this minimally invasive surgery or intervention, deflection and guided probes can be used to reach areas not easily reachable. Operating through natural ways, as the arteries, this technique is being used successfully to treat brain aneurisms.

3.5 Intracavity Interventions

When the intervention cannot be carried out by means of needles and more versatile instruments are needed with two, three or four DoF, endoscopic techniques are required. There are two endoscopic techniques, one based on the use of natural orifices (NOTES) and the second based on small incisions to access the abdominal and thoracic cavity (laparoscopy) or the joints between bones (arthroscopy).

These techniques were introduced in the seventies operating the instruments manually. At present, these instruments can be guided by teleoperated robots, so as they can take advantage of assisted teleoperation techniques. These robots should be multiarm (2, 3 or 4), each with several DoF, not only for tool positioning but also to increase accessibility and to make the pose of each arm compatible with the patient in the operating table.

NOTES are used in intra vaginal interventions, ear, laparoscopy for abdominal, prostate, heart, gynecology or arthroscopy, knee specially.

3.6 Orthopedics

Orthopedic surgery uses as end effectors drilling, cutting or milling tools to operate over bone tissues. These techniques are oriented to bone repair either

with prosthesis implants, subjecting or immobilizing boards or to reconstruct bones with grafts after oncologic surgery, for example.

For these procedures CAD/CAM techniques can be used, with similar methods than those used in industry. Reference frames registration between anatomical elements in the operating table and the CAD model previously obtained from CT images are used to make task planning possible.

4 SENSOR REQUIREMENTS

Based on these different scenarios, robotics requires different kind of sensors to be able to implement the required control strategies. Two kinds of sensors are needed: 3D geometrical positioning sensors (navigators) and physical interaction characteristics (force and torque sensors).

Positioning anatomical parts in the 3D space is not simple, especially when dealing with not immobilized rigid elements, soft, deformable or rhythmically moving tissues. The success of surgical robots rely on their capability for adequately sensing positions either using physical contact sensors (optical or magnetic techniques) or remote sensing, specially vision. Current limitations of computer vision strongly condition its advances.

Apart from being able to control the robot, not only geometrical strategies are necessary to generate trajectories, but additional force control techniques are required to avoid injures, as for instance, necrosis. On the other hand, force sensors provide the information required to generate *haptic* information to be feedback to the surgeon. .

5 INTERFACES REQUIREMENTS

The requirements of an interface for surgical applications does not imply uniquely the interpretation of human will to control the teleoperated robot, but also to provide some feedback of the task going on to the surgeon. Thus, an interface constitutes a complete system, fig. 5, consisting of master devices adequate for every specific kind of surgery, actuators to feedback information to the surgeon, monitoring devices, and the computing power to process the information coming both, from the teleoperated system to provide the adequate information and from the human operator to provide the adequate control orders.

The schema of fig.5 shows that an interface can be a complete system that in some environments should provide certain intelligence level and, as indicated in the schema, even generate synthetic information to improve human operator's perception.

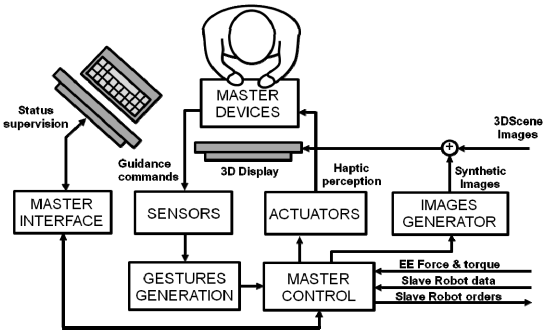


Figure 5: Schema of the master station in a teleoperated system.

Besides considering the best kind of master devices, either classical manual devices or hand free systems, the information coming from the robot side are key to achieve an efficient and smooth interaction. Feedback information can be either visual: direct images, augmented images or virtual, besides other kind of numerical and graphic data; or haptic, based on the force data measured in the robot working environment.

6 CONCLUSIONS

Interfaces are key components in teleoperated systems, or in robotic systems that work in close cooperation with humans. In the field of surgery, the surgeon faces the problem of dealing with complex systems while they are performing their own job, surgery. On the other side, their specialty is far from informatics and mechatronic systems. Thus, the design of a human robot interface in such fields should consider the three parts that compose the working environment, as shown in fig. 4, the user, the interface and the robotic system. While the surgeon has to pay complete attention to the intervention itself, the interface should provide the means of reacting to the humans' will, to interpret their needs, and to supply any kind of information that can help them to take decisions.

Teleoperation and its interface with the human operator can take different configurations according

to the application needs, considering both, the distance from the master to the slave and the typology of the application. In each case, the availability of the required teleoperation assistance functions to improve human and system performances is essential. Together with such assistance, the information fed back to the user, visual, haptic or even sound, can be intelligently processed to constitute a significant help for the whole process.

BRIEF BIOGRAPHY

Alicia Casals is professor at the Technical University of Catalonia (UPC), in the Automatic Control and Computer Engineering Department. She is currently leading the research group on Robotics and Medical imaging Program of the Institute for Biomedical Engineering of Catalonia, and is member of the research group GRINS: Intelligent Robotics and Systems at UPC. The research is oriented to improve human robot interaction through multimodal perception, focused mainly in the area of medical robotics. In this field she is working both in rehabilitation, assistance and surgical applications. Her background is in Electrical and Electronic Engineering and PhD in Computer Vision. From 2001 to 2008 she was the coordinator of the Education and Training key area within Euron, the Network of Excellence: European Robotics Network, and RAS Vice President for Membership in the period 2008-2009. From the developed research projects she won different awards, Award to a social invention (Mundo Electrónico), International Award *Barcelona'92* (Barcelona City_Hall), *Ciutat de Barcelona* Award 1998 (Barcelona City_Hall), and *Narcis Monturiol* Medal from the Catalan Government as recognition of the research trajectory 1999. From 2007 Prof. Casals is member of the *Institut d'Estudis Catalans*, the Academy of Catalonia.

MAKING MICROROBOTS MOVE

Bradley J. Nelson
Institute of Robotics and Intelligent Systems, ETH Zurich, Zurich, Switzerland

Keywords: Nanorobotics, Microrobotics, Nanocoils, Magnetic actuation.

Abstract: Our group has recently demonstrated three distinct types of microrobots of progressively smaller size that are wirelessly powered and controlled by magnetic fields. For larger scale microrobots, from 1mm to 500 μm, we microassemble three dimensional devices that precisely respond to torques and forces generated by magnetic fields and field gradients. In the 500 μm to 200 μm range, we have developed a process for microfabricating robots that harvest magnetic energy from an oscillating field using a resonance technique. At even smaller scales, down to micron dimensions, we have developed microrobots we call Artificial Bacterial Flagella (ABF) that are of a similar size and shape as natural bacterial flagella, and that swim using a similar low Reynolds number helical swimming strategy. ABF are made from a thin-film self-scrolling process. In this paper I describe why we want to do this, how each microrobot works, as well as the benefits of each strategy.

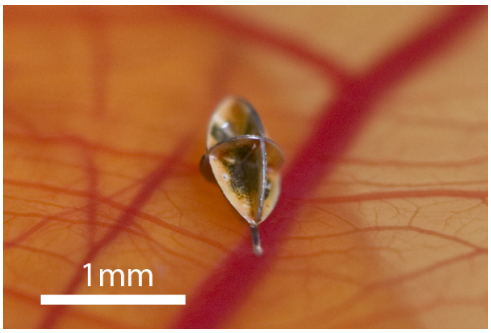
1 INTRODUCTION

Micro and nanorobotics have the potential to dramatically change many aspects of medicine by navigating bodily fluids to perform targeted diagnosis and therapy and by manipulating cells and molecules. In the past few years, we have developed three new approaches to wirelessly controlling microscale structures with high precision over long distances in liquid environments (Figure 1). Because the distance from which these structures can be controlled is relatively large, the structures can not only be used as tools for manipulating other micro and nanoscale structures, similar to particle trapping techniques, but can also serve as vehicles for targeted delivery to locations deep within the human body. The microrobots we have developed are non-spherical. Therefore, both their position and orientation can be precisely controlled, removing another limitation of particle trapping. Unprecedented control in multiple degrees of freedom has been achieved with field strengths as low as 1 mT.

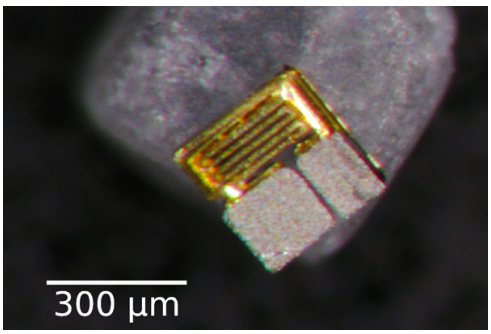
2 MEDICAL MICROROBOTS

Minimally invasive medical techniques are linked with a variety of patient oriented benefits ranging from reduction of recovery time, medical complications, infection risks, and post-operative pain, to lower hospitalization costs, shorter hospital stays, and increased quality of care [1-4]. Microrobotic devices have the potential for improved accessibility compared to current clinical tools, and medical tasks performed by them can even become practically noninvasive. They will perform tasks that are either difficult or impossible with current methods. Rather than acting as autonomous agents that navigate the body diagnosing and solving problems, microrobots will more likely act as new technical tools for clinicians, continuing to capitalize on the clinicians cognitive skill, which is their greatest asset.

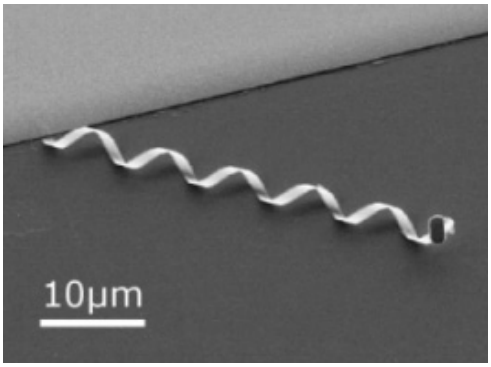
In recent years there has been significant progress on robot-assisted colonoscopy and on wireless miniature robots for use in the GI tract (Kazzim et al., 2006). Motivated by capsule endoscopes that are already in clinical use, a number of technologies have been explored to expand the capabilities of these devices, ranging from wireless GI pressure monitoring systems and lab-on-a-chip devices equipped with pH and temperature sensors (Johannessen et al., 2006) to the addition of legs and other mechanisms for controlled locomotion (Menciassi et al., 2007). The size of these devices approaches a few centimeters, capitalizing on the relatively large size of the GI tract. By further reducing device size and creating microrobots with a



(a) Octomag



(b) Magmite



(c) ABF

Figure 1: Three different types of microrobots at varying orders of magnitude. a) The Octomag robot (shown puncturing a small vein) is controlled using magnetic fields and field gradients. b) The Magmite (sitting on a grain of salt) is powered by oscillating magnetic fields that excite a spring mass system to resonance that harvests the impact energy. c) Artificial Bacterial Flagella (ABF) are propelled through fluid by rotating a magnetic field to generate torques on the magnetic metal head of the device.

maximum dimension of only a few millimeters or less, additional locations in the human body becomavailable for wireless intervention. Natural pathways such as the circulatory system, the urinary system, and the central nervous system become available, enabling intervention with minimal trauma.

As we downscale robots to submillimeter dimensions, the relative importance of physical effects changes (Wautelet, 2001). As device size is reduced, surface effects and fluid viscosity dominate over inertia and other volumetric effects, and power storage becomes a key issue. Furthermore, microrobots, like microorganisms, swim in a low-Reynolds-number regime, requiring swimming methods that differ from macroscale swimmers (Purcell, 1977). This places strong constraints on the development of medical microrobots. In traditional robotics, it is often easy to compartmentalize aspects of robot design such as kinematics, power, and control. In the design of wireless microrobots, fabrication is fundamentally limited by scaling issues, and power and control are often inextricably linked. Engineers must give up intuition gained from observing and designing in the macroscale physical world, and instead rely on analysis and simulation to explore microrobot design. Even then, only experimental results will demonstrate the efficacy of a given microrobot strategy, as the world experienced by the microrobot may be quite difficult to accurately model.

3 OCTOMAG MICROROBOTS

The Octomag microrobot (Yesin et al., 2006), shown in Figure 1, was the first microrobot we developed and is primarily intended for ophthalmic surgery. An external magnetic field acts to align the robot along the long (“easy”) axis, and a field gradient is generated to pull or push the microrobot. The winged shape acts to reduce the side-ways drift of the microrobot by increasing the fluid drag along the axes perpendicular to the long axis (Abbott et al., 2007). The relatively large size of the device is compatible with using gradient fields for propulsion at distances suitable for use within the body and is targeted at controlling the device in a viscous medium.

The robot is a three-dimensional structure built by microassembling individual parts, which allows for the combination of incompatible materials and processes for the integration of MEMS based sensors and actuators. The principle advantage of the hybrid design is that the individual parts of the assembly can be produced with standard MEMS manufacturing processes that create planar geometries. In this way, different subsystems of the robot can be manufactured using the most suitable process for the purpose. Robot parts have been made with electroplated nickel, single crystal silicon, polymer, and laser cut steel.

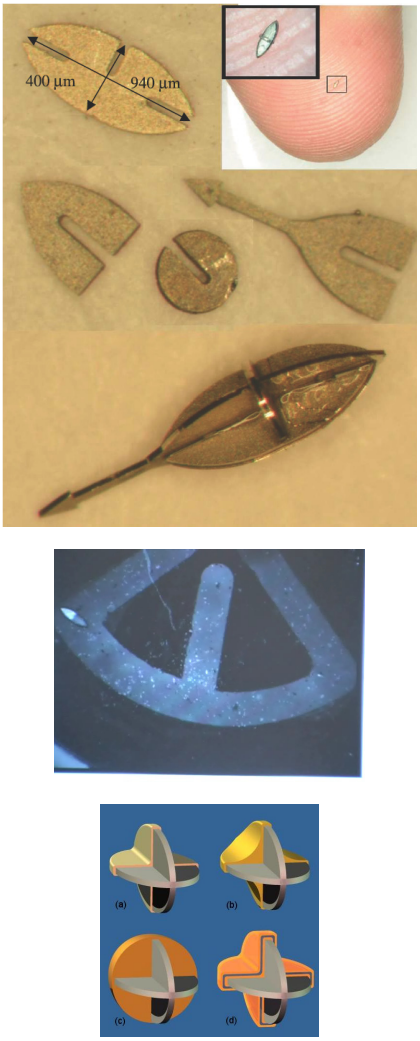


Figure 2: (a) Octomag microrobots are microassembled from 50µm thick electroplated nickel. (b) The sub-mm devices can be precisely controlled to navigate through tiny mazes, and (c) can be coated to ensure biocompatibility and for transport of cargo, such as drugs to be unloaded by diffusion (Yesin et al., 2006).

4 MAGMITES

With decreasing size, gradient propulsion becomes infeasible due to the force generated being related to the magnetic medium's volume, which decreases rapidly with size. To overcome this limitation, we have developed a second propulsion mechanism that harnesses the interactive forces between small magnetic bodies in a uniform magnetic field to drive a spring mechanism to resonance(Vollmers et al., 2008). This energy is then rectified to move the robot through its environment.

The resonant nature of the actuator enables the device to move with fields below 2 mT which is roughly 50x that of the Earth's magnetic field. This locomotion mechanism has been demonstrated on both structured and unstructured surfaces and is controllable enough to repeatedly and precisely follow trajectories. The frequency selectivity of the spring mass resonating structure allows multiple robotic agents to be used on the same substrate to perform tasks. Although this propulsion method was initially designed for operation in air, it has demonstrated its ability to perform in aqueous environments and manipulate glass microspheres on the order of 50 µm.

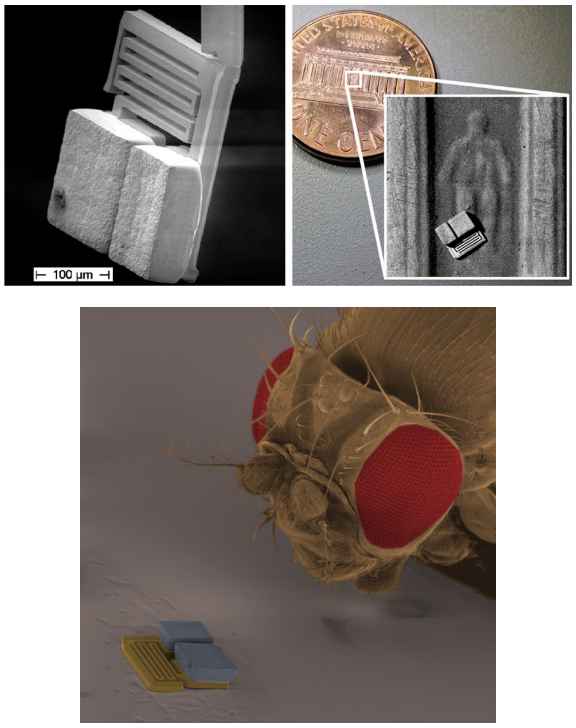


Figure 3: Magmite robots consist of two Ni masses separated by a gold spring. The robot shown measures 300µm square, 70µm thick, and is dwarfed by Drosophila melanogaster (Vollmers et al., 2008).

5 ARTIFICIAL BACTERIAL FLAGELLA

As sizes decrease further, the resonant frequencies of the mechanical structures required for the resonant magnetic actuator increase to tens of kHz and become difficult to generate at sufficient strength. At this scale, torque on the magnetic bodies becomes one of the predominant forces that can be generated.

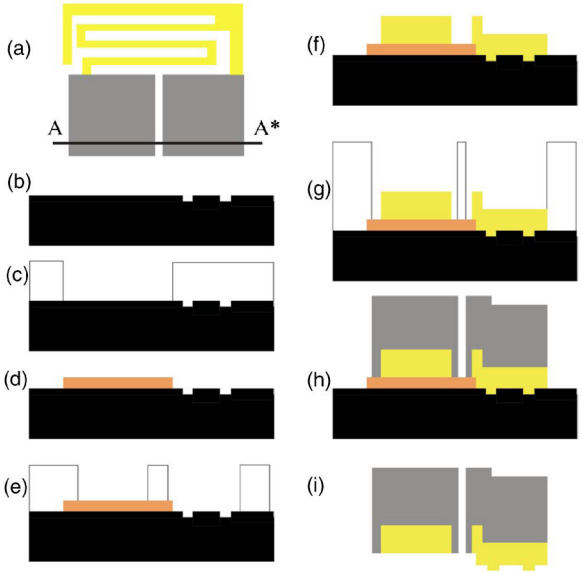


Figure 4: Fabrication sequence with cross section shown along line A-A* (a). Holes for dimples (b) are etched in a wafer before a Ti/Cu adhesion/seed layer is evaporated onto the surface. Photoresist is applied (c) to define thick electroplated copper islands (d). The springs and frame are defined (e) and plated (f) before a final layer of photoresist (g) defines the nickel bodies (h). The device is released from the wafer by etching the sacrificial copper layer (i).

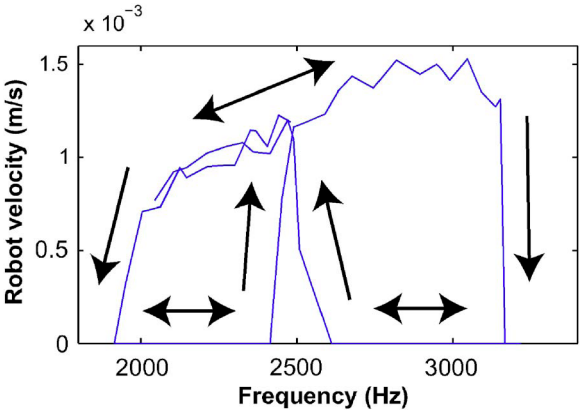


Figure 5: Robot velocity as a function of frequency with a driving field of 2.2 mT. When moving, the robot never comes to a complete rest to allow static friction can take effect. Driving with a frequency too far from resonance reduces the system energy and at some point the robot sticks to the substrate. Moving back toward resonance increases the absorbed energy, allowing the robot to begin moving again.

Taking inspiration from nature, this torque can be leveraged to create artificial bacterial flagella (Zhang et al., 2009).

The helical swimming robot consists of two parts: a helical tail and a magnetic metal head. The tails are 27 to 42 nm thick, less than 2 µm wide, and coil into diameters smaller than 3 µm. The robots are fabricated by a self-scrolling technique (Zhang et al., 2006). The helical swimming microrobots are propelled and steered precisely in water by a rotating magnetic field on the order of 1 to 2 mT. As the robot's principle dimensions approach those of individual cells, many of the experimental methods used with the robots parallel those used by their biological counterparts.

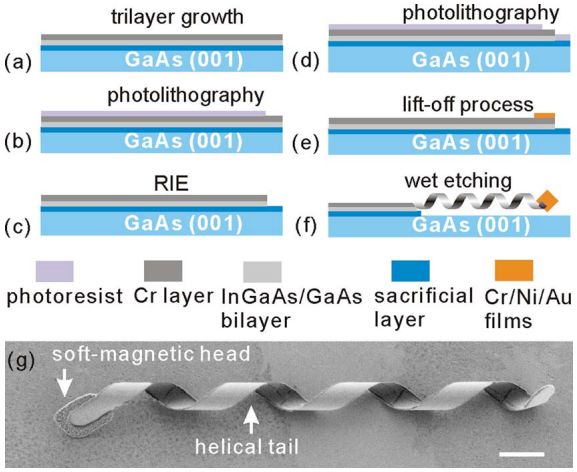


Figure 6: [(a)–(f)] Fabrication procedure of the ABF with InGaAs/GaAs/Cr helical tail. (g) FESEM image of an untethered ABF. The scale bar is 4 µm (Zhang et al., 2009).

6 SUMMARY

Recent advances in microbotics have demonstrated new capabilities in wirelessly controlling microscale structures with high precision over long distances in liquid environments. These breakthroughs make it possible to experimentally investigate the use of these microrobots for manipulating micro and nano size structures in as many as six degrees-of-freedom. Applications to nanomedicine in areas related to targeted medical therapies and molecular manipulation are clear, though many challenges must be addressed. To functionalize these devices and to improve their performance capabilities, fundamental issues in the role surface forces play must be addressed; biocompatibility must be ensured; loading and diffusion of biomolecules must be investigated; and interactions with and manipulation of tissue and macromolecules must be considered. There is a lot yet to do.

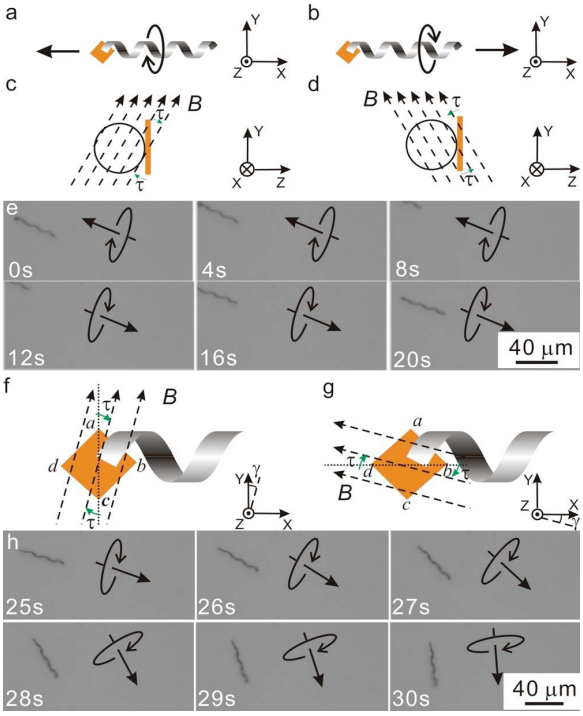


Figure 7: ABF swimming motion controlled by magnetic fields with field strength of 2.0 mT. [(a)-(d)] Schematic of a left-handed ABF swimming forward and backward. With the field B continuously rotating perpendicular to the X axis of the ABF, a misalignment angle between the field and the thin magnetic head will induce a magnetic torque (t) that attempts to align the ABF head with the field, resulting in rotation and propulsion of the ABF. (e) Optical microscope images of the forward/ backward motion of an ABF controlled by magnetic fields. The commanded translation and rotation directions of the ABF are indicated by the arrows. (f) If the field is rotated about the Z axis by an angle $|\gamma|<45°$ with respect to the easy axis ac of the head, then the ABF is steered as it is propelled, as the easy axis ac attempts to align with the field. This is the steering principle used during normal operation of the ABF. (g) If the field is rotated about the Z axis by an angle $|\gamma|<45°$ with respect to the easy axis bd, the ABF will instantaneously attempt to rotate perpendicular to the helix axis. However, steering using the bd easy axis is not possible simultaneously with forward/backward propulsion. (h) Optical microscope images of the turning motion of an ABF controlled by magnetic fields. The commanded translation and rotation directions of the ABF are indicated by the arrows.

REFERENCES

F. Tendick, S. S. Sastry, R. S. Fearing, and M. Cohn., "Applications of micromechatronics in minimally invasive surgery," *IEEE/ASME Transactions on Mechatronics,* vol. 3, pp. 34-42, 1998.

P. Dario, M. C. Carrozza, A. Benvenuto, and A. Menciassi, "Micro-systems in biomedical applications," *Journal of Micromechanics and Microengineering,* vol. 10, pp. 235-244, 2000.

M. J. Mack, "Minimally invasive and robotic surgery," *Journal of American Medical Association,* vol. 285, pp. 568-572, 2001.

M. C. Carrozza, P. Dario, and L. P. S. Jay, "Micromechatronics in surgery," *Transactions of the Institute of Measurement and Control,* vol. 25, pp. 309-327, 2003.

I. Kassim, L. Phee, W. S. Ng, F. Gong, P. Dario, and C. A. Mosse, "Locomotion techniques for robotic colonoscopy," *IEEE Engineering in Medicine and Biology Magazine,* pp. 49-56, May/June 2006.

E. A. Johannessen, L. Wang, S. W. J. Reid, D. R. S. Cumming, and J. M. and Cooper, "Implementation of radiotelemetry in a lab-in-a-pill format," *Lab on a Chip,* vol. 6, pp. 39-45, 2006.

A. Menciassi, D. Accoto, S. Gorini, and P. Dario, "Development of a biomimetic miniature robotic crawler," *Autonomous Robots,* vol. 21, pp. 155-163, 2007.

M. Wautelet, "Scaling laws in the macro-, micro- and nanoworlds," *European Journal of Physics,* vol. 22, pp. 601-611, Nov 2001.

E. M. Purcell, "Life at low Reynolds numbers," *American Journal of Physics,* vol. 45, pp. 3-11, 1977.

K. B. Yesin, K. Vollmers and B.J. Nelson, "Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields," *International Journal of Robotics Research,* vol. 25, pp. 527-536, 2006.

J. J. Abbott, O. Ergeneman, M. Kummer, A. M. Hirt, and B. J. Nelson, " Modeling magnetic torque and force for controlled manipulation of soft-magnetic bodies," *IEEE Transactions on Robotics,* vol. 23, pp. 1247-1252, December 2007.

K. Vollmers, D. Frutiger, B. E. Kratochvil, and B. J. Nelson, "Wireless resonant magnetic actuation for untethered microrobots," *Applied Physics Letters,* vol. 92, April 2008.

L. Zhang, J.J. Abbott, L.X. Dong, B.E. Kratochvil, D.J. Bell, D.J. and B.J. Nelson, "Artificial bacterial flagella: Fabrication and magnetic control," *Applied Physics Letters,* vol. 94, February 2009.

L. Zhang, L. X. Dong, D. J. Bell, B. J. Nelson, C. Schoenenberger, and D. Gruetzmacher, "Fabrication and characterization of freestanding Si/Cr micro- and nanospirals," *Microelectron. Eng.,* vol. 83, pp. 1237-1240, Apr-Sep 2006.

BRIEF BIOGRAPHY

Brad Nelson is the Professor of Robotics and Intelligent Systems at ETH Zürich. His primary research focus is on microrobotics and nanorobotics with an emphasis on applications in biology and

medicine. He received a B.S.M.E. from the University of Illinois at Urbana-Champaign and an M.S.M.E. from the University of Minnesota. He has worked as an engineer at Honeywell and Motorola and served as a United States Peace Corps Volunteer in Botswana, Africa, before obtaining a Ph.D. in Robotics from Carnegie Mellon University in 1995. He was an Assistant Professor at the University of Illinois at Chicago (1995-1998) and an Associate Professor at the University of Minnesota (1998-2002). He became a Full Professor at ETH Zürich in 2002.

Prof. Nelson has been awarded a McKnight Land-Grant Professorship and is a recipient of the Office of Naval Research Young Investigator Award, the National Science Foundation Faculty Early Career Development (CAREER) Award, the McKnight Presidential Fellows Award, and the Bronze Tablet. He was elected as a Robotics and Automation Society Distinguished Lecturer in 2003 and 2008 and won Best Paper Awards at major robotics conferences and journals in 2004, 2005, 2006, 2007, 2008 and 2009. He was named to the 2005 “Scientific American 50,” Scientific American magazine’s annual list recognizing fifty outstanding acts of leadership in science and technology from the past year for his efforts in nanotube manufacturing. His laboratory won the 2007 and 2009 RoboCup Nanogram Competition, both times the event has been held. He serves on the editorial boards of several journals, has served as the head of the Department of Mechanical and Process Engineering from 2005–2007, and is currently the Chairman of the ETH Electron Microscopy Center (EMEZ).

DYNAMIC MODELING OF ROBOTS USING RECURSIVE NEWTON-EULER TECHNIQUES

Wisama Khalil
Ecole Centrale de Nantes, IRCCyN UMR CNRS 6597, 1 Rue de la Noë, 44321 Nantes, France
Wisama.khalil@irccyn.ec-nantes.fr

Keywords: Dynamic Modelling, Newton-Euler, Recursive Calculation, Tree Structure, Parallel Robots, Flexible Joints, Mobile Robots.

Abstract: This paper present the use of recursive Newton-Euler to model different robotics systems. The main advantages of this technique are the facility of implementation by numerical or symbolical programming and providing models with reduced number of operations. In this paper the inverse and direct dynamic models of different robotics systems will be presented. At first we start by rigid tree structure robots, then these algorithms will be generalized for closed loop robots, parallel robots, and robots with lumped elasticity. At the end the case of robots with moving base will be treated.

1 INTRODUCTION

The dynamic modelling of robots is an important topic for the design, simulation, and control of robots. Different techniques have been proposed and used by the robotics community. In this paper we show that the use of Newton-Euler recursive technique for different robotics systems is easy to develop and programme. The proposed algorithm can be extended to many types of structures; serial, tree structure, closed, parallel, with a fixed base or with moving platform. The same technique can be used for robots with lumped elasticity or flexible links.
In section 2 we will recall the method used to describe the kinematics of the structure, and then in section 3 we present the inverse and the direct dynamic modeling of tree structure rigid robots which are considered as the base methods. The following sections present the generalization to the other systems.

2 DESCRIPTION OF THE KINEMATICS OF ROBOTS

The geometry of the structures will be described using the Modified Denavit and Hartenberg method as proposed in (Khalil and Kleinfinger, 1986). This method can take into account tree structures and closed loop robots. Its use facilitates the calculation of the base inertial parameters of robots (Gautier and Khalil, 1988, Khalil W., Bennis F., 1994, Khalil and Bennis, 1995).

2.1 Geometric Description of Tree Structure Robots

A tree structure robot is composed of n+1 links and n joints. Link 0 is the base and link n is a terminal link. The joints are either revolute or prismatic, rigid or elastic. The links are numbered consecutively from the base, link 0, to the terminal links. Joint j connects link j to link a(j), where a(j) denotes the link antecedent to link j. A frame R_i is attached to each link i such that (Figure 1):

- $\mathbf{z}_i$ is along the axis of joint i;
- $\mathbf{x}_i$ is taken along the common normal between $\mathbf{z}_i$ and one of the succeeding joint axes, which are fixed on link i.

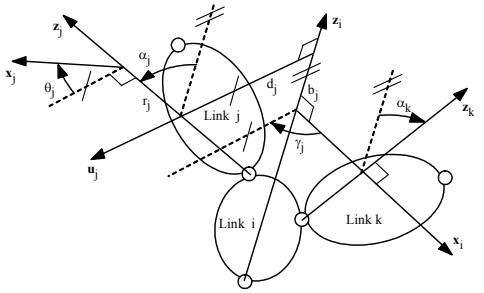


Figure 1: Geometric parameters for a link i.

In general the homogeneous transformation matrix ${}^{i}\mathbf{T}_j$, which defines the frame R_j relative to frame R_i is obtained as a function of six geometric parameters (γ_j , b_j, α_j, d_j, θ_j, r_j). Thus ${}^{i}\mathbf{T}_j$ is obtained as:

$${}^{i}\mathbf{T}_j = \mathbf{Rot}(\mathbf{z}, \gamma_j)\ \mathbf{Tran}(\mathbf{z}, b_j)\ \mathbf{Rot}(\mathbf{x}, \alpha_j)\ \mathbf{Tran}(\mathbf{x}, d_j)\ \mathbf{Rot}(\mathbf{z}, \theta_j)\ \mathbf{Tran}(\mathbf{z}, r_j)$$

After developing, this matrix can be partitioned as follows:

$${}^{i}\mathbf{T}_j = \begin{bmatrix} {}^{i}\mathbf{R}_j & {}^{i}\mathbf{P}_j \\ \mathbf{0}_{1x3} & 0 \end{bmatrix} \quad (1)$$

Where **R** defines the (3×3) rotation matrix and P defines the (3×1) vector defining the position of the origin of frame j with respect to frame i.

If $\mathbf{x}_i$ is along the common normal between $\mathbf{z}_i$ and $\mathbf{z}_j$, the parameters γ_j and b_j will be equal to zero.

The joint variable of joint j is denoted by:

$$q_j = \bar{\sigma}_j \theta_j + \sigma_j r_j$$

where $\sigma_j = 0$ if joint j is revolute, $\sigma_j = 1$ if joint j is prismatic, and $\bar{\sigma}_j = 1 - \sigma$. We set $\sigma_j = 2$ to define a frame R_j fixed with respect to frame a(j). In this case, q_j and $\bar{\sigma}_j$ are not defined.

The serial structure is a special case of a tree structure where a(j)=j-1, $\gamma_j = 0$, and $b_j = 0$ for all j=1,..,n.

2.2 Description of Closed Loop Structure

The system is composed of L joints and n + 1 links, where link 0 is the fixed base and L > n. The number of independent closed loops is equal to:

$$B = L - n$$

The joints are either active (motorized) or passive. The number of active joints is denoted N. The position and orientation of all the links can be determined as a function of the active joint variables.

To determine the geometric parameters of a mechanism with closed chains, we proceed as follows:

a) Construct an equivalent tree structure having n joints by virtually cutting each closed chain at one of its passive joints. Define the geometric parameters of the tree structure as given in section 2.1.

b) For each cut joint define two supplementary frames on one of the links connected by this joint. Assuming that a cut joint is numbered k (where k=n+1,…, L) and that the links connected by joint k are numbered i and j (where i and j <n) the frames will be defined as follows (Figure 2):

- frame R_k is defined fixed on link j such that a(k)=i, the axis $\mathbf{z}_k$ is along the axis of joint k, and $\mathbf{x}_k$ is along the common normal between $\mathbf{z}_k$ and $\mathbf{z}_j$. The matrix ${}^{i}\mathbf{T}_k$ will be determined using the general parameters γ_k, b_k, α_k, d_k, θ_k, r_k.

- frame R_{k+B} is aligned with R_k that is to say it is fixed on link j, but a(k+B)= j. The geometric parameters defining R_{k+B} are constant, we note that r_{k+B} and θ_{k+B} are zero since $\mathbf{x}_{k+B}$ is normal to $\mathbf{z}_j$.

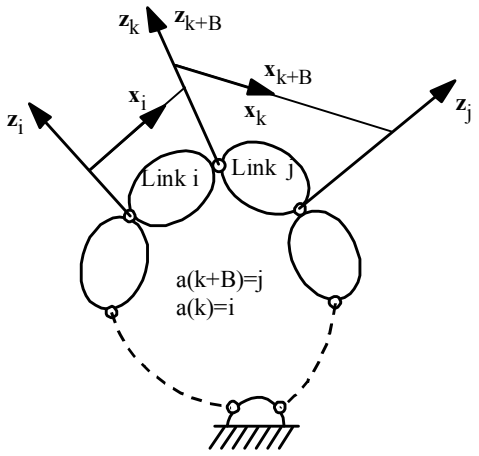


Figure 2: Frames of a cut joint k.

The joint variables are denotes as:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{tr} \\ \mathbf{q}_c \end{bmatrix}, \mathbf{q}_{tr} = \begin{bmatrix} \mathbf{q}_a \\ \mathbf{q}_p \end{bmatrix} \quad (2)$$

- $\mathbf{q}_{tr}$ vector containing the tree structure joint variables;
- $\mathbf{q}_a$ vector containing the N active joint variables;
- $\mathbf{q}_p$ vector containing the p=n−N passive joint variables of the equivalent tree structure;
- $\mathbf{q}_c$ vector containing the B variables of the cut joints.

Only the N active variables $\mathbf{q}_a$ are independent. Since R_k and R_{k+B} are aligned, the geometric constraint equations for each loop, which can be used to calculate the passive joint variables in terms of the active joint variables, can be written as:

$${}^{k+B}\mathbf{T}_j \ldots {}^{i}\mathbf{T}_k = \mathbf{I}_4 \quad (3)$$

The kinematic constraint equations are obtained by using the fact that the screw of frame k is equal to that of frame k+B:

$$\begin{aligned} {}^{0}\mathbb{V}_{k} &= {}^{0}\mathbb{V}_{k+B} \\ \mathbf{J}_{k}\,\dot{\mathbf{q}}_{\mathbf{b1}} &= \mathbf{J}_{k+B}\dot{\mathbf{q}}_{\mathbf{b2}} \end{aligned} \tag{4}$$

$\mathbb{V}_j$ (6×1) kinematic screw vector of frame j, given by:

$$\mathbb{V}_j = \begin{bmatrix} \mathbf{V}_j^{\mathrm{T}} & \boldsymbol{\omega}_j^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{5}$$

$\mathbf{V}_j$ linear velocity of the origin of frame R_j,
$\boldsymbol{\omega}_j$ angular velocity of frame j,
$\dot{\mathbf{q}}_{\mathbf{b1}}$ joint velocities from the base to frame k, through branch 1,
$\dot{\mathbf{q}}_{\mathbf{b2}}$ joint velocities from the base to frame k through branch 2.

3 DYNAMIC MODELING OF TREE STRUCTURE ROBOTS

3.1 Introduction

The most common in use methods to calculate the dynamic models are the Lagrange equations and the Newton Euler Equations (Craig 1986, Khalil and Dombre 2002, Angeles 2006).

The Lagrange equation is given as:

$$\boldsymbol{\Gamma} = \frac{\mathrm{d}}{\mathrm{dt}}\left[\frac{\partial L}{\partial \dot{\mathbf{q}}}\right]^{T} - \left[\frac{\partial L}{\partial \mathbf{q}}\right]^{T} \tag{6}$$

where Γ is the joint torques and forces, L is the Lagrangian of the robot defined as the difference between the kinetic energy E and the potential energy U of the system : L = E – U. After developing we obtain:

$$\boldsymbol{\Gamma} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) \tag{7}$$

where **A** is the inertia matrix of the robot and **H** is the Coriolis, Centrifuge and gravity torques.

Solving the previous equation to find Γ in terms of $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ is known as the inverse dynamic problem, and solving it to obtain $\ddot{\mathbf{q}}$ in terms of $(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\Gamma})$ is known as the direct dynamic model. The inverse dynamic model is obtained by substituting $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ into (7), whereas the direct model needs to inverse the inertia matrix.

$$\ddot{\mathbf{q}} = -\left(\mathbf{A}\right)^{-1}\left(\boldsymbol{\Gamma} - \mathbf{H}\right) \tag{8}$$

The calculation of the Lagrange equations for systems with big number of degrees of freedom using developed symbolic methods is time consuming, and the obtained model will need more time to execute with respect to that of recursive methods. The recursive Newton-Euler algorithms have been shown to be an excellent tool to model rigid robots (Khalil and Kleinfinger, 1987, Khalil and Creusot, 1987, Khosla, 1987). In (Hollerbach, 1980) an efficient recursive Lagrange algorithm is presented but without achieving better performances than that of Newton-Euler.

The Newton-Euler equations giving the external forces and moments on a link j about the origin of frame j are written as:

$${}^{j}\mathbb{F}_j = {}^{j}\mathbb{J}_j\,{}^{j}\dot{\mathbb{V}}_j + \begin{bmatrix} {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\boldsymbol{\omega}_j \times {}^{j}\mathbf{MS}_j\right) \\ {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\mathbf{J}_j\,{}^{j}\boldsymbol{\omega}_j\right) \end{bmatrix} \tag{9}$$

where

$${}^{j}\mathbb{F}_j = \begin{bmatrix} {}^{j}\mathbf{F}_j \\ {}^{j}\mathbf{M}_j \end{bmatrix} \tag{10}$$

$\boldsymbol{\omega}_j$ the angular velocity of link j;
$\dot{\mathbf{V}}_j$ the linear acceleration of the origin of frame j;
$\mathbb{F}_j$ total external wrench on link j;
$\mathbf{F}_j$ total external forces on link j;
$\mathbf{M}_j$ total external moments on link j about O_j;
$\mathbb{J}_j$ (6×6) inertia matrix of link j:

$${}^{j}\mathbb{J}_j = \begin{bmatrix} \mathrm{M}_j\mathbf{I}_3 & -{}^{j}\mathbf{M}\hat{\mathbf{S}}_j \\ {}^{j}\mathbf{M}\hat{\mathbf{S}}_j & {}^{j}\mathbf{J}_j \end{bmatrix} \tag{11}$$

Where M_j, $\mathbf{MS_j}$ and $\mathbf{J_j}$ are the standard inertial parameters of link j. They are respectively, the mass, the first moments, and the inertia matrix about the origin.

3.2 Calculation of the Inverse Dynamics using Recursive NE Algorithm

The algorithm consists of two recursive computations (Luh, Walker and Paul, 1980): forward recursion and backward recursion. The forward equations, from link 1 to link n, compute the link velocities and accelerations and consequently the dynamic wrench on each link. The backward equations, from link n to the base, provide the reaction wrenches on the links and consequently the joint torques.

This method gives the joint torques in terms of the joint positions, velocities and accelerations without explicitly computing the matrices **A** and **H**. That is to say the algorithm will be denoted by:

$$\boldsymbol{\Gamma} = \mathrm{NE}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{f}_e,\mathbf{m}_e) \tag{12}$$

Where $\mathbf{f}_e$ and $\mathbf{m}_e$ are the external forces and moments of the links of the robot on the environment.

The forward recursive equations are based on the following equations (Khalil and Dombre 2002):

$$ {}^{j}\mathbb{V}_j = {}^{j}\mathbb{T}_i \, {}^{i}\mathbb{V}_i + \dot{q}_j \, {}^{j}\mathbb{a}_j \quad (13) $$

$$ {}^{j}\dot{\mathbb{V}}_j = {}^{j}\mathbb{T}_i \, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\boldsymbol{\gamma}_j + \ddot{q}_j \, {}^{j}\mathbb{a}_j \quad (14) $$

$$ {}^{j}\boldsymbol{\gamma}_j = \begin{bmatrix} {}^{j}\mathbf{R}_i \left[{}^{i}\boldsymbol{\omega}_i \times \left({}^{i}\boldsymbol{\omega}_i \times {}^{i}\mathbf{P}_j \right) \right] + 2\sigma_j \left({}^{j}\boldsymbol{\omega}_i \times \dot{q}_j \, {}^{j}\mathbf{a}_j \right) \\ \bar{\sigma}_j \, {}^{j}\boldsymbol{\omega}_i \times \dot{q}_j \, {}^{j}\mathbf{a}_j \end{bmatrix} \quad (15) $$

where

${}^{j}\mathbb{a}_j$ is the (6x1) column matrix given as :

$$ {}^{j}\mathbb{a}_j = [\, 0 \;\; 0 \; \sigma_j \;\; 0 \;\; 0 \;\; \bar{\sigma}_j \,]^T \quad (16) $$

${}^{j}\mathbb{T}_i$ and the screw transformation matrix is:

$$ {}^{j}\mathbb{T}_i = \begin{bmatrix} {}^{j}\mathbf{R}_i & -{}^{j}\mathbf{R}_i \, {}^{i}\hat{\mathbf{P}}_j \\ \mathbf{0}_{3x3} & {}^{j}\mathbf{R}_i \end{bmatrix} \quad (17) $$

The forward algorithm is given for j=1,…, n, with i = a(j), as follows:

$$ {}^{j}\boldsymbol{\omega}_i = {}^{j}\mathbf{R}_i \, {}^{i}\boldsymbol{\omega}_i \quad (18) $$

$$ {}^{j}\boldsymbol{\omega}_j = {}^{j}\boldsymbol{\omega}_i + \bar{\sigma}_j \, \dot{q}_j \, {}^{j}\mathbf{a}_j \quad (19) $$

$$ {}^{j}\dot{\boldsymbol{\omega}}_j = {}^{j}\mathbf{R}_i \, {}^{i}\dot{\boldsymbol{\omega}}_i + \bar{\sigma}_j \, (\ddot{q}_j \, {}^{j}\mathbf{a}_j + {}^{j}\boldsymbol{\omega}_i \times \dot{q}_j \, {}^{j}\mathbf{a}_j) \quad (20) $$

$$ {}^{j}\dot{\mathbf{V}}_j = {}^{j}\mathbf{R}_i \, ({}^{i}\dot{\mathbf{V}}_i + {}^{i}\mathbf{U}_i \, {}^{i}\mathbf{P}_j) + \sigma_j \, (\ddot{q}_j \, {}^{j}\mathbf{a}_j + 2 \, {}^{j}\boldsymbol{\omega}_i \times \dot{q}_j \, {}^{j}\mathbf{a}_j) \quad (21) $$

$$ {}^{j}\mathbf{F}_j = M_j \, {}^{j}\dot{\mathbf{V}}_j + {}^{j}\mathbf{U}_j \, {}^{j}\mathbf{MS}_j \quad (22) $$

$$ {}^{j}\mathbf{M}_j = {}^{j}\mathbf{J}_j \, {}^{j}\dot{\boldsymbol{\omega}}_j + {}^{j}\boldsymbol{\omega}_j \times ({}^{j}\mathbf{J}_j \, {}^{j}\boldsymbol{\omega}_j) + {}^{j}\mathbf{MS}_j \times {}^{j}\dot{\mathbf{V}}_j \quad (23) $$

with

$$ {}^{j}\mathbf{U}_j = {}^{j}\hat{\dot{\boldsymbol{\omega}}}_j + {}^{j}\hat{\boldsymbol{\omega}}_j \, {}^{j}\hat{\boldsymbol{\omega}}_j $$

and where $\mathbf{a}_j$ is the unit vector along the $\mathbf{z}_j$ axis which is the axis of joint j.

The matrix $\hat{\mathbf{w}}$ defines the 3×3 vector product matrix associated to the (3×1) vector $\mathbf{W}$ such that:

$$ \hat{\mathbf{w}} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \quad (24) $$

$$ \mathbf{w} \times \mathbf{v} = \hat{\mathbf{w}} \, \mathbf{v} $$

These equations are initialized by $\boldsymbol{\omega}_0 = \mathbf{0}$, $\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$, $\dot{\mathbf{V}}_0 = -\mathbf{g}$, ${}^{0}\mathbf{U}_0 = 0$, with $\mathbf{g}$ is the acceleration of gravity.

Initialising the linear acceleration $\dot{\mathbf{V}}_0$ by $-\mathbf{g}$ will take automatically the effect of gravity forces on all the links of the structure.

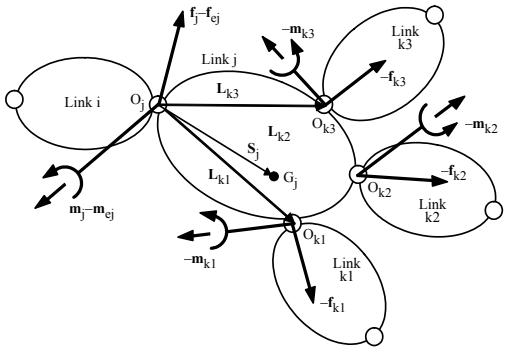


Figure 3: Forces and moments acting on a link j.

The backward recursive equations are deduced from the resultant forces and moments on link j around the origin of link j (Figure 3).

$$ {}^{j}\mathbb{f}_j = {}^{j}\mathbb{F}_j + \sum_k {}^{k}\mathbb{T}_j^T \, {}^{k}\mathbb{f}_k + {}^{j}\mathbb{f}_{ej} $$

(25)

Where a(k)=j,

The backward equations can be calculated for j=n,…,1:

$$ {}^{j}\mathbf{f}_j = {}^{j}\mathbf{F}_j + {}^{j}\mathbf{f}_{ej} \quad (26) $$

$$ {}^{j}\mathbf{m}_j = {}^{j}\mathbf{M}_j + {}^{j}\mathbf{m}_{ej} \quad (27) $$

$$ {}^{i}\mathbf{f}_j = {}^{i}\mathbf{R}_j \, {}^{j}\mathbf{f}_j \quad (28) $$

$$ {}^{i}\mathbf{f}_{ei} = {}^{i}\mathbf{f}_{ei} + {}^{i}\mathbf{f}_j \quad (29) $$

$$ {}^{i}\mathbf{m}_{ei} = {}^{i}\mathbf{m}_{ei} + {}^{i}\mathbf{R}_j \, {}^{j}\mathbf{m}_j + {}^{i}\mathbf{P}_j \times {}^{i}\mathbf{f}_j \quad (30) $$

$$ \Gamma_j = (\sigma_j \, {}^{j}\mathbf{f}_j + \bar{\sigma}_j \, {}^{j}\mathbf{m}_j)^T {}^{j}\mathbf{a}_j + Ia_j \, \ddot{q}_j + F_{sj} \, \mathrm{sign}(\dot{q}_j) + F_{vj} \, \dot{q}_j \quad (31) $$

Where:

$\mathbf{f}_j$ and $\mathbf{m}_j$ are the reaction forces and moments of link a(j) on link j respectively, Ia_j is the inertia of the rotor and transmission gears of the motor of joint j, F_{sj} and F_{vj} are the coulomb and viscous friction parameters respectively, ${}^{j}\mathbf{f}_{ej}$ and ${}^{j}\mathbf{m}_{ej}$ are the external forces and moments of link j on the environment.

This algorithm is easy to program numerically or symbolically. The computational cost is linear with the number of degrees of freedom of the robot. To reduce the number of operations of the calculation of this model the base inertial parameters can be used instead of the standard inertial parameters and the technique of customized symbolic method can be applied (Khosla 1986 , Khalil and Kleinfinger 1987, Khalil and Creusot 1997).

3.3 Computation of the Direct Dynamic Model

The computation of the direct dynamic model is employed to carry out simulations for the purpose of testing the robot performances and studying the control laws. During simulation, the dynamic equations are solved for the joint accelerations given the input torques and the state of the robot (joint positions and velocities). Through integration of the joint accelerations, the robot trajectory is then determined.

The direct dynamic model can be obtained from Lagrange equation (8) as follows:

$$\ddot{\mathbf{q}} = \mathbf{A}^{-1} [\boldsymbol{\Gamma} - \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})]$$

Two methods based on Newton-Euler methods can be used to obtain the dynamic model: the first is based on calculating the **A** and **H** matrices using Newton-Euler inverse dynamic model in order to calculate the joint accelerations by (8); the second method is based on a recursive Newton-Euler algorithm that does not explicitly calculate the matrix **A** and has a computational cost that varies linearly with the number of degrees of freedom of the robot. For tree structure robots, the second method is more efficient, but the first method can be used for closed loop robots and other complicated systems. That is why we will present both methods.

3.3.1 Using the Inverse Dynamic Model to Calculate the Direct Dynamic Model

In this method the matrices $\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})$ and $\mathbf{A}(\mathbf{q})$ are calculated using the inverse model by giving special values for the joint accelerations, joint velocities, external forces, friction, gravity (Walker and Orin 1982).

By comparing equations (7) and (12) we deduce that $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ is equal to $\boldsymbol{\Gamma}$ if $\ddot{\mathbf{q}} = \mathbf{0}$, and that the i^{th} column of **A** is equal to $\boldsymbol{\Gamma}$ if:

$$\ddot{\mathbf{q}} = \mathbf{u}_i, \dot{\mathbf{q}} = \mathbf{0}, \mathbf{g} = \mathbf{0}, \mathbf{f}_{ej} = \mathbf{0}, \mathbf{m}_{ej} = \mathbf{0}$$

where $\mathbf{u}_i$ is the (nx1) unit vector whose i^{th} element is equal to 1, and the other elements are zeros. Iterating the procedure for i = 1,…, n leads to the construction of the entire inertia matrix.

To reduce the computational complexity of this algorithm, we can make use of the base inertial parameters and the customized symbolic techniques. Moreover, we can take advantage of the fact that the inertia matrix **A** is symmetric.

3.3.2 Recursive NE Computation of the Direct Dynamic Model

This method is based on the recursive Newton-Euler equations and does not use explicitly the inertia matrix of the robot (Armstrong 1979, (Featherstone 1983, (Brandl, Johanni and Otter, 1986).

Using (9) and (25) the equilibrium equations of link j can be written as:

$${}^j\mathbb{J}_j\,{}^j\dot{\mathbb{V}}_j = {}^j\mathbf{f}_j + {}^j\boldsymbol{\beta}_j - \sum_{\mathbf{k}} {}^k\mathbb{T}_j^T\,{}^k\mathbf{f}_k \qquad (32)$$

where k denote the links articulated on link j such that a(k)=j, and

$${}^j\boldsymbol{\beta}_j = -{}^j\mathbf{f}_{ej} - \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times \left({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{MS}_j \right) \\ {}^j\boldsymbol{\omega}_j \times \left({}^j\mathbf{J}_j\,{}^j\boldsymbol{\omega}_j \right) \end{bmatrix} \qquad (33)$$

The joint accelerations are obtained as a result of three recursive computations:

i) first forward computations for j = 1, ..., n: in this step, we compute the screw transformation matrices ${}^j\mathbb{T}_i$, the link angular velocities ${}^j\boldsymbol{\omega}_j$ as well as ${}^j\boldsymbol{\gamma}_j$ and ${}^j\boldsymbol{\beta}_j$ vectors, which appear in the link accelerations and the link wrenches equations respectively when $\ddot{\mathbf{q}} = \mathbf{0}$;

$${}^j\boldsymbol{\gamma}_j = \begin{bmatrix} {}^j\mathbf{R}_i \left[{}^i\boldsymbol{\omega}_i \times \left({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_j \right) \right] + 2\sigma_j ({}^j\boldsymbol{\omega}_i \times \dot{q}_j\,{}^j\mathbf{a}_j) \\ \bar{\sigma}_j\,{}^j\boldsymbol{\omega}_i \times \dot{q}_j\,{}^j\mathbf{a}_j \end{bmatrix} \qquad (34)$$

$${}^j\boldsymbol{\beta}_j = -{}^j\mathbf{f}_{ej} - \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times \left({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{MS}_j \right) \\ {}^j\boldsymbol{\omega}_j \times \left({}^j\mathbf{J}_j\,{}^j\boldsymbol{\omega}_j \right) \end{bmatrix} \qquad (35)$$

ii) backward recursive computation: in this step we calculate the elements H_j, ${}^j\mathbb{J}_j^*$, ${}^j\boldsymbol{\beta}_j^*$, ${}^j\mathbb{K}_j$, ${}^j\boldsymbol{\alpha}_j$ which express $\ddot{q}_j$ and ${}^j\mathbb{f}_j^*$ in terms of ${}^i\dot{\mathbb{V}}_i$ in the third recursive equations. These equations are demonstrated in the following sub-section.

For j = n ... 1, compute:

$$H_j = ({}^j\mathbb{a}_j^T\,{}^j\mathbb{J}_j^*\,{}^j\mathbb{a}_j + Ia_j) \qquad (36)$$

$${}^j\mathbb{K}_j = {}^j\mathbb{J}_j^* - {}^j\mathbb{J}_j^*\,{}^j\mathbb{a}_j\,H_j^{-1}\,{}^j\mathbb{a}_j^T\,{}^j\mathbb{J}_j^* \qquad (37)$$

$${}^j\boldsymbol{\alpha}_j = {}^j\mathbb{K}_j\,{}^j\boldsymbol{\gamma}_j + {}^j\mathbb{J}_j^*\,{}^j\mathbb{a}_j\,H_j^{-1}\,(\tau_j + {}^j\mathbb{a}_j^T\,{}^j\boldsymbol{\beta}_j^*) - {}^j\boldsymbol{\beta}_j^* \qquad (38)$$

If $a(j) \neq 0$, calculate also:

$${}^i\boldsymbol{\beta}_i^* = {}^i\boldsymbol{\beta}_i^* - {}^j\mathbb{T}_i^T\,{}^j\boldsymbol{\alpha}_j \qquad (39)$$

$$ {}^{i}\mathbb{J}_i^* = {}^{i}\mathbb{J}_i^* + {}^{j}\mathbb{T}_i^T \, {}^{j}\mathbb{K}_j \, {}^{j}\mathbb{T}_i \quad (40) $$

These equations are initialized by

$${}^{j}\mathbb{J}_j^* = {}^{j}\mathbb{J}_j \text{ and } {}^{j}\boldsymbol{\beta}_j^* = {}^{j}\boldsymbol{\beta}_j.$$

iii) second forward recursive computations. Since the acceleration of the base is known ($\dot{\mathbf{V}}_0 = -\mathbf{g}$, $\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$ for fixed base), the third recursive computation gives $\ddot{q}_j$ and ${}^{j}\mathbb{f}_j$ (if needed) for j = 1… n. as follows:

$$ \ddot{q}_j = H_j^{-1} \left[-{}^{j}\mathbb{a}_j^T \, {}^{j}\mathbb{J}_j^* \left({}^{j}\mathbb{T}_i \, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\boldsymbol{\gamma}_j \right) + \tau_j + {}^{j}\mathbb{a}_j^T \, {}^{j}\boldsymbol{\beta}_j^* \right] \quad (41) $$

$$ {}^{j}\mathbb{f}_j = \begin{bmatrix} {}^{j}\mathbf{f}_j \\ {}^{j}\mathbf{m}_j \end{bmatrix} = {}^{j}\mathbb{K}_j \, {}^{j}\mathbb{T}_i \, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\boldsymbol{\alpha}_j \quad (42) $$

$$ {}^{j}\dot{\mathbb{V}}_j = {}^{j}\mathbb{T}_i \, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\mathbb{a}_j \, \ddot{q}_j + {}^{j}\boldsymbol{\gamma}_j \quad (43) $$

where

$$ \tau_j = \Gamma_j - F_{sj} \, \mathrm{sign}(\dot{q}_j) - F_{vj} \, \dot{q}_j \quad (44) $$

Calculation of the elements of the backward recursive equations

To simplify the notations, we consider the case of a serial structure of n joints. Expressing the acceleration of link n in terms of the acceleration of link n-1, and since ${}^{n+1}\mathbb{f}_{n+1} = \mathbf{0}$, we obtain:

$$ {}^{n}\mathbb{J}_n \left({}^{n}\mathbb{T}_{n-1} \, {}^{n-1}\dot{\mathbb{V}}_{n-1} + \ddot{q}_n \, {}^{n}\mathbb{a}_n + {}^{n}\boldsymbol{\gamma}_n \right) = {}^{n}\mathbb{f}_n + {}^{n}\boldsymbol{\beta}_n \quad (45) $$

Since:

$${}^{j}\mathbb{a}_j^T \, {}^{j}\mathbb{f}_j = \tau_j - Ia_j \, \ddot{q}_j$$

$$\tau_j = \Gamma_j - F_{sj} \, \mathrm{sign}(\dot{q}_j) - F_{vj} \, \dot{q}_j$$

We obtain the joint acceleration of joint n:

$$ \ddot{q}_n = H_n^{-1} \left(-{}^{n}\mathbb{a}_n^T \, {}^{n}\mathbb{J}_n \left({}^{n}\mathbb{T}_{n-1} \, {}^{n-1}\dot{\mathbb{V}}_{n-1} + {}^{n}\boldsymbol{\gamma}_n \right) + \tau_n + {}^{n}\mathbb{a}_n^T \, {}^{n}\boldsymbol{\beta}_n \right) \quad (46) $$

where H_n is a scalar given as:

$$ H_n = \left({}^{n}\mathbb{a}_n^T \, {}^{n}\mathbb{J}_n \, {}^{n}\mathbb{a}_n + Ia_n \right) \quad (47) $$

Substituting for $\ddot{q}_n$ from (46) and (45), we obtain the dynamic wrench ${}^{n}\mathbb{f}_n$ as:

$$ {}^{n}\mathbb{f}_n = \begin{bmatrix} {}^{n}\mathbf{f}_n \\ {}^{n}\mathbf{m}_n \end{bmatrix} = {}^{n}\mathbb{K}_n \, {}^{n}\mathbb{T}_{n-1} \, {}^{n-1}\dot{\mathbb{V}}_{n-1} + {}^{n}\boldsymbol{\alpha}_n \quad (48) $$

where:

$$ {}^{n}\mathbb{K}_n = {}^{n}\mathbb{J}_n - {}^{n}\mathbb{J}_n \, {}^{n}\mathbb{a}_n \, H_n^{-1} \, {}^{n}\mathbb{a}_n^T \, {}^{n}\mathbb{J}_n \quad (49) $$

$$ {}^{n}\boldsymbol{\alpha}_n = {}^{n}\mathbb{K}_n \, {}^{n}\boldsymbol{\gamma}_n + {}^{n}\mathbb{J}_n \, {}^{n}\mathbb{a}_n \, H_n^{-1} \left(\tau_n + {}^{n}\mathbb{a}_n^T \, {}^{n}\boldsymbol{\beta}_n \right) - {}^{n}\boldsymbol{\beta}_n \quad (50) $$

We now have $\ddot{q}_n$ and ${}^{n}\mathbb{f}_n$ in terms of ${}^{n-1}\dot{\mathbb{V}}_{n-1}$. Iterating the procedure for j = n – 1, we obtain:

$$ {}^{n-1}\mathbb{J}_{n-1} \, {}^{n-1}\dot{\mathbb{V}}_{n-1} = {}^{n-1}\mathbb{f}_{n-1} + {}^{n}\mathbb{T}_{n-1}^T \, {}^{n}\mathbb{f}_n + {}^{n-1}\boldsymbol{\beta}_{n-1} \quad (51) $$

which can be rewritten as:

$$ {}^{n-1}\mathbb{J}_{n-1}^* \left({}^{n-1}\mathbb{T}_{n-2} \, {}^{n-2}\dot{\mathbb{V}}_{n-2} + \ddot{q}_{n-1} \, {}^{n-1}\mathbb{a}_{n-1} + {}^{n-1}\boldsymbol{\gamma}_{n-1} \right) = {}^{n-1}\mathbb{f}_{n-1} + {}^{n-1}\boldsymbol{\beta}_{n-1}^* \quad (52) $$

where:

$$ {}^{n-1}\mathbb{J}_{n-1}^* = {}^{n-1}\mathbb{J}_{n-1} + {}^{n}\mathbb{T}_{n-1}^T \, {}^{n}\mathbb{K}_n \, {}^{n}\mathbb{T}_{n-1} \quad (53) $$

$$ {}^{n-1}\boldsymbol{\beta}_{n-1}^* = {}^{n-1}\boldsymbol{\beta}_{n-1} - {}^{n}\mathbb{T}_{n-1}^T \, {}^{n}\boldsymbol{\alpha}_n \quad (54) $$

Equation (52) has the same form as (45). Thus, we can express $\ddot{q}_{n-1}$ and ${}^{n-1}\mathbb{f}_{n-1}$ in terms of ${}^{n-2}\dot{\mathbb{V}}_{n-2}$. Iterating this procedure for j = n – 2, …, 1, we obtain $\ddot{q}_j$ and ${}^{j}\mathbb{f}_j$ in terms of ${}^{j-1}\dot{\mathbb{V}}_{j-1}$ for j = n – 1, ..., 1 as given by equations (41) and (43) which represent the general case.

4 INVERSE DYNAMIC MODELING OF CLOSED LOOP ROBOTS

The computation of the Inverse dynamic model of closed loop robots can be obtained by first calculating the inverse dynamic model of the equivalent tree structure robot, in which the joint variables satisfy the constraints of the loop. Then the closed loop torques of the active joints $\boldsymbol{\Gamma}_c$ are obtained by projecting the tree structure torques $\boldsymbol{\Gamma}_{tr}$ on the motorized joints using the transpose of the Jacobian matrix of the tree structure variables (or velocities) in terms of the active joint variables (or velocities).

$$ \boldsymbol{\Gamma}_c = \mathbf{G}^T \boldsymbol{\Gamma}_{tr} \left(\mathbf{q}_{tr}, \dot{\mathbf{q}}_{tr}, \ddot{\mathbf{q}}_{tr} \right) \quad (55) $$

where:

$$ \mathbf{G} = \frac{\partial \mathbf{q}_{tr}}{\partial \mathbf{q}_a} = \frac{\partial \dot{\mathbf{q}}_{tr}}{\partial \dot{\mathbf{q}}_a} \quad (56) $$

It can be written also as:

$$ \boldsymbol{\Gamma}_c = \boldsymbol{\Gamma}_a + \frac{\partial \dot{\mathbf{q}}_p}{\partial \dot{\mathbf{q}}_a} \boldsymbol{\Gamma}_p \quad (57) $$

Where:

$\boldsymbol{\Gamma}_a$ and $\boldsymbol{\Gamma}_p$ are the torque of actuated and passive joints of the tree structure.

The kinematics Jacobian matrix can be obtained from (4) representing the kinematics closed loop constraints.

There is no recursive method to obtain the direct dynamic model of closed loop robots. It can be

computed using the inverse dynamic model by a procedure similar to that given in section (3.3.1) in order to obtain the matrices $\mathbf{A}_c$ and $\mathbf{H}_c$ of the following relation:

$$\boldsymbol{\Gamma}_c = \mathbf{A}_c(\mathbf{q}_{tr})\ddot{\mathbf{q}}_a + \mathbf{H}_c(\mathbf{q}_{tr},\dot{\mathbf{q}}_{tr}) \quad (58)$$

5 INVERSE DYNAMIC MODELING OF PARALLEL ROBOTS

A parallel robot is a complex multi-body system having several closed loops. It is composed of a moving platform connected to a fixed base by parallel legs. The dynamic model can be obtained as described in the previous section, but in this section we present a method that takes into account the parallel structure. To simplify the notations we will present her the case of parallel robots with six degrees of freedom. Examples concerning reduced mobility robots are given in (Khalil and Ibrahim 2007).

The robot is composed of a fixed base and a mobile platform. They are connected using m parallel legs.

The inverse dynamic model gives the forces and torques of motorized joints as a function of the desired trajectory of the mobile platform.

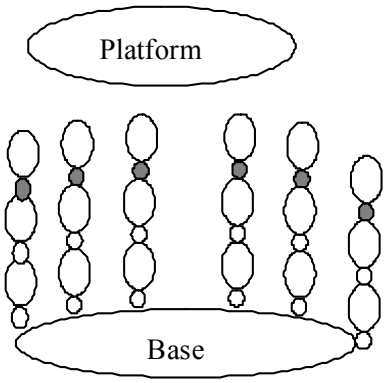


Figure 4: Parallel structure after separating the platform.

To obtain the dynamic models of parallel robots, we exploit their structural characteristics by decomposing the system into two subsystems: the platform and the legs.

The dynamics of the platform is calculated as a function of the Cartesian variables (spatial Cartesian position, velocity and acceleration of the platform), whereas the dynamics of the legs are calculated as a function of the joint variables of the legs $(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i)$ for i=1,…,m. The active joint torques are obtained by the sum of these dynamics and projecting them on the active joint axes.

To project the dynamics of the platform on the active joint space we multiply it by the transpose of the robot Jacobian matrix, which gives the platform screw $\mathbb{V}_p$ in terms of the motorized joint velocities $\dot{\mathbf{q}}_a$, and to project the leg dynamics on the active joint space we use the Jacobian between these two spaces. Thus the dynamic model of the parallel structure is given by the following equation:

$$\boldsymbol{\Gamma} = \mathbf{J}_P^T \mathbb{F}_P + \sum_{i=1}^{m} \left(\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} \right)^T \boldsymbol{\Gamma}_i \quad (59)$$

where

$\mathbb{F}_P$ is the total forces and moments on the platform,

$\mathbf{J}_P$ is the (6×n) kinematics Jacobian matrix of the robot, which gives the platform velocity $\mathbf{V}_P$ (translational and angular) as a function of the active joint velocities:

$$\mathbb{V}_P = \mathbf{J}_P \dot{\mathbf{q}}_a \quad (60)$$

$\boldsymbol{\Gamma}_i$ is the inverse dynamic model of leg i, it is a function of $(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i)$, which can be obtained in terms of the platform location, velocity and acceleration, using the inverse kinematic models of the legs. We note that $\mathbf{q}_i$ does not include the passive joint variables connecting the legs to the platform.

In this section we suppose n=6, thus $\mathbf{J}_P$ is (6×6) matrix.

The calculation of $\mathbf{J}_p$ is obtained by inverting $\mathbf{J}_p^{-1}$, which is easy to obtain for most parallel structures.

$\mathbb{F}_P$ is calculated by the Newton-Euler equation (9).

The calculation of $\partial \dot{\mathbf{q}}_i / \partial \dot{\mathbf{q}}_a$ is carried out by the following relation, which exploits the parallel structure of the robot:

$$\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} = \frac{\partial \dot{\mathbf{q}}_i}{\partial \mathbf{v}_i} \frac{\partial \mathbf{v}_i}{\partial \mathbb{V}_P} \frac{\partial \mathbb{V}_P}{\partial \dot{\mathbf{q}}_a} \quad (61)$$

with: $\mathbf{v}_i$ is the Cartesian velocity transferred from leg i to the platform.

We can rewrite (61) as:

$$\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} = \mathbf{J}_i^{-1} \mathbf{J}_{vi} \mathbf{J}_r \quad (62)$$

$\mathbf{J}_i$ is the kinematic Jacobian matrix of leg i such that:

$$\mathbf{v}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \quad (63)$$

$\mathbf{J}_{vi}$ gives $\mathbf{v}_i$ as a function of $\mathbb{V}_P$:

$$\mathbf{v}_i = \mathbf{J}_{vi}\mathbb{V}_P \tag{64}$$

For the Gough-Stewart platform (where the mobile platform is connected to the legs using spherical joints), we obtain:

$$\mathbf{J}_{vi} = \frac{\partial \mathbf{v}_i}{\partial \mathbb{V}_P} = \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{P}}_i \end{bmatrix} \tag{65}$$

Where P_i is the vector between the origin of the platform frame and the centre of the spherical joint linking the platform with leg i.

Finally the inverse dynamic model of the robot is given by the following form:

$$\mathbf{\Gamma} = \mathbf{J}_p^T \left[\mathbb{F}_P + \sum_{i=1}^{m} \mathbf{J}_{vi}^T \mathbf{J}_i^{-T} \mathbf{\Gamma}_i \right] \tag{66}$$

We note that the term between the brackets in (66) represents the dynamic model of the robot expressed in the Cartesian space of the platform frame (Khalil and Guegain, 2002).

6 INVERSE DYNAMIC MODELING OF ROBOTS WITH ELASTIC JOINTS

In this section we tree structure robots with lumped elasticity or flexible joints. The system can be described using Modified Denavit and Hartenberg method presented in section 2. Each joint could be either elastic or rigid (Khalil and Gautier, 2000).

6.1 Lagrange Dynamic Form

The general form of the dynamic model of a system with flexible joints has the same form as (7). It can be rewritten as:

$$\mathbf{\Gamma} = \mathbf{A}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \tag{67}$$

It can be partitioned as follows:

$$\Gamma = \begin{bmatrix} \Gamma_r \\ \Gamma_f \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{H}_r \\ \mathbf{H}_f \end{bmatrix} \tag{68}$$

Where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are the (n×1) vectors of positions, velocities, and accelerations of rigid and elastic joints;

$\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ is the (n×1) vector of Coriolis, centrifugal and gravity forces,

$\mathbf{A}(\mathbf{q})$ is the (n×n) inertia matrix of the system,

$\mathbf{\Gamma}_r$ is the vector of rigid joint torques,

$\mathbf{\Gamma}_r$ is the vector of elastic joint torques.

If joint j is flexible:

$$\Gamma_j = -\Delta q_j K_j \tag{69}$$

where K_j is the stiffness of the elastic joint,

$$\Delta q_j = q_j - q_{0j} \tag{70}$$

q_{0j} is the joint position corresponding to zero elasticity force.

In the case of a system with elasticity, the direct dynamic model has the same outputs as in the case of rigid bodies; it gives the joint accelerations as a function of the joint torques and of the system state variables ($\mathbf{q}$, $\dot{\mathbf{q}}$). It can be calculated using (68) by calculation the inverse of $\mathbf{A}$.

In the case of a system with elasticity, the inverse dynamic model calculates the input torques and the elastic accelerations as a function of the joint positions, velocities and rigid joint accelerations. It is to be noted that the accelerations of the elastic variables cannot be specified independently. Using (68) to calculate the inverse model, we have first to calculate the acceleration elastic accelerations from the second row:

$$\Gamma_f = \begin{bmatrix} \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_f \end{bmatrix} + \mathbf{H}_f \tag{71}$$

then we can calculate the rigid joint torques from the first row.

6.2 Direct Dynamics of Systems with Flexible Joints using Recursive NE

The direct dynamic model of system with flexible joints can be calculated using the recursive direct dynamic model algorithm of rigid joints presented in section (3.3) after putting $\Gamma_j = -\Delta q_j K_j$ for the elastic joints.

Remark: We note that in case of rigid non motorized joint, the same algorithm can be used after putting $\Gamma_j = 0$.

6.3 Inverse Dynamics of Systems with Flexible Joints using Recursive NE

The recursive inverse dynamic algorithm of rigid links cannot be used for system with flexible joints since the accelerations of the flexible joints are unknown. On the contrary it can be used to obtain the **A** and **H** matrices as explained in section (3.3.1),

then we can proceed as explained in 6.1 for the calculation of $\ddot{\mathbf{q}}_f$ and $\mathbf{\Gamma}_r$.

We propose here a recursive algorithm to solve this problem (Khalil and Gautier 2000). This algorithm consists of three recursive steps.

i) The first forward iteration is exactly the same as that of the direct dynamic model (section 3.3).

ii) The second backward recursive equations calculate the matrices giving the elastic accelerations $\ddot{q}_j$ and ${}^j\mathbb{f}_j$ as a function of ${}^{a(j)}\dot{\mathbb{V}}_{a(j)}$. These matrices can be defined using a similar procedure as in section (3.3). They can be calculated for j =n, ...,1, as follows:

- If joint j is elastic:

$$H_j = {}^j\mathbb{a}_j^T\, {}^j\mathbb{J}_j^*\, {}^j\mathbb{a}_j \tag{72}$$

$${}^j\mathbb{K}_j = {}^j\mathbb{J}_j^* - {}^j\mathbb{J}_j^*\, {}^j\mathbb{a}_j\, H_j^{-1}\, {}^j\mathbb{a}_j^T\, {}^j\mathbb{J}_j^* \tag{73}$$

$${}^j\boldsymbol{\alpha}_j = {}^j\mathbb{K}_j\, {}^j\boldsymbol{\gamma}_j + {}^j\mathbb{J}_j^*\, {}^j\mathbb{a}_j\, H_j^{-1}\, (-K_j\, \Delta q_j + {}^j\mathbb{a}_j^T\, {}^j\boldsymbol{\beta}_j^*) - {}^j\boldsymbol{\beta}_j^* \tag{74}$$

- If joint j is rigid:

$${}^j\mathbb{K}_j = {}^j\mathbb{J}_j^* \tag{75}$$

$${}^j\boldsymbol{\alpha}_j = {}^j\mathbb{K}_j\, {}^j\boldsymbol{\gamma}_j + {}^j\mathbb{J}_j^*\, {}^j\mathbb{a}_j\, \ddot{q}_j - {}^j\boldsymbol{\beta}_j^* \tag{76}$$

if a(j) ≠ 0, calculate:

$${}^i\boldsymbol{\beta}_i^* = {}^i\boldsymbol{\beta}_i^* - {}^j\mathbb{T}_i^T\, {}^j\boldsymbol{\alpha}_j \tag{77}$$

$${}^i\mathbb{J}_i^* = {}^i\mathbb{J}_i^* + {}^j\mathbb{T}_i^T\, {}^j\mathbb{K}_j\, {}^j\mathbb{T}_i \tag{78}$$

The previous equations are initialized by:

$${}^j\mathbb{J}_j^* = {}^j\mathbb{J}_j, \text{ and } {}^j\boldsymbol{\beta}_j^* = {}^j\boldsymbol{\beta}_j.$$

The third recursive equations (for j = 1, ..., n) calculate $\ddot{q}_j$ for the elastic joints and the joint torques for the rigid joints using the following equation:

$${}^j\mathbb{f}_j = \begin{bmatrix} {}^j\mathbf{f}_j \\ {}^j\mathbf{m}_j \end{bmatrix} = {}^j\mathbb{K}_j\, {}^j\mathbb{T}_i\, {}^i\dot{\mathbb{V}}_i + {}^j\boldsymbol{\alpha}_j \tag{79}$$

- if j is elastic:

$$\ddot{q}_j = H_j^{-1}[-{}^j\mathbb{a}_j^T\, {}^j\mathbb{J}_j^*\, ({}^j\mathbb{T}_i\, {}^i\dot{\mathbb{V}}_i + {}^j\boldsymbol{\gamma}_j) - K_j\, \Delta q_j + {}^j\mathbb{a}_j^T\, {}^j\boldsymbol{\beta}_j^* \tag{80}$$

$${}^j\dot{\mathbb{V}}_j = {}^j\mathbb{T}_i\, {}^i\dot{\mathbb{V}}_i + {}^j\mathbb{a}_j\, \ddot{q}_j + {}^j\boldsymbol{\gamma}_j \tag{81}$$

- if j is rigid

$$\mathbf{\Gamma}_j = (\sigma_j\, {}^j\mathbf{f}_j + \bar{\sigma}_j\, {}^j\mathbf{m}_j)^T\, {}^j\mathbf{a}_j + Ia_j\, \ddot{q}_j \tag{82}$$

7 DYNAMIC MODELING OF ROBOTS WITH MOVING BASE

The structure treated in this section includes a big number of systems such as: cars, mobile robots, mobile manipulators, walking robots, Humanoid robots, eel like robots (Khalil W., G. Gallot G., Boyer F., 2007), snakes like robots, flying robots, spatial vehicle, etc. The difference between all of these systems will be in the calculation of the interaction forces with the environment. In the previous sections the base is fixed thus the acceleration of the base is equal to zero, whereas in the case of a mobile base system the acceleration of the base must be determined in both direct and inverse dynamic models. The proposed recursive dynamic models are easy to implement and calculate using numerical calculation. The inverse dynamic model, which is used in general in the control problems, can be used in simulation too when the objective is to study the evolution of the base giving joint positions, velocities and accelerations of the other joints. The direct dynamic model can be used in simulation when the joint torques are specified.

We use the same notations of section 2 to describe the structure. The base fixed frame R_0 is defined wrt the world fixed frame R_w by the transformation matrix ${}^w\mathbf{T}_0$. This matrix is supposed known at t = 0, it will be updated by integrating the base acceleration. The velocity and acceleration of the base are represented by the (6x1) vectors $\mathbb{V}_0$ and $\dot{\mathbb{V}}_0$ respectively.

The Cartesian velocities and accelerations of the links are calculated using the recursive equations (13)-(17).

7.1 General form of the Dynamic Models

The dynamic model of a robot with moving base can be represented by the following relation:

$$\begin{bmatrix} \mathbf{0}_{6x1} \\ \Gamma \end{bmatrix} = \mathbf{A} \begin{bmatrix} {}^0\dot{\mathbb{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{H} \tag{83}$$

$\mathbf{\Gamma}$ $(n \times 1)$ vector of joint torques,
$\mathbf{q}$ $(n \times 1)$ vector of joint positions,
$\mathbf{A}$ is the $(6+n)\times(6+n)$ inertia matrix of the robot, it can be partitioned as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \tag{84}$$

$\mathbf{A}_{11}$ is the (6×6) inertia matrix of the composed link 0, which is composed of the inertia of all the links referred to frame R_0 (the base).

$\mathbf{A}_{22}$ is the $(n \times n)$ inertia matrix of the other links when the head is fixed,

$\mathbf{A}_{12}$ is the $(6 \times n)$ coupled inertia matrix of the joints and the base. It reflects the effect of the joint accelerations on the base motion, and the dual effect of base accelerations on the joint motions.

$\mathbf{H}$ is the $(n+6) \times 1$ vector representing the Coriolis, centrifugal, gravity and external forces effect on the robot. Its elements are functions of the base and joint velocities and the external forces. This vector can be partitioned as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix} \tag{85}$$

where:

$\mathbf{H}_1$ the Coriolis, centrifugal, gravity and external forces on the base.

$\mathbf{H}_2$ the Coriolis, centrifugal, gravity and external forces on the links 1,…,n.

The inverse dynamic model gives the joint torques and the base acceleration in terms of the desired trajectory (position, velocity and acceleration) of the articulated system (links 1 to n) and the base position and velocity. Using equation (83) and (84), the inverse dynamic model is solved by using the first row of equation (83) to obtain the base acceleration:

$$^0\dot{\mathbb{V}}_0 = -\left(\mathbf{A}_{11}\right)^{-1}\left(\mathbf{H}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}}\right) \tag{86}$$

Then the second row of (83), can be used to find the joint torques:

$$\Gamma = \mathbf{A}_{12}^{\mathrm{T}}\,\dot{\mathbb{V}}_0 + \mathbf{A}_{22}\,\ddot{\mathbf{q}} + \mathbf{H}_2 \tag{87}$$

The direct dynamic model gives the joint accelerations and the base acceleration in terms of the position and velocity of the base and the articulated system and the joint input torques. Thus using (83), the direct dynamic model is solved as follows:

$$\begin{bmatrix} \dot{\mathbb{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} = \mathbf{A}^{-1}\begin{bmatrix} -\mathbf{H}_1 \\ \Gamma - \mathbf{H}_2 \end{bmatrix} \tag{88}$$

The calculation of **A** and **H** can be done by Lagrange method. They can also be calculated using the inverse dynamic model of tree structure of section (3.2) and using the procedure of section (3.3.1). The base can be taken into account by either of the following methods:

- The velocity and acceleration of the base will be the initial conditions $\dot{\mathbb{V}}_0$ and $\boldsymbol{\omega}_0$ for the forward recursive calculation. The backward recursive calculation must continue to j=0, where this new iteration will obtain the 6 equations of Newton-Euler equations of the base.
- We can assign link 1 to be the base, and suppose that link 0 is a virtual link whose inertial parameters are equal to zero but has the velocity and acceleration of the base. This can be done by putting σ_2=2. The six equations of the base will be those of $\mathrm{f}_1 = 0$;

Solving the inverse and direct dynamic problems using **A** and **H** may be very time consuming for systems with big number of degrees of freedom (as the eel like robot). Therefore, we propose here to use a recursive method, which is easy to programme, and its computational complexity is linear wrt the number of degrees of freedom.

The recursive Newton-Euler algorithm is based on the kinematic equations presented in section 3.

7.2 Recursive NE Calculation of the Inverse Dynamic Model of Robots with Mobile Base

The inverse dynamic algorithm in this case consists of three recursive equations (a forward, then a backward, then a forward).

i) Forward recursive calculation:

In this step we calculate the screw transformation matrices, link velocities, and the elements of the accelerations and external wrenches on the links, which are independent of the acceleration of the robot base ($\dot{\mathbf{V}}_0, \dot{\boldsymbol{\omega}}_0$). Thus we calculate for j=1,…,n: $^{j}\mathbb{T}_{i}$, $^{j}\mathbb{V}_{j}$ and $^{j}\boldsymbol{\gamma}_{j}$ using equations (13)-(17). We calculate also $^{j}\boldsymbol{\beta}_{j}$ representing the elements of the Newton-Euler equations, which are independent of the base acceleration in equations (14) and (15) such that:

$$^{j}\boldsymbol{\zeta}_{j} = {}^{j}\boldsymbol{\gamma}_{j} + \ddot{\mathrm{q}}_{j}\,{}^{j}\mathbf{a}_{j} \tag{89}$$

$$^{j}\boldsymbol{\beta}_{j} = -{}^{j}\mathbb{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_{j} \times \left({}^{j}\boldsymbol{\omega}_{j} \times {}^{j}\mathbf{MS}_{j}\right) \\ {}^{j}\boldsymbol{\omega}_{j} \times \left({}^{j}\mathbf{J}_{j}\,{}^{j}\boldsymbol{\omega}_{j}\right) \end{bmatrix} \tag{90}$$

ii) Backward recursive equations:

In this step we obtain the base acceleration using the inertial parameters of the composite link 0, where the composite link j consists of the links j, j+1, …, n.

We note that (32), giving the equilibrium equation of link j, can be rewritten using (90) as:

$$ {}^{j}\mathbf{f}_j = {}^{j}\mathbb{J}_j\, {}^{j}\dot{\mathbb{V}}_j - {}^{j}\boldsymbol{\beta}_j + \sum_{k} {}^{k}\mathbb{T}_j^{T}\, {}^{k}\mathbf{f}_k \qquad (91) $$

Applying the Newton-Euler equations on the composite link j, we obtain:

$$ {}^{j}\mathbf{f}_j = {}^{j}\mathbb{J}_j\, {}^{j}\dot{\mathbb{V}}_j - {}^{j}\boldsymbol{\beta}_j + \sum_{s(j)} {}^{s(j)}\mathbb{T}_j^{T}\left({}^{s(j)}\mathbb{J}_{s(k)}\, {}^{s(j)}\dot{\mathbb{V}}_{s(j)} - {}^{s(j)}\boldsymbol{\beta}_{s(j)} \right) \qquad (92) $$

Where s(k) means all the links succeeding joint j, that is to say joining j to any terminal link.

Substituting for ${}^{s(j)}\dot{\mathbb{V}}_{s(j)}$ in terms of ${}^{j}\dot{\mathbb{V}}_j$ using (14), we obtain:

$$ {}^{s(j)}\dot{\mathbb{V}}_{s(j)} = {}^{s(j)}\mathbb{T}_j\, {}^{j}\dot{\mathbb{V}}_j + \sum_{r} {}^{s(j)}\mathbb{T}_r\, {}^{r}\boldsymbol{\zeta}_r \qquad (93) $$

Where r denotes all links between j and s(j).

From (92), we obtain:

$$ {}^{j}\mathbf{f}_j = {}^{j}\mathbb{J}_j^{c}\, {}^{j}\dot{\mathbb{V}}_j - {}^{j}\boldsymbol{\beta}_j^{c} \qquad (94) $$

with:

$$ {}^{j}\mathbb{J}_j^{c} = {}^{j}\mathbb{J}_j^{c} + \sum_{k} {}^{k}\mathbb{T}_j^{T}\, {}^{k}\mathbb{J}_k^{c}\, {}^{k}\mathbb{T}_j \qquad (95) $$

$$ {}^{j}\boldsymbol{\beta}_j^{c} = {}^{j}\boldsymbol{\beta}_j^{c} - \sum_{k} {}^{k}\mathbb{T}_j^{T}\, {}^{k}\boldsymbol{\beta}_k^{c} + {}^{k}\mathbb{T}_j^{T}\, {}^{k}\mathbb{J}_k^{c}\, {}^{k}\boldsymbol{\zeta}_k \qquad (96) $$

${}^{j}\mathbb{J}_j^{c}$ is the inertial matrix of the composite link j.

For j = 0, and supposing ${}^{0}\mathbf{f}_0$ is equal to zero, we obtain using (94):

$$ {}^{0}\dot{\mathbb{V}}_0 = \left({}^{0}\mathbb{J}_0^{c} \right)^{-1} {}^{0}\boldsymbol{\beta}_0^{c} \qquad (97) $$

To conclude, the recursive equations of this step consist of initialising ${}^{n}\mathbb{J}_n^{c} = {}^{n}\mathbb{J}_n$, ${}^{n}\boldsymbol{\beta}_n^{c} = {}^{n}\boldsymbol{\beta}_n$ and then calculating (95)-(96) for j = n,…, 0. At the end ${}^{0}\dot{\mathbb{V}}_0$ is calculated by (97).

Comparing (97) with (68) we can deduce that $\mathbf{A}_{11}$ is equal to ${}^{0}\mathbb{J}_0^{c}$, whereas ${}^{0}\boldsymbol{\beta}_0^{c}$ is equal to $(\mathbf{H}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}})$.

iii) Forward recursive equations:

After calculating ${}^{0}\dot{\mathbb{V}}_0$, the wrench ${}^{j}\mathbf{f}_j$ and the joint torques are obtained using equations (6) and (22) for j= 1,…, n as:

$$ {}^{j}\dot{\mathbb{V}}_j = {}^{j}\mathbb{T}_i\, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\boldsymbol{\zeta}_j \qquad (98) $$

$$ {}^{j}\mathbf{f}_j = \begin{bmatrix} {}^{j}\mathbf{f}_j \\ {}^{j}\mathbf{m}_j \end{bmatrix} = {}^{j}\mathbb{J}_j^{c}\, {}^{j}\dot{\mathbb{V}}_j - {}^{j}\boldsymbol{\beta}_j^{c} \qquad (99) $$

The joint torque is calculated by projecting ${}^{j}\mathbf{f}_j$ on the joint axis, and by taking into account the friction and the actuators inertia:

$$ \Gamma_j = {}^{j}\mathbf{f}_j^{T}\, {}^{j}\mathbf{a}_j + F_{sj}\,\mathrm{sign}\left(\dot{q}_j\right) + F_{vj}\,\dot{q}_j + I_{aj}\,\ddot{q}_j \qquad (100) $$

It is to be noted that the inverse dynamic model algorithm can be used in the dynamic simulation of the mobile robot when the objective is to study the effect of the joint motions on the base. In this case the joint positions, velocities and accelerations trajectories are given. At each sampling time the acceleration of the base will be integrated to provide the angular and linear velocities for the next sampling time.

7.3 Recursive Direct Dynamic Model

The direct dynamic model consists of three recursive calculations in the same order as those of the inverse dynamic model (forward, backward and forward):

i) Forward recursive equations:

We calculate the link Cartesian velocities using (13) and the terms of Cartesian accelerations and equilibrium equations of the links that are independent of the accelerations of the base and of the joints. We calculate the following recursive equations for j = 1, ..., n:

$$ {}^{j}\boldsymbol{\gamma}_j = \begin{bmatrix} {}^{j}\mathbf{R}_i\left[{}^{i}\boldsymbol{\omega}_i \times \left({}^{i}\boldsymbol{\omega}_i \times {}^{i}\mathbf{P}_j \right) \right] + 2\sigma_j \left({}^{i}\boldsymbol{\omega}_i \times \dot{q}_j\, {}^{j}\mathbf{a}_j \right) \\ \bar{\sigma}_j\, {}^{j}\boldsymbol{\omega}_i \times \dot{q}_j\, {}^{j}\mathbf{a}_j \end{bmatrix} \qquad (101) $$

$$ {}^{j}\boldsymbol{\beta}_j = -{}^{j}\mathbf{f}_{ej} - \begin{bmatrix} {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\boldsymbol{\omega}_j \times {}^{j}\mathbf{MS}_j \right) \\ {}^{j}\boldsymbol{\omega}_j \times \left({}^{j}\mathbf{J}_j\, {}^{j}\boldsymbol{\omega}_j \right) \end{bmatrix} \qquad (102) $$

ii) Backward recursive equations:

In this second step, we first initialise ${}^{n}\mathbb{J}_n^{*} = {}^{n}\mathbb{J}_n$, ${}^{n}\boldsymbol{\beta}_n^{*} = {}^{n}\boldsymbol{\beta}_n$ and then we calculate for j = n,…,1 the following elements, which permit to calculate ${}^{j}\mathbf{f}_j$ and $\ddot{q}_j$ in terms of ${}^{i}\dot{\mathbb{V}}_i$ and will be used in the third recursive equations (these matrices can be obtained using a similar procedure as for the direct dynamic model of rigid links):

$$ H_j = {}^{j}\mathbf{a}_j^{T}\, {}^{j}\mathbb{J}_j^{*}\, {}^{j}\mathbf{a}_j + Ia_j \qquad (103) $$

$$ {}^{j}\mathbb{K}_j = {}^{j}\mathbb{J}_j^* - {}^{j}\mathbb{J}_j^* \, {}^{j}a_j H_j^{-1} \, {}^{j}a_j^T \, {}^{j}\mathbb{J}_j^* \quad (104) $$

$$ {}^{i}\mathbb{J}_i^* = {}^{i}\mathbb{J}_i + {}^{j}\mathbb{T}_i^T \, {}^{j}\mathbb{K}_j \, {}^{j}\mathbb{T}_i \quad (105) $$

$$ \tau_j = \Gamma_j - F_{sj} \, \mathrm{sign}(\dot{q}_j) - F_{vj} \, \dot{q}_j \quad (106) $$

$$ {}^{j}\boldsymbol{\alpha}_j = {}^{j}\mathbb{K}_j \, {}^{j}\boldsymbol{\gamma}_j + {}^{j}\mathbb{J}_j^* \, {}^{j}a_j H_j^{-1} \left(\tau_j + {}^{j}a_j^T \, {}^{j}\boldsymbol{\beta}_j^* \right) - {}^{j}\boldsymbol{\beta}_j^* \quad (107) $$

$$ {}^{i}\boldsymbol{\beta}_i^* = {}^{i}\boldsymbol{\beta}_i - {}^{j}\mathbb{T}_i^T \, {}^{j}\boldsymbol{\alpha}_j \quad (108) $$

iii) Forward recursive equations:

At first, the base acceleration is calculated by the following relation:

$$ {}^{0}\dot{\mathbb{V}}_0 = \left({}^{0}\mathbb{J}_0^* \right)^{-1} {}^{0}\boldsymbol{\beta}_0^* \quad (109) $$

We note that ${}^{0}\boldsymbol{\beta}_0^*$ is a function of $\boldsymbol{\tau}$, whereas ${}^{0}\boldsymbol{\beta}_0^c$ (used in the inverse model) is a function of $\ddot{\mathbf{q}}$.

$\ddot{q}_j$ and ${}^{j}f_j$ (if desired) are calculated for j=1,…,n using the following equations:

$$ \ddot{q}_j = H_j^{-1} \left[- {}^{j}a_j^T \, {}^{j}\mathbb{J}_j^* \left({}^{j}\dot{\mathbb{V}}_{j-1} + {}^{j}\boldsymbol{\gamma}_j \right) + \tau_j + {}^{j}a_j^T \, {}^{j}\boldsymbol{\beta}_j^* \right] \quad (110) $$

$$ {}^{j}f_j = {}^{j}\mathbb{K}_j \, {}^{j}\mathbb{T}_i \, {}^{i}\dot{\mathbb{V}}_i + {}^{j}\alpha_j \quad (111) $$

where:

$$ {}^{j}\dot{\mathbb{V}}_j = {}^{j}\dot{\mathbb{V}}_{j-1} + {}^{j}a_j \ddot{q}_j + {}^{j}\boldsymbol{\gamma}_j \quad (112) $$

8 CONCLUSIONS

This paper presents the inverse and direct dynamic modeling of different robotics systems. The dynamic models are developed using the recursive Newton-Euler formalism. The inverse model provides the torque of the joint and the acceleration of the free degrees of freedom such as the elastic joints, or the acceleration of the base in case of mobile base.

The direct model provides the joint acceleration of the joints including those of the free degrees of freedom.

These algorithms constitute the generalization of the algorithms of articulated manipulators to the other cases.

The proposed methods have been applied on more complicated systems such as:

- flexible link robots (Boyer and Khalil, 1998),
- Micro continuous system (Boyer, Porez and Khalil, 2006),
- hybrid structure, where the robot is composed of parallel modules, which are connected in serie, (Ibrahim, Khalil 2010).

REFERENCES

Angeles J. 2002. *Fundamentals of Robotic Mechanical Systems*. Second edition, Springer-Verlag, New York.

Armstrong W.W., 1979. Recursive solution to the equation of motion of an N-links manipulator. In *Proc. 5th World Congress on Theory of Machines and Mechanisms*, p. 1343-1346.

Boyer, F., Khalil,W, 1998. An efficient calculation of flexible manipulator inverse dynamic. In, *Int. Journal of Robotics Research,* vol. 17, No.3, pp.282-293

Boyer, F., Porez M., Khalil, W. 2006. Macro-continuous torque algorithm for a three-dimensional eel-like robot. In *IEEE Robotics transaction*, vol.22, No.4, 2006, pp.763-775.

Brandl H., Johanni R., Otter M., .1986. A very efficient algorithm for the simulation of robots and multibody systems without inversion of the mass matrix. In *Proc. IFAC Symp. on Theory of Robots*, Vienne, p. 365-370.

Craig J.J., 1986. *Introduction to robotics: mechanics and control*. Addison Wesley Publishing Company, Reading.

Featherstone R., 1983. The calculation of robot dynamics using articulated-body inertias. In the *Int. J. of Robotics Research*, Vol. 2(3), p. 87-101.

Gautier Gautier,M., Khalil W. 1990. Direct calculation of minimum set of inertial parameters of serial robots. In *IEEE Trans. on Robotics and Automation*, Vol. RA-6(3), p. 368-373.

Ibrahim, O., Khalil, W, 2010. Inverse and direct dynamic models of Hybride robots. In *Mechanism and machine theory,* Volume 45, Issue 4, p. 627-640.

Luh J.Y.S., Walker M.W., Paul R.C.P.,1980. On-line computational scheme for mechanical manipulators. In *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 102(2) p. 69-76.

Khalil W., Kleinfinger J.-F., 1986. A new geometric notation for open and closed-loop robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, p. 1174-1180.

Khalil W., Kleinfinger J.-F., 1987. Minimum operations and minimum parameters of the dynamic model of tree structure robots. In *IEEE J. of Robotics and Automation*, Vol. RA-3(6), p. 517-526.

Khalil W., Bennis F., 1994. Comments on Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. In *IEEE Trans. on Rob.& Automation*, Vol. RA-10(1), p. 78-79.

Khalil W., Creusot D., 1997. SYMORO+: a system for the symbolic modelling of robots. In *Robotica*, Vol. 15, p. 153-161.

Khalil W., Gautier M., 2000. Modeling of mechanical systems with lumped elasticity", In *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, p. 3965-3970.

Khalil, W., Dombre, E. 2002. *Modeling identification and control of robots*. Hermes, Penton-Sciences, London.

Khalil W. and Guegan S.,2004. Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. In

IEEE Transactions on Robotics and Automation, 20(4), p. 754-762.

Khalil W., G. Gallot G., Boyer F., 2007. Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot. In *IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and reviews, Vol. 37, N° 6.*

Khalil W., Ibrahim O., 2007**.** General solution for the Dynamic modeling of parallel robots. In *Journal of Intelligent and Robotic Systems*, Vol.49, pp.19-37.

Khosla P.K., 1986. Real-time control and identification of direct drive manipulators. Ph. D. Thesis, Carnegie Mellon.

Walker M.W., Orin D.E., 1982. Efficient dynamic computer simulation of robotics mechanism. In *Trans. of ASME, J. of Dynamic Systems, Measurement, and Control*, Vol. 104, p. 205-211.

BRIEF BIOGRAPHY

Wisama Khalil received the Ph.D. and the "Doctorat d'Etat" degrees in robotics and control engineering from the University of Montpellier, France, in 1976 and 1978, respectively. Since 1983, he has been a Professor at the Automatic Control and Robotics Department, Ecole Centrale de Nantes, France. He is the coordinator of Erasmus Mundus master course EMARO "European Master in Advanced Robotics". He is carrying out his research within the Robotics team, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN). His current research interests include modeling, control, and identification of robots. He has more than 100 publictions in journals and international conferences.

EMOTIVE DRIVER ADVISORY SYSTEM

Oleg Gusikhin
Ford Motor Company, Research and Innovation Center
2101 Village Road, Dearborn, MI 48121, U.S.A.

EXTENDED ABSTRACT

In 2007, Ford, in cooperation with Microsoft, introduced an in-car communication and entertainment system, SYNC. This system enables Bluetooth and USB connectivity for consumer phones and MP3 players and allows hands-free voice-activated control of brought-in devices. Since its initial introduction, there has been rapid growth of SYNC-enabled services, such as remote monitoring of vehicle health, personalized traffic reports, weather, news, and turn-by-turn directions utilizing data-over-voice technology. Furthermore, SYNC takes advantage of existing networking capabilities of smart phones/PDAs by providing a SYNC API to mobile application developers.

The Emotive Driver Advisory System (EDAS) is a Ford Research project that fills the technology pipeline for future SYNC versions, exploiting advances in information technology and consumer electronics to enhance the driver's experience. EDAS was inspired by recent developments in affective computing, open mic grammar-based speech recognition, embodied conversational agents, and humanoid robotics focusing on personalization and context-aware adaptive and intelligent behavior. The EDAS concept was revealed at the 2009 Consumer Electronics Show and the 2009 North American International Auto Show as EVA, Emotive Voice Activation.

The core elements of EDAS include an emotive and natural spoken dialogue system and an AVATAR-based visual interface integrated with adaptive vehicle controls and cloud-based infotainment. The system connects the vehicle, the driver, and the environment, while providing the dialogue strategy best suited for the given driving context and emotive status of the driver.

Voice interaction is the prevalent method for the driver to interface with vehicle systems for hands-free, eyes-free communication. The effectiveness of such communication depends on the quality and sophistication of both speech recognition and speech generation. The EDAS spoken dialogue system allows recognition of the driver's commands in an open mic, natural, non-hierarchical manner. In turn, the system response depends on the driving environment, as well as the driver's status. The responses can be more extensive and engaging in open road conditions, while concise in high traffic situations. The ability to recognize the driver's emotions and generate emotions in response can further improve such communication. The spoken interface is augmented by the AVATAR as a universal intelligent gauge. The AVATAR supplements the emotive intent in delivering system messages, as well as providing non-verbal cues into the status of the active task.

The system leverages cloud-based infotainment, allowing for personalized, context-aware and interactive delivery of infotainment services. The ability to maintain connectivity between vehicle systems and the internet not only gives access to a vast amount of up-to-date information, but also allows outsourcing of computationally intensive tasks to a remote server, tapping into the power of cloud computing. Specifically, we demonstrate how EDAS enhances four most common in-vehicle infotainment activities: points of interest, news radio, music and refueling notification and advice.

BRIEF BIOGRAPHY

Dr. Oleg Gusikhin is a Technical Leader at Ford Manufacturing, Vehicle Design and Safety Research Laboratory. He received his Ph.D. from the St. Petersburg Institute of Informatics and Automation of Russian Academy of Sciences and an MBA from the Ross Business School at the University of Michigan. For over 15 years, he has been working at Ford Motor Company in different functional areas including Information Technology, Advanced Electronics Manufacturing, and Research & Advanced Engineering. During his tenure at Ford, Dr. Gusikhin has been involved in the design and implementation of advanced information technology and intelligent controls for manufacturing and vehicle systems. Dr. Gusikhin is a recipient of 2004 Henry Ford Technology Award and two Ford

Research and Advanced Engineering Technical Achievement Awards. He holds 2 patens and is a co-author of 8 patent applications on advanced vehicle infotainment technology.

FINGERTIP FORCE MEASUREMENT BY IMAGING THE FINGERNAIL

John Hollerbach
University of Utah, U.S.A.

Abstract: Shear and normal forces from fingertip contact with a surface are measured by external camera images of the fingernail. Due to mechanical interaction between the surface, fingertip bone, and fingernail, regions of tension or compression are set up that result in reddening or whitening due to blood flow. The effect is quantitative enough to serve as a transducer of fingertip force. Due to individual differences, calibration is required for the highest accuracy. Automated calibration is achieved by use of a magnetically levitated haptic interface probe.

BRIEF BIOGRAPHY

John M. Hollerbach is Professor of Computing, and Research Professor of Mechanical Engineering, at the University of Utah. He also directs the Robotics Track, a joint graduate program between the School of Computing and Department of Mechanical Engineering. From 1989-1994 he was the Natural Sciences and Engineering/Canadian Institute for Advanced Research Professor of Robotics at McGill University, jointly in the Departments of Mechanical Engineering and Biomedical Engineering. From 1982-1989 he was on the faculty of the Department of Brain and Cognitive Sciences and a member of the Artificial Intelligence Laboratory at MIT; from 1978-1982 he was a Research Scientist. He received his BS in chemistry ('68) and MS in mathematics ('69) from the University of Michigan, and SM ('75) and PhD ('78) from MIT in Computer Science. He is presently the Vice President for Technical Activities of the IEEE Robotics and Automation Society, and Editor of the International Journal of Robotics Research.

SIGNAL PROCESSING, SYSTEMS MODELING AND CONTROL

FULL PAPERS

LINEARIZING CONTROL OF YEAST AND BACTERIA FED-BATCH CULTURES

A Comparison of Adaptive and Robust Strategies

Laurent Dewasme, Alain Vande Wouwer
Service d'Automatique, Université de Mons, 31 Boulevard Dolez, 7000 Mons, Belgium
{laurent.dewasme, Alain.VandeWouwer}@umons.ac.be

Daniel Coutinho
Group of Automation and Control Systems, PUCRS, Av. Ipiranga 6681, 90619-900, Porto Alegre, Brazil
dcoutinho@pucrs.br

Keywords: Nonlinear robust control, Adaptive control, Fermentation process, Biotechnology.

Abstract: Linearizing control is a popular approach to control bioprocesses, which has received considerable attention is the past several years. This control approach is however quite sensitive to modeling uncertainties, thus requiring some on-line parametric adaptation so as to ensure performance. In this study, this usual adaptive strategy is compared in terms of implementation and performance to a robust strategy, where the controller has a fixed parametrization which is determined using a LMI framework so as to ensure robust stability and performance. Fed-batch cultures of yeast and bacteria are considered as application examples.

1 INTRODUCTION

The culture of host recombinant micro-organisms is nowadays a very important way of producing biopharmaceuticals. Fed-batch operation is popular in industrial practice, since it is advantageous from an operational and control point of view. The off-line determination of the feeding profile is usually sub-optimal as some security margin has to be provided in order to avoid an excess of substrate leading to the accumulation of inhibitory by-products (inhibition of the cell respiratory capacity), namely ethanol for yeast cultures and acetate for bacteria cultures.

To optimize the culture conditions and to avoid high concentrations of inhibitory by-products, a closed-loop solution is required, and a wide diversity of approaches, e.g., (Pomerleau, 1990; Chen et al., 1995; Rocha, 2003; Renard and Wouwer, 2008; Dewasme et al., 2009a; Dewasme et al., 2009b) have been considered.

In particular, linearizing control (Bastin and Dochain, 1990) is a very popular approach, which has been applied successfully in a number of case studies. However, linearizing control requires the knowledge of an accurate model, and on-line parametric adaptation is usually implemented so as to ensure performance. Whereas parametric adaptation is a simple approach, it does not guarantee stability in the presence of unmodeled dynamics.

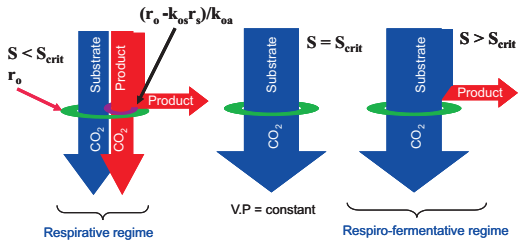


Figure 1: Illustration of Sonnleitner's bottleneck assumption for cells limited respiratory capacity.

In this study, another approach is also considered, which is based on nonlinear robust control and the used of Linear Matrix Inequalities (LMIs) to design the free linear dynamics so as to ensure robust stability and performance. A comparison of the adaptive and robust control approaches is provided in terms of implementation, and simulation tests shows the respective advantages and limitations of both strategies.

2 MECHANISTIC MODEL

In this study, we consider a generic model that would, in principle, allow the representation of the culture of different strains presenting an overflow metabolism (yeasts, bacteria, animal cells, etc). This model describes therefore the cell catabolism through the following three main reactions:

Substrate oxidation :

$$k_{S1}\mathrm{S} + k_{O1}\mathrm{O} \xrightarrow{r_1 X} k_{X1}\mathrm{X} + k_{C1}\mathrm{C} \quad (1a)$$

Overflow reaction (typically fermentation) :

$$k_{S2}\mathrm{S} + k_{O2}\mathrm{O} \xrightarrow{r_2 X} k_{X2}\mathrm{X} + k_{P2}\mathrm{P} + k_{C2}\mathrm{C} \quad (1b)$$

Metabolite product oxidation :

$$k_{P3}\mathrm{P} + k_{O3}\mathrm{O} \xrightarrow{r_3 X} k_{X3}\mathrm{X} + k_{C3}\mathrm{C} \quad (1c)$$

where X, S, P, O and C are, respectively, the concentration in the culture medium of biomass, substrate (typically glucose or glycerol), product (i.e. ethanol or methanol in yeast cultures, acetate in bacteria cultures or lactate in animal cells cultures), dissolved oxygen and carbon dioxide. $k_{\xi i}$ (i=1,2,3) are the yield coefficients and r_1, r_2 and r_3 are the nonlinear specific growth rates given by:

$$r_1 = \frac{\min(r_S, r_{S_{crit}})}{k_{S1}} \quad (2)$$

$$r_2 = \frac{\max(0, r_S - r_{S_{crit}})}{k_{S2}} \quad (3)$$

$$r_3 = \frac{\max\left(0, \min\left(r_P, \frac{k_{os}(r_{S_{crit}} - r_S)}{k_{oa}}\right)\right)}{k_{P3}} \quad (4)$$

where the kinetic terms associated with the substrate consumption r_S, the critical substrate consumption $r_{S_{crit}}$ (generally dependent on the cells oxidative or respiratory capacity r_O) and the product oxidative rate r_P are given by:

$$r_S = \mu_S \frac{\mathrm{S}}{\mathrm{S} + K_S} \quad (5a)$$

$$r_{S_{crit}} = \frac{r_O}{k_{os}} = \frac{\mu_O}{k_{os}} \frac{\mathrm{O}}{\mathrm{O} + K_O} \frac{Ki_P}{Ki_P + P} \quad (5b)$$

$$r_P = \mu_P \frac{\mathrm{P}}{\mathrm{P} + K_P} \quad (5c)$$

These expressions take the classical form of Monod laws where μ_S, μ_O and μ_P are the maximal values of specific growth rates, K_S, K_O and K_P are the saturation constants of the corresponding element, and Ki_P is the inhibition constant. k_{os} and k_{oa} represent the coefficients characterizing respectively the yield between the oxygen and substrate consumptions, and the yield between the acetate and oxygen consumptions.

This kinetic model is based on Sonnleitner's bottleneck assumption (Sonnleitner and Käppeli, 1986) which was developed for a yeast strain *Saccharomyces cerevisiae* (Figure 1). During a culture, the cells are likely to change their metabolism because of their limited respiratory capacity. When the substrate is in excess (concentration $S > S_{crit}$), the cells produce a metabolite product P through fermentation, and the culture is said in respiro-fermentative (RF) regime. On the other hand, when the substrate becomes limiting (concentration $S < S_{crit}$), the available substrate (typically glucose), and possibly the metabolite P (as a substitute carbon source), if present in the culture medium, are oxidized. The culture is then said in respirative (R) regime.

Component-wise mass balances give the following differential equations :

$$\frac{dX}{dt} = (k_{X1}r_1 + k_{X2}r_2 + k_{X3}r_3)X - DX \quad (6a)$$

$$\frac{dS}{dt} = -(k_{S1}r_1 + k_{S2}r_2)X + DS_{in} - DS \quad (6b)$$

$$\frac{dP}{dt} = (k_{P2}r_2 - k_{P3}r_3)X - DP \quad (6c)$$

$$\frac{dO}{dt} = -(k_{O1}r_1 + k_{O2}r_2 + k_{O3}r_3)X - DO + OTR \quad (6d)$$

$$\frac{dC}{dt} = (k_{C1}r_1 + k_{C2}r_2 + k_{C3}r_3)X - DC - CTR \quad (6e)$$

$$\frac{dV}{dt} = F_{in} \quad (6f)$$

where S_{in} is the substrate concentration in the feed, F_{in} is the inlet feed rate, V is the culture medium volume and D is the dilution rate ($D = F_{in}/V$). OTR and CTR represent respectively the oxygen transfer rate from the gas phase to the liquid phase and the carbon transfer rate from the liquid phase to the gas phase. Classical models of OTR and CTR are given by:

$$OTR = k_L a(O_{sat} - O) \quad (7a)$$

$$CTR = k_L a(P - P_{sat}) \quad (7b)$$

where $k_L a$ is the volumetric transfer coefficient and, O_{sat} and P_{sat} are respectively the dissolved oxygen and carbon dioxide concentrations at saturation.

3 A SUBOPTIMAL STRATEGY

The maximum of productivity is obtained at the edge between the respirative and respiro-fermentative

regimes, where the quantity of by-product is constant and equal to zero ($VP = 0$). Unfortunately, evaluating accurately the volume is a difficult task as it depends on the inlet and outlet flows including F_{in} but also the added base quantity for pH control and several gas flow rates. Moreover, maintaining the quantity of by-product constant in a fed-batch process means that the by-product concentration has to decrease while the volume increases. So, even if the volume is correctly measured, VP becomes unmeasurable once P reaches the sensitivity level of the by-product probe. For those practical limitations, a sub-optimal strategy is elaborated through the control of the by-product concentration around a low value P^* depending on the sensitivity of commercially available probes (for instance, a general order for ethanol probe is $0.1 g/l$), and requiring only an estimation of the volume by integration of the feed rate.

The basic principle of the controller is thus to regulate the by-product at a constant low setpoint, leading to a self-optimizing control in the sense of (Skogestad, 2004) and ensuring that the culture operates in the respiro-fermentative regime, close to the biological optimum, i.e., close to the edge with the respirative regime.

4 LINEARIZING CONTROL STRATEGY

The component-wise mass balances of reaction scheme (1) lead to the following state-space representation

$$\dot{x} = Kr(x)X + Ax - ux + B(u) \tag{8}$$

where $x = [\ X \quad S \quad P \quad O \quad C \quad V\ ]'$ is the state vector, $r(x) = [\ r_1 \quad r_2 \quad r_3\ ]'$ is the vector of reaction rates, and $u = D = F_{in}/V$ is the control input (the dilution rate). The matrices K and A, and the vector function $B(\cdot)$ are given by:

$$K = \begin{bmatrix} k_{X1} & k_{X2} & k_{X3} \\ -k_{S1} & -k_{S2} & 0 \\ 0 & k_{P2} & -k_{P3} \\ -k_{O1} & -k_{O2} & -k_{O3} \\ k_{C1} & k_{C2} & k_{C3} \\ 0 & 0 & 0 \end{bmatrix}, \; B(u) = \begin{bmatrix} 0 \\ S_{in}\, u \\ 0 \\ k_L a\, O_{sat} \\ k_L a\, P_{sat} \\ 0 \end{bmatrix}, \tag{9}$$

$$A = \begin{bmatrix} 0_{3\times3} & 0_{3\times2} & 0_{3\times1} \\ 0_{2\times2} & -k_L a\, I_{2\times2} & 0_{2\times2} \\ 0_{1\times3} & 0_{1\times2} & 0 \end{bmatrix},$$

A feedback linearizing controller is illustrated in Figure 2. In a first step, this controller is derived assuming a perfect process knowledge. The basic idea

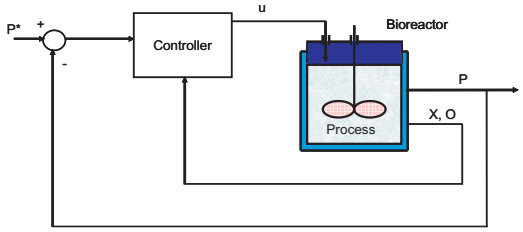


Figure 2: Linearizing control scheme.

is to derive a nonlinear controller, which allows a linearization of the process behavior ((Chen et al., 1995; Pomerleau, 1990)).

As the theoretical value of S_{crit} is very small (below $0.1\ g/l$) and assuming a quasi-steady state of S (i.e. considering that there is no accumulation of glucose when operating the bioreactor in the neighborhood of the optimal operating conditions), the small quantity of substrate VS is almost instantaneously consumed by the cells ($\frac{d(VS)}{dt} \approx 0$ and $S \approx 0$) and (6b) becomes:

$$k_{S2} r_2 X = -k_{S1} r_1 X + S_{in} u \tag{10}$$

where r_1 and r_2 are nonlinear functions of S, P and O as given by (2-3).

Replacing $r_2 X$ by (10) in the mass balance equation for P (6c), we obtain:

$$\dot{P} = -\frac{k_{P2} k_{S1}}{k_{S2}} r_1 X - k_{P3} r_3 X - u\left(P - \frac{k_{P2}}{k_{S2}} S_{in}\right) \tag{11}$$

A first-order linear reference model is imposed:

$$\frac{d(P^* - P)}{dt} = -\lambda(P^* - P)\ ,\ \lambda > 0 \tag{12}$$

and a constant setpoint is considered so that:

$$\frac{dP}{dt} = \lambda(P^* - P)\ ,\ \lambda > 0 \tag{13}$$

Equating (13) and (11), the following control law is obtained:

$$F_{in} = V \frac{\lambda(P^* - P) + (\frac{k_{P2} k_{S1}}{k_{S2}} r_1 + k_{P3} r_3) X}{\frac{k_{P2}}{k_{S2}} S_{in} - P} \tag{14}$$

where $\frac{k_{P2} k_{S1}}{k_{S2}} r_1$ and $k_{P3} r_3$, the kinetic expressions, contain several uncertain parameters.

4.1 A Classical Adaptive Strategy

In (Chen et al., 1995), the parameter uncertainties are handled using an on-line estimation of the kinetic term $\frac{k_{P2} k_{S1}}{k_{S2}} r_1 + k_{P3} r_3$ in the linearizing control law (14). In this study, the biomass concentration X is supposed to be measured using a probe (for instance

a optical density probe or a conductance probe, which are nowadays widely available), whereas in (Chen et al., 1995), an asymptotic observer is used to estimate this component concentration. The following adaptive scheme is therefore a simplified version of the original algorithm.

$$F_{in} = V\frac{\lambda(P^* - P) + \hat{\theta}X}{\frac{k_{P2}}{k_{S2}}S_{in} - P} \tag{15}$$

A direct adaptive scheme as described in (Bastin and Dochain, 1990) is used. Consider the following Lyapunov function candidate:

$$V(t) = \frac{1}{2}\left(\tilde{P}^2 + \frac{\tilde{\theta}^2}{\gamma}\right) \tag{16}$$

where $\tilde{P} = P^* - P$, $\tilde{\theta} = \theta - \hat{\theta}$ and γ is a strictly positive scalar. The specific growth rates r_1 and r_3 (and, of course, the pseudo-stoichiometric coefficient k_4) are assumed to be constant so that θ variations are negligible ($\frac{d\theta}{dt} = 0$).

Using the Lyapunov stability theory, the time derivative of the Lyapunov candidate function should be negative for the closed-loop system to be stable:

$$\frac{dV}{dt} = \frac{d\tilde{P}}{dt}\tilde{P} + \tilde{\theta}\frac{d\tilde{\theta}}{dt}\frac{1}{\gamma} \tag{17}$$

Considering (13) and a possible parameter mismatch ($\hat{\theta} \neq \theta$):

$$\frac{d\tilde{P}}{dt} = -\lambda\tilde{P} - \tilde{\theta}X \tag{18}$$

so that (17) becomes:

$$\frac{dV}{dt} = -\lambda\tilde{P}^2 - \tilde{P}\tilde{\theta}X - \tilde{\theta}\frac{d\hat{\theta}}{dt}\frac{1}{\gamma} \tag{19}$$

Choosing the following θ adaptive law cancels the second and the third terms:

$$\frac{d\hat{\theta}}{dt} = \gamma X\tilde{P} \tag{20}$$

4.2 A Robust Strategy

Structural and parametric uncertainties can be lumped into a global parametric error:

$$\delta = \bar{\theta} - \theta \tag{21}$$

where δ is a nonlinear function of (S,P,O) representing possible inexact cancellations of nonlinear terms due to model uncertainties and $\bar{\theta}$ represents the hypothetical exact unknown value. Rewriting the kinetic term in (15) using the new expression taken from (21), we obtain:

$$u = F_{in} = V\frac{\lambda(P^* - P) + \bar{\theta}X - \delta X}{\frac{k_{P2}}{k_{S2}}S_{in} - P} \tag{22}$$

which corresponds to the perturbed reference system:

$$\dot{P} = \lambda(P^* - P) - \delta X \tag{23}$$

Borrowing the ideas of the *Quasi-LPV* approach (Leith and Leithead, 2000), we bound the time-varying parameter δ which is supposed to belong to a known set $\Delta := \{\delta : \underline{\delta} \leq \delta \leq \overline{\delta}\}$ with $\underline{\delta}$ and $\overline{\delta}$ respectively representing the minimal and maximal admissible uncertainties.

The parameter λ is designed to ensure some robustness and tracking performance to the overall closed-loop system, which is modeled as follows:

$$\mathcal{M} : \left\{ \begin{array}{rcl} \dot{P} & = & -\lambda z - \delta X \\ z & = & P^* - P \end{array} \right. \tag{24}$$

where $z = P^* - P$ is the performance output.

Let $w = \left[\begin{array}{ccc} P^* & X \end{array}\right]' \subset \mathcal{L}_{2,[0,T]}$ be the disturbance input to the system $\mathcal{M}$, $a(\lambda,\delta) = \left[\begin{array}{cc} \lambda & -\delta \end{array}\right]$ and $c = \left[\begin{array}{cc} 1 & 0 \end{array}\right]$. The closed-loop system (24) can be rewritten:

$$\mathcal{M} : \left\{ \begin{array}{rcl} \dot{P} & = & -\lambda P + a(\lambda,\delta)w \\ z & = & -P + c\,w\,,\ \delta \in \Delta \end{array} \right. \tag{25}$$

Consider the finite horizon (for instance, between the instant 0 and the time T) $\mathcal{L}_2$-gain of system $\mathcal{M}$ (M. Green and D.J.N. Limebeer, 1994), representing the worst-case of the ratio of $\|z\|_{2,[0,T]}$ (i.e., the finite horizon 2-norm of the tracking error) and $\|w\|_{2,[0,T]}$ (i.e., the finite horizon 2-norm of the disturbance input), which is defined as:

$$\|\mathcal{M}_{wz}\|_{\infty,[0,T]} = \sup_{\delta\in\Delta, 0\neq w\subset\mathcal{L}_{2,[0,T]}} \frac{\|z\|_{2,[0,T]}}{\|w\|_{2,[0,T]}} \tag{26}$$

Thus, the parameter λ is designed based on the $\mathcal{H}_\infty$ control theory (M. Green and D.J.N. Limebeer, 1994; Skogestad and Postlethwaite, 2001). Let $\alpha > 0$ be an upper limiting of $\|\mathcal{M}_{wz}\|_{\infty,[0,T]}$. Thus, the problem is to find α such that:

$$\min_{\lambda,\delta\in\Delta} \alpha : \|\mathcal{M}_{wz}\|_{\infty,[0,T]} \leq \alpha \tag{27}$$

while ensuring the robust stability of system (25).

This optimization problem can be written in terms of linear matrix inequalities (*LMIs*) and solved using readily available toolboxes, e.g., SeDuMi (Sturm et al., 2006) can be applied to solve the problem. These constraints can be easily obtained via a quadratic Lyapunov function (S.Boyd, L.El-Ghaoui, E.Feron and V.Balakrishnan, 1994)

$$V(P) = P'QP = QP^2 \tag{28}$$

where Q is a strictly positive symmetric matrix (i.e., $Q = Q' \succ 0$) and " $'$ " corresponds to the transposition matrix operation.

The minimization in (27) is then equivalent to:

$$\min \alpha \ : \ V(P) \succ 0 \ , \ \dot{V}(P) + \frac{1}{\alpha} z'z - \alpha w'w \prec 0 \quad (29)$$

where, using (25) and (28), the time derivative of $V(P)$ is given by:

$$\begin{aligned} \dot{V}(P) &= \dot{P}'QP + P'Q\dot{P} \\ &= (-\lambda P + aw)'QP + P'Q(-\lambda P + aw) \\ &= -\lambda P'QP + (aw)'QP - \lambda P'QP + P'Qaw \\ &= -2\lambda P'QP + a'w'QP + P'Qaw \qquad (30) \end{aligned}$$

Using (30) in (29), the following expression is obtained:

$$\begin{bmatrix} P \\ w \end{bmatrix}' \begin{bmatrix} -2m & Qa \\ a'Q & -\alpha I_{n_w} \end{bmatrix} \begin{bmatrix} P \\ w \end{bmatrix} - \frac{1}{\alpha} zz' \prec 0 \quad (31)$$

where $m = \lambda Q$ and I_{n_w} is the unity matrix of dimension $n_w \times n_w$ and n_w is the dimension of w.

Now, consider the following lemma (*Schur Complement*):

Lemma 1. The following matrix inequalities are equivalent

$$\begin{aligned} &(i) \quad T > 0, R - ST^{-1}S' \succ 0 \\ &(ii) \quad R > 0, T - S'R^{-1}S \succ 0 \\ &(iii) \quad \begin{bmatrix} R & S \\ S' & T \end{bmatrix} \succ 0 \end{aligned}$$

Hence, using the expression of z, a and c in (25) and Lemma 1, the optimization problem in (27) can be written as follows:

$$\min_{Q,m} \alpha \ : \ \alpha > 0 \ , \ Q = Q' > 0 \ \text{ and }$$
$$\begin{bmatrix} -2m & m & -\delta Q & -1 \\ m & -\alpha & 0 & 1 \\ -\delta Q & 0 & -\alpha & 0 \\ -1 & 1 & 0 & -\alpha \end{bmatrix} \prec 0 \qquad (32)$$

If there exists a feasible solution to the above optimization problem for all δ evaluated at the vertices of Δ, then (27) is satisfied and $\lambda = mQ^{-1}$.

Remark 1. Quadratic Lyapunov functions may be conservative for assessing the stability of parameter-dependent systems (G. Chesi and Vicino, 2004). However, a parameter-independent Lyapunov function is considered in this study for two main reasons:

1. λ is parametrized with the Lyapunov matrix Q so as to obtain a convex design condition. A parameter-independent matrix Q therefore results in a parameter-independent control law;
2. the variation of δ is a priori unknown.

Remark 2. This method is likely to be conservative, as the parameter δ has to bound the nonlinearities of the inexactly cancelled terms. Less conservative results can be obtained by considering the approach of (D.F. Coutinho, M. Fu, A. Trofino and P. Danès, 2008) to deal with the nonlinearities at the cost of a larger computational effort.

5 NUMERICAL RESULTS

In this section, for comparing the adaptive and robust linearizing control strategies, several numerical simulations considering small-scale bacteria and yeast cultures (respectively in 5 and 20 $[l]$ bioreactors) are performed. The first simulation set is dedicated to yeast cultures with initial and operating conditions: $X_0 = 0.4 g/l$, $S_0 = 0.5 g/l$, $E_0 = 0.8 g/l$, $O_0 = O_{sat} = 0.035 g/l$, $C_0 = C_{sat} = 1.286 g/l$, $V_0 = 6.8 l$, $S_{in} = 350 g/l$. The second simulation set is dedicated to bacteria cultures with initial and operating conditions: $X_0 = 0.4 g/l$, $S_0 = 0.05 g/l$, $A_0 = 0.8 g/l$, $O_0 = O_{sat} = 0.035 g/l$, $C_0 = C_{sat} = 1.286 g/l$, $V_0 = 3.5 l$, $S_{in} = 250 g/l$

The values of all model parameters are listed in Tables 1, 2, 3 and 4. Note that, for yeast cultures, coefficients k_{os} and k_{oa} are simply replaced by k_{O1} and k_{03} while $k_{O2} = 0$, in accordance with the model of (Sonnleitner and Käppeli, 1986). For the bacteria model, parameters values are taken from (Rocha, 2003) and slightly modified to adapt the yield coefficient normalization to the proposed reaction scheme (1) and kinetic model (with a slight difference in the formulation of r_3).

The state variables are assumed available (i.e., measured) online for feedback. The adaptive and robust linearizing feedback controllers proposed in section 4 aim at tracking the byproduct set-point (E^* and $A^* = 1\ g/l$) which is chosen sufficiently low so as to stay in the neighborhood of the optimal trajectory but also sufficiently high to avoid probe sensitivity limitations. In this setup, a noisy byproduct measurement is considered.

To design the parameter λ in (23) via the optimization problem (27), the parameters K_S, K_P, K_O, K_{iP} and μ_S, μ_O are assumed to be respectively varying of $\pm 100\%$ and $\pm 15\%$ from their nominal values. Simulating the operating conditions of the control strategy in (22), we may infer that $\overline{\delta} = -\underline{\delta} = 0.5/3600\ s^{-1}$ for yeast cultures and $\overline{\delta} = -\underline{\delta} = 0.1/3600\ s^{-1}$ for bacteria cultures. In light of (25) and (27), these constraints yield for yeasts and bacteria, respectively to $\lambda = 0.0056$ and $\lambda = 0.0046$.

Concerning the adaptive control law, $\lambda = 1$ and

Table 1: Yield coefficients values of Sonnleitner and Käppeli for *S. cerevisiae* model (Sonnleitner and Käppeli, 1986)

Yield coefficients	Values	Units
k_{X1}	0,49	$g\,of\,X/g\,of\,S$
k_{X2}	0,05	$g\,of\,X/g\,of\,S$
k_{X3}	0,72	$g\,of\,X/g\,of\,E$
k_{S1}	1	
k_{S2}	1	
k_{P2}	0,48	$g\,of\,E/g\,of\,S$
k_{P3}	1	
k_{O1}	0,3968	$g\,of\,O_2/g\,of\,S$
k_{O2}	0	$g\,of\,O_2/g\,of\,S$
k_{O3}	1,104	$g\,of\,O_2/g\,of\,E$
k_{C1}	0,5897	$g\,of\,CO_2/g\,of\,S$
k_{C2}	0,4621	$g\,of\,CO_2/g\,of\,S$
k_{C3}	0,6249	$g\,of\,CO_2/g\,of\,E$

Table 2: Kinetic coefficients values of Sonnleitner and Käppeli for the *S. cerevisiae* model (Sonnleitner and Käppeli, 1986)

Kinetic coefficients	Values	Units
μ_O	0,256	$g\,of\,O_2/g\,of\,X\ /h$
μ_S	3,5	$g\,of\,S/g\,of\,X\ /h$
K_O	0,0001	$g\,of\,O_2/l$
K_S	0,1	$g\,of\,S/l$
K_E	0,1	$g\,of\,E/l$
Ki_E	10	$g\,of\,E/l$

$\gamma = 0.05$ for yeast cultures while $\lambda = 2$ and $\gamma = 0.25$ for bacteria cultures. Note also that the sampling period is chosen equal to 0.1 *h*.

Before discussing the results of the proposed methods, it is interesting to observe the performance of a plain linearizing controller, i.e. without adaptation or robustification, applied to the yeast process in the presence of modeling errors. For instance, consider the situation where the user selects a relatively high gain $\lambda = 1$, and $\hat{\theta}$ is fixed to $k_{P2}/2$. Figure 3 illustrates the consequences of such choices. Even if the controller behaves correctly during the first hours, the divergence of the ethanol signal during the last hours will impact the quality of the culture.

Figure 4 shows now the closed-loop response of biomass X, ethanol E concentrations, and the inlet feed rate F_{in}, for five different values of the kinetic parameters (which were randomly chosen) in yeast cultures under a robust control strategy. In all simulation runs, a white noise is added to the ethanol concentration measurement with a standard deviation of $\pm 0.1\ [g/l]$ and the culture is considered as always evolving in the optimal operating conditions in which $r_1 = \frac{r_O}{k_{O1}}$ and $r_3 = 0$ so that the hypothetical parameter

Table 3: Yield coefficients values of Rocha's *E.coli* model (Rocha, 2003)

Yield coefficients	Values	Units
k_{X1}	1	
k_{X2}	1	
k_{X3}	1	
k_{S1}	0,316	$g\,of\,S/g\,of\,X$
k_{S2}	0,04	$g\,of\,S/g\,of\,X$
k_{P2}	0,157	$g\,of\,A/g\,of\,X$
k_{P3}	0,432	$g\,of\,A/g\,of\,X$
k_{O1}	0,339	$g\,of\,O_2/g\,of\,X$
k_{O2}	0,471	$g\,of\,O_2/g\,of\,X$
k_{O3}	0,955	$g\,of\,O_2/g\,of\,X$
k_{C1}	0,405	$g\,of\,CO_2/g\,of\,X$
k_{C2}	0,754	$g\,of\,CO_2/g\,of\,X$
k_{C3}	1,03	$g\,of\,CO_2/g\,of\,X$
k_{os}	2,02	$g\,of\,O_2/g\,of\,X$
k_{oa}	1,996	$g\,of\,O_2/g\,of\,X$

Table 4: Kinetic coefficients values of Rocha's *E.coli* model (Rocha, 2003)

Kinetic coefficients	Values	Units
μ_O	0,7218	$g\,of\,O_2/g\,of\,X\ /h$
μ_S	1,832	$g\,of\,S/g\,of\,X\ /h$
K_O	0,0001	$g\,of\,O_2/l$
K_S	0,1428	$g\,of\,S/l$
K_A	0,5236	$g\,of\,A/l$
Ki_A	6,952	$g\,of\,A/l$

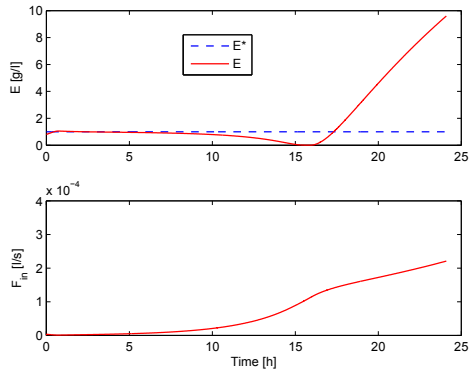


Figure 3: Yeast cultures – ethanol concentration and feed rate when the controller is designed using a plain linearizing control approach (no adaptation and no robustification) in the presence of modeling errors.

$\bar{\theta}$ in (22) is taken as

$$\bar{\theta} = \frac{k_{P2}\tilde{k}_{S1}}{k_{S2}} r_1 + k_{P3}\tilde{r}_3 \approx \frac{\frac{k_{P2}k_{S1}}{k_{S2}} r_O}{k_{O1}} \qquad (33)$$

Figure 4 shows that during the start-up phase, F_{in} saturates to 0, leading to an ethanol overshoot (see

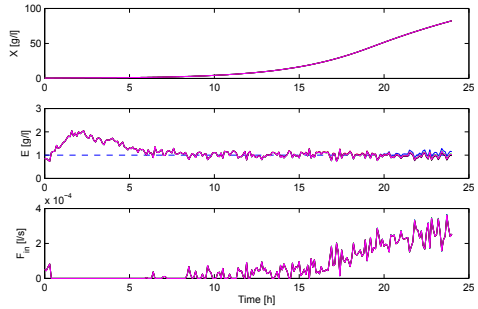


Figure 4: Yeast cultures – biomass and ethanol concentrations, and feed rate – robust control strategy – results of 5 runs with random parameter variations and a noise standard deviation of ± 0.1 $[g/l]$.

Figure 4). The different curves are more or less indistinguishable (the same noise signal is applied during the 5 runs) except in the last hours where the consequences of model errors appear. Nevertheless, these results are very satisfactory as model errors have a negligible influence.

Figures 5 and 6 show the results of a simulation performed with the same initial and operating conditions with the adaptive strategy, in the ideal case where there is no measurement noise, whereas Figures 7 and 8 correspond to a noise standard deviation of ± 0.05 $[g/l]$ added to the ethanol concentration measurements. Due to sensitivity problems of the adaptive law, higher noise levels usually lead to computational failures. When the parameter adaptation performs well, the productivity of the adaptive and robust strategies is more or less the same, i.e., a biomass concentration of approximately 80 g/l is obtained within 24 hours.

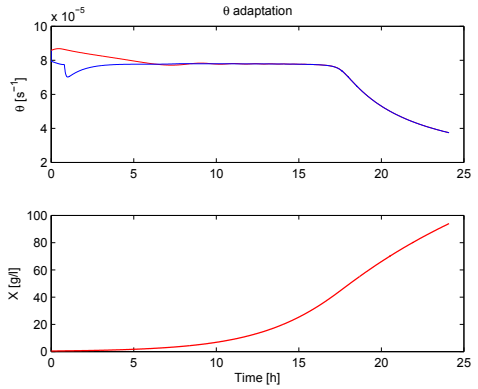


Figure 5: Yeast cultures – θ adaptation and biomass concentration – adaptive control strategy – no measurement noise.

Figure 9 shows the closed-loop response of biomass X, acetate A concentrations, and inlet feed rate F_{in}, for five different values of the kinetic parameters which are randomly chosen, in the bacteria

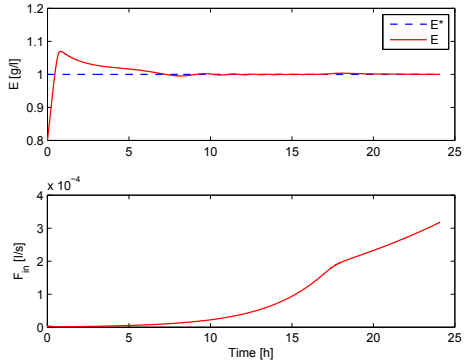


Figure 6: Yeast cultures – ethanol concentration and feed flow rate – adaptive control strategy – no measurement noise.

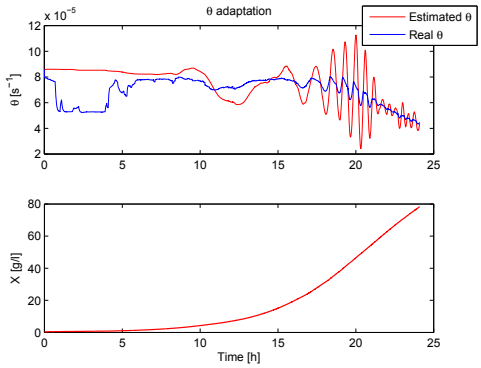


Figure 7: Yeast cultures – θ adaptation and biomass concentration – adaptive control strategy – noise standard deviation of ± 0.05 $[g/l]$.

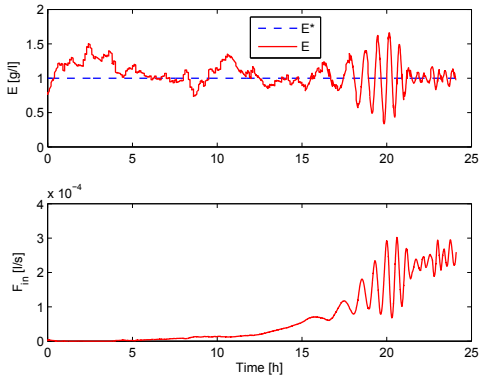


Figure 8: Yeast cultures – ethanol concentration and feed flow rate – adaptive control strategy – noise standard deviation of ± 0.05 $[g/l]$.

cultures under a robust control strategy. Figures 10 and 11 show similar simulation runs with the adaptive strategy. The same comments concerning the noise sensitivity apply.

Note that the productivity is lower in the bacteria

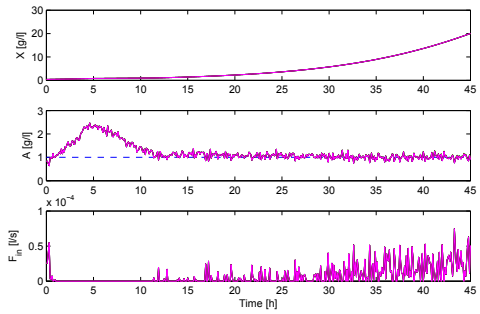


Figure 9: Bacteria cultures – biomass and acetate concentrations, and feed rate – robust control strategy – results of 5 runs with random parameter variations and a noise standard deviation of ± 0.1 $[g/l]$.

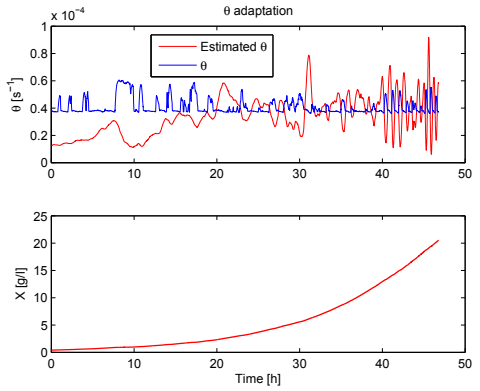


Figure 10: Bacteria cultures – θ adaptation and biomass concentration – adaptive control strategy – noise standard deviation of ± 0.05 $[g/l]$.

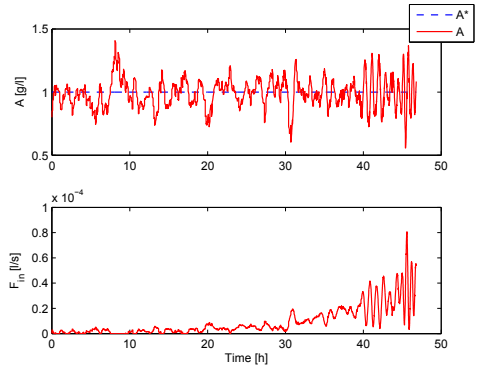


Figure 11: Bacteria cultures – acetate concentration and feed flow rate – adaptive control strategy – noise standard deviation of ± 0.05 $[g/l]$.

cultures (for biological and operating reasons, bacteria strains lead to reaction rates and, therefore, growth rates that are smaller than yeast reaction rates). However, from a control point of view, results are satisfactory in both cases.

6 CONCLUSIONS

Linearizing control is a powerful approach to the control of fed-batch bioprocesses. In most applications reported in the literature, on-line parameter adaptation is proposed in order to ensure the control performance despite modeling uncertainties. On-line parameter adaptation is however sensitive to measurement noise, and requires some kind of tuning. On the other hand, robust control provides an easy design procedure, based on well established computational procedures using the LMI formalism. Large parametric and structural uncertainties, as well as measurement noise levels can be dealt with.

ACKNOWLEDGEMENTS

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Program, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

REFERENCES

Bastin, G. and Dochain, D. (1990). *On-Line Estimation and Adaptive Control of Bioreactors*, volume 1 of *Process Measurement and Control*. Elsevier, Amsterdam.

Chen, L., Bastin, G., and van V. Breusegem (1995). A case study of adaptive nonlinear regulation of fed-batch biological reactors. *Automatica*, 31(1):55–65.

Dewasme, L., Richelle, A., Dehottay, P., Georges, P., Remy, M., Bogaerts, P., and Wouwer, A. V. (2009a). Linear robust control of s. cerevisiae fed-batch cultures at different scales. *In press, Biochemical Engineering Journal*.

Dewasme, L., Wouwer, A. V., Srinivasan, B., and Perrier, M. (2009b). Adaptive extremum-seeking control of fed-batch cultures of micro-organisms exhibiting overflow metabolism. In *Inproceedings of the ADCHEM conference in Istanbul (Turkey)*.

D.F. Coutinho, M. Fu, A. Trofino and P. Danès (2008). L2-gain analysis and control of uncertain nonlinear systems with bounded disturbance inputs. *Int'l J. Robust Nonlinear Contr.*, 18(1):88–110.

G. Chesi, A. Garulli, A. T. and Vicino, A. (2004). Robust analysis of LFR systems through homogeneous polynomial Lyapunov functions. *49*, (7):1211–1215.

Leith, D. and Leithead, W. (2000). Survey of gain-scheduling analysis and design. *International Journal of Control*, 73:1001–1025.

M. Green and D.J.N. Limebeer (1994). *Linear Robust Control*. Prentice Hall.

Pomerleau, Y. (1990). *Modélisation et commande d'un procédé fed-batch de culture des levures pain*. PhD thesis, Département de génie chimique. Université de Montréal.

Renard, F. and Wouwer, A. V. (2008). Robust adaptive control of yeast fed-batch cultures. *Comp. and Chem. Eng.*, 32:1238–1248.

Rocha, I. (2003). *Model-based strategies for computer-aided operation of recombinant E. coli fermentation*. PhD thesis, Universidade do Minho.

S.Boyd, L.El-Ghaoui, E.Feron and V.Balakrishnan (1994). *Linear Matrix Inequalities in System and Control Theory*. SIAM.

Skogestad, S. (2004). Control structure design for complete chemical plants. *Computers and Chemical Engineering*, 28(1-2):219–234.

Skogestad, S. and Postlethwaite, I. (2001). *Multivariable Feedback Control - Analysis and Design*. John Wiley & Sons, New York, NJ.

Sonnleitner, B. and Käppeli, O. (1986). Growth of S*accharomyces cerevisiae* is controlled by its limited respiratory capacity : Formulation and verification of a hypothesis. *Biotechnol. Bioeng.*, 28:927–937.

Sturm, J. F., Romanko, O., and Plik, I. (2006). SeDuMi, version 1.1R3. Online – http://sedumi.mcmaster.ca/.

IMAGE MOTION ESTIMATION USING OPTIMAL FLOW CONTROL

Annette Stahl * and Ole Morten Aamo
Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Norway
anstahl@itk.ntnu.no, aamo@ntnu.no

Keywords: Motion estimation, Optimal control, Physical prior, Optimisation.

Abstract: In this paper we present an optimal control approach for image motion estimation in an explorative and novel way. The variational formulation incorporates physical prior knowledge by giving preference to motion fields that satisfy appropriate equations of motion. Although the framework presented is flexible, we employ the Burgers equation from fluid mechanics as physical prior knowledge in this study. Our control based formulation evaluates entire spatio-temporal image sequences of moving objects. In order to explore the capability of the algorithm to obtain desired image motion estimations, we perform numerical experiments on synthetic and real image sequences. The comparison of our results with other well-known methods demonstrates the ability of the optical control formulation to determine image motion from video and image sequences, and indicates improved performance.

1 INTRODUCTION

In this work we are concerned with motion estimation of objects in image sequences. The understanding and reconstruction of dynamic motion in image scenes is one of the key problems in computer vision and robotics. We present an attempt to adopt control methods from the field of applied mathematics in a new form to image sequence processing and to provide preliminary evaluations of the capability of this approach.

We describe motion as the displacement vector field of pixels between consecutive frames of an image sequence. In the literature this is known as *optical flow* (Jain et al., 1995). In computer vision local and global approaches are used to compute the optical flow field of image sequences. Local approaches are designed to compute the optical flow at a certain pixel position by using only the image information in the local neighbourhood of this specific pixel (Lucas and Kanade, 1981). Variational optical flow methods represent global optimisation problems which can be used to recover the flow field from an image sequence as a global minimiser of an appropriate energy functional. Usually, these energy functionals consist of two terms: a *data term* that imposes the result to be consistent with the measurement (here the brightness constancy assumption) and a *regularisation term* which imposes additional constraints like global or piecewise smoothness to the optical flow field.

One of the first variational methods for motion analysis was introduced by (Horn and Schunck, 1981) and incorporates a *homogeneous* regularisation term, where the optical flow is enforced to vary smoothly in space. This leads to an undesired blurring across motion discontinuities. Therefore, regularisation terms were introduced to regularise the flow in an *image-driven* (Schnörr, 1991; Alvarez et al., 1999) or *flow-driven* (Deriche et al., 1995) way, where the flow is prevented from smoothing across object or motion boundaries, respectively. A systematic classification of these approaches can be found in (Weickert and Schnörr, 2001a).

Most of the variational approaches incorporate a purely *spatial* regularisation of the flow. However, some efforts have been made to incorporate *temporal* smoothness (Nagel, 1990). The work of (Weickert and Schnörr, 2001b) investigates an extension of spatial flow-driven regularisation terms to spatio-temporal flow-driven regularisers. Time is considered as a third dimension analogue to the two spatial dimensions. These approaches improve both the robustness and the accuracy of the motion estimation but the

*This research was supported by an *Alain Bensoussan* fellowship from the European Research Consortium for Informatics and Mathematics (ERCIM) and a fellowship from the Irish Research Council for Science, Engineering and Technology (IRCSET).

flow computation involves the data of the full image sequence at once.

Note that all these approaches do not incorporate physical prior knowledge about the motion itself. In contrast our approach incorporates a space-time regularisation using physical prior knowledge in a control framework that draws on the literature on the control of distributed parameter systems in connection with fluid dynamics (Gunzburger, 2002).

The ideas of two existing control approaches that are related to motion computation of image sequences are presented by (Ruhnau and Schnörr, 2007) and (Borzi et al., 2002). Ruhnau and Schnörr presented an optical flow estimation approach for particle image velocimetry that is based on a control formulation subject to physical constraints (Stokes equation). Their aim is to estimate the velocities of particles in image sequences of fluids rather than to estimate motion in every day image scenes.

The basic idea of (Borzi et al., 2002) is to estimate both an optical flow field u and a rectified image function I satisfying the brightness constancy assumption. Note that in their approach Y_k (and not I) denotes the sampled images of the image sequence. The most significant difference to our optical flow approach is that they do not only estimate the optical flow u, but also I_k which is an approximation of the captured grey value distributions Y_k, where k specifies the frame number within the image sequence. As part of the first-order necessary optimality conditions of the Lagrangian functional their optimal control formulation does not require a differentiation of the image data.

In contrast to that approach, we interpret the grey values of a scene as a "fictive fluid" - assuming that its motion can be described by an appropriate physical model, in this work realised with the Burgers equation of fluid mechanics. We adopt the well established variational optical flow approach of (Horn and Schunck, 1981) and add a distributed control exploiting the Burgers equation resulting in a constrained minimisation problem. The obtained objective functional has to be minimised with respect to the optical flow and control variables subject to the model equation over the entire flow domain in space and time. Our approach estimates not only the optical flow data from an image sequence, but it also estimates a force driven by the Burgers equation. The force field indicates the violation of the equation and can indicate accelerated motions like starting or stopping events or the change of the motion direction. Therefore one can exploit this feature as an indicator of unexpected motion events, taking place in the image sequence.

The initially constrained optimisation problem is reformulated - exploiting Lagrange multipliers - into an unconstrained problem allowing to obtain the associated first-order optimality system. This results in a forward-backward system with appropriate initial and boundary conditions. To solve the optimality system we uncouple the forward and backward computation as described in (Gunzburger, 2002) leading to an iterative solution scheme.

2 APPROACH

Before we start to describe the approach in more detail we first exemplify the notation and components of our control formulation.

We define a grey value of a certain pixel within an image sequence by a real valued one-time continuously differentiable C^1 image function $I(x,t)$, where $x = (x_1, x_2)^\top$ denotes the location within some rectangular image domain Ω and $t \in [0,T]$ labels the corresponding frame at time t. In particular, the function $I(x_1,x_2,t)$ denotes the intensity of a pixel at position $(x_1,x_2)^\top$ in the image frame at time t. The optical flow field is denoted by a two-dimensional vector field $u = (u_1(x,t), u_2(x,t))^\top$, which describes the intensity changes between images.

We formulate our motion estimation problem within a variational framework. We minimise an energy functional E, which consists of a data and a regularisation term:

Data Term. We make use of the following *data term*

$$\int_\Omega (\partial_t I + u \cdot \nabla I)^2 \, dx, \tag{1}$$

which comprises the optical flow constraint (Horn and Schunck, 1981) and provides the link between the given image data, the observed intensity I and the desired velocity field u. Note that the optical flow constraint equation represents the requirement that the intensity of an object point stays constant along its motion trajectory. Problem (1) is ill-posed as any vector field u satisfying $u \cdot \nabla I = -\partial_t I$, is a minimiser. Therefore a regularisation term is added to introduce additional constraints for the flow field u to obtain an unique solution.

Regularisation Term. We incorporate the regularisation term from (Horn and Schunck, 1981)

$$\int_\Omega \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) \, dx, \quad 0 < \alpha \in \mathbb{R}, \tag{2}$$

to enforce spatial smoothness of the optical flow field, preferring neighbouring optical flow vectors to

be similar. The regularisation parameter α adjusts the relative importance of the smoothness term to the data term. With an increasing value of α the vector field is forced to become smoother. We are aware that regulariser like the L1-regulariser used for example in (Wedel et al., 2009) allows for sharper discontinuities in the flow field. Our decision to use the L2-regulariser in the motion estimation was mainly driven by the idea to keep the approach clear and numerically simple. However, the replacement of the quadratic homogeneous smoothness term could improve the accurateness of the computed motion boundaries.

Physical Prior. Considering a constant moving object one can determine that structures are transported by a velocity field and along with it the velocity field is transported by itself. A physical model equation, which describes this behaviour is the *Burgers equation* and allows to model the movement of rigid objects.

The inviscid Burgers equation

$$\frac{D}{Dt}u = \partial_t u + (u\cdot\nabla)u = 0\,, \quad u(x,0) = u_0 \qquad (3)$$

has been studied and successfully applied for many decades in aero- and fluid dynamics (Burgers, 1948; Hirsch, 2000) as a simplified model for turbulence, boundary layer behaviour, shock wave formation and mass transport. It contains the convection term from the fundamental equations of fluid mechanics, the Navier-Stokes equations.

As a physical interpretation, u in (3) may be regarded as a vector of conserved (fictive) quantities or states, with corresponding density functions u_1, u_2 as components. The material derivative $\frac{D}{Dt}$ yields the acceleration of moving particles. The nonlinear term $(u\cdot\nabla)u$ is known as the inertia term of the transport process described by (3). See Figure 1 for an illustration of the transport. We found that our approach even with the constant velocity assumptions of our physical prior predicts the non-uniform motion pattern quite well as shown in our numerical results (cf. Sec. 4.2).

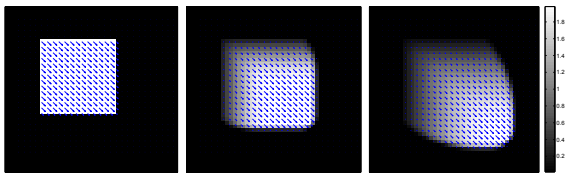


Figure 1: Illustration of the transportation of a vector field with equation (3) at times $t = 0, 5, 10$. Gray values visualise vector magnitudes. Fictive particles move along a shock front in the lower right direction. In the absence of any further external information, a region of rarefaction arises due to mass conservation, acting like a short-time memory.

2.1 Optimal Control Formulation

In the following sections we explain our optimal control approach. Foundations exploiting fluid dynamical methods can be found in the book of Gunzburger (Gunzburger, 2002).

We obtain our spatial-temporal control approach as follows: Additionally to the smoothness term we introduce a control f, that is distributed in space and time, which means that it acts over the entire optical flow domain $\Omega\times[0,T]$. The magnitude of the control is bounded due to penalisation within the objective functional. The resulting optimisation problem is to minimise

$$E(u,f) = \frac{1}{2}\int\limits_{\Omega\times[0,T]} \Big\{(\partial_t I + u\cdot\nabla I)^2 \qquad (4)$$
$$+ \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) + \beta|f|^2\Big\}\,dxdt,$$

subject to the equations of motion

$$\begin{cases} \partial_t u + (u\cdot\nabla)u &= f & \text{in } (0,T)\times\Omega, \\ \partial_n u &= 0 & \text{on } (0,T)\times\Gamma. \end{cases} \qquad (5)$$

We intend to find an optimal state $u = (u_1,u_2)^\top$ and an optimal control $f = (f_1,f_2)^\top$, such that the functional $E(u,f)$ is minimised and u and f satisfy the Burgers equation (5).

The objective of this formulation is to determine a body force f (the control !) that leads to a velocity field u which fits to the apparent motion in the image sequence, and at the same time satisfies physical prior knowledge in terms of the given equations of motion.

2.2 Optimality System

In order to obtain the velocity field u and the control f we recast the constrained optimisation problem (4) - (5) into an unconstrained optimisation problem. Introducing the Lagrange multiplier or adjoint variable $w = (w_1(x,t), w_2(x,t))^\top$ yields the following Lagrangian functional

$$L(u,f,w) = E(u,f) - \int\limits_{\Omega\times[0,T]} w^\top(\partial_t u + (u\cdot\nabla)u - f)\,dxdt. \qquad (6)$$

To solve this functional we have to derive the first-order necessary conditions. This results in the following optimality system (7)-(9) from which the optimal

state u, adjoints w, and the optimal control f can be determined such that $L(u, f, w)$ is rendered stationary.

$$\begin{cases} \partial_t u + (u \cdot \nabla)u &= f & \text{in } \Omega \times [0,T], \\ \partial_n u &= 0 & \text{on } \Gamma \times [0,T], \\ u|_{t=0} &= u_0 & \text{in } \Omega, \end{cases} \tag{7}$$

$$\begin{cases} -\partial_t w - (u \cdot \nabla)w - w\nabla \cdot u + (\nabla U)^\top w \\ = \nabla I(\partial_t I + u \cdot \nabla I) - \alpha \Delta u & \text{in } \Omega \times [0,T], \\ w = 0 & \text{on } \Gamma \times [0,T], \\ w|_{t=T} = 0 & \text{in } \Omega, \end{cases} \tag{8}$$

$$\begin{cases} \beta f + w &= 0 & \text{in } \Omega \times [0,T], \\ f &= 0 & \text{on } \Gamma \times [0,T], \\ f|_{t=T} &= 0 & \text{in } \Omega, \end{cases} \tag{9}$$

where $(\nabla U)^\top$ the transposed Jacobian matrix.

The state equation (7) is obtained by derivation of the Lagrangian functional (6) in the direction of the Lagrange multiplier, and turns out to be identical to the Burgers equation (5) itself. The adjoint equation (8) specifies the first-order necessary conditions with respect to the state variables u. The optimality condition (9) is the necessary condition that the gradient of the objective function – with respect to the control f – vanishes at the optimum. It also includes the initial and terminal conditions.

The optimality system (7)-(9) is a coupled system which turns out to be - due to the large number of unknowns - prohibitively expensive to solve directly, but can be solved iteratively as described in the next section.

2.3 Algorithm

We solve the optimality system (7)-(9) using an iterative gradient descent method (with step length adoption) which decouples the state and adjoint computation. It consists of the iterative solution of the state and adjoint equation in such a way that the state equation is computed forward in time with appropriate initial condition u_0 and the adjoint equation is computed backward in time with terminal condition $w_{t=T} = 0$. The optimality condition is used to update the control f with the adjoint variable w. The control f is then used to compute the actual state u. Additionally, the step length is adjusted ensuring that the actual energy of the objective functional (4) decreases. Note that we choose the start value for f to be zero in the very first iteration.

In our pseudo code description of Algorithm 1, variable s denotes the step-size that is adapted by the algorithm and ε the threshold which is used to decide if the relative difference of the energy is small enough to be seen as converged.

In the initial step of the algorithm the flow fields u for all consecutive image frames and the terminal condition of the adjoint variable for the last frame ($w_{t=T}$) are set to zero. The first step of the iteration loop solves the adjoint equation (8) for w backwards in time using the terminal condition on w and the flow field u. Then, the optimality condition (9) is used to update the control field for all frames, allowing the state equation (8) to be solved for u forward in time using the new control field. The iteration loop continues until the decline in E is negligible.

Algorithm 1: Gradient algorithm with automatic step-length selection.

1: set $u = 0$, $\varepsilon = 10^{-8}$, and $s := s_0$ (initial step)
2: **repeat**
3: solve the adjoint equation (8) for w
4: update f: $f_m = f_{m-1} - s(\beta f_{m-1} + w)$
5: solve the state equation (7) for u
6: **if** $E(u, f_m) \geq E(u, f_{m-1})$ **then**
7: $s := 0.5s$
8: GOTO 4
9: **else**
10: $s := 1.5s$
11: **end if**
12: **until** $|E(u, f_m) - E(u, f_{m-1})|/|E(u, f_m)| < \varepsilon$

3 NUMERICAL SOLUTION

In this part, we summarised the numerical discretisation methods employed in solving the optimality system (7)-(9). For more details, we refer to (Colella and Puckett, 1998).

Discretisation of the State Equation. Within the numerical implementation of the nonlinear state system equation (7) we have to cope with over- and undershoots, with shock formations, with the compliance of conditions (entropy-, monotony-, CFL-condition, etc.) and different discretisation schemes. We use the second-order conservative Godunov scheme for our implementation. The fluxes are numerically computed by solving the equations at pixel edges. The correct behaviour at discontinuities is obtained by using solutions of the appropriate Riemann problem.

Discretisation of the Adjoint Equation. The numerical implementation of the time-dependent adjoint system (8) in the domain Ω is done by using a second-order predictor-corrector finite difference

scheme. The basic idea behind this is that all methods with an accuracy larger than the order one will produce spurious oscillations in the vicinity of large gradients, while being second-order accurate in regions where the solution is smooth. To prevent such oscillations the slopes of Fromm's method are replaced by the slopes of the Van Leers scheme. The Van Leer scheme *detects* discontinuities and modifies its behaviour in such locations accordingly. The implication of this is that this method retains the high-order accuracy of Fromm's scheme in smooth regions, but near discontinuities the discretised evolution equation drops to first-order accuracy.

4 EXPERIMENTS

In this section we first illustrate the control performance of our optical flow approach on a real-world 2D image sequence. Secondly, we evaluate the following motions which violate the incorporated motion assumption: rotation, translation in combination with scaling. Finally, we present the results for noisy image data showing the influence of the temporal regularisation in the control approach and provide a comparison with error measures obtained by the approach from (Horn and Schunck, 1981) and the dynamic optical flow approach from (Stahl et al., 2006).

4.1 Control - Force

We illustrate the control behaviour of our approach for a real-world 2D image sequence with an unexpected motion. The image sequence consists of 10 image frames and shows a moving hand which starts to move and then stops again. Figure 2 depicts the starting (left column) and stopping (right column) event of the sequence. The first row shows the velocity estimates u, and the second row shows the force fields. The force field f nicely indicates the deviation of the expected motion from the observed motion. This is evident in the second row of Figure 4, where the force field acts in the direction of the moving hand as the hand accelerates into motion (left picture), while it turns in the opposite direction as the hand stops (right picture).

4.2 Non-uniform Motion

In this section we provide an evaluation of our approach on the basis of two well known synthetic image sequences for which the ground truth motion data is available. To allow for a quantitative comparison we provide the results we obtain for the Horn and Schunck as well. The image sequences we use show global motion patterns such as rotation, translation and divergence.

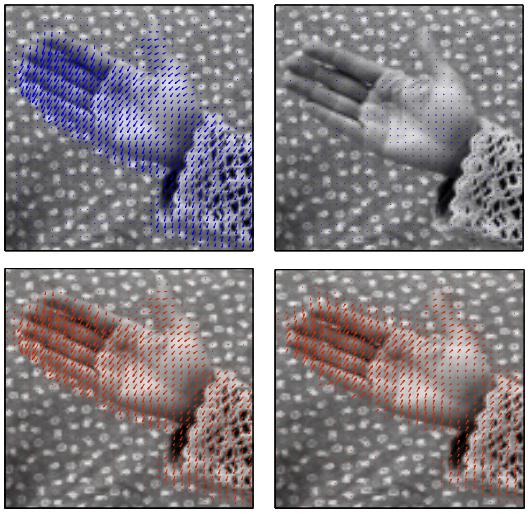


Figure 2: "Waving hand" sequence: Unexpected events. **Top:** A waving hand stops. The estimated optical flow field u for a starting (left) and stopping (right) event is depicted in blue. **Bottom:** The corresponding control field f is shown in red. The force acts when the hand starts to move (left) and reacts into the opposite direction of the flow field (right) when it stops and forces the flow field into the observed state of no motion (parameters: $\alpha = 0.01$, $\beta = 0.0001$).

In particular we evaluate our approach on the grey value versions of the following two image sequences: the "rotating sphere" sequence (McCane et al., 2001) and the "Yosemite" sequence (available at ftp://ftp.csd.uwo.ca/pub/vision).

The "rotating sphere" sequence contains a curling vector field and is shown in Figure 3. This sequence consists of 45 frames, where a sphere rotates in front of a stationary background.

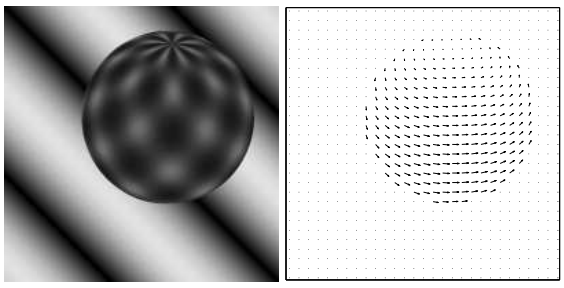


Figure 3: The synthetic "rotating sphere" sequence. The sphere rotates in front of a stationary background. **Left:** Gray value version of frame 6 that is used in our computations. **Right:** Vector plot of the ground truth data.

The computed vector fields obtained by the Horn and Schunck approach and our approach (4)-(5) for the "rotating sphere" sequence is shown in Figure 4.

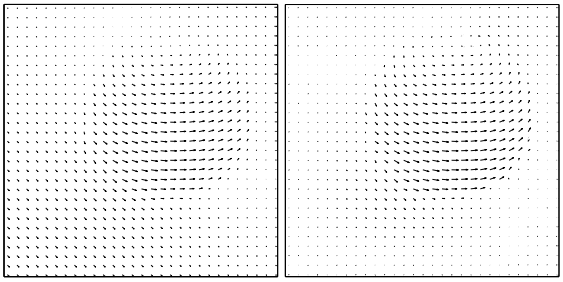


Figure 4: The synthetic "rotating sphere" sequence. Computational Results for the Horn and Schunck approach and the control based approach. **Left:** Result Horn and Schunck (RMSE = 0.395). **Right:** Result control based approach (RMSE = 0.192).

The motion estimation results for the "Yosemite" sequence are shown in section 4.3.

The results show that even for sequences which violate the constant velocity assumption of the model equation we obtain good results. However, due to the flexibility of our variational approach it should be possible to model such motion patterns by incorporation of a suitable model equation.

4.3 Temporal Regularisation

To investigate the impact of the temporal regularisation to the robustness of our approach under noise, we choose the "Yosemite" sequence with different Gaussian noise levels $\sigma = 0$, 10, 20 and 40 (cf. Fig. 5). The

Figure 5: **Top Left:** Yosemite sequence. **Top Right:** We added Gaussian noise with standard deviation $\sigma = 40$.

sequence exhibits divergent and translational motion combined with illumination changes. To investigate the performance of our approach we compare the root mean square error (RMSE)

$$RMSE(u_o, u_e) = \frac{1}{|\Omega|} \int_\Omega \sqrt{(u_o - u_e)^2} dx$$

and the average angular error (AAE)

$$AAE(u_o, u_e) = \frac{1}{|\Omega|} \int_\Omega \arccos\left(\frac{u_o \cdot u_e}{|u_o||u_e|}\right) dx,$$

where $|\cdot|$ denotes the Euclidean norm, $u_o = (u_{o_1}, u_{o_2}, 1)^\top$ the original optical flow vectors, and $u_e = (u_{e_1}, u_{e_2}, 1)^\top$ the estimated optical flow vectors (compare (Barron et al., 1994)). Note that the time dimension is set to 1 corresponding to the distance of one frame.

This measure is currently used as a kind of standard to provide accuracy measures for optical flow results.

We compare the errors of the optical flow computation obtained for three different approaches with optimised parameters. In particular these are the homogeneous spatial regularised approach from (Horn and Schunck, 1981), the spatio-temporal dynamic image motion approach from (Stahl et al., 2006), and our control based image motion approach (4) - (5). The control approach results in a improved vector field, which is based on the forward-backward computation, which incorporates additional knowledge of the future frames leading to an improved temporal regularisation. The result for a single frame in the highly noisy Yosemite sequence is shown in Figure 6. The

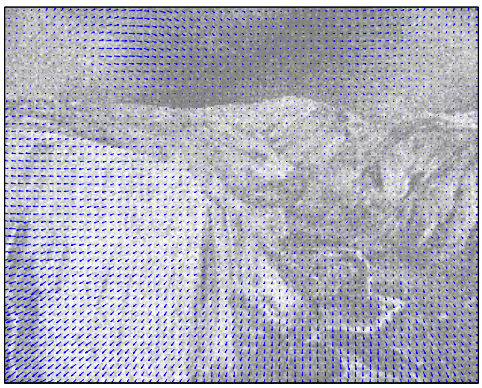


Figure 6: Temporal regularisation. We added Gaussian noise with standard derivation $\sigma = 40$ to the "Yosemite" sequence. The shown high quality optical flow field is obtained by the control based optical flow approach (4) - (5) (parameters: $\alpha = 0.05$ and $\beta = 0.000003$).

results for the computed errors (RMSE and AAE) for all three approaches with increasing noise level are shown in Table 1. The purely spatial regularised approach from Horn and Schunck and the absence of physical prior knowledge leads to the higher error values with increasing noise levels. In contrast to the spatio-temporal dynamic image motion approach (Stahl et al., 2006) a higher noise level requires the selection of a smaller β regularisation parameters for the control part of the objective functional. The consistently lower error indicates an improved global motion prediction in our control approach (4)-(5) exerting a better temporal regularisation. Our explanation for this observation is that the control approach incor-

Table 1: Performance of our control approach (C) in comparison with the Horn and Schunck approach (HS) and the dynamic image motion approach (Dy) in presence of noise: We added random Gaussian noise with zero mean and standard deviation $\sigma = 0$, 10, 20, and 40 to the Yosemite image sequence.

σ	app.	α	β	RMSE	AAE
0	HS	0.005	-	0.177	3.04°
	Dy	0.006	0.00002	0.178	3.09°
	C	0.007	0.0005	**0.169**	**2.88°**
10	HS	0.008	-	0.283	5.74°
	Dy	0.01	0.0003	0.275	5.68°
	C	0.009	0.0001	**0.243**	**4.92°**
20	HS	0.02	-	0.429	8.61°
	Dy	0.025	0.001	0.395	7.54°
	C	0.02	0.00001	**0.350**	**6.67°**
40	HS	0.05	-	0.640	13.27°
	Dy	0.05	0.005	0.523	9.89°
	C	0.05	0.000003	**0.497**	**9.16°**

porates also future knowledge of the image sequence instead of using only past information with a prediction as in (Stahl et al., 2006).

5 CONCLUSIONS

We have presented an optimal control approach to image motion estimation including physical prior knowledge in a novel and exploratory way. It leads to an unconstrained optimisation problem, where the optimality system - from which the optimal state and the optimal control are determined - can be solved using an iterative gradient descent method. The forward-backward structure of the model allows for a *robust* estimation of the coherent flows by including *prior knowledge* that enforce spatio-temporal smoothness of the minimising vector field.

In the case that the image measurements indicate changes of the current velocity distribution, fictive control forces modify the system state accordingly. The presence of such forces may serve as an indicator notifying a higher-level processing stage about unexpected motion events in video sequences.

The comparison of our results with the approach from (Horn and Schunck, 1981) and the approach from (Stahl et al., 2006) demonstrates the ability of the control formulation to determine image motion from video sequences, and shows improved performance, especially for highly noisy image data. Our further work will include the modification of the Burgers equation to achieve better motion boundaries in the rarefaction area and the reformulation of the approach to a receding horizon formulation.

ACKNOWLEDGEMENTS

We would like to thank Dr. Christian Schellewald, Dr. Paul Ruhnau, Prof. Christoph Schnörr, and Prof. Øyvind Stavdahl for some inspiring discussions and comments.

REFERENCES

Alvarez, L., Esclarìn, J., Lefebure, M., and Sànchez, J. (1999). A PDE model for computing the optical flow. In *Proceedings of CEDYA XVI*, pages 1349–1356.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43–77.

Borzi, A., Ito, K., and Kunisch, K. (2002). Optimal control formulation for determining optical flow. *SIAM J. Sci. Comput.*, 24(3):818–847.

Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:171–199.

Colella, P. and Puckett, E. G. (1998). *Modern Numerical Methods for Fluid Flow*. Lecture Notes, Dep. of Mech. Eng., Uni. of California, Berkeley, CA. http://www.rzg.mpg.de/ bds/numerics/cfd-lectures.html.

Deriche, R., Kornprobst, P., and Aubert, G. (1995). Optical-flow estimation while preserving its discontinuities: A variational approach. In *ACCV*, pages 71–80.

Gunzburger, M. (2002). *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics.

Hirsch, C. (2000). *Numerical Computation of Internal and External Flows (Vol.I+II)*. John Wiley & Sons.

Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. McGraw-Hill, Inc.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision (darpa). In *Proc. of the 1981 DARPA Image Understanding Workshop*, pages 121–130.

McCane, B., Novins, K., Crannitch, D., and Galvin, B. (2001). On benchmarking optical flow. *Comput. Vis. Image Underst.*, 84(1):126–143.

Nagel, H. H. (1990). Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In *Proc. of the first european conf. on computer vision*, pages 139–148. Springer.

Ruhnau, P. and Schnörr, C. (2007). Optical Stokes flow: An imaging-based control approach. *Experiments in Fluids*, 42:61–78.

Schnörr, C. (1991). Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *Int. J. of Computer Vision*, 6(1):25–38.

Stahl, A., Ruhnau, P., and Schnörr, C. (2006). *A Distributed Parameter Approach to Dynamic Image Motion*. Int. Workshop on The Representation and Use of Prior Knowledge in Vision. ECCV Workshop.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An improved algorithm for tv-l1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18, 2008. Revised Papers*, pages 23–45. Springer-Verlag.

Weickert, J. and Schnörr, C. (2001a). A theoretical framework for convex regularizers in PDE-based computation of image motion. *Int. J. of Computer Vision*, 45(3):245–264.

Weickert, J. and Schnörr, C. (2001b). Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging and Vision*, 14(3):245–255.

STABILITY ANALYSIS FOR BACTERIAL LINEAR METABOLIC PATHWAYS WITH MONOTONE CONTROL SYSTEM THEORY

Nacim Meslem, Vincent Fromion, Anne Goelzer and Laurent Tournier
INRA, Mathematics, Informatics and Genome Laboratory
Jouy-en-Josas, Domaine de Vilvert 78352 Jouy-en-Josas Cedex, France
{*nmeslem, vfromion, agoelzer, ltournier*}*@jouy.inra.fr*

Keywords: Monotone control systems, Negative feedback theorem, Linear bacterial metabolic pathways.

Abstract: In this work we give technical conditions which guarantee the global attractivity of bacterial linear metabolic pathways (reversible and irreversible structures) where both genetic and enzymatic controls involve the end product through metabolic effectors. To reach this goal, we use the negative feedback theorem of the monotone control systems theory, and we represent all conditions needed to apply the negative feedback theorem to the bacterial linear metabolic pathways in convenient deduced forms.

1 INTRODUCTION

The bacterial metabolic machinery and its regulation make up a complex system involving many cellular components such as metabolites and enzymes. In this paper, we focus on the dynamical behavior of the control structures used in a large number of bacterial biosynthesis pathways where both the genetic and enzymatic controls involve the last product as metabolite effector (Goelzer et al., 2008). Stability analysis of these biological structures is recognized as an issue of great importance in order to deduce key biological properties of the bacterial metabolic pathways. In the literature, many studies focused on the analysis of the metabolic and genetic networks separately. For instance, using the stability results about cyclic dynamical systems (Tyson and Othmer, 1978), (Sanchez, 2009), (Arcak and Sontag, 2006), one can state nice stability conditions of the irreversible linear metabolic pathways with allosteric regulation. One can also use the stability results about tridiagonal systems (Angeli and Sontag, 2008), (Wang et al., 2008) to analyze the stability of the reversible metabolic pathways. However, few works have considered structures with both genetic and allosteric regulation. Thus, in this paper we investigate stability of the common structures shared by many bacteria cells and yeasts. These structures are called *end product structures*, because both genetic and enzymatic controls involve the end product of the pathway (Grundy et al., 2003), (Gollnick et al., 2005), (Goelzer et al., 2008).

We will use the monotone control system theory developed in (Angeli and Sontag, 2003) to deal with stability issue of biological systems. In particular, the negative feedback theorem has been applied to a model of Mitogen-Activated Protein Kinase (MAPK) cascades in (Angeli and Sontag, 2003), and more recently to Goldbeter's circadian model (Angeli and Sontag, 2008). The main contribution of this work consists in providing technical conditions to check all the required assumptions to apply the negative feedback theorem to *end product structures* (under irreversible and reversible forms).

This paper is structured as follows. Section 2 presents the mathematical models for the linear reversible and irreversible bacterial metabolic pathways and states the main results of this paper which consist in propositions 1 and 2. Section 3 recalls some definitions and properties of monotone control systems theory and introduces the negative feedback theorem. Section 4 addresses the stability analysis of the dynamical models introduced in section 2 and proves the two propositions.

2 LINEAR METABOLIC PATHWAYS

Consider a linear pathway with n metabolites involved in enzymatic reactions, an input flux ν_1 and an output flux ν_n as depicted in Figure 1. Each X_i and $\mathbb{E}_i$ correspond to a metabolite and an enzyme respectively. We assume that the pool X_1 of the first

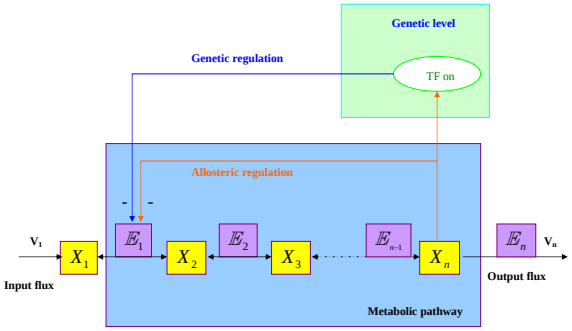


Figure 1: End product control linear structure.

metabolite is maintained by the input flux ν_1 which corresponds to a supply flux. Hence its concentration $\bar{x}_1$ is strictly positive constant. The output of the pathway is the flux ν_n which corresponds to the bacterium requirement for the metabolite X_n. Hereafter, for each $i \in \{2,\dots,n\}$ we denote by x_i the nonnegative concentration of the metabolite X_i, and by E_i the assumed constant positive concentration of the enzyme $\mathbb{E}_i$. The three phenomena, *enzymatic reactions, allosteric regulation and genetic regulation* (with respect to E_1), presented in Figure 1 can be described by a set of interconnected nonlinear differential equations. In the sequel, we analyze global stability of two types of the interconnected differential equations, namely the reversible and irreversible metabolic pathways.

2.1 Reversible Pathways

The common *end product structure* of linear reversible metabolic pathways is described by the following dynamical system:

$$\left\{\begin{array}{lcll} \dot{x}_2 & = & E_1 f_1(\bar{x}_1,x_2,x_n) - E_2 f_2(x_2,x_3) \\ \dot{x}_3 & = & E_2 f_2(x_2,x_3) - E_3 f_3(x_3,x_4) \\ \vdots & \vdots & \vdots & \vdots \\ \dot{x}_n & = & E_{n-1} f_{n-1}(x_{n-1},x_n) - E_n f_n(x_n) \\ \dot{E}_1 & = & g(x_n) - \mu E_1 \end{array}\right. \quad (1)$$

where the Lipschitz functions f_i denote the reaction rates of the enzymes $\mathbb{E}_i$. Note that, in the reversible structures all reaction rates depend on the product and substrate concentrations and have the following properties:

- *For the first enzyme*: we assume that the metabolite X_n modulates the activity of the enzyme $\mathbb{E}_1$ through, for example, an allosteric effect. The function $f_1(x_1,x_2,x_n)$ is increasing in its first argument and decreasing with respect to its second and third arguments, and we have for any $x_1 > 0$, $x_2 \geq 0$ and $x_n \geq 0$, $f_1(x_1,x_2,x_n) > 0$ and for any $x_n \geq 0$, $f_1(0,0,x_n) = 0$. In addition, there exists $M_1 > 0$ such that for any $x_1 > 0$, $x_2 \geq 0$ and $x_n \geq 0$, $f_1(x_1,x_2,x_n) \in [0,\ M_1)$. We also assume that for any $x_1 > 0$ and $x_n \geq 0$ there exists $x_2^* > 0$ such that $f_1(x_1,x_2^*,x_n) = 0$. Finally, for any $x_1 > 0$ and $x_2 > 0$ we have,
$$\lim_{x_n \to +\infty} f_1(x_1,x_2,x_n) = 0.$$
- *For the intermediate enzymes*: $f_i, i \in \{2,\dots,n-1\}$, is increasing in x_i and decreasing in x_{i+1}. For any $x_i > 0$, $f_i(x_i,0) > 0$, and for any $x_{i+1} > 0$, $f_i(0,x_{i+1}) < 0$ and $f_i(0,0) = 0$. Moreover, there exists $M_i > 0$ and $M_i' \geq 0$ such that for any $x_i > 0$ and $x_{i+1} \geq 0$, $f_i(x_i,x_{i+1}) \in (-M_i',\ M_i)$. Finally, we assume that for any $x_i > 0$ there exists $x_{i+1}^* > 0$ such that $f_i(x_i,x_{i+1}^*) = 0$.
- *For the final enzyme*: $\mathbb{E}_n$ describes the properties of the remainder part of the metabolic network and summarizes the relation between the flux supplied by the pathway and the final concentration. The properties of f_n mainly depends on the properties of the next modules, and generally f_n is a strictly increasing, positive and bounded function in x_n such that
$$f_n(0) = 0,\ \lim_{x_n \to +\infty} f_n(x_n) = M_n.$$

The dynamics of the enzyme concentrations during the exponential growth phase are mostly the result of two phenomena: (i) the *de novo* production (ii) the *dilution* effect caused by the increase of the cell volume. For this, in the last equation of (1), we have considered that the control of the concentration of the first enzyme is regulated by the concentration of the final metabolite x_n, where μ is the growth rate of the bacterium assumed to be in the exponential growth phase. The term $g(x_n)$ corresponds to the instantaneous production of the enzyme E_1 modulated by a metabolite (implicitly through a transcription factor). The continuous function $g(.)$ is positive strictly decreasing in the end product x_n with $g(0) = g_{max}$, $g_{max} > 0$ and

$$\lim_{x \to +\infty} g(x) = 0.$$

After the detailed description of the dynamical model of the linear reversible metabolic pathway, we state below the main results of this paper about its global attractivity.

Stability Results. Let us start by setting three hypotheses and then we introduce our first proposition.

- Hypothesis $\mathcal{H}_1$: The $n-1 \times n-1$ Tridiagonal matrix,

$$\mathbf{Q}=\begin{bmatrix} q_{2,2} & q_{2,3} & 0 & \dots & \dots & 0 \\ q_{3,2} & q_{3,3} & q_{3,4} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & q_{n-1,n-2} & q_{n-1,n-1} & q_{n-1,n} \\ \vdots & \dots & \dots & 0 & q_{n,n-1} & q_{n,n} \end{bmatrix}$$

where $\forall i, j \in \{2,\dots,n\}$

$$q_{i,j} = \sup\left(\frac{\partial(E_{i-1}f_{i-1}(.) - E_i f_i(.))}{\partial x_j}\right),$$

is Hurwitz.

- Hypothesis $\mathcal{H}_2$: The inequality $E_{n-1}M_{n-1} \leq E_nM_n$ is verified.
- Hypothesis $\mathcal{H}_3$: The graph of the scalar function

$$T(u) = g \circ k_y(u)$$

and that of its reciprocal function $T^{-1}(u)$ have a unique intersection point on the open interval $u \in (0, g_{max})$.

The scalar function $k_y(.)$ is the *static input-output characteristic* associated to the monotone part of (1) (resp. (2)), see Definition 2 in subsection 3.1.

Proposition 1. *If $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ are satisfied, then for any $\bar{x}_1$ and E_n, the reversible end product structure* (1) *has globally attractive equilibrium.*

2.2 Irreversible Pathways

The main difference between the irreversible and the reversible metabolic pathways is in the reaction rates f_i for the first and intermediate enzymes. Indeed, here we assume that the reaction rates depend only on the substrate concentration and have the following properties:

- *For the first enzyme.* We assume that the function f_1 is increasing in its first argument and decreasing in its second argument and for any $x_1 > 0$,

$$\lim_{x_n \to +\infty} f_1(x_1, x_n) = 0.$$

In addition, we have for any $x_n \geq 0$, $f_1(0, x_n) = 0$ and there exists $M_1 > 0$ such that for any $x_1 > 0$ and $x_n \geq 0$, $f_1(x_1, x_n) \in [0, M_1)$.

- *For the intermediate enzymes*: $f_i\ i \in \{2,\dots,n-1\}$ is strictly increasing in x_i and $f_i(0) = 0$. Moreover, there exists $M_i > 0$ such that

$$\lim_{x_i \to +\infty} f_i(x_i) = M_i.$$

Then, the *end product structure* of the linear irreversible metabolic pathways is described by the following dynamical system

$$\begin{cases} \dot{x}_2 &= E_1 f_1(\bar{x}_1, x_n) - E_2 f_2(x_2) \\ \dot{x}_3 &= E_2 f_2(x_2) - E_3 f_3(x_3) \\ \vdots & \vdots \quad \vdots \quad \vdots \\ \dot{x}_n &= E_{n-1} f_{n-1}(x_{n-1}) - E_n f_n(x_n) \\ \dot{E}_1 &= g(x_n) - \mu E_1. \end{cases} \tag{2}$$

Stability Results. Now, we state the contribution of this paper concerning the global attractivity of the irreversible metabolic pathway (2).

- Hypothesis $\mathcal{H}_4$: for each $i \in \{2,\dots,n\}$ the inequality is verified $\overline{E}_1 M_1 \leq E_i M_i$, where $\overline{E}_1$ is the upper bound of all solutions $E_1(t)$.

Proposition 2. *The irreversible end product structure* (2) *has globally attractive equilibrium for any $\bar{x}_1$ and E_n if hypotheses $\mathcal{H}_3$ and $\mathcal{H}_4$ are satisfied.*

To prove Proposition 1 and Proposition 2, we will use the monotone control system theory, in particular the negative feedback theorem. Thus, we present briefly this theory in the next section and then we give the proofs in section 4.

3 MONOTONE CONTROL SYSTEMS

Monotone control systems theory (Angeli and Sontag, 2003) is an extension of the autonomous monotone system theory (Smith, 1995). Briefly, monotone control system is a dynamical system on an ordered metric space which has the property that ordered initial states and ordered inputs generate ordered state trajectories and ordered outputs. In other words, a controlled dynamical system (3),

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}) \end{cases}, \ \mathbf{x}(t_0) = cst, \tag{3}$$

where $\mathbf{x}(t) \in \mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{U} \subseteq \mathbb{R}^m$, is said monotone if the following implication holds: $\forall (\mathbf{x}_1(t_0),\ \mathbf{x}_2(t_0)) \in \mathbb{X}^2$ and $\forall (\mathbf{u}_1(t),\ \mathbf{u}_2(t)) \in \mathbb{U}^2$,

$$\begin{aligned} &\mathbf{x}_1(t_0) \preceq \mathbf{x}_2(t_0), \mathbf{u}_1(t) \preceq \mathbf{u}_2(t) \Rightarrow \\ &\quad \mathbf{x}_1(t, \mathbf{x}_1(t_0), \mathbf{u}_1(t)) \preceq \mathbf{x}_2(t, \mathbf{x}_2(t_0), \mathbf{u}_2(t))\ \forall t \geqslant t_0 \end{aligned} \tag{4}$$

where $\mathbf{x}(t, \mathbf{x}(t_0), \mathbf{u}(t))$ represent the state trajectory generated by (3) with $\mathbf{x}(t_0)$ as initial state and $\mathbf{u}(t)$ as input. The dimensions of the vectors $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ are respectively n, m and p.

Here, we consider that $\preceq$ is the classical lower or equal comparison operator $\leq$, applied component by

component. Systems that are monotone with respect to this order are called cooperative systems, as all state variables have a positive influence on one other and the inputs act positively on state variables.

Proposition 3. *The dynamical system* (3) *is cooperative if and only if the following properties hold:*

$$\begin{array}{ll} \frac{\partial f_i}{\partial x_j}(\mathbf{x},\mathbf{u}) \geq 0 & \forall \mathbf{x} \in \mathbb{X}, \forall \mathbf{u} \in \mathbb{U}, \forall i \neq j \\ \frac{\partial f_i}{\partial u_j}(\mathbf{x},\mathbf{u}) \geq 0 & \forall \mathbf{x} \in \mathbb{X}, \forall \mathbf{u} \in \mathbb{U}, \forall i, j \\ \frac{\partial h_i}{\partial x_j}(\mathbf{x}) \geq 0 & \forall \mathbf{x} \in \mathbb{X}, \forall i, j \end{array} \tag{5}$$

Proof. See (Angeli and Sontag, 2003; Angeli and Sontag, 2004).

After this brief recall about monotone control systems, we now introduce in the next the negative feedback theorem which states stability conditions for monotone control systems with negative feedback.

3.1 Stability Analysis with Monotone Control System

Recently, the negative feedback theorem of the monotone control system theory is used to analyze stability of several biological systems. Indeed, this theorem allows, under some conditions, to obtain the globally attractive stable steady state of non-monotone dynamical systems. Here we give some definitions and assumptions needed to state the negative feedback theorem.

Definition 1 (Angeli and Sontag, 2003)**.** *We say that the SISO dynamical system* (3) *($m = p = 1$) admits an input to state static characteristic $\mathbf{k}_x(.) : \mathbb{U} \to \mathbb{X}$ if, for each constant input $u \in \mathbb{U}$, there exists a unique globally asymptotically stable equilibrium noted $\mathbf{k}_x(u)$.*

Definition 2 (Angeli and Sontag, 2003)**.** *SISO system with an input-state characteristic and with a continuous output map $y = h(\mathbf{x})$ has an input to output characteristic defined as the composite function $k_y(u) = (h \circ \mathbf{k}_x)(u)$.*

Note that, if the system (3) (with $m = p = 1$) is cooperative and admits a *static input-state characteristic* $\mathbf{k}_x$ and *static input-output characteristic* k_y, then $\mathbf{k}_x$ and k_y must be increasing with respect to u, viz.

$$\forall\ (u_1,u_2) \in \mathbb{U}^2,\ u_1 \geq u_2 \quad \Leftrightarrow \quad \begin{array}{l} \mathbf{k}_x(u_1) \geq \mathbf{k}_x(u_2), \\ k_y(u_1) \geq k_y(u_2). \end{array}$$

Assumptions. Consider the non-monotone autonomous system given by (6)

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}), \tag{6}$$

and let us state the following assumptions,

- $\mathcal{H}_5$: Any state trajectory generated by system (6) is bounded.
- $\mathcal{H}_6$: System (6) is decomposable into an open loop SISO monotone control system (7)

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x},u) \\ y(t) & = & h(\mathbf{x}), \end{array} \right. \tag{7}$$

  closed by a monotone decreasing feedback law $f_b : y \longrightarrow u$ as depicted in Figure 2.

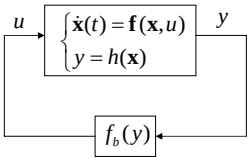


Figure 2: System (6) in closed loop configuration.

- $\mathcal{H}_7$: Open loop system (7) admits a well-defined static *input-output characteristic* $k_y(.)$.

Then, we can introduce the negative feedback theorem.

Theorem 1. *Let* (8) *be a discrete scalar dynamical system associated to the continuous non-monotone system* (6)

$$u_{j+1} = (f_b \circ k_y)(u_j). \tag{8}$$

If this iteration has a globally attractive fixed point u^* on an open interval $\mathcal{U}_x$, then the autonomous system (6)*, provided that the assumptions $\mathcal{H}_5$, $\mathcal{H}_6$ and $\mathcal{H}_7$ are satisfied, has a globally attracting steady state $\mathbf{x}^* = \mathbf{k}_x(u^*)$.*

Proof. See (Angeli and Sontag, 2003).

Hereafter, we give proofs of our main results stated in subsections 2.1 and 2.2.

4 PROOF OF THE MAIN RESULTS

In this section, we prove that propositions 1 and 2 are consequences of Theorem 1. We start with the irreversible metabolic pathways, for which the *static input-state characteristic* of its monotone part is easier to establish. Then we will focus on the reversible pathways.

4.1 Irreversible Structure

In this subsection we will show that the technical Proposition 2 is a consequence of Theorem 1.

Checking Assumption $\mathcal{H}_5$. First of all, let us prove the boundedness of the controlled enzyme E_1 which is governed by the following differential equation

$$\dot{E}_1 = g(x_n) - \mu E_1. \tag{9}$$

By definition we know that $g(.)$ is bounded, viz. $\forall x_n,\ g(x_n) \in (0, g_{max}]$. Then, for any x_n the solution $E_1(t)$ of (9) is framed by

$$\check{E}_1(t) \leq E_1(t) \leq \hat{E}_1(t),$$

where $\check{E}_1(t)$ and $\hat{E}_1(t)$ are respectively the solutions of the following stable linear differential equations

$$\dot{\check{E}}_1 = -\mu \check{E}_1 \text{ and } \dot{\hat{E}}_1 = g_{max} - \mu \hat{E}_1.$$

Thus, there exists

$$\overline{E}_1 > 0 \mid \forall t \geq 0,\ E_1(t) \leq \overline{E}_1.$$

Now, consider the first differential equation of (2)

$$\begin{array}{rcl} \dot{x}_2 & = & E_1 f_1(\bar{x}_1, x_n) - E_2 f_2(x_2) \\ & \leq & \overline{E}_1 M_1 - E_2 f_2(x_2). \end{array}$$

We know that $f_2(.)$ is positive increasing and bounded. Then if

$$\overline{E}_1 M_1 \leq E_2 M_2, \tag{10}$$

there exists $x_2^\star$ such that $E_2 f_2(x_2^\star) = \overline{E}_1 M_1$, and we obtain

$$\forall x_2 > x_2^\star,\ \dot{x}_2 \leq 0,$$

namely the solution $x_2(t)$ decreases towards $x_2^\star$ and then the metabolite concentration x_2 is bounded. In addition, for any initial condition $x_2(t_0)$ there exists $t^\star \geq t_0$ such that,

$$\forall t \geq t^\star,\ E_2 f_2(x_2(t)) \leq E_2 f_2(x_2^\star) = \overline{E}_1 M_1.$$

To proof the boundedness of the remainder metabolite concentrations, we use mathematical induction. Assume that x_i is bounded, viz. the following inequality is satisfied

$$\overline{E}_1 M_1 \leq E_i M_i, \tag{11}$$

and there exists $(t^\star, x_i^\star)$ such that for all $t \geq t^\star$

$$E_i f_i(x_i(t)) \leq E_i f_i(x_i^\star) = \cdots = E_2 f_2(x_2^\star) = \overline{E}_1 M_1.$$

Then, for $t \geq t^\star$ the dynamics of the next metabolite concentration x_{i+1} is bounded by

$$\begin{array}{rcl} \dot{x}_{i+1} & = & E_i f_i(x_i) - E_{i+1} f_{i+1}(x_{i+1}) \\ & \leq & E_i f_i(x_i^\star) - E_{i+1} f_{i+1}(x_{i+1}) \\ & = & \overline{E}_1 M_1 - E_{i+1} f_{i+1}(x_{i+1}). \end{array}$$

Hence we show, with the same way used to prove the boundedness of x_2, that inequality (12) guarantees the boundedness of the metabolite concentration x_{i+1}.

$$\overline{E}_1 M_1 \leq E_{i+1} M_{i+1}. \tag{12}$$

Therefore $\mathcal{H}_4$ guarantees the boundedness of the all state trajectories generated by (2), namely $\mathcal{H}_5$.

Checking Assumption $\mathcal{H}_6$. System (2) is not monotone. However, we can regard it as a cooperative controlled system (13), which has a triangular Jacobian matrix $DF(\mathbf{x})$ with nonnegative off-diagonal entries, closed by a negative feedback (14),

- Open loop (cooperative system)

$$\left\{ \begin{array}{rcl} \dot{x}_2 & = & E_1 f_1(\bar{x}_1, g^{-1}(u)) - E_2 f_2(x_2) \\ \dot{x}_3 & = & E_2 f_2(x_2) - E_3 f_3(x_3) \\ \vdots & \vdots & \vdots \quad \vdots \\ \dot{x}_n & = & E_{n-1} f_{n-1}(x_{n-1}) - E_n f_n(x_n) \\ \dot{E}_1 & = & u - \mu E_1 \\ y & = & x_n \end{array} \right. \tag{13}$$

- Negative feedback

$$u = g(y) \tag{14}$$

where $g^{-1}(.)$ is the reciprocal function of $g(.)$ and $u \in (0, g_{max})$ since $g(.) \in (0, g_{max}]$. This verifies $\mathcal{H}_6$.

Checking Assumption $\mathcal{H}_7$. The *static input-state characteristic* $\mathbf{k}_x(u)$ of (13) is computed at steady states corresponding to constant inputs u. Thus, we vanish all the time derivatives of (13) to obtain:

$$\mathbf{k}_x^T(u) = \left[f_2^{-1}\left(\tfrac{f_1(\bar{x}_1, g^{-1}(u))u}{E_2\mu}\right), \ldots, f_n^{-1}\left(\tfrac{f_1(\bar{x}_1, g^{-1}(u))u}{E_n\mu}\right), \tfrac{u}{\mu} \right] \tag{15}$$

and for the *static input-output characteristic* we have:

$$k_y(u) = f_n^{-1}\left(\frac{f_1(\bar{x}_1, g^{-1}(u))u}{E_n\mu}\right) \tag{16}$$

Since functions $f_i(.), i = 2, \ldots, n$ are bounded, the existence of (15) is conditioned by the following inequalities:

$$\forall i,\ \forall u \in (0, g_{max}),\ \ \frac{f_1(\bar{x}_1, g^{-1}(u))u}{E_i\mu} \leq M_i$$

which are always true if assumption $\mathcal{H}_4$ is verified. Moreover, as system (13) is cooperative, both static characteristics ((15) and (16)) are increasing with respect to u.

Now, to prove that for each constant input $u \in (0, g_{max})$ there exists a unique globally asymptotically stable equilibrium point $\mathbf{k}_x(u)$ for (13), we consider separately the dynamics of the enzymatic reactions $(\dot{x}_2, \ldots, \dot{x}_n)^T$ and that of the genetic regulation $\dot{E}_1$.

- The growth rate μ of the bacteria is constantly positive. Then for each constant input u all the solutions generated by the dynamics of the genetic regulation converge asymptotically to $\frac{u}{\mu}$.
- The Jacobian matrix $DF(\mathbf{x})$ of the dynamics of the enzymatic reactions is a lower triangular matrix with nonnegative off-diagonal entries and real

negative eigenvalues. Then $-DF(\mathbf{x})$ is a *M-Matrix* (Berman and Plemmons, 1994) and there exists a diagonal matrix $\mathbf{P} = diag(p_1,\dots,p_n)$ with $p_i > 0$ such that

$$\exists \varepsilon > 0,\ \forall \mathbf{x},\ \ \mathbf{P}DF(\mathbf{x}) + DF(\mathbf{x})^T\mathbf{P} < -\varepsilon \mathbf{I}_{n-1}. \tag{17}$$

Consequently, we can state that the dynamics of the enzymatic reactions have a well defined quadratic Lyapunov function:

$$V(\mathbf{z}) = \mathbf{z}^T\mathbf{P}\mathbf{z},$$

where $\mathbf{z} = \mathbf{x} - \mathbf{x}^*$, $x_i^* = (k_x(u))_i,\ i = 2,\dots,n$ and

$$\begin{aligned} \dot{V}(\mathbf{z}) &= \mathbf{z}^T\mathbf{P}[\mathbf{f}(\mathbf{x},u) - \mathbf{f}(\mathbf{x}^*,u)] \\ &= \mathbf{z}^T \textstyle\int_0^1 \mathbf{P}DF(\lambda\mathbf{z}+\mathbf{x})\mathbf{z}d\lambda \\ &= \tfrac{1}{2}\mathbf{z}^T \textstyle\int_0^1 (\mathbf{P}DF(\lambda\mathbf{z}+\mathbf{x}) + DF(\lambda\mathbf{z}+\mathbf{x})^T\mathbf{P})d\lambda\mathbf{z} \\ &\leq -\tfrac{1}{2}\varepsilon \parallel \mathbf{z} \parallel^2 \end{aligned} \tag{18}$$

Hence, for each constant input $u \in (0, g_{max})$, any solution of the open loop system (13) converges asymptotically to the unique steady state given by (15). This verifies assumption $\mathcal{H}_7$.

Now, to complete the proof that the Proposition 2 is consequence of Theorem 1, we will show that assumption $\mathcal{H}_3$ implies the global attractivity of the following scalar discrete dynamical system

$$u_{j+1} = g(f_n^{-1}(\frac{f_1(\bar{x}_1, g^{-1}(u_j))u_j}{E_n\mu})) \tag{19}$$

To do so, *(i)* we prove existence and unicity of a fixed point u^* for (19); and *(ii)* we give convenient condition which guarantee its global attractivity.

Existence and Unicity. To prove this property, it is sufficient to show that the curves of the functions $g^{-1}(u)$ and $k_y(u)$ have a unique intersection point over the interval $(0, g_{max})$. Since:

- $k_y(u)$ is is monotone increasing with respect to u and for $u = 0$, $k_y(0) \geq 0$ and $\lim_{u\to g_{max}} k_y(u) = +\infty$
- $g^{-1}(u)$ is monotone decreasing with respect to u and $\lim_{u\to 0} g^{-1}(u) = +\infty$ and for $u = g_{max}$, $g^{-1}(g_{max}) = 0$,

then the two curves have a unique intersection point u^* (see Figure 3) which present the unique fixed point of (19).

Global Attractivity. Denote by T^2 the composite function

$$T^2(u) = (T \circ T)(u),$$

where $T(u) = (g \circ k_y)(u)$. The following proposition gives the necessary and sufficient condition for the global attractivity of the unique equilibrium of (19).

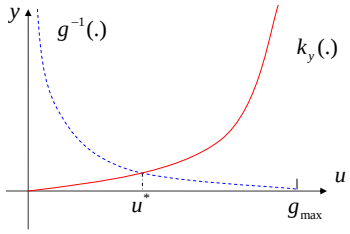


Figure 3: Graphical proof of the existence and unicity of the fixed point u^* for the discrete system (19).

Proposition 4. *If u^* is also the unique fixed point of $T^2(u)$ on $(0, g_{max})$, That is*

$$\forall u \in (0, g_{max}), T^2(u) = u \Leftrightarrow u = u^*, \tag{20}$$

then (19) *converges to its unique fixed point.*
Proof: see (Enciso and Sontag, 2006).
In practice, we can check condition (20) by graphical test ($\mathcal{H}_3$). Indeed, if the graph of $T(u)$ and that of $T^{-1}(u)$ have a unique intersection point u^* over $(0, g_{max})$, then the composite function $T^2(u)$ has unique fixed point u^*. This completes the proof.

4.2 Reversible Structure

Now, consider the reversible metabolic pathways (1) and we prove that Proposition 1 is a consequence of Theorem 1.

Checking Assumption $\mathcal{H}_5$: First, note that the enzyme E_1 is bounded (see proof given in subsection 4.1). Now, to analyze the boundedness of all the metabolite concentrations of (1), we proceed by step and we show that if any metabolite concentration x_i is bounded then the metabolite concentration x_{i-1} is also bounded. We start by x_2, and we consider the first differential equation of (1),

$$\begin{aligned} \dot{x}_2 &= E_1 f_1(\bar{x}_1, x_2, x_n) - E_2 f_2(x_2, x_3) \\ &\leq \overline{E}_1 f_1(\bar{x}_1, x_2, 0) - E_2 f_2(x_2, x_3). \end{aligned}$$

We assume that x_3 is bounded ($\forall t > 0, x_3(t) \leq \bar{x}_3$), then by definition there exists $x_2^\star$ such that:

$$f_1(\bar{x}_1, x_2^\star, 0) = 0 \ \text{ and } \ f_2(x_2^\star, \bar{x}_3) \geq 0,$$

and thus at $x_2^\star$ we obtain $\dot{x}_2 \leq 0$. Hence the threshold $x_2^\star$ is repulsive, and so we have proved that the boundedness of x_3 implies the boundedness of x_2.

Now, for any metabolite concentration x_i, $i \in \{3,\dots,n-1\}$ we have x_{i-1} bounded with bound $\bar{x}_{i-1}$, and we assume that x_{i+1} is bounded with bound $\bar{x}_{i+1}$. Then the dynamics of x_i is bounded by:

$$\begin{aligned} \dot{x}_i &= E_{i-1} f_{i-1}(x_{i-1}, x_i) - E_i f_i(x_i, x_{i+1}) \\ &\leq E_{i-1} f_{i-1}(\bar{x}_{i-1}, x_i) - E_i f_i(x_i, \bar{x}_{i+1}), \end{aligned}$$

and by definition we have

$$\exists x_i^\star \mid f_{i-1}(\bar{x}_{i-1}, x_i^\star) \leq 0 \text{ and } f_i(x_i^\star, \bar{x}_{i+1}) \geq 0.$$

Hence for $x_i^\star$ we obtain $\dot{x}_i \leq 0$, and so the threshold $x_i^\star$ is repulsive. Thus, we have proved that $\forall i \in \{2,\ldots,n-1\}$ the boundedness of x_{i+1} implies the boundedness of x_i. Lastly, consider the dynamics of the concentration of the end product x_n,

$$\begin{array}{rcl} \dot{x}_n & = & E_{n-1} f_{n-1}(x_{n-1}, x_n) - E_n f_n(x_n) \\ & \leq & E_{n-1} M_{n-1} - E_n f_n(x_n). \end{array}$$

Since $f_n(.)$ is positive increasing and bounded with respect to x_n, it is clear that if $E_{n-1}M_{n-1} \leq E_n M_n$ we obtain

$$\exists \bar{x}_n \mid \forall x_n \geq \bar{x}_n \Rightarrow \dot{x}_n \leq 0$$

independently of the values of x_{n-1}. Consequently, if $\mathcal{H}_2$ is true, then all the state trajectories generated by (1) are bounded and so assumption $\mathcal{H}_5$ is verified.

Checking assumption $\mathcal{H}_6$: As in the case of the irreversible metabolic pathways, structure (1) is not monotone. Nevertheless, we can decompose it into an open loop cooperative controlled system (21), which has tridiagonal Jacobian matrix $DF(x)$ with nonnegative off-diagonal entries, closed by a negative feedback (22).

- Open loop (cooperative system)

$$\left\{ \begin{array}{rcl} \dot{x}_2 & = & E_1 f_1(\bar{x}_1, x_2, g^{-1}(u)) - E_2 f_2(x_2, x_3) \\ \dot{x}_3 & = & E_2 f_2(x_2, x_3) - E_3 f_3(x_3, x_4) \\ \vdots & \vdots & \vdots \qquad\qquad \vdots \\ \dot{x}_n & = & E_{n-1} f_{n-1}(x_{n-1}, x_n) - E_n f_n(x_n) \\ \dot{E}_1 & = & u - \mu E_1 \\ y & = & x_n \end{array} \right. \tag{21}$$

- Negative feedback

$$u = g(y) \tag{22}$$

where $g^{-1}(.)$ and $g(.)$ are the same as in the irreversible case and also $u \in (0, g_{max})$. Hence, assumption $\mathcal{H}_6$ is intrinsically satisfied.

Checking assumption $\mathcal{H}_7$: In the reversible context, build the *static input-state characteristic* is not explicit as in the irreversible case. However, to establish this characteristic we use the monotonicity property of all reaction rates $f_i(.,.),\ i \in \{1,\ldots,n\}$. First, we show that at steady state there exists a binary relation between each metabolite concentration $x_i,\ i \in \{3,\ldots,n\}$ and x_2. Second, we show that the metabolite concentration x_2 is an increasing function of the constant input u.

- Consider the dynamics corresponding to the last pool X_n. Since: *(i)* the function $f_n(x_n)$ is monotone increasing in x_n with $f_n(0) = 0$, *(ii)* the function $f_{n-1}(x_{n-1}, x_n)$ is decreasing in x_n and *(iii)* for any x_{n-1} there exists x_n^* such that $f_{n-1}(x_{n-1}, x_n^*) = 0$,

$$\forall x_{n-1},\ \exists x_n \mid E_{n-1} f_{n-1}(x_{n-1}, x_n) = E_n f_n(x_n).$$

In other words, we can say that there exists a monotone increasing function $H_n(.)$ with respect to x_{n-1} such that:

$$x_n = H_n(x_{n-1}). \tag{23}$$

- According to the previous stage, we can write

$$f_n(x_n) = f_n(H_n(x_{n-1})).$$

Thus, since $H_n(.)$ is monotone increasing in x_{n-1}, $f_n(.)$ is also monotone increasing in x_{n-1}. Now, consider the dynamics of the pool X_{n-1}. By definition $f_{n-2}(x_{n-2}, x_{n-1})$ is decreasing in x_{n-1} and for any x_{n-2} there exists x_{n-1}^* such that $f_{n-2}(x_{n-2}, x_{n-1}^*) = 0$. Hence, we deduce: $\forall x_{n-2}$,

$$\exists x_{n-1} \mid E_{n-2} f_{n-2}(x_{n-2}, x_{n-1}) = E_n f_n(H_n(x_{n-1})).$$

Therefore, there exists a monotone increasing function $H_{n-1}(.)$ with respect to x_{n-2} such that:

$$x_{n-1} = H_{n-1}(x_{n-2}) \text{ and } x_n = H_n(H_{n-1}(x_{n-2})). \tag{24}$$

- Then we repeat this reasoning to obtain at steady state the following relations between x_2 and all the metabolic concentrations $x_i,\ i \in \{3,\ldots,n\}$:

$$\begin{array}{l} x_3 = H_3(x_2) \\ x_4 = H_4(H_3(x_2)) \\ \vdots \\ x_n = H_n(H_{n-1}(\ldots H_3(x_2)\ )) \end{array} \tag{25}$$

where all H_i are increasing functions.

- Lastly, the enzyme's dynamics vanished while $E_1 = \frac{u}{\mu}$. Thus, it is possible to build at the steady state a monotone relationship between the concentration of the pool X_2 and the input u. Indeed, as we have shown previously, *(i)* the monotone decreasing property of the function $\frac{u}{\mu} f_1(\bar{x}_1, x_2, g^{-1}(u))$ in x_2, *(ii)* the monotone increasing property of the function $f_n(H_n(H_{n-1}(\ldots H_3(x_2)\ ))$ in x_2, and *(iii)* the existence of x_2^* such that $f_1(\bar{x}_1, x_2^*, g^{-1}(u)) = 0$ allow to state: $\forall u,\ \exists x_2$ such that,

$$\frac{u}{\mu} f_1(\bar{x}_1, x_2, g^{-1}(u)) = f_n(H_n(H_{n-1}(\ldots H_3(x_2)\ ))$$

Then, at the steady state there exists a monotone increasing function $H_2(.)$ with respect to u such that:

$$x_2 = H_2(u). \tag{26}$$

Hence, the *static input-state characteristic* of the system (21) is given by:

$$\mathbf{k}_x^T(u) = \left[H_2(u), H_3(H_2(u)), \ldots, H_n(H_{n-1}(\ldots H_2(u)\ )), \frac{u}{\mu}\right] \tag{27}$$

and its *input-output characteristic* is obtained by the composition law between (27) and the output equation of (21),

$$k_y(u) = H_n(H_{n-1}(\ldots H_2(u)\ )). \tag{28}$$

Now, we must prove that for each constant input u the vector $[\mathbf{x}^{*T}, \frac{u}{\mu}] = \mathbf{k}_x^T(u)$ is the globally asymptotically stable equilibrium point for the open loop system (21). To do so, we use the same analysis as in the irreversible case. First, we separate the two dynamics (enzymatic reaction, genetic regulation) and we deduce that for each constant input u all the solutions generated by the dynamics of the genetic regulation ($\dot{E}_1$) converge to $\frac{u}{\mu}$. Second, hypothesis $\mathcal{H}_1$ claims the existence of Tridiagonal Hurwitz matrix $\mathbf{Q}$ with nonnegative off-diagonal entries such that for all $\mathbf{x}$ the Jacobian matrix $DF(\mathbf{x})$ of the dynamics of the enzymatic reactions $(\dot{x}_2, \ldots, \dot{x}_n)$ is bounded by, $DF(\mathbf{x}) \leq \mathbf{Q}$. Then there exists a diagonal matrix $\mathbf{N} = diag(n_1, \ldots, n_n)$ with $n_i > 0$ and a real number $\varepsilon > 0$ such that $\forall \mathbf{x}$

$$\begin{array}{rcl} \mathbf{N}DF(\mathbf{x}) + DF^T(\mathbf{x})\mathbf{N} & \leq & \mathbf{NQ} + \mathbf{Q}^T\mathbf{N} \\ & \leq & -\varepsilon \mathbf{I}_{n-1} \end{array} \tag{29}$$

because $-\mathbf{Q}$ is a *M-Matrix* (Berman and Plemmons, 1994). Thus, the dynamics of the enzymatic reactions admits as Lyapunov function the quadratic form

$$V(\mathbf{z}) = \mathbf{z}^T\mathbf{N}\mathbf{z},$$

where $\mathbf{z} = \mathbf{x} - \mathbf{x}^*$. See previous demonstration of (18). Therefore, under assumption $\mathcal{H}_1$, relation (27) gives the globally asymptotically stable steady state of the open loop system (21) for each constant input u. This verifies assumption $\mathcal{H}_7$.

Finally, as we have shown in the context of irreversible metabolic pathways (here $\mathbf{k}_x(.)$, $k_y(.)$ and $g^{-1}(.)$ have the same properties with respect to u as in the irreversible context), we can check the global convergence of the following scalar discrete time dynamical system

$$u_{j+1} = g(H_n(H_{n-1}(\ldots H_2(u_j)\ )), \tag{30}$$

to its unique fixed point $u^* \in (0, g_{max})$ by the same graphical test stated in assumption ($\mathcal{H}_3$). This completes the proof that Proposition 1 is a consequence of Theorem 1.

5 CONCLUSIONS

We have used in this paper the negative feedback theorem of monotone control SISO systems theory, to give technical propositions which prove global attractivity of linear metabolic pathways. For future works, we will consider the stability analysis for dynamical systems through monotone control MIMO systems. That will allow us to tackle the stability issue for complex bacterial metabolic networks.

REFERENCES

Angeli, D. and Sontag, E. D. (2003). Monotone control systems. *IEEE transactions on automatic control*, 48:1684–1698.

Angeli, D. and Sontag, E. D. (2004). Multi-stability in monotone input/output systems. *Systems and Control Letters*, 51:185–202.

Angeli, D. and Sontag, E. D. (2008). Oscillations in i/o monotone systems under negative feedback. *IEEE transactions on automatic control*, 55:166–176.

Arcak, M. and Sontag, E. D. (2006). Diagonal stability of a class of cyclic systems and its connection with the secant criterion. *Automatica*, 42:1531–1537.

Berman, A. and Plemmons, R. (1994). *Nonnegative Matrices in the Mathematical Sciences*. Society for Industrial and applied Mathematics, Philadelphia.

Enciso, G. A. and Sontag, E. D. (2006). Global attractivity, i/o monotone small-gain theorems, and biological delay systems. *Discrete and Continuous Dynamical Systems*, 14:549–578.

Goelzer, A., Bekkal-Brikci, F., Martin-Verstraete, I., Noirot, P., Bessiéres, P., Aymerich, S., and Fromion, V. (2008). Reconstruction and analysis of the genetic and metabolic regulatory networks of the central metabolism of bacillus subtilis. *BMC Systems Biology*, doi:10.1186/1752-0509-2-20.

Gollnick, P., Babitzke, P., Antson, A., and CA, C. (2005). Complexity in regulation of tryptophan biosynthesis in bacillus subtilis. *Annual review of genetics*, 39:47–68.

Grundy, F., Lehman, S., and Henkin, T. (2003). The l box regulon: lysine sensing by leader rnas of bacterial lysine biosynthesis genes. *PNAS*, 100(21):12057–12062.

Sanchez, L. (2009). Global asymptotic stability of the goodwin system with repression. *Nonlinear analysis : real world applications*, 10(4):2151–2156.

Smith, H. L. (1995). *Monotone dynamical systems: An Introduction to the theory of competitive and cooperative systems*. Mathematical Surveys and Monographs Vol. 41, AMS, Providence, RI.

Tyson, J. and Othmer, H. (1978). The dynamics of feedback control circuits in biological pathways. In *R. Rosen, and F.M. Snell (Eds.), Progress in theoretical biology (Vol. 5)*, pages 1–62, New York: Academic Press.

Wang, L., de Leenheer, P., and Sontag, E. (2008). Global stability for monotone tridiagonal systems with negative feedback. In *Proc. IEEE Conf. Decision and Control CDC*, pages 4091–4096, Cancun,Mexico.

SIMPLE DERIVATION OF A STATE OBSERVER OF LINEAR TIME-VARYING DISCRETE SYSTEMS

Yasuhiko Mutoh
Department of Applied Science and Engineering, Sophia University, 7-1, Kioicho, Chiyoda-ku, Tokyo, Japan
y_mutou@sophia.ac.jp

Keywords: Pole Placement, State Observer, Linear Time-Varying System, Discrete System.

Abstract: In this paper, a simple calculation method to derive the Luenberger observer for linear time-varying discrete systems is presented. For this purpose, the simple design method of the pole placement for linear time-varying discrete systems is proposed. It is shown that the pole placement controller can be derived simply by finding some particular "output signal" such that the relative degree from the input to this new output is equal to the order of the system. Using this fact, the feedback gain vector can be calculated directly from plant parameters without transforming the system into any standard form.
Then, this method is applied to the design of the observer, i.e., because of the duality of linear time-varying discrete system, the state observer can be derived by simple calculations.

1 INTRODUCTION

The design of the state observer for linear time-varying discrete systems is well established. As for the continuous case, the condition for a system to be a state observer is very simple. However, different from the time-invariant case, calculation procedure to obtain the observer gain is not straightforward. This paper gives a simple calculation method to design the state observer for linear time-varying discrete systems.

Since the design of the observer is based on the pole placement technique, simplified calculation method to derive the pole placement feedback gain vector for linear time-varying discrete systems is considered first. We define the pole placement of linear time-varying discrete systems as follows. The problem is to find a time-varying state feedback gain for linear discrete time-varying discrete system, so that the closed loop system is equivalent to the time-invariant system with desired poles.

Usually, the pole placement design procedure needs the change of variable to the Flobenius standard form, and hence, is very complicated. To simplify this procedure, it will be shown that the pole placement controller can be derived simply by finding some particular "output signals" such that the relative degree from the input to this output is equal to the order of the system [4]. Using this fact, the feedback gain vector can be calculated directly from plant parameters without transforming the system into any standard form.

Because of the duality of the linear discrete time-varying system, the simplified pole placement technique can be applied to the design of the state observer for linear discrete time-varying discrete systems.

In the sequel, the simple pole placement technique is proposed in Section 2, and then, this method is used to the observer design problem in Section 3.

2 POLE PLACEMENT OF LINEAR DISCRETE TIME-VARYING SYSTEMS

Consider the following linear time-varying discrete system with a single input.

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{1}$$

Here, $x \in R^n$ and $u \in R^1$ are the state variable and the input signal respectively. $A(k) \in R^{n\times n}$ and $b(k) \in R^n$ are time-varying parameter matrices. The problem is to find the state feedback

$$u = h^T(k)x(k) \tag{2}$$

which makes the closed loop system equivalent to the time invariant linear system with arbitrarily stable poles.

Definition 1. The system (1) is called completely reachable in step n from the origin, if for any $x_1 \in R^n$, there exists a finite input $u(m)$ $(m = k, \cdots, k+n-1)$ such that $x(k) = 0$ and $x(k+n) = x_1$.

Lemma 1. The system (1) is completely reachable in step n from the origin, if and only if

$$\begin{aligned} &rank\,[\, b(k+n-1), \Phi(k+n,k+n-1)b(k+n-2), \\ &\quad \cdots, \Phi(k+n,k+1)b(k)\,] \\ &\qquad = rank\, U_R(k) = n, \qquad {}^{\forall}k \end{aligned} \tag{3}$$

where $\Phi(i,j)$ is the transition matrix from $k = j$ to $k = i$, i.e.,

$$\Phi(i,j) = A(i-1)A(i-2)\cdots A(j) \quad i > j \tag{4}$$

$\nabla\nabla$

Now, consider the problem of finding a new output signal $y(k)$ such that the relative degree from $u(k)$ to $y(k)$ is n. Here, $y(k)$ has the following form.

$$y(k) = c^T(k)x(k) \tag{5}$$

Then, the problem is to find a vector $c(k) \in R^n$ that satisfies this condition.

Lemma 2. The relative degree from u to y defined by (5) is n, if and only if

$$\begin{aligned} &c^T(k+1)b(k) = 0 \\ &c^T(k+2)\Phi(k+2,k+1)b(k) = 0 \\ &\qquad \vdots \\ &c^T(k+n-1)\Phi(k+n-1,k+1)b(k) = 0 \\ &c^T(k+n)\Phi(k+n,k+1)b(k) = 1 \end{aligned} \tag{6}$$

(Here, $c^T(k+n)\Phi(k+n,k+1)b(k) = 1$ without loss of generality.) $\nabla\nabla$

Proof : This is obvious by checking $y(k+1)$, $\cdots$, $y(k+n)$.

If the system (1) is completely reachable in step n, there exists a vector $c(k)$ such that the relative degree from $u(k)$ to $y(k) = c^T(k)x(k)$ is n. And, from (6), such a vector, $c(k)$, is obtained by

$$\begin{aligned} c^T(k) &= [0, \cdots, 0, 1]\,[\, b(k-1), \Phi(k,k-1)b(k-2), \\ &\qquad \cdots, \Phi(k,k+1-n)b(k-n)\,]^{-1} \\ &= [0, 0, \cdots, 1]U_R^{-1}(k-n) \end{aligned} \tag{7}$$

The next step is to derive the state feedback for the arbitrary pole placement.

The new output, $y(k) = c^T(k)x(k)$, with $c(k)$ obtained by (7), satisfies the following equations.

$$\begin{aligned} y(k) &= c^T(k)x(k) \\ y(k+1) &= c^T(k+1)\Phi(k+1,k)x(k) \\ &\vdots \\ y(k+n-1) &= c^T(k+n-1)\Phi(k+n-2,k)x(k) \\ y(k+n) &= c^T(k+n)\Phi(k+n-1,k)x(k) + u(k) \end{aligned} \tag{8}$$

Let $q(z)$ be a desired stable polynomial of z-operator, i.e.,

$$q(z) = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_0 \tag{9}$$

By multiplying $y(k+i)$ by α_i $(i = 0, \cdots, n-1)$ and then summing them up, the following equation is obtained from (8).

$$q(p)y(k) = d^T(k)x(k) + u(k) \tag{10}$$

where $d(k) \in R^n$ is defined by the following.

$$\begin{aligned} d^T(k) &= [\, \alpha_0, \alpha_1, \cdots, \alpha_{n-1}, 1\,] \\ &\quad \times \begin{bmatrix} c^T(k) \\ c^T(k+1)\Phi(k+1,k) \\ \vdots \\ c^T(k+n)\Phi(k+n,k) \end{bmatrix} \end{aligned} \tag{11}$$

Hence, the state feedback,

$$u = -d^T(k)x(k) + r(k) \tag{12}$$

makes the closed loop system as follows.

$$q(z)y(k) = r(k) \tag{13}$$

where $r(k)$ is an external input signal.

This control system can be summarized as follows. The given system is

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{14}$$

and, using (4), (9), and (11) the state feedback for the pole placement is given by

$$u(k) = -d^T(k)x(k). \tag{15}$$

Then, the closed loop system becomes

$$x(k+1) = (A(k) - b(k)d^T(k))x(k). \tag{16}$$

Let $T(k)$ be the time varying matrix defined by

$$T(k) = \begin{bmatrix} c^T(k) \\ c^T(k+1)\Phi(k+1,k) \\ \vdots \\ c^T(k+n-1)\Phi(k+n-1,k) \end{bmatrix} \tag{17}$$

and define the new state variable $w(k)$ by the following equations.

$$x(k) = T(k)w(k), \qquad w = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{bmatrix} \tag{18}$$

Using the above, (16) is transformed into

$$\begin{aligned} & w(k+1) \\ & = T^{-1}(k+1)(A(k)-b(k)d^T(k))T(k)w(k) \\ & = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ \vdots & & & 1 \\ -\alpha_0 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix} w(k) \\ & = A^* w(k) \end{aligned} \tag{19}$$

This implies that the closed loop system is equivalent to the time invariant linear system which has the desired closed loop poles ($\det(zI-A^*) = q(z)$).

Theorem 2. If the system (1) is completely reachable in step n, then, the matrix for the change of variable, $T(k)$, given by (17) is nonsingular for all k. $\nabla\nabla$

Example 1.
Consider the following unstable system.

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{20}$$

where

$$\begin{aligned} A(k) &= \begin{bmatrix} 1 & 2+cos0.1k \\ 2+sin0.2k & 2 \end{bmatrix} \\ b(k) &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{21}$$

From (7), $c^T(k)$ is obtained as follows.

$$\begin{aligned} c^T(k) &= [\,0,\,1\,][\,b(k-1),\,A(k-1)b(k-2)\,]^{-1} \\ &= \begin{bmatrix} \dfrac{1}{2+cos0.1(k-1)} & 0 \end{bmatrix} \end{aligned} \tag{22}$$

The purpose is to design the state feedback so that the closed loop system is equivalent to the linear time invariant system with $\lambda_1 = 0.4$ and $\lambda_2 = 0.5$ as its closed loop poles. This implies that the desired closed loop characteristic polynomial is

$$q(z) = z^2 + 0.9z + 0.2.$$

From (11),

$$\begin{aligned} d^T(k) &= [\,0.2,\,0.9,\,1\,] \\ &\quad \times \begin{bmatrix} c^T(k) \\ c^T(k+1)A(k) \\ c^T(k+2)A(k+1)A(k) \end{bmatrix} \\ &= [\; d_1(k) \quad d_2(k) \;] \end{aligned} \tag{23}$$

In the above, $d_1(k)$ and $d_2(k)$ are given by

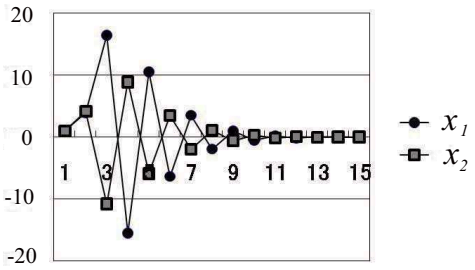


Figure 1: Responce of the state variable (x) of the system.

$$\begin{aligned} d_1(k) &= \frac{0.2}{\gamma(k-1)} + \frac{0.9}{\gamma(k)} + \frac{1}{\gamma(k+1)} + 2 + sin0.2k \\ d_2(k) &= 0.9 + \frac{\gamma(k)}{\gamma(k+1)} + 2 \end{aligned}$$

where

$$\gamma(k) = 2 + cos0.1k$$

Fig.1 shows the simulation results.

3 STATE OBSERVER

In this section, we consider the design of the observer for the following linear time-varying system.

$$\begin{aligned} x(k+1) &= A(k)x(k) + b(k)u(k) \\ y(k) &= g^T(k)x(k) \end{aligned} \tag{24}$$

Here, $y(k) \in R$ is the output signal of this system. The problem is to design the full order state observer for (24). Consider the following system as a candidate of the observer.

$$\begin{aligned} \hat{x}(k+1) &= F(k)\hat{x}(k) + b(k)u(k) + h(k)y(k) \\ &= F(k)\hat{x}(k) + b(k)u(k) + h(k)g^T(k)x(k) \end{aligned} \tag{25}$$

where $F(k) \in R^{n\times n}$, and $h(k) \in R^n$. Define the state error $e(k) \in R^n$ by

$$e = x(k) - \hat{x}(k) \tag{26}$$

Then, $e(k)$ satisfies the following error equation.

$$\begin{aligned} e(k+1) &= F(k)e(k) + (A(k) - F(k) \\ &\quad -h(k)g^T(k))x(k) \end{aligned} \tag{27}$$

Hence, (25) is a state observer of (24) if $F(k)$ and $h(k)$ satisfy the following condition.

$$\begin{aligned} F(k) &= A(k) - h(k)g^T(k) \\ F(k) &: \text{ arbitrarily stable matrix} \end{aligned} \tag{28}$$

Then, the problem is to find $h(k)$ such that F(k) is equevalent to a constant matrix F^* with arbitrarily stable poles. Consider the pole placement control problem of the following system.

$$w(k+1) = A^T(-k)w(k) + g(-k)v(k) \tag{29}$$

where $w(k) \in R^n$ and $v(k) \in R^1$ are the state variable and an input signal.

Let $\Psi(i,j)$ be the state transient matrix of the system (29). Then, we have the following relation.

$$\Phi^T(i,j) = \Psi(-j,-i) \tag{30}$$

Definition 2. The system (24) is called completely obsermable in step n, if from $y(k), y(k+1), \cdots, y(k+n-1)$, the state, $x(k)$, can be determined uniquely for any k.

Lemma 3. The system (24) is completely observable in step n, if and only if

$$\begin{aligned} &rank \begin{bmatrix} g^T(k) \\ g^T(k+1)\Phi(k+1,k) \\ \vdots \\ g^T(k+n-1)\Phi(k+n-1,k) \end{bmatrix} \\ &= rank\, U_o(k) = n, \quad \forall k \end{aligned} \tag{31}$$

From the property of the duality of the time varying discrete system, if the pair $(A(k), g^T(k))$ is completely observable in step n, the pair $(A^T(-k), g(-k))$ is completely reachable in step n. Then, if the pair $(A(k), g^T(k))$ is completely observable in step n, the system (29) has a state feedback

$$v(k) = h^T(-k)w(k) \tag{32}$$

such that the closed loop system is equevalent to the linear time invariant system with arbitrarily stable poles.

This implies that for some state transformation matrix, $P(-k) \in R^n$,

$$\begin{aligned} &P^{-1}((k+1))(A^T(-k) - g(-k)h^T(-k))P(k) \\ &= F^{*T} \end{aligned} \tag{33}$$

where, F^{*T} is a constant matrix with arbitrarily stable poles. From this and the duality, we have the following equation.

$$\begin{aligned} &P^{-1}(-k)(A(k) - h(k)g^T(k))P(-(k+1)) \\ &= F^* \end{aligned} \tag{34}$$

Hence, using this $h(k)$, the state observer for the system (24) is obtained.

Example 2.
Consider the following system.

$$\begin{aligned} x(k+1) &= A(k)x(k) + b(k)u(k) \\ y(k) &= g^T(k)x(k) \end{aligned} \tag{35}$$

where

$$\begin{aligned} A(k) &= \begin{bmatrix} 0 & 1 \\ -0.7 & -(1.2 + 0.5cos0.4k) \end{bmatrix} \\ b(k) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad g^T(k) = [\,2,\ 1\,] \end{aligned} \tag{36}$$

The dual system matrices are as follows.

$$\begin{aligned} A^T(-k) &= \begin{bmatrix} 0 & -0.7 \\ 1 & -(1.2 + 0.5cos0.4(-k)) \end{bmatrix} \\ g(-k) &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} \end{aligned} \tag{37}$$

From (7), $c^T(-k)$ for the new output matrix is obtained as

$$\begin{aligned} c^T(-k) &= [\,0,\ 1\,][\, g(-(k-1)), \\ &\qquad A^T(-(k-1))g(-(k-2))\,]^{-1} \\ &= \frac{1}{\gamma(-(k-1))} \begin{bmatrix} -1 & 2 \end{bmatrix} \end{aligned} \tag{38}$$

where,

$$\begin{aligned} \gamma(-k) &= 4.7 - 2\lambda(k) \\ \lambda(k) &= 1.2 + 0.5cos0.4k. \end{aligned} \tag{39}$$

The purpose is to design the state feedback so that the closed loop system is equivalent to the linear time invariant system with $\lambda_1 = 0.3$ and $\lambda_2 = 0.4$ as its closed loop poles. This implies that the desired closed loop characteristic polynomial is

$$q(z) = z^2 + 0.7z + 0.12.$$

From (11), $d^T(-k)$ is calculated as follows.

$$\begin{aligned} d^T(-k) &= [\,0.12,\ 0.7,\ 1\,] \\ &\qquad \times \begin{bmatrix} c^T(-k) \\ c^T(-(k+1))A(-k) \\ c^T(-(k+2))A(-(k+1))A(-k) \end{bmatrix} \\ &= \begin{bmatrix} d_1(-k) & d_2(-k) \end{bmatrix} \end{aligned} \tag{40}$$

Here, $d_1(k)$ and $d_2(k)$ are

$$\begin{aligned} d_1(k) &= -\frac{0.12}{\gamma(k-1)} + \frac{0.7}{\gamma(k)} \\ &\qquad + \frac{1}{\gamma(k+1)}(0.2 - \lambda(k+1)) \\ d_2(k) &= \frac{0.12}{\gamma(k-1)} + \frac{0.7}{\gamma(k)}(0.2 - \lambda(k)) \\ &\qquad + \frac{1}{\gamma(k+1)}\{-0.2 - (0.2 + \lambda(k+1))\lambda(k)\} \end{aligned}$$

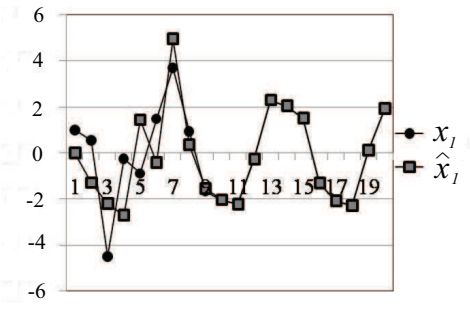


Figure 2: Responce of $x_1(k)$ and $\hat{x}_1(k)$.

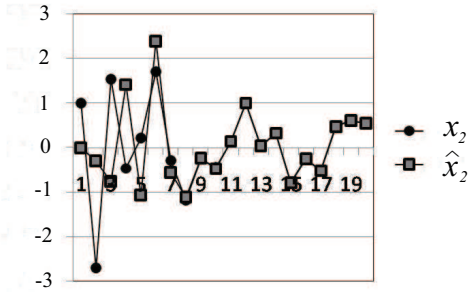


Figure 3: Responce of $x_2(k)$ and $\hat{x}_2(k)$.

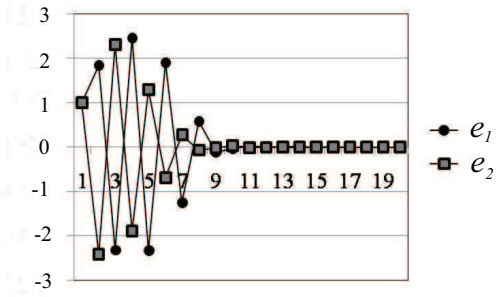


Figure 4: Responce of the estimation error ($e_1(k) = x_1(k) - \hat{x}_1(k)$, $e_2(k) = x_2(k) - \hat{x}_2(k)$).

Hence, the observer gain vector, $h(k)$, is obtained as

$$h(k) = -d(k) \tag{41}$$

and, using this $h(k)$, the observer is

$$\begin{aligned} \hat{x}(k+1) &= \{A(k) - h(k)g^T(k)\}\hat{x}(k) \\ &\quad + b(k)u(k) + h(k)y(k) \end{aligned} \tag{42}$$

Fig.2 $\sim$ 4 show the simulation results with $u(k) = 2cos(0.9k)$. The initial condition of the plant is $x_1(1) = x_2(1) = 1$.

4 CONCLUSIONS

In this paper, a simple design method for the state observer for linear time-varying discrete systems is proposed. We first proposed the simple derivation method of the pole placement state feedback gain for liner time-varying discrete system. Feedback gain can be calculated directly from the plant parameters without the transformation of the system into any standard form, which makes the design procedure very simple. This technique is applied to the observer design procedure using the duality of the linear time-varying system. The author appreciates the helpful comments of the anonymous reviewers.

REFERENCES

Kwakaernaak H. C., *Linear Optimal Control Systems*. Wiley-Interscience, 1972

Chi-Tsong Chen C., *Linear System Theory and Design (Third edition)*. Oxford University Press, 1999

T. Kailath C., *Linear Systems*. Prentice-Hall, 1980

Tse E., and Athans M., *Optimal Minimal-Order Observer-Estimators for Discrete Linear Time-Varying System*. IEEE, Transaction on AC, AC-15, 4, 416–426, 1970

Mutoh Y, *Simple Design of the State Observer for Linear Time-Varying Systems*. 6-th ICINCO, 2009

ROBUSTNESS OF ISS SYSTEMS TO INPUTS WITH LIMITED MOVING AVERAGE, WITH APPLICATION TO SPACECRAFT FORMATIONS

Esten Ingar Grøtli[a], Antoine Chaillet[b], Elena Panteley[c] and Jan Tommy Gravdahl[a]
[a] *Dept. of Eng. Cybernetics, NTNU, O. S. Bragstads plass 2D, 7491 Trondheim, Norway*
[b] *Univ. Paris Sud 11 - L2S - EECI - Supélec, 3 rue Joliot-Curie, 91192 Gif Sur Yvette, France*
[c] *CNRS - L2S, Supélec, 3 rue Joliot-Curie, 91192 Gif Sur Yvette, France*
grotli@itk.ntnu.no, antoine.chaillet@supelec.fr, panteley@lss.supelec.fr, tommy.gravdahl@itk.ntnu.no

Keywords: Robustness, ISS, Moving average of disturbances, Spacecraft formation.

Abstract: We provide a theoretical framework that fits realistic challenges related to spacecraft formation with disturbances. We show that the input-to-state stability of such systems guarantees some robustness with respect to a class of signals with bounded average-energy, which encompasses the typical disturbances acting on spacecraft formations. Solutions are shown to converge to the desired formation, up to an offset which is somewhat proportional to the considered moving average of disturbances. The approach provides a tighter evaluation of the disturbances' influence, which allows for the use of more parsimonious control gains.

1 INTRODUCTION

Spacecraft formation control is a relatively new and active field of research. Formations, characterized by the ability to maintain relative positions without real-time ground commands, are motivated by the aim of placing measuring equipment further apart than what is possible on a single spacecraft. This is desirable as the resolution of measurements often are proportional to the baseline length, meaning that either a large monolithic spacecraft or a formation of smaller, but accurately controlled spacecraft, may be used. Monolithic spacecraft architecture that satisfy the demand of resolution are often both impractical and costly to develop and to launch. On the other hand, smaller spacecrafts may be standardized and have lower development cost. In addition they may be of a lower collective weight and/or of smaller collective size such that cheaper launch vehicles can be used. There is also the possibility for them to piggyback with other commercial spacecraft. These advantages, come at the cost of an increased complexity. From a control design perspective, a crucial challenge is to maintain a predefined relative trajectory, even in presence of disturbances. Most of these disturbances are hard to model in a precise manner. Only statistical or averaged characteristics of the perturbing signals (*e.g.* amplitude, energy, average energy, etc.) are typically available. These perturbing signals may have diverse origins:

- *Intervehicle Interference*. In close formation or spacecraft rendezvous, thruster firings and exhaust gases may influence other spacecraft.
- *Solar wind and Radiation.* Particles and radiation expelled from the sun influence the spacecraft and are highly dependent on the solar activity (Wertz, 1978), which is difficult to predict (Hanslmeier et al., 1999).
- *Small Debris.* While large debris would typically mean the end of the mission, some space trash, including paint flakes, dust, coolant and even small needles[1], is small enough to "only" deteriorate the performance, see (NASA, 1999).
- *Micrometeroids.* The damages caused by micrometeroids may be limited due to their tiny size, but constant high velocity impacts also degrade the performance of the spacecraft through momentum transfer (Schäfer, 2006).
- *Gravitational Disturbances.* Even gravitational

[1] Project West Ford was a test carried out in the early 1960s, where 480 million needles were placed in orbit, with the aim to create an artificial ionosphere above the Earth to allow global radio communication, (Overhage and Radford, 1964).

models including higher order zonal harmonics, can only achieve a limited level of accuracy due to the shape and inhomogeneity of the Earth. In addition comes the gravitational perturbation due to other gravitating bodies such as the Sun and the Moon.

- *Actuator Mismatch.* There will commonly be a mismatch between the actuation computed by the control algorithm, and the actual actuation that the thrusters can provide. This mismatch is particularly present if the control algorithm is based on continuous dynamics, without taking into account pulse based thrusters.

Nonlinear control theory provides instruments to guarantee a prescribed precision in spite of these disturbances. Input-to-state stability (ISS) is a concept introduced in (Sontag, 1989), which has been thoroughly treated in the literature: see for instance the survey (Sontag, 2008) and references therein. Roughly speaking, this robustness property ensures asymptotic stability, up to a term that is "proportional" to the *amplitude* of the disturbing signal. Similarly, its integral extension, iISS (Sontag, 1998), links the convergence of the state to a measure of the *energy* that is fed by the disturbance into the system. However, in the original works on ISS and iISS, both these notions require that these indicators (amplitude or energy) be finite to guarantee some robustness. In particular, while this concept has proved useful in many control application, ISS may yield very conservative estimates when the disturbing signals come with high amplitude even if their *moving average* is reasonable.

These limitations have already been pointed out and partially addressed in the literature. In (Angeli and Nešić, 2001), the notions of "Power ISS" and "Power iISS" are introduced to estimate more tightly the influence of the power or moving average of the exogenous input on the *power* of the state. Under the assumption of local stability for the zero-input system, these properties are shown to be actually equivalent to ISS and iISS respectively. Nonetheless, for a generic class of input signals, no hard bound on the state norm can be derived for this work.

Other works have focused on quantitative aspects of ISS, such as (Praly and Wang, 1996), (Grüne, 2002) and (Grüne, 2004). All these three papers solve the problem by introducing a "memory fading" effect in the input term of the ISS formulation. In (Praly and Wang, 1996) the perturbation is first fed into a linear scalar system whose output then enters the right hand side of the ISS estimate. The resulting property is referred to as exp-ISS and is shown to be equivalent to ISS. In (Grüne, 2002) and (Grüne, 2004) the concept of input-to-state dynamical stability (ISDS) is introduced and exploited. In the ISDS state estimate, the value of the perturbation at each time instant is used as the initial value of a one-dimensional system, thus generalizing the original idea of Praly and Wang. The quantitative knowledge of how past values of the input signal influence the system allows, in particular, to guarantee an explicit decay rate of the state for vanishing perturbations.

In this paper, our objective is to guarantee hard bound on the state norm for ISS systems in presence of signals with possibly unbounded amplitude and/or energy. We enlarge the class of signals to which ISS systems are robust, by simply conducting a tighter analysis on these systems. In the spirit of (Angeli and Nešić, 2001), and in contrast to most previous works on ISS and iISS, the considered class of disturbances is defined based on their moving average. We show that any ISS system is robust to such a class of perturbations. When an explicitly Lyapunov function is known, we explicitly estimate the maximum disturbances' moving average that can be tolerated for a given precision. These results are presented in Section 2. We then apply this new analysis result to the control of spacecraft formations. To this end, we exploit the Lyapunov function available for such systems to identify the class of signals to which the formation is robust. This class includes all kind of perturbing effects described above. This study is detailed, and illustrated by simulations, in Section 3.

Notation and Terminology

A continuous function $\alpha : \mathbb{R}_{\geq} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ ($\alpha \in \mathcal{K}$), if it is strictly increasing and $\alpha(0) = 0$. If, in addition, $\alpha(s) \to \infty$ as $s \to \infty$, then α is of class $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty$). A continuous function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{KL}$ if, $\beta(\cdot, t) \in \mathcal{K}$ for any $t \in \mathbb{R}_{\geq 0}$, and $\beta(s, \cdot)$ is decreasing and tends to zero as s tends to infinity. The solutions of the differential equation $\dot{x} = f(x,u)$ with initial condition $x_0 \in \mathbb{R}^n$ is denoted by $x(\cdot; x_0, u)$. We use $|\cdot|$ for the Euclidean norm of vectors and the induced norm of matrices. The closed ball in $\mathbb{R}^n$ of radius δ centered at the origin is denoted by $\mathcal{B}_\delta$, i.e. $\mathcal{B}_\delta := \{x \in \mathbb{R}^n : |x| \leq \delta\}$. $|\cdot|_\delta$ denotes the distance to the ball $\mathcal{B}_\delta$, that is $|x|_\delta := \inf_{z \in \mathcal{B}_\delta} |x - z|$. $\mathcal{U}$ denotes the set of all measurable locally essentially bounded signals $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^p$. For a signal $u \in \mathcal{U}$, $\|u\|_\infty := \text{ess sup}_{t \geq 0} |u(t)|$. The maximum and minimum eigenvalue of a symmetric matrix A is denoted by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$, respectively. I_n and 0_n denote the identity and null matrices of $\mathbb{R}^{n \times n}$ respectively.

2 ISS SYSTEMS AND SIGNALS WITH LOW MOVING AVERAGE

2.1 Preliminaries

We start by recalling some classical definitions related to the stability and robustness of nonlinear systems of the form

$$\dot{x} = f(x,u), \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathcal{U}$ and $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ is locally Lipschitz and satisfies $f(0,0) = 0$.

Definition 1. *Let δ be a positive constant and u be a given signal in $\mathcal{U}$. The ball $\mathcal{B}_\delta$ is said to be globally asymptotically stable (GAS) for (1) if there exists a class $\mathcal{KL}$ function β such that the solution of (1) from any initial state $x_0 \in \mathbb{R}^n$ satisfies*

$$|x(t;x_0,u)| \leq \delta + \beta(|x_0|,t), \quad \forall t \geq 0. \tag{2}$$

Definition 2. *The ball $\mathcal{B}_\delta$ is said to be globally exponentially stable (GES) for (1) if the conditions of Definition 1 hold with $\beta(r,s) = k_1 r e^{-k_2 s}$ for some positive constants k_1 and k_2.*

We next recall the definition of ISS, originally introduced in (Sontag, 1989).

Definition 3. *The system $\dot{x} = f(x,u)$ is said to be input-to-state stable (ISS) if there exist $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that, for all $x_0 \in \mathbb{R}^n$ and all $u \in \mathcal{U}$, the solution of (1) satisfies*

$$|x(t;x_0,u)| \leq \beta(|x_0|,t) + \gamma(\|u\|_\infty), \quad \forall t \geq 0. \tag{3}$$

ISS thus imposes an asymptotic decay of the norm of the state up to a function of the *amplitude* $\|u\|_\infty$ of the input signal.

We also recall the following well-known Lyapunov characterization of ISS, originally established in (Praly and Wang, 1996) and thus extending the original characterization proposed by Sontag in (Sontag and Wang, 1995).

Proposition 1. *The system (1) is ISS if and only if there exist $\underline{\alpha}, \overline{\alpha}, \gamma \in \mathcal{K}_\infty$ and $\kappa > 0$ such that, for all $x \in \mathbb{R}^n$ and all $u \in \mathbb{R}^p$,*

$$\underline{\alpha}(|x|) \leq V(x) \leq \overline{\alpha}(|x|) \tag{4}$$

$$\frac{\partial V}{\partial x}(x) f(x,u) \leq -\kappa V(x) + \gamma(|u|). \tag{5}$$

γ is then called a supply rate *for (1).*

Remark 1. *Since ISS implies iISS (cf. (Sontag, 1998)), it can be shown that the solutions of any ISS system with supply rate γ satisfies, for all $x_0 \in \mathbb{R}^n$,*

$$|x(t;x_0,u)| \leq \beta(|x_0|,t) + \eta\left(\int_0^t \gamma(|u(\tau)|)d\tau\right), \quad \forall t \geq 0, \tag{6}$$

where $\beta \in \mathcal{KL}$ and $\eta \in \mathcal{K}_\infty$. The above integral can be seen as a measure, through the function γ, of the energy *of the input signal u.*

The above remark establishes a link between a measure of the energy fed into the system and the norm of the state: for ISS (and iISS) systems, if this input energy is small, then the state will eventually be small. However, Inequalities (3) and (6) do not provide any information on the behavior of the system when the amplitude (for (3)) and/or the energy (for (6)) of the input signal is not finite.

From an applicative viewpoint, the precision guaranteed by (3) and (6) involve the *maximum value* and the *total energy* of the input. These estimates may be conservative and thus lead to the design of greedy control laws, with negative consequences on the energy consumption and actuators solicitation. This issue is particularly relevant for spacecraft formations in view of the inherent fuel limitation and limited power of the thrusters.

(Angeli and Nešić, 2001) has started to tackle this problem by introducing ISS and iISS-like properties for input signals with limited *power*, thus not necessarily bounded in amplitude nor in energy. For systems that are stable when no input is applied, the authors show that ISS (resp. iISS) is equivalent to "power ISS" (resp. "power iISS") and "moving average ISS" (resp. "moving average iISS"). In general terms, these properties evaluate the influence of the *amplitude* (resp. the *energy*) of the input signal on the *power* or *moving average* of the state. However, as stressed by the authors themselves, these estimates do not guarantee in general any *hard bound* on the state norm. Here, we consider a slightly more restrictive class of input signals under which such a hard bound can be guaranteed. Namely, we consider input signals with bounded moving average.

Definition 4. *Given some constants $E, T > 0$ and some function $\gamma \in \mathcal{K}$, the set $\mathcal{W}_\gamma(E,T)$ denotes the set of all signals $u \in \mathcal{U}$ satisfying*

$$\int_t^{t+T} \gamma(|u(s)|)ds \leq E, \quad \forall t \in \mathbb{R}_{\geq 0}.$$

The main concern here is the measure E of the maximum energy that can be fed into the system over a moving time window of given length T. These quantities are the only information on the disturbances that will be taken into account in the control design. More parsimonious control laws than those based on their amplitude or energy can therefore be expected. We stress that signals of this class are not necessarily globally essentially bounded, nor are they required to have a finite energy, as illustrated by the following examples. Robustness to this class of sig-

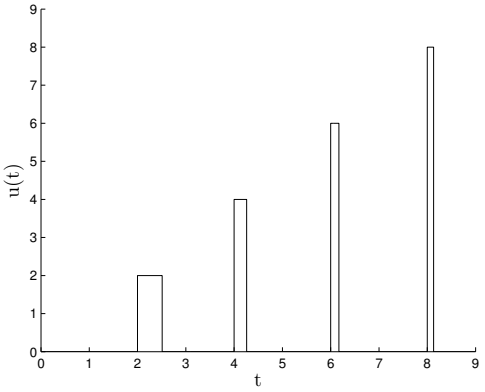


Figure 1: An example of unbounded signal with bounded moving average.

nals thus constitutes an extension of the typical properties of ISS systems.

Example 1.

1. *Unbounded signals: given any $T > 0$ and any $\gamma \in \mathcal{K}$, the following signal belongs to $\mathcal{W}_\gamma(1,T)$ and satisfies $\limsup_{t\to\infty} |u(t)| = +\infty$:*

$$u(t) := \begin{cases} 2k & \text{if} \quad t \in [2kT; 2kT + \frac{1}{2k}], k \in \mathbb{N} \\ 0 & \text{otherwise.} \end{cases}$$

   The signal for $T = 1$ is illustrated in Figure 1.

2. *Essentially bounded signals: given any $T > 0$ and any $\gamma \in \mathcal{K}$, if $\|u\|_\infty$ is finite then it holds that $u \in \mathcal{W}_\gamma(T\gamma(\|u\|_\infty), T)$. We stress that this includes signals with infinite energy (think for instance of constant non-zero signals).*

2.2 Robustness of ISS Systems to Signals in the Class $\mathcal{W}$

The following result establishes that the impact of an exogenous signal on the qualitative behavior of an ISS systems is negligible if the moving average of this signal is sufficiently low.

Theorem 1. *Assume that the system $\dot{x} = f(x,u)$ is ISS. Then there exists a function $\gamma \in \mathcal{K}_\infty$ and, given any precision $\delta > 0$ and any time window $T > 0$, there exists a positive average energy $E(T,\delta)$ such that the ball $\mathcal{B}_\delta$ is GAS for any $u \in \mathcal{W}_\gamma(E,T)$.*

The above result, proved in Section 4, adds another brick in the wall of nice properties induced by ISS, cf. (Sontag, 2008) and references therein. It ensures that, provided that a steady-state error δ can be tolerated, every ISS system is robust to a class of disturbances with sufficiently small moving average.

If an ISS Lyapunov function is known for the system, then an explicit bound on the tolerable average excitation can be provided based on the prooflines of Theorem 1. More precisely, we state the following result.

Corollary 1. *Assume there exists a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, class $\mathcal{K}_\infty$ functions γ, $\underline{\alpha}$ and $\overline{\alpha}$ and a positive constant κ such that (4) and (5) hold for all $x \in \mathbb{R}^n$ and all $u \in \mathbb{R}^p$. Given any precision $\delta > 0$ and any time window $T > 0$, let E denote any average energy satisfying*

$$E(T,\delta) \leq \frac{\underline{\alpha}(\delta)}{2} \frac{e^{\kappa T} - 1}{2e^{\kappa T} - 1}. \tag{7}$$

Then the ball $\mathcal{B}_\delta$ is GAS for $\dot{x} = f(x,u)$ for any $u \in \mathcal{W}_\gamma(E,T)$.

The above statement shows that, by knowing a Lyapunov function associated to the ISS of a system, and in particular its dissipation rate γ, one is able to explicitly identify the class $\mathcal{W}_\gamma(E,T)$ to which it is robust up to the prescribed precision δ.

In a similar way, we can state sufficient condition for global exponential stability of some neighborhood of the origin. This result follows also trivially from the proof of Theorem 1.

Corollary 2. *If the conditions of Corollary 1 are satisfied with $\underline{\alpha}(s) = \underline{c}s^p$ and $\overline{\alpha}(s) = \overline{c}s^p$, with $\underline{c}, \overline{c}, p$ positive constants, then, given any $T, \delta > 0$, the ball $\mathcal{B}_\delta$ is GES for (1) with any signal $u \in \mathcal{W}_\gamma(E,T)$ provided that*

$$E(T,\delta) \leq \frac{\underline{c}\delta^p}{2} \frac{e^{\kappa T} - 1}{2e^{\kappa T} - 1}.$$

3 ILLUSTRATION: SPACECRAFT FORMATION CONTROL

We now exploit the results developed in Section 2 to demonstrate the robustness of a spacecraft formation control in a leader-follower configuration, when only position is measured. The focus on output feedback in this illustration is motivated by the fact that velocity measurements in space may not be easily achieved, *e.g.* because the spacecraft cannot be equipped with the necessary sensors for such measurements due to space constraints or budget limits. The models described in this section have strong resemblance with the model of a robot manipulator. Our control design is therefore be based on control algorithms already validated for robot manipulators, in particular (Berghuis and Nijmeijer, 1993) and (Paden and Panja, 1988). We stress that he proposed study is made for two spacecraft only, but can easily be extended to formations involving more spacecrafts.

3.1 Spacecraft Models

The spacecraft models presented in this section are similar to the ones derived in (Ploen et al., 2004). All coordinates, both for the leader and the follower spacecraft, are expressed in an orbital frame, which origin relative to the center of Earth is given by $\vec{r}_o$, and satisfies Newton's gravitational law

$$\ddot{\vec{r}}_o = -\frac{\mu}{|\vec{r}_o|^3}\vec{r}_o\,,$$

μ being the gravitational constant of Earth. The unit vectors are such that $\vec{o}_1 := \vec{r}_o/|\vec{r}_o|$ points in the anti-nadir direction, $\vec{o}_3 := (\vec{r}_o \times \dot{\vec{r}}_o)/|\vec{r}_o \times \dot{\vec{r}}_o|$ points in the direction of the orbit normal, and finally $\vec{o}_2 := \vec{o}_3 \times \vec{o}_1$ completes the right-handed orthogonal frame. We let ν_o denote the true-anomaly of this reference frame and assume the following:

Assumption 1. *The true anomaly rate $\dot{\nu}_o$ and true anomaly rate-of-change $\ddot{\nu}_o$ of the reference frame satisfy $\|\dot{\nu}_o\|_\infty \leq \beta_{\dot{\nu}_o}$ and $\|\ddot{\nu}_o\|_\infty \leq \beta_{\ddot{\nu}_o}$, for some positive constants $\beta_{\dot{\nu}_o}$ and $\beta_{\ddot{\nu}_o}$.*

Note that this assumption is naturally satisfied when the reference frame is following a Keplerian orbit, but it also holds for any sufficiently smooth reference trajectory. We define the following quantities:

$$C(\dot{\nu}_o) := 2\dot{\nu}_o\bar{C}, \quad \bar{C} := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$D(\dot{\nu}_o, \ddot{\nu}_o) := \dot{\nu}_o^2\bar{D} + \ddot{\nu}_o\bar{C}, \quad \bar{D} := \mathrm{diag}(-1,-1,0),$$

and

$$n(r_o, p) := \mu\left(\frac{r_o + p}{|r_o + p|^3} - \frac{r_o}{|r_o|^3}\right).$$

In the above reference frame, the dynamics ruling the evolution of the coordinate $p \in \mathbb{R}^3$ of the leader spacecraft is then given by

$$\ddot{p} + C(\dot{\nu}_o)\,\dot{p} + D(\dot{\nu}_o, \ddot{\nu}_o)\,p + n(r_o, p) = F_l \quad (8)$$

with $F_l := (u_l + d_l)/m_l$, while the evolution of the relative position ρ of the follower spacecraft with respect to the leader is given by

$$\ddot{\rho} + C(\dot{\nu}_o)\,\dot{\rho} + D(\dot{\nu}_o, \ddot{\nu}_o)\,\rho + n(r_o + p, \rho) = F_f - F_l\,, \quad (9)$$

where $F_f := (u_f + d_f)/m_f$, and where subscripts l and f stand for the leader and follower spacecraft respectively, m_l and m_f are the spacecrafts' masses, u_l and u_f are the control inputs, and d_l and d_f denote all exogenous perturbations acting on the spacecrafts (*e.g.*, as detailed in the Introduction; intervehicle interference, small impacts, solar wind, etc.).

3.2 Control of the Leader Spacecraft

We now propose a controller whose goal is to make the leader spacecraft follow a given trajectory $p_d : \mathbb{R}_{\geq 0} \to \mathbb{R}^3$ relative to the reference frame. In other words, its aim is to decrease the tracking error defined as $e_l := p - p_d$. To derive this controller, we rely on the position p of the leader only. No measurement on its velocity is required. The latter will be estimated through the derivative of the some position estimate $\hat{p}$ in order to avoid brute force derivation of the measurement p. We therefore define $\tilde{p} := p - \hat{p}$ as the estimation error. Similarly to (Berghuis, 1993), the controller is given by:

$$\begin{aligned} u_l =& m_l\Big[\ddot{p}_d + C(\dot{\nu}_o)\,\dot{p}_d + D(\dot{\nu}_o, \ddot{\nu}_o)\,p + n(r_o, p) \\ & - k_l(\dot{p}_0 - \dot{p}_r)\Big] \end{aligned} \quad (10)$$

$$\dot{p}_r = \dot{p}_d - \ell_l e_l \quad (11)$$

$$\dot{p}_0 = \dot{\hat{p}} - \ell_l\tilde{p}, \quad (12)$$

where k_l and ℓ_l denote positive gains. The velocity estimator is given by

$$\dot{\hat{p}} = a_l + (l_l + \ell_l)\,\tilde{p} \quad (13)$$

$$\dot{a}_l = \ddot{p}_d + l_l\ell_l\tilde{p}\,, \quad (14)$$

where l_l denotes another positive gain. Define $X_l := \left(e_l^\top, \dot{e}_l^\top, \tilde{p}^\top, \dot{\tilde{p}}^\top\right)^\top \in \mathbb{R}^{12}$ and $d := (d_l^\top, d_f^\top)^\top \in \mathbb{R}^6$. Then the leader dynamics takes the form of a perturbed linear time-varying system:

$$\dot{X}_l = A_l(\dot{\nu}_o(t))X_l + B_l d\,, \quad (15)$$

where $A_l \in \mathbb{R}^{12\times 12}$ and $B_l \in \mathbb{R}^{12\times 6}$ refer to the following matrices

$$A_l(\dot{\nu}_o) := \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 \\ a_{21} & a_{22}(\dot{\nu}_o) & a_{23} & a_{24} \\ 0_3 & 0_3 & 0_3 & I_3 \\ a_{41} & a_{42}(\dot{\nu}_o) & a_{43} & a_{44} \end{bmatrix}, \quad (16)$$

$$B_l := \frac{1}{m_l}\begin{bmatrix} 0_3 & 0_3 \\ I_3 & 0_3 \\ 0_3 & 0_3 \\ I_3 & 0_3 \end{bmatrix},$$

where out of notational compactness, the following matrices are defined: $a_{21} := a_{41} := -k_l\ell_l I_3$, $a_{22} := a_{42} := -C(\dot{\nu}_o) - k_l I_3$, $a_{23} := k_l\ell_l I_3$, $a_{24} := k_l I_3$, $a_{43} := (k_l - l_l)\ell_l I_3$ and $a_{44} := (k_l - l_l - \ell_l)I_3$.

3.3 Control of the Follower Spacecraft

We next propose a controller to make the follower spacecraft track a desired trajectory $\rho_d : \mathbb{R}_{\geq 0} \to \mathbb{R}^3$ relative to the leader. In the same way as for the leader

spacecraft, let $\dot{\hat{\rho}} \in \mathbb{R}^3$ denote the estimated velocity of the follower with respect to the leader, let $e_f := \rho - \rho_d$ denote the tracking error and let $\tilde{\rho} := \rho - \hat{\rho}$ be the estimation error. We use the following control law:

$$\begin{aligned} u_f =& m_f \Big[\ddot{p}_d + \ddot{\rho}_d + C(\dot{v}_o)(\dot{p}_d + \dot{\rho}_d) \\ &+ D(\dot{v}_o, \ddot{v}_o)(p+\rho) + n(r_o + p, \rho) + n(r_o, p) \\ &- k_l(\dot{p}_0 - \dot{p}_r) - k_f(\dot{\rho}_0 - \dot{\rho}_r) \Big] \end{aligned} \tag{17}$$

$$\dot{\rho}_r = \dot{\rho}_d - \ell_f e_f \tag{18}$$

$$\dot{\rho}_0 = \dot{\hat{\rho}} - \ell_f \tilde{\rho}, \tag{19}$$

with the observer being given by

$$\dot{\hat{\rho}} = a_f + (l_f + \ell_f)\tilde{\rho} \tag{20}$$

$$\dot{a}_f = \ddot{\rho}_d + l_f \ell_f \tilde{\rho} \tag{21}$$

where k_f, l_f and ℓ_f denote positive tuning gains. We stress that, in order to implement (17), (11)-(14) must also be implemented in follower spacecraft control algorithm. Define $X_f := (e_f^\top, \dot{e}_f^\top, \tilde{\rho}^\top, \dot{\tilde{\rho}}^\top)^\top \in \mathbb{R}^{12}$. Combining (9) and (17)-(21) and inserting the leader spacecraft controller u_l (10), we can summarize the follower spacecraft's dynamics by

$$\dot{X}_f = A_f(\dot{v}_o(t))X_f + B_f d, \tag{22}$$

where $A_f(\dot{v}_o)$ can be obtained from $A_l(\dot{v}_o)$ (cf. (16)) by simply substituting the subscripts l by f in the expression of the submatrices a_{ij}, and

$$B_f := \frac{1}{m_l m_f} \begin{bmatrix} 0_3 & 0_3 \\ -m_f I_3 & m_l I_3 \\ 0_3 & 0_3 \\ -m_f I_3 & m_l I_3 \end{bmatrix}.$$

3.4 Robustness Analysis of the Overall Formation

We are now ready to state the following result, which establishes the robustness of the controlled formation to a wide class of disturbances.

Proposition 2. *Let Assumption 1 hold. Let the controller of the leader spacecraft be given by (10)-(14) and the controller of the follower spacecraft be given by (17)-(21) with, for each $i \in \{l, f\}$, $l_i \geq 2k_i$, $k_i > 2k_i^\star$ and (for simplicity) $\ell_i \geq 1$, where*

$$k_i^\star := \ell_i + \beta_{\dot{v}_o}\sqrt{2\ell_i^2 + 1} + \left(1 + \frac{m_f^2}{m_l^2}\right)\frac{2\left(l_i^2+1\right)}{m_i^2}. \tag{23}$$

Given any precision $\delta > 0$ and any time window $T > 0$, consider any average energy satisfying

$$E \leq \frac{1}{4}\min_{i\in\{l,f\}}\left\{\ell_i^2 - \frac{1}{2}\sqrt{4\ell_i^4+1} + \frac{1}{2}\right\}\delta^2 \frac{e^{\kappa T}-1}{2e^{\kappa T}-1}, \tag{24}$$

where

$$\kappa := \frac{\min_{i\in\{l,f\}} k_i^\star / \max_{i\in\{l,f\}}\left\{\frac{k_i}{\ell_i}\right\}}{\max_{i\in\{l,f\}}\left\{\ell_i^2 + \frac{1}{2}\sqrt{4\ell_i^4+1} + \frac{1}{2}\right\}}. \tag{25}$$

Then, for any $d \in \mathcal{W}_\gamma(E,T)$ where $\gamma(s) := s^2$, the ball $\mathcal{B}_\delta$ is GES for the overall formation summarized by (15) and (22).

Proof. Let the overall dynamics be condensed into $\dot{X} = AX + Bd$ with $X := (X_1^\top, X_2^\top)^\top$, $A := \mathrm{diag}(A_l, A_f)$ and $B := (B_l^\top, B_f^\top)^\top$. The proof is done by applying Corollary 2. Consider the Lyapunov function candidate

$$V(X) := \frac{1}{2}\sum_{i\in\{l,f\}} V_i(X_i)$$

where $V_i(X_i) := X_i^\top W_i^\top R_i W_i X_i$, $R_i := \mathrm{diag}((2k_i/\ell_i - 1)I_3, I_3, 2k_i/\ell_i I_3, I_3)$ and

$$W_i := \begin{bmatrix} \ell_i I_3 & 0_3 & 0_3 & 0_3 \\ \ell_i I_3 & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \ell_i I_3 & 0_3 \\ 0_3 & 0_3 & \ell_i I_3 & I_3 \end{bmatrix}.$$

It can be shown that the time derivative of the Lyapunov function candidate can be written as

$$\begin{aligned} \dot{V} &= \sum_{i\in\{l,f\}} X_i^\top W_i^\top R_i W_i A_i X_i + X_i^\top W_i^\top R_i W B_i d \\ &= -\left(\sum_{i\in\{l,f\}} X_i^\top (Q_i + S_i) X_i - X_i^\top W_i^\top R_i W_i B_i d\right) \end{aligned}$$

where $Q_i := \mathrm{diag}(k_i\ell_i^2 I_3, (k_i - \ell_i)I_3, k_i\ell_i^2 I_3, k_i I_3)$,

$$S_i := \frac{1}{2}\begin{bmatrix} 0_3 & C(\dot{v}_o)\ell_i & 0_3 & 0_3 \\ C^\top(\dot{v}_o)\ell_i & 0_3 & C^\top(\dot{v}_o)\ell_i & C^\top(\dot{v}_o) \\ 0_3 & C(\dot{v}_o)\ell_i & \ell^2 s_i & \ell s_i \\ 0_3 & C(\dot{v}_o) & \ell s_i & s_i \end{bmatrix},$$

where $s_i := 2(l_i - 2k_i)I_3$. Since $l_i \geq 2k_i$, $-X_i^\top S_i X_i \leq \|\dot{v}_o\|_\infty (2\ell_i^2 + 1)^{1/2}|X_i|^2$. Furthermore, $\lambda_{\min}(Q_i) = \min\{k_i - \ell_i, k_i\ell_i^2\} = k_i - \ell_i$ for $\ell_i \geq 1$, $|W_l^\top R_l W_l B_l| = (2(l_l^2+1))^{1/2}/m_l$, $|W_f^\top R_f W_f B_f| = (2(m_f^2 + m_l^2)(l_f^2 + 1))^{1/2}/(m_l m_f)$, and invoking Assumption 1, we get that the derivative of the Lyapunov function can be upper bounded as:

$$\begin{aligned} \dot{V} \leq & -\sum_{i\in\{l,f\}}\left(k_i - \ell_i - \beta_{\dot{v}_o}\sqrt{2\ell_i^2+1}|X_i|^2\right) \\ &+ \frac{\sqrt{2\left(l_l^2+1\right)}}{m_l}|X_l|\,|d| \\ &+ \frac{\sqrt{2(m_f^2+m_l^2)(l_f^2+1)}}{m_l m_f}|X_f|\,|d|. \end{aligned}$$

By Young's inequality it follows that

$$\dot{V} \leq -\sum_{i\in\{l,f\}} \Big[k_i - \ell_i - \beta_{\dot{\nu}_o}\sqrt{2\ell_i^2+1} - \Big(1+\frac{m_f^2}{m_l^2}\Big)\frac{2\left(l_i^2+1\right)}{m_i^2}\Big]|X_i|^2 + |d|^2 .$$

If we chose $k_l > 2k_l^\star$ and $k_f > 2k_f^\star$ as given in the statement of Proposition 2, R_l, R_f, Q_l, Q_f are all positive definite matrices. Furthermore, it can be shown that $\underline{c}|X|^2 \leq V(X) \leq \overline{c}|X|^2$, where

$$\underline{c} := \frac{1}{2}\min_{i\in\{l,f\}}\left\{\ell_i^2 - \frac{1}{2}\sqrt{4\ell_i^4+1} + \frac{1}{2}\right\} \tag{26}$$

$$\overline{c} := \max_{i\in\{l,f\}}\left\{\frac{k_i}{\ell_i}\right\}\max_{i\in\{l,f\}}\left\{\ell_i^2 + \frac{1}{2}\sqrt{4\ell_i^4+1} + \frac{1}{2}\right\}. \tag{27}$$

Using these inequalities, we get that

$$\begin{aligned}\dot{V} &\leq -\min_{i\in\{l,f\}}\{k_i^\star\}\left(|X_l|^2 + |X_f|^2\right) + |d|^2 \\ &\leq -\kappa V(x) + |d|^2\end{aligned}$$

with the constant κ defined in (25). Hence, the conditions of Corollary 2 are satisfied, with $\underline{c}$ and $\overline{c}$ defined in (26)-(27) and $\gamma(s) = s^2$, and the conclusion follows.

3.5 Simulations

Let the reference orbit be an eccentric orbit with radius of perigee $r_p = 10^7 m$ and radius of apogee $r_a = 3\times 10^7 m$, which can be generated by numerical integration of

$$\ddot{r}_o = -\frac{\mu}{|r_o|^3} r_o, \tag{28}$$

with $r_o(0) = (r_p, 0, 0)$ and $\dot{r}_o(0) = (0, v_p, 0)$, and where

$$v_p = \sqrt{2\mu\left(\frac{1}{r_p} - \frac{1}{(r_p+r_a)}\right)}.$$

The true anomaly ν_o of the reference frame can be obtained by numerical integration of the equation

$$\ddot{\nu}_o(t) = \frac{-2\mu e_o\left(1+e_o\cos\nu_o(t)\right)^3\sin\nu_o(t)}{\left(\frac{1}{2}(r_p+r_a)(1-e_o^2)\right)^3}.$$

From this expression, and the eccentricity, which can be calculated from r_a and r_p to be $e_o = 0.5$, we see that the constant $\beta_{\dot{\nu}_o}$ in Assumption 1 can be chosen as $\beta_{\dot{\nu}_o} = 4\times 10^{-7}$. From the analytical equivalent for $\dot{\nu}_o$,

$$\dot{\nu}_o(t) = \frac{\sqrt{\mu}\left(1+e_o\cos\nu_o(t)\right)^2}{\left(\frac{1}{2}(r_p+r_a)(1-e_o^2)\right)^{3/2}},$$

we see that the constant $\beta_{\dot{\nu}_o}$ in Assumption 1 can be chosen as $\beta_{\dot{\nu}_o} = 8\times 10^{-4}$. Since the reference frame is initially at perigee, $\nu_o(0) = 0$ and $\dot{\nu}_o(0) = v_p/r_p$. For simplicity, we choose the desired trajectory of the leader spacecraft to coincide with the reference orbit, *i.e.* $p_d(\cdot) \equiv (0,0,0)^\top$. The initial values of the leader spacecraft are $p_l(0) = (2,-2,3)^\top$ and $\dot{p}_l(0) = (0.4,-0.8,-0.2)^\top$. The initial values of the observer are chosen as $\hat{p}(0) = (0,0,0)^\top$ and $a_l(0) = (0,0,0)^\top$.

The reference trajectory of the follower spacecraft are chosen as the solutions of a special case of the Clohessy-Wiltshire equations, cf. (Clohessy and Wiltshire, 1960). We use

$$\rho_d(t) = \begin{bmatrix} 10\cos\nu_o(t) \\ -20\sin\nu_o(t) \\ 0 \end{bmatrix}. \tag{29}$$

This choice imposes that the two spacecrafts evolve in the same orbital plane, and that the follower spacecraft will make a full rotation about the leader spacecraft per orbit around the Earth. The initial values of the follower spacecraft are $\rho(0) = (9,-1,2)^\top$ and $\dot{\rho}(0) = (-0.3, 0.2, 0.6)^\top$. The initial parameters of the observer are chosen to be $\hat{\rho}(0) = \rho_d(0) = (10,0,0)^\top$ and $a_f(0) = (0,0,0)^\top$. We use $m_f = m_l = 25$ kg both in the model and the control structure.

The choice of control gains are based on the analysis in Section 3. First we pick $\ell_i = 1$, $i \in \{l,f\}$. Then, by using that $\beta_{\dot{\nu}_o} = 8\times 10^{-4}$, we find that $k_i^\star = 1.0014 + 0.0064(l_i^2+1)$ from (23). Since k_i should satisfy $k_i > 2k_i^\star$ and $l_i \geq 2k_i$, we chose $k_i = 2.3$ and $l_i = 4.6$, $i \in \{l,f\}$. With these choices, we find from (25) that $\kappa \approx 0.1899$. Over a 10 second interval (*i.e.* T=10), the average excitation must satisfy $E(T,\delta) \leq 0.0439\,\delta^2$, according to (24). We consider two types of disturbances acting on the spacecraft: "impacts" and continuous disturbances. The "impacts" have random amplitude, but with maximum of 1.5 N in each direction of the Cartesian frame. For simplicity, we assume that at most one impact can occur over each 10 second interval, and we assume that the duration of each impact is 0.1s. The continuous part is taken as sinusoids, also acting in each direction of the Cartesian frame, and are chosen to be $(0.1\sin 0.01t, 0.25\sin 0.03t, 0.3\sin 0.04t)^\top$ for both spacecraft. The motivation for choosing the same kind of continuous disturbance for both spacecraft, is that this disturbance is typically due to gravitational perturbation, which at least for close formations, have the same effect on both spacecraft. Notice from (9) that the relative dynamics are influenced by disturbances acting on the leader and follower spacecraft, so the effect of the continuous part of the disturbance on the relative dynamics is zero. It can easily

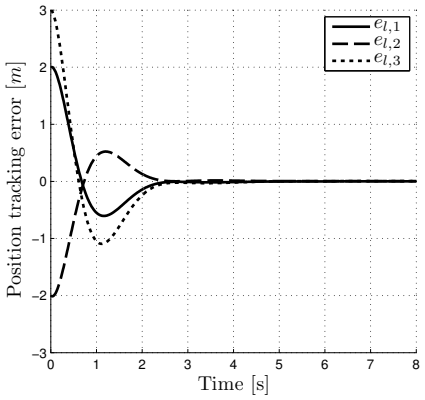


Figure 2: Position tracking error of the leader spacecraft.

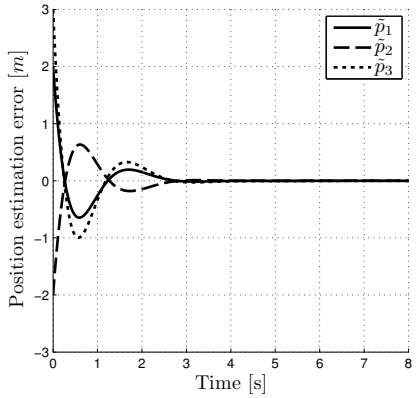


Figure 3: Position estimation error of the leader spacecraft.

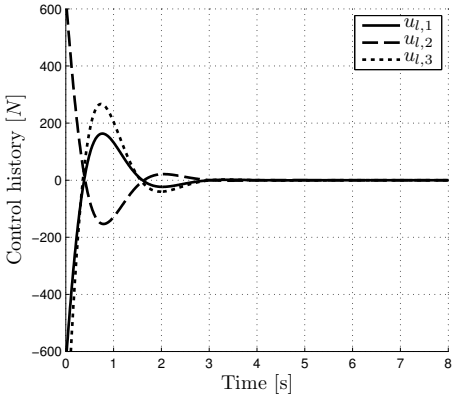


Figure 4: Control actuation of the leader spacecraft.

be shown that the disturbances satisfy the following:

$$\int_t^{t+10} |d(\tau)|^2 d\tau \leq 1.42\,, \quad \forall t \geq 0\,.$$

Figure 2, 3 and 4 show the position tracking error, position estimation error and control history of the leader spacecraft, whereas Figure 6, 7 and 8 are the equivalent figures for the follower spacecraft. Figure 5 and 9 show the effect of d_l and $d_l - d_f$ acting on the formation. Notice in Figure 9 that the effect of the continuous part of the disturbance is canceled out (since we consider relative dynamics and both spacecraft are influenced by the same continuous disturbance), whereas the effect of the impacts has increased compared to the effect of the impacts

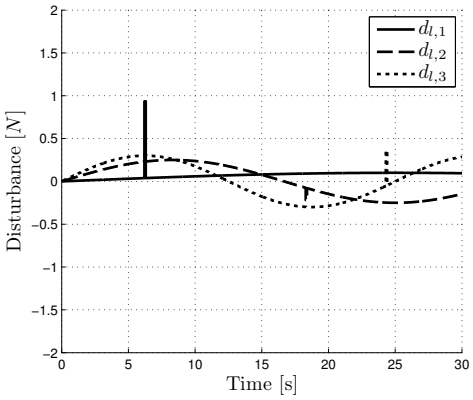


Figure 5: Disturbances acting on the leader spacecraft.

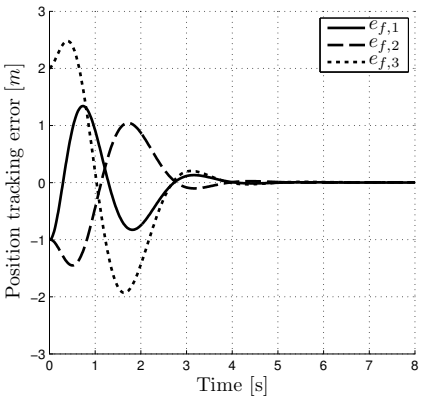


Figure 6: Position tracking error of the follower spacecraft.

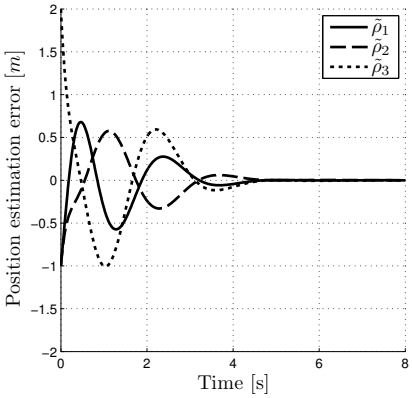


Figure 7: Position estimation error of the follower spacecraft.

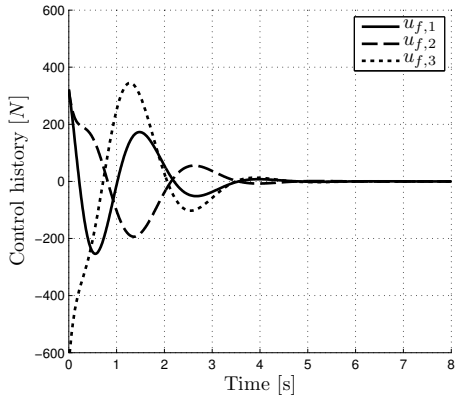


Figure 8: Control actuation of the leader spacecraft.

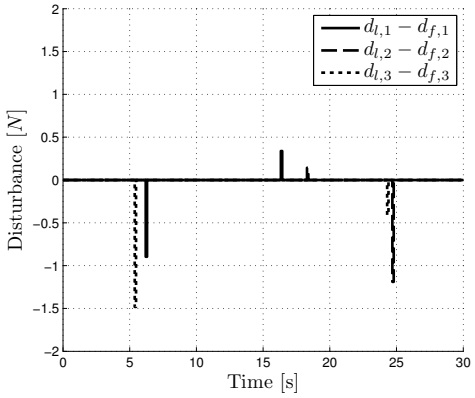


Figure 9: Disturbances acting on the follower spacecraft.

on the leader spacecraft. The control gains have been chosen based on the Lyapunov analysis. This yields in general very conservative constraints on the choice of control gains, and also conservative estimates of the disturbances the control system is able to handle. As shown in Figure 4, and in particular Figure 8, this leads to large transients in the actuation. We stress that the control gains proposed by this approach is still much smaller that those obtained through a classical ISS approach (*i.e.* relying on the disturbance magnitude).

4 PROOF OF THEOREM 1

In view of (Praly and Wang, 1996, Lemma 11) and (Angeli et al., 2000, Remark 2.4), there exists a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, class $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}$ and γ, and a positive constant κ such that, for all $x \in \mathbb{R}^n$ and all $u \in \mathbb{R}^m$,

$$\underline{\alpha}(|x|) \leq V(x) \leq \overline{\alpha}(|x|) \tag{30}$$

$$\frac{\partial V}{\partial x}(x) f(x,u) \leq -\kappa V(x) + \gamma(|u|)\,. \tag{31}$$

Let $w(t) := V(x(t;x_0,u))$. Then it holds in view of (31) that

$$\begin{aligned} \dot{w}(t) &= \dot{V}(x(t;x_0,u)) \\ &\leq -\kappa V(x(t;x_0,u)) + \gamma(|u(t)|) \\ &\leq -\kappa w(t) + \gamma(|u(t)|)\,. \end{aligned}$$

In particular, it holds that, for all $t \geq 0$,

$$w(t) \leq w(0)e^{-\kappa t} + \int_0^t \gamma(|u(s)|)ds\,. \tag{32}$$

Assuming that u belongs to the class $\mathcal{W}_\gamma(E,T)$, for some arbitrary constants $E, T > 0$, it follows that

$$w(T) \leq w(0)e^{-\kappa T} + \int_0^T \gamma(|u(s)|)ds \leq w(0)e^{-\kappa T} + E\,.$$

Considering this inequality recursively, it follows that, for each $\ell \in \mathbb{N}_{\geq 1}$,

$$\begin{aligned} w(\ell T) &\leq w(0)e^{-\ell\kappa T} + E\sum_{j=0}^{k-1} e^{-j\kappa T} \\ &\leq w(0)e^{-\ell\kappa T} + E\sum_{j\geq 0} e^{-j\kappa T} \\ &\leq w(0)e^{-\ell\kappa T} + E\frac{e^{\kappa T}}{e^{\kappa T}-1}\,. \end{aligned} \tag{33}$$

Given any $t \geq 0$, pick ℓ as $\lfloor t/T \rfloor$ and define $t' := t - \ell T$. Note that $t' \in [0,T]$. It follows from (32) that

$$w(t) \leq w(\ell T)e^{-\kappa t'} + \int_{\ell T}^t \gamma(|u(s)|)ds \leq w(\ell T)e^{-\kappa t'} + E\,,$$

which, in view of (33), implies that

$$\begin{aligned} w(t) &\leq \left(w(0)e^{-\ell\kappa T} + E\frac{e^{\kappa T}}{e^{\kappa T}-1}\right)e^{-t'} + E \\ &\leq w(0)e^{-k(\ell T + t')} + E\left(1 + \frac{e^{\kappa T}}{e^{\kappa T}-1}\right) \\ &\leq w(0)e^{-\kappa t} + \frac{2e^{\kappa T}-1}{e^{\kappa T}-1}E\,. \end{aligned}$$

Recalling that $w(t) = V(x(t;x_0,u))$, it follows that

$$V(x(t;x_0,u)) \leq V(x_0)e^{-\kappa t} + \frac{2e^{\kappa T}-1}{e^{\kappa T}-1}E\,,$$

which implies, in view of (30), that

$$\underline{\alpha}(|x(t;x_0,u)|) \leq \overline{\alpha}(|x_0|)e^{-\kappa t} + \frac{2e^{\kappa T}-1}{e^{\kappa T}-1}E\,,$$

Recalling that $\underline{\alpha}^{-1}(a+b) \leq \underline{\alpha}^{-1}(2a) + \underline{\alpha}^{-1}(2b)$ as $\underline{\alpha} \in \mathcal{K}_\infty$, we finally obtain that, given any $x_0 \in \mathbb{R}^n$, any $u \in \mathcal{W}_\gamma(E,T)$ and any $t \geq 0$,

$$|x(t;x_0,u)| \leq \underline{\alpha}^{-1}\left(2\overline{\alpha}(|x_0|)e^{-\kappa t}\right) + \underline{\alpha}^{-1}\left(2E\frac{2e^{\kappa T}-1}{e^{\kappa T}-1}\right)\,. \tag{34}$$

Given any $T, \delta \geq 0$, the following choice of E:

$$E(T,\delta) \leq \frac{\underline{\alpha}(\delta)}{2} \frac{e^{\kappa T}-1}{2e^{\kappa T}-1}. \tag{35}$$

ensures that

$$\underline{\alpha}^{-1}\left(2E\frac{2e^{\kappa T}-1}{e^{\kappa T}-1}\right) \leq \delta$$

and the conclusion follows in view of (34) with the $\mathcal{KL}$ function

$$\beta(s,t) := \underline{\alpha}^{-1}\left(2\overline{\alpha}(s)e^{-\kappa t}\right), \quad \forall s,t \geq 0.$$

REFERENCES

Angeli, D. and Nešić, D. (2001). Power characterizations of input-to-state stability and integral input-to-state stability. *IEEE Transactions on Automatic Control*, 48:1298–1303.

Angeli, D., Sontag, E. D., and Wang, Y. (2000). A characterization of integral input-to-state stability. *IEEE Transactions on Automatic Control*, 45:1082–1097.

Berghuis, H. (1993). *Model-based Robot Control: from Theory to Practice*. PhD thesis, Universiteit Twente.

Berghuis, H. and Nijmeijer, H. (1993). A passivity approach to controller-observer design for robots. *IEEE Transactions on Robotics and Automation*, 9(6):740–754.

Clohessy, W. H. and Wiltshire, R. S. (1960). Terminal guidance system for satellite rendezvous. *Journal of Aerospace Sciences*, 27:9.

Grüne, L. (2002). Input-to-state dynamical stability and its Lyapunov function characterization. *IEEE Transactions on Automatic Control*, 47:1499–1504.

Grüne, L. (2004). Quantitative aspects of the input-to-state-stability property. In de Queiroz, M., Malisoff, M., and Wolenski, P., editors, *Optimal Control, Stabilization and Nonsmooth Analysis*, pages 215–230. Springer-Verlag.

Hanslmeier, A., Denkmayr, K., and Weiss, P. (1999). Longterm prediction of solar activity using the combined method. *Solar Physics*, 184:213–218.

NASA (1999). On space debris. Technical report, NASA. ISBN: 92-1-100813-1.

Overhage, C. F. J. and Radford, W. H. (1964). The Lincoln Laboratory West Ford Program - An historical perspective. *Proceeding of the IEEE*, 52:452–454.

Paden, B. and Panja, R. (1988). Globally asymptotically stable 'PD+' controller for robot manipulators. *International Journal of Control*, 47(6):1697–1712.

Ploen, S. R., Scharf, D. P., Hadaegh, F. Y., and Acikmese, A. B. (2004). Dynamics of earth orbiting formations. In *Proc. of AIAA Guidance, Navigation and Control Conference*.

Praly, L. and Wang, Y. (1996). Stabilization in spite of matched unmodeled dynamics and an equivalent definition of input-to-state stability. *Mathematics of Control, Signals, and Systems*, 9:1–33.

Schäfer, F. (2006). The threat of space debris and micrometeoroids to spacecraft operations. *ERCIM NEWS*, (65):27–29.

Sontag, E. D. (1989). Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34:435–443.

Sontag, E. D. (1998). Comments on integral variants of ISS. *Systems & Control Letters*, 34:93–100.

Sontag, E. D. (2008). Input to state stability: Basic concepts and results. In Nistri, P. and Stefani, G., editors, *Nonlinear and Optimal Control Theory*, pages 163–220. Springer-Verlag, Berlin.

Sontag, E. D. and Wang, Y. (1995). On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24:351–359.

Wertz, J. R., editor (1978). *Spacecraft attitude determination and control*. D. Reidel Publishing company. ISBN: 9027709599.

A ROBUST LIMITED-INFORMATION FEEDBACK FOR A CLASS OF UNCERTAIN NONLINEAR SYSTEMS

Alessio Franci
LSS - Université Paris Sud - Supélec, 3 rue Joliot-Curie, 91192 Gif sur Yvette, France
alessio.franci@lss.supelec.fr

Antoine Chaillet
EECI - LSS - Université Paris Sud - Supélec, 3 rue Joliot-Curie, 91192 Gif sur Yvette, France
antoine.chaillet@supelec.fr

Keywords: Limited-information feedback, Robustness, Nonlinear systems.

Abstract: We propose a variant of the recently introduced strategy for stabilization with limited information recently introduced in (Liberzon and Hespanha, 2005) and analyze its robustness properties. We show that, if the nominal plant can be made Input-to-State Stable (ISS) with respect to measurement errors, parameter uncertainty and exogenous disturbances, then this robustness is preserved with this quantized feedback. More precisely, if a sufficient bandwidth is available on the communication network, then the resulting closed-loop is shown to be semiglobally Input-to-State practically Stable (ISpS).

1 INTRODUCTION

The always greater use of digital communication devices for control applications makes quantization a crucial issue. The limitations on the communication rate between the plant sensors and the controller imposes to develop new approaches that are able to guarantee good performance even when only limited information on the plant's state is available. Despite strong technological improvements, the bit rate available for a given control application may indeed be strongly limited due to scalability or energy-saving concerns, or due to harsh environment constraints. Stabilization in this context becomes particularly challenging in presence of model uncertainties, measurement errors or exogenous disturbances.

These observations explain why limited-information control feedback has been widely studied recently: (Nair et al., 2007; Hespanha et al., 2007; Liberzon, 2009) and references therein for representative examples. An important literature already exists for linear systems, (Montestruque and Antsaklis, 2004; Liberzon, 2003; Petersen and Savkin, 2001; Nair and Evans, 2004; Jaglin et al., 2008; Jaglin et al., 2009). In particular, the results of (Liberzon and Nešić , 2007) provide a coding/decoding strategy that achieves Input-to-State Stabilization of quantized linear control systems. The proposed control strategy relies on a discrete time zoom-in/zoom-out procedure. This construction is based on the *exact* sampled dynamics of the system, or at most on its discrete time approximation. This is why the closed-loop system may lack robustness with respect to parameter uncertainties. These results were subsequently generalized to nonlinear systems in (Kameneva and Nešić , 2008).

In (Sharon and Liberzon, 2007), Input-to-State Stabilization of quantized linear and nonlinear systems is achieved in the framework of continuous time quantized control systems, that is exploiting hybrid dynamics. It is based on a generalization of the dynamic quantization approach developed in (Liberzon and Hespanha, 2005) and (Persis and Isidori, 2004) for ISS and global asymptotically stable systems respectively. For nonlinear systems, this control strategy leads to local ISS. However, model uncertainties can seriously compromise the efficiency of the proposed algorithm and no estimates of the domain of attraction can be obtained in general. We detail these limitation in Section 5.

The purpose of this paper is to propose an alternative dynamic quantization strategy, able to cope with (time-varying) model uncertainties. It is based on a simple and natural modification of the one proposed in (Liberzon and Hespanha, 2005). We show that the quantized control strategy ensures

semiglobal Input-to-State practical Stability. More precisely, any compact set of initial conditions and for any bounded time-varying measurement error, disturbance and model uncertainty, it is possible to achieve the desired robustness properties by properly tuning the controller parameters. On the other hand, practical stability here does not guarantee convergence to the origin for vanishing perturbations, although the size of the stable subset depends on the tuning parameters and can be somewhat reduced, provided a sufficient knowledge on the intensity of the perturbations. The main contributions of our work are the robustness to model uncertainties and its semiglobal characterization for nonlinear systems.

The rest of the paper is organized as follows. In Section 2 we introduce the needed notation. In Section 3 we formally state the problem. In Section 4 we introduce our dynamic quantization strategy. We then present the main results of the paper and comment them in Section 5. In Section 6 we check their application on the illustrative example of a DC motor with nonlinear load. Proofs are given in Section 8.

2 NOTATION

For a set $A \subset \mathbb{R}$, and $a \in A$, $A_{\geq a}$ denotes the set $\{x \in A : x \geq a\}$. $|x|$ denotes the *infinity norm* of the vector x, that is, if $x \in \mathbb{R}^n$, $|x| := \max_{i=1,\dots,n} |x_i|$. $B(x,R)$ refers to the closed ball of radius R centered at x in this norm, *i.e.* $B(x,R) := \{z \in \mathbb{R}^n : |x-z| \leq R\}$. $\|x\|$ is the *infinity norm* of the signal $x(\cdot)$, that is, if $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, $\|x\| = \text{esssup}_{t \geq 0} |x(t)|$. A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{K}$ if it is increasing and $\alpha(0) = 0$. It is said to be of class $\mathcal{K}_\infty$ if it is of class $\mathcal{K}$ and $\alpha(s) \to \infty$ as $s \to \infty$. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{KL}$ if $\beta(\cdot,t) \in \mathcal{K}$ for any fixed $t \geq 0$ and $\beta(s,\cdot)$ is continuous, decreasing and tends to zero at infinity for any fixed $s \geq 0$.

3 PROBLEM STATEMENT

We are interested in the robustness properties of nonlinear plants of the form

$$\dot{x} = f(x,\mu,u,d), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^h \to \mathbb{R}^n$ is a locally Lipschitz function, $\mu : \mathbb{R}_{\geq 0} \to \mathcal{P} \subset \mathbb{R}^p$ is a vector of (possibly time-varying) parameters, $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is a control input and $d : \mathbb{R}_{\geq 0} \to \mathcal{D} \subset \mathbb{R}^h$ is a vector of measurable and locally essentially bounded exogenous perturbations. We assume that $f(0,\mu,0,0) = 0$ for all $\mu \in \mathcal{P}$.

Limited-information feedback imposes that only an estimate of the state is available to the controller. This estimate is elaborated based on an encoded measurement of the actual state. This encoded symbol is then sent over the communication channel. The communication channel is defined by its constant sampling period τ and by the number of symbols N^n, $N \in \mathbb{N}_{>0}$, that can be transmitted at each sampling time $k\tau$, $k \in \mathbb{N}$. We will assume, in this paper, that the communication channel is noiseless and delay-free. The overall structure of the controlled systems can be summarized by Figure 1.

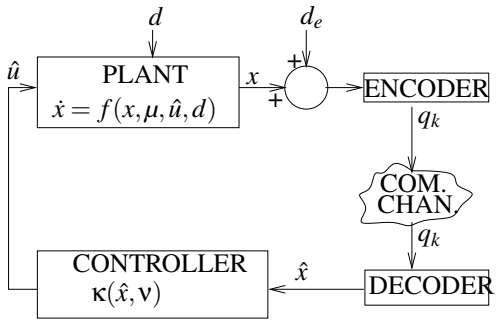


Figure 1: Limited information feedback with exogenous perturbations, measurement errors and uncertainties.

At each reception of a symbol, that is, at each time instant $k\tau$, $k \in \mathbb{N}$, the decoder computes the state estimate that will be used in the applied feedback law. The decoding is necessarily imprecise due to the limited bandwidth of the channel. This imprecision is reinforced by the uncertainty on the plant parameter μ, by the presence of exogenous disturbances d and by the possible measurement errors d_e. We assume that only a *constant*[1] approximation $\nu \in \mathcal{P}$ of the (possibly time-varying) parameter vector μ is available and define $\mu - \nu =: d_p \in \mathbb{R}^p$ as the parameter uncertainty. Our first assumption imposes that, without communication constraints, the plant (1) can be stabilized by a state-feedback law that makes it ISS with respect to exogenous disturbances, parameter uncertainties and measurement errors.

Assumption 1 (ISS of the nominal plant)**.** *There exists a continuous feedback law $\kappa : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$, a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ and class $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}, \alpha, \chi, \Gamma, \gamma$ such that, for all $x \in \mathbb{R}^n$, $d \in \mathcal{D}$, $d_p \in \mathcal{P}$ and $d_e \in \mathbb{R}^n$,*

$$\underline{\alpha}(|x|) \leq V(x) \leq \overline{\alpha}(|x|),$$

$$|x| \geq \chi(|d|) + \Gamma(|d_p|) + \gamma(|d_e|) \quad \Rightarrow \tag{2a}$$

[1] In a second stage one may think of implementing an adaptive control strategy.

$$\frac{\partial V}{\partial x} f(x,\mu,\kappa(x+d_e,\mu+d_p),d) \leq -\alpha(|x|). \quad (2b)$$

Based on the Lyapunov characterization of ISS systems (Sontag and Wang, 1995), condition (2) is equivalent to ISS of (1) with respect to d, d_e, d_p (at least locally as far as d and d_p are concerned). Assumption 1 therefore constitutes a strong requirement, but the following remarks may help establishing it in some particular contexts.

Remark 1 (Systems in strict feedback form)**.** *For all systems in strict feedback form it is possible to achieve conditions of Assumption 1. Indeed, back-stepping allows to iteratively make each subsystem ISS with respect to (d,d_p,d_e), using part of the state as a "virtual" control input. See (Freeman and Kokotovich, 1993) for details.*

Remark 2 (Globally Lipschitz systems)**.** *The conditions of Assumption 1 can be achieved for all systems which can be stabilized by a* globally *Lipschitz state feedback that makes it ISS with respect to actuation errors. Indeed if $\mathcal{L}$ denotes the global Lipschitz constant of the nominal control law κ, then the effects due to parameter uncertainties d_p and measurement errors d_e can be described explicitly as an input disturbance $\tilde{d}$ satisfying $|\tilde{d}| \leq \mathcal{L}|d_p| + \mathcal{L}|d_e|$. Hence, if $\gamma \in \mathcal{K}_\infty$ is the ISS gain, the presence of measurement errors and parameter uncertainties simply adds $\gamma(|\tilde{d}|) \leq \gamma(2\mathcal{L}|d_p|) + \gamma(2\mathcal{L}|d_e|)$ to the solution estimate of the closed-loop solutions, hence proving ISS with respect to (d_p,d_e). Note that all systems which can be made ISS by differentiable bounded control trivially satisfy this global Lipschitz condition. Cf. e.g. (A.Isidori, 1999, Chapters 12,13,14)*$\triangleleft$

4 QUANTIZED CONTROLLER

In this section, we extend the encoding-decoding procedure presented in (Liberzon and Hespanha, 2005) and (Persis and Isidori, 2004) to take into account exogenous disturbances, measurement errors and parameter uncertainties. We assume that measurement errors are bounded by some constant $E > 0$ such that

$$\|d_e\| \leq E. \quad (3)$$

4.1 Quantization Region

Given an estimate $\hat{x}$ of the actual state x, the quantization region Q is defined by its centroid $\hat{x}$ and its radius $L > 0$ as

$$Q := B(\hat{x},L).$$

Due to measurement errors, the information available to the encoder about the system state, *i.e.* $x+d_e$, belongs to the quantization region Q if and only if the *estimation error* $e := x - \hat{x} + d_e$ is small enough, that is $|e| \leq |x-\hat{x}| + |E| \leq L$. Note that the presence of the estimation error e results from the combined effects of quantization $(x-\hat{x})$ and measurement errors (d_e). Given the number N^n of symbols that can be transmitted through the communication channel, we partition the quantization region into N^n identical hypercubes. Q is then updated according to the encoding-decoding procedure described below.

4.2 Dynamics of the Encoder

At each step $k \in \mathbb{N}$, the *centroid update law* is given by the following hybrid dynamics

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x},\nu,\kappa(\hat{x},\nu),0), \quad \forall t \in [k\tau,(k+1)\tau), & (4a)\\ \hat{x}(k\tau) &= \hat{c}(k\tau), \quad k \neq 0, & (4b)\\ \hat{x}(0^-) &= 0, & (4c) \end{aligned}$$

where $\hat{c}(k\tau)$ is the centroid of the sub-region of $Q(k\tau)$ in which $x(k\tau)+d_e$ lies. This sub-region is identified by the variable q_k, which constitutes the output of the encoder. In other words, $q_k \in \mathbb{N}_{\leq N^n}$ denotes the index of the sub-region of $Q(k\tau)$ to which $x(k\tau)+d_e$ belongs. Then, given some $\Lambda > 1$ (we will make it precise in the sequel) and any ball of initial conditions $B(0,\Delta)$ with $\Delta > 0$, the *radius update law* is given, at each step $k \in \mathbb{N}$, by the following dynamics

$$\begin{aligned} L((k+1)\tau) &= \Lambda\left(\frac{L(k\tau)}{N} + E\right) + E, & (5a)\\ L(0) &= \Delta + E. & (5b) \end{aligned}$$

This radius update law is a natural extension of the algorithms proposed in (Liberzon and Hespanha, 2005; Persis and Isidori, 2004). It takes into account possible measurement errors. We will show in the sequel (cf. Claim 2) that such a dynamics leads to a sequence $\{L(k\tau)\}_{k\in\mathbb{N}}$ that decreases up to a constant depending on E, Λ and N. This in turn imposes a decrease of the estimation error, modulo the measurement errors, as long as dynamics (5) applies. The idea behind this dynamics can be roughly summarized as follows. The parameter $\Lambda > 1$ accounts for the expansiveness between sampling times. The constant E appearing inside the brackets of (5a) accounts for the case in which the encoder individuates a wrong sub-region due to measurements errors. In such a situation the error between the real and the measured state is indeed less than the size of the sub-region, $L(k\tau)/N$, plus the measurement error. The second E appearing in (5a) prevents the measured state from falling out of the quantization region while the real one is inside.

Note that, as long as $x(k\tau)+d_e$ lies in $Q(k\tau)$, the estimation error satisfies $|e(k\tau)| \leq \bar{e}(k\tau)$, where

$$\bar{e}(k\tau) := \frac{L(k\tau)}{N} + E, \quad \forall k \in \mathbb{N}, \tag{6}$$

is the *maximum quantization error*. Hence, at each sampling time, the quantization procedure individuates a hypercube $B(\hat{x}(k\tau), \bar{e}(k\tau))$ to which $x(k\tau)$ belongs, provided that $x(k\tau)+d_e \in Q(k\tau)$.

However, due to uncertainties and disturbances, it may happen that $x(k\tau)+d_e$ falls out of the quantization region anyway. Indeed, the expansion factor Λ in (5a) ensures that the updated quantization region is large enough to contain the measured state only if the quantization error is large compared to the disturbances (see the proof of Theorem 1 for details). This situation is defined as an *overflow*. It is represented by the symbol $q_k = 0$. In particular, as detailed in the proof of Theorem 1, an overflow can happen at time $(k+1)\tau$ only if the maximum quantization error (6) at time $k\tau$ is strictly smaller than the size of perturbations and uncertainties. If an overflow occurs at the k_0th sampling time, $k_0 \in \mathbb{N}_{>0}$, the encoder updates the quantization region as follows:

$$\hat{x}(k_0\tau) = \hat{x}(k_0\tau^-), \tag{7a}$$
$$L((k_0+1)\tau) = \Lambda(\overline{E}+E)+E, \tag{7b}$$

where $\overline{E} \in \mathbb{R}_{>0}$ will be defined later on. This means that the hypercube individuated by the quantization procedure (to which $x(k\tau)$) belongs, is no longer $B(\hat{x}(k\tau), \frac{L(k\tau)}{N}+E)$, but rather $B(\hat{x}(k\tau), \overline{E}+E)$, while the rest of the update law remains as in (4),(5).

4.3 Dynamics of the Decoder

By implementing the same evolution laws as (4),(5),(7), the decoder is able to reconstruct the evolution of the state estimate $\hat{x}$ from the knowledge of $\{q_k\}_{k\in\mathbb{N}}$.

4.4 Controller

Inspired by the principle of certainty equivalence, and in view of Assumption 1, the applied control input is given by

$$\hat{u}(t) = \kappa(\hat{x}(t), \nu), \tag{8}$$

where $\hat{x}(\cdot)$ is given by (4),(5),(7).

5 MAIN RESULTS

Our first result establishes robustness properties of the closed-loop system with the proposed limited-information feedback in the case the number of transmittable bits is fixed and the sampling period can be adjusted arbitrarily.

Theorem 1 (Fixed N)**.** *Let Assumption 1 hold for the system (1). Then, there exist class $\mathcal{K}_\infty$ functions $\overline{\chi}, \overline{\Gamma}, \overline{\gamma}$ and, given any compact sets $\mathcal{P} \subset \mathbb{R}^p$ and $\mathcal{D} \subset \mathbb{R}^d$, any constant $\Delta \in \mathbb{R}_{>0}$ and any $N \in \mathbb{N}_{>1}$, there exist positive constants $\tau, \Lambda, \overline{E}, E$ and a class $\mathcal{KL}$ function β such that the trajectories of the closed-loop system*

$$\dot{x} = f(x, \mu, \hat{u}, d), \tag{9}$$

where $\hat{u}(t)$ is the output of the digital controller defined by (4),(5), (7),(8), satisfy, for all $x(0) \in B(0,\Delta)$, all $\nu \in \mathcal{P}$, all $\mu : \mathbb{R}_{\geq 0} \to \mathcal{P}$, all $d : \mathbb{R}_{\geq 0} \to \mathcal{D}$ and all $d_e : \mathbb{R}_{\geq 0} \to B(0,E)$,

$$|x(t)| \leq \overline{\beta}(\Delta, t) + \overline{\chi}(\|d\|) + \overline{\Gamma}(\|\mu - \nu\|) + \delta, \tag{10}$$

where $\delta := \overline{\gamma}\left(\Lambda(\overline{E}+E) + \left(2 + \frac{\Lambda+1}{1-\frac{\Lambda}{N}}\right)E\right)$.

Theorem 1 states that (9) is semiglobally ISpS (Input-to-State practically Stable) in the sense of (Jiang et al., 1994) with the proposed quantized control strategy. Our proof, provided in Section 8.1, is constructive. The utilized bit-rate, given by $\frac{\log_2(N^n)}{\tau}$, is fixed by the condition that quantization resolution, given by N^{-1}, is small enough to compensate for the expansiveness of the system between sampling times Λ. An upper bound on this expansiveness expressed in terms of the (local) Lipschitz constant of the system $\mathcal{L}$, is given by

$$\Lambda := e^{\mathcal{L}\tau}.$$

Based on the size of the initial conditions Δ, our control strategy permits to build a forward invariant region where the constant $\mathcal{L}$ can be computed (cf. Claim 1 below). The explicit condition on the data rate used in the proof is then given by

$$\Lambda N^{-1} < 1.$$

It is interesting to note that the required data-rate is the same as in (Liberzon and Hespanha, 2005), modulo the size of the constructed forward invariant region.

The comparison functions involved in (10) can be explicitly given

$$\begin{aligned}
\overline{\beta}(\cdot,t) &= \underline{\alpha}^{-1}(\overline{\alpha}(2\gamma(\cdot)\gamma(e^{-\lambda t}))),\\
\overline{\chi}(\cdot) &= \underline{\alpha}^{-1}(\overline{\alpha}(4\chi(\cdot))),\\
\overline{\Gamma}(\cdot) &= \underline{\alpha}^{-1}(\overline{\alpha}(8\Gamma(\cdot))),\\
\overline{\gamma}(\cdot) &= \underline{\alpha}^{-1}(\overline{\alpha}(8\gamma(\cdot)))\,.
\end{aligned}$$

where $\lambda := -\frac{1}{\tau}\ln\left(\frac{\Lambda}{N}\right)$, and the $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}, \gamma, \chi$ and Γ are defined as in Assumption 1. We note that, since the function $\overline{\chi}$ does not depend on the parameters of the of the controller, but only on the nominal

comparison functions introduced in Assumption 1, it is possible for a class of control and disturbance affine systems to find a continuous feedback law for any desired ISS attenuation gain $\overline{\chi}$ (Praly and Wang, 1996; Teel and Praly, 1998).

Due to the particular design of the encoding-decoding procedure, measurement errors no longer appear as an input. Indeed, their effects are embedded in the last term of (10), which depends only on the parameters of the digital controller. As already anticipated, the constant Λ is an estimate of the expansion of the system between two successive sampling times. On the other hand, the constants $\overline{E}$ and E are proportional to the upper bound on the size of disturbances-uncertainties and measurement errors, respectively. In particular E is defined in (3) and

$$\overline{E} = \Lambda \max \left\{ \sup_{\mu,\nu\in\mathcal{P}} |\mu-\nu|, \sup_{d\in\mathcal{D}} |d| \right\}.$$

Hence, the last term in (10), which constitutes an upper bound to the steady-state error, is a continuous function of the known upper bound on the size of exogenous disturbances, that vanishes at zero. This guarantees that the steady-state error is small if disturbances are small, provided a sufficient knowledge of the plant. Moreover, when no perturbations apply, we recover the exact same result as (Liberzon and Hespanha, 2005).

Robustness to model uncertainties is the main contribution of this work if compared to the existing representative examples in the literature ((Kameneva and Nešić , 2008) and (Sharon and Liberzon, 2007)). In (Kameneva and Nešić , 2008) this lack of robustness is due to the digital nature of the controller, which is based on the *exact* dynamics or at most on its discrete time approximation (cf. Equation (2) in that reference). In (Sharon and Liberzon, 2007) this possible lack of robustness comes from the fact that ISS of the quantized closed-loop system is achieved through a cascade reasoning from the quantization error (which is ISS with respect to external disturbances thanks to the particular encoding/decoding strategy) to the system's state (which is ISS by hypothesis). This is possible because the evolution of the quantization error is shown to be independent from both the controller's and the system's state (cf. Equation (12) in that reference). This is no longer achievable if one introduces parametric uncertainties, as the state of the controller is fed back in the evolution equation of the quantization error. However, it would be interesting to study if this lack of robustness persists if under Assumption 1. Then, it would be worth comparing the "gains" given by the different methods. These studies are not presented here.

Another contribution compared to (Sharon and Liberzon, 2007) is the non-local characterization of robustness, which turns out to be semiglobal. In the statement of Theorem 2 in that reference, which gives an extension of the proposed algorithm to nonlinear systems, the admissible set of initial conditions and external disturbances are built starting from the $\mathcal{K}_\infty$ functions $\overline{\beta}_{cl}$ and $\overline{\gamma}_{cl}$, whose explicit expression depends on the Lipschitz constant of the system (cf. proof of Theorem 1 in that reference). Indeed, given a region where to define the Lipschitz constant ($|x| < l_x$ and $|w| < l_w$), it is possible to find the size of the ball of admissible initial conditions Δ and allowed disturbances ε by satisfying the two relations $\overline{\beta}_{cl}(\delta)+\overline{\gamma}(\varepsilon) < l_x$ and $\varepsilon < l_w$. It follows that the value of Δ and ε cannot be chosen *a priori*, and may result impossible to be arbitrarily enlarged, depending on the explicit expression of the two $\mathcal{K}_\infty$ functions $\overline{\beta}_{cl}$ and $\overline{\gamma}_{cl}$. On the other hand, given a compact set of initial conditions and a bound on the size of exogenous disturbances, it is not possible either to build the $\mathcal{K}_\infty$ functions used in the statement of the theorem, as it is not possible to build an "overshoot" region in which the Lipschitz constant would be defined. These observations show that in (Sharon and Liberzon, 2007) the extension to nonlinear systems is only local. In this paper we give a constructive way to build the overshoot region starting from an arbitrary ball of initial conditions and an arbitrary size for the exogenous disturbances.

However, considering the superior performances in the steady-state error of the algorithm proposed in (Sharon and Liberzon, 2007) (ISS instead of ISpS), one may think of implementing some switching strategy between the two methods to benefit from the advantages of each procedure. In a first step the state would be estimated with the algorithm proposed here even in the case of parametric uncertainties. In a second time, once the parameters of the systems have been identified and the state has entered a sufficiently small region around the origin, one would switch to the algorithm proposed in (Sharon and Liberzon, 2007).

In case of overflow, the size of quantization region is set to $\Lambda(\overline{E}+E)+E$ (cf. (7)). It may happen, in particular for large sampling periods or highly nonlinear systems, that Λ gets big. In this case, the quantization error may become very large as $\overline{E}$ depends linearly on Λ (cf. (21)), leading to a drop in performances. This can be easily avoided by using a suitable E in the encoding-decoding procedure. Indeed it follows from Claim 2 (see below) that, as long as no overflow occurs, the size of the quantization region converges to

$$Q_\infty := \sigma_\infty E,$$

where σ_∞ denotes a positive constant (see (23) below). It follows that the maximum quantization error (6) converges from above to

$$\bar{e}_\infty = \left(\frac{\sigma_\infty}{N} + 1\right) E. \tag{12}$$

As we show in the sequel (cf. (19)) an overflow can occur only if the maximum quantization error gets smaller than some constant $\bar{\eta}$ (defined in (16)), which denotes an upper bound on the size of perturbations and uncertainties. In other words, it suffices to set E such that

$$\bar{e}_\infty \geq \bar{\eta} \tag{13}$$

to avoid overflows. We then have the following theorem, whose proof follows directly from that of Theorem 1, together with Equations (12) and (13).

Theorem 2 (Fixed N - no overflows)**.** *Under the assumptions of Theorem 1, the design parameters $\tau, \Lambda, \bar{E}, E$ can be picked in such a way that (10) holds with $\delta = \bar{\gamma}\left(\left(1 + \frac{\Lambda+1}{1-\frac{\Lambda}{N}}\right) E\right)$.*

We point out that the size of the steady state error δ defined in Theorem 2 can be either larger or smaller than the one obtained in Theorem 1, depending on the parameters involved.

We finally state a similar result for the case when the sampling period is imposed by technological constraints and we can only adjust the number of transmittable bits. In this context, it appears that, due to the presence of exogenous perturbations, τ cannot be chosen arbitrarily large, as it happens in the ideal case (cf. (Liberzon and Hespanha, 2005)). This fact is detailed in the proof, given in Section 8.4.

Theorem 3 (Fixed sampling period)**.** *Let Assumption 1 hold for the system (1). Then, there exist class $\mathcal{K}_\infty$ functions $\bar{\chi}, \bar{\Gamma}, \bar{\gamma}$ and, given any compact sets $\mathcal{P} \subset \mathbb{R}^p$, $\mathcal{D} \subset \mathbb{R}^d$ and any $\Delta \in \mathbb{R}_{>0}$, there exists a time $\tau_{max} \in \mathbb{R}_{>0}$ such that, for all $\tau \in (0, \tau_{max})$, there exist positive constants $N, \Lambda, \bar{E}, E$ and a class $\mathcal{KL}$ function β such that trajectories of the closed-loop system (9), where $\hat{u}(t)$ is the output of the digital controller defined by equations (4),(5), (7),(8), satisfy (10) for all $x(0) \in B(0,\Delta)$, all $\nu \in \mathcal{P}$, all $\mu : \mathbb{R}_{\geq 0} \to \mathcal{P}$, all $d : \mathbb{R}_{\geq 0} \to \mathcal{D}$ and all $d_e : \mathbb{R}_{\geq 0} \to B(0,E)$. That is, (9) is semiglobally ISpS.*

We stress that the functions involved in the trajectories estimate of this result are the same as for Theorem 1.

Remark 3. *Theorems 1, 2 and 3 can be easily generalized to non-constant sampling periods, provided that the time between two samples does not exceed the value τ defined in the above statements.* ◁

6 ILLUSTRATIVE EXAMPLE

We check the application of our strategy on the control of a model of a DC motor with a load modeled as a nonlinear torque. The uncertainty on the load is modeled by unknown time-varying variables μ and d_1. Actuator errors are represented by an exogenous disturbance d_2:

$$\begin{aligned} \dot{x}_1 &= x_2 + \mu x_1^3 + d_1 \\ \dot{x}_2 &= u + d_2. \end{aligned}$$

For the needs of the numerical simulations, we have chosen $\mu(t) = 1 + P\sin(t)$, $d_1(t) = D\sin(t)$ and $d_2(t) = D\cos(t)$. At each sampling time the measurement available to the encoder is perturbed by the measurement error $d_e(t) = E(\sin(t), \cos(t))^T$. The system being in strict feedback form, we follow (Freeman and Kokotovich, 1993) to construct a continuous ISS feedback law. We assume that only 2 bytes can be transmitted at each sampling time. Our aim is to stabilize every solution starting in $B(0,10)$ (*i.e* $\Delta = 10$) assuming the following values for the perturbation amplitudes, $P = 0.5$, $D = 1.0$, $E = 0.1$. Note that this correspond to a 50% uncertainty on the load parameter. Our control scheme with parameters $\tau = 0.1s$, $\Lambda = 64$, $\bar{E} = 64$, E and $\nu = 1$ successfully stabilizes the system (cf. Figure 2).

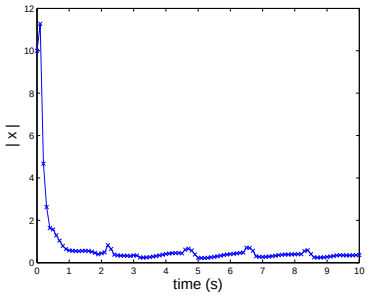


Figure 2: Evolution of the norm of the state from $x(0) = [-10, -10]$.

Since $E > 0$, the size of the quantization region remains sufficiently large (*i.e* $\bar{e}_\infty \geq \bar{\eta}$, cf. (13)) and no overflow occurs, as stated by Theorem 2. If an overflow had occurred then the size of the quantization region would have jumped to $\Lambda(\bar{E} + E) + E$ (cf. (7)), leading to a big drop in performances. This illustrates the fact that, even when no measurement error applies, setting an appropriate $E > 0$ may be very profitable in practice.

For $\Delta = 5$, and the same parameters, the sampling period can be taken as large as $0.3s$. We point out that the good performance of our strategy are also due to the ISS characteristics of the feedback strategy provided in (Freeman and Kokotovich, 1993).

In conclusion, with only two bytes, with a sampling period of the order of the plant's time scale, and with disturbances magnitude of the order of the nominal controlled dynamics values, our proposed approach succeeds in stabilizing the system, with a steady-state error of the same magnitude as the perturbations.

7 CONCLUSIONS

The proposed strategy for limited-information feedback control of nonlinear plants is shown to be robust to exogenous disturbances and measurement errors, even in presence of parametric uncertainty. Its application is illustrated by the numerical simulation of a DC motor control. Possible future extensions concern *output* feedback (see (Sharon and Liberzon, 2008) for a representative example) and robustness to delays.

8 PROOF OF THE MAIN RESULTS

8.1 Proof of Theorem 1

Contraction of the quantization region: Suppose there are no measurement errors, *i.e* $E = 0$. Then we want the estimation error to decrease as long as no overflow occurs, that is, as long as $x(k\tau) \in Q(k\tau)$, we impose $L(k\tau) < L((k-1)\tau)$. This means, in view of (5), that

$$\frac{\Lambda}{N} < 1. \tag{15}$$

Divergence between sampling times: During the time intervals separating two consecutive sampling times, the estimation error may increase. To evaluate this expansion, let us assume that a number $\hat{W} > 0$ is known[2] such that $x(t) + d_e \in B(0,\hat{W})$ and $\hat{x}(t) \in B(0,\hat{W})$ for all $t \geq 0$. In this case, it results from the continuity of κ that, for all $t \geq 0$, $|\hat{u}(t)| \leq \max_{\hat{x}\in B(0,\hat{W}),\nu\in\mathcal{P}} |\kappa(\hat{x},\nu)| =: U < \infty$. Let $\mathcal{L}(\hat{W})$ be the Lipschitz constant of f over the region $\{(x,\mu,u,d) \in \mathbb{R}^{n+p+m+h} : |x| \leq \hat{W}, \mu \in \mathcal{P}, |u| \leq U, d \in \mathcal{D}\}$, then, in view of (4) and exploiting the Bellman-Gronwell Lemma, it holds that, for all $x,\hat{x} \in B(0,\hat{W})$, $\frac{d}{dt}|e(t)| \leq |f(x,\mu(t),\hat{u},d(t)) - f(\hat{x},\nu,\hat{u},0)| \leq \mathcal{L}(\hat{W}) \max\{|e(t)|,\eta(t)\}$, where $\eta(t) = \max\{|\mu(t)-\nu|,|d(t)|\}$. Note that $\eta(t) \leq \overline{\eta}$, where

$$\overline{\eta} = \max\left\{ \sup_{\mu',\nu'\in\mathcal{P}} |\mu'-\nu'|, \sup_{d'\in\mathcal{D}} |d'| \right\}. \tag{16}$$

[2]We will demonstrate the existence of such $\hat{W}$, by constructing it, in the sequel (cf. Claim 1).

Hence, it holds that, for all $t \neq k\tau$, $k \in \mathbb{N}$,
$|e(t)| \geq \eta(t) \quad \Rightarrow \quad \frac{d}{dt}|e(t)| = \mathcal{L}(\hat{W})|e(t)|$, and
$|e(t)| < \eta(t) \quad \Rightarrow \quad \frac{d}{dt}|e(t)| = \mathcal{L}(\hat{W})\eta(t)$.
By the fact that $|e(0)|e^{\mathcal{L}(\hat{W})t} < |e(0)| + \overline{\eta}\mathcal{L}(\hat{W})t$ only if $|e(0)| < \overline{\eta}\frac{\mathcal{L}(\hat{W})t}{e^{\mathcal{L}(\hat{W})t}-1} \leq \overline{\eta}$, it follows that, for all $t \in [0,\tau]$,

$$|e(0)| \geq \overline{\eta} \;\Rightarrow\; |e(t)| \leq |e(0)|e^{\mathcal{L}(\hat{W})t}. \tag{17}$$

Defining $\Lambda = e^{\mathcal{L}(\hat{W})\tau}$, we can give a natural interpretation to (5). The constant N^{-1} describes the effect of measuring the state, which individuates a smaller hypercube to which the state belongs, while Λ describes the increase in the size of this hypercube during the time before the next sampling to make sure the state belongs to the new quantization region. Recalling that $|e(k\tau)| \leq \bar{e}(k\tau)$ for all $k \in \mathbb{N}$, we claim that

$$\bar{e}(k\tau) \geq \overline{\eta} \;\Rightarrow\; |e((k+1)\tau^-| \leq \Lambda\bar{e}(k\tau), \tag{18}$$

that is $x((k+1)\tau) \in Q((k+1)\tau)$ and, consequently, $q_{k+1} > 0$ (*i.e.* no overflow at step $k+1$). Indeed, note that if $\overline{\eta} \leq |e(k\tau)| \leq \bar{e}(k\tau)$, then (18) follows from (17), while, if $|e(k\tau)| < \overline{\eta}$, there exists $\tilde{t} := \min\{t \geq 0 : |e(k\tau)| + \overline{\eta}\mathcal{L}(\hat{W})t \geq \overline{\eta}\}$. If $\tilde{t} \geq \tau$ then $|e((k+1)\tau^-)| < \overline{\eta} < \Lambda\bar{e}(k\tau)$, while, if $\tilde{t} < \tau$, then, by (17), $|e((k+1)\tau^-)| \leq \overline{\eta}e^{(\tau-\tilde{t})\mathcal{L}(\hat{W})} < \Lambda\overline{\eta} \leq \Lambda\bar{e}(k\tau)$, which shows (18).

Furthermore, by reversing (18), we obtain that $|e((k+1)\tau^-| > \Lambda\bar{e}(k\tau)$ only if $\bar{e}(k\tau) < \overline{\eta}$, that is, defining $E = \sup_{d'_e\in\mathcal{E}} |d_e|$, $|e((k+1)\tau^-| + E > L((k+1)\tau)$ only if $\bar{e}(k\tau) < \overline{\eta}$, which implies that an overflow may occur only when the maximum quantization error (6) is below the bound $\overline{\eta}$:

$$q_{k+1} = 0 \quad \Rightarrow \quad \bar{e}(k\tau) < \overline{\eta}. \tag{19}$$

Moreover, we establish the following upper bound on the size of the estimation error right before an overflow:

$$q_{k+1} = 0 \quad \Rightarrow \quad |e((k+1)\tau^-)| < \Lambda\overline{\eta}. \tag{20}$$

Note that, from (19), there exists a time $\tilde{t}' := \min\{t > 0 : |e(k\tau)| + \overline{\eta}\mathcal{L}(\hat{W})t \geq \overline{\eta}\}$. If $\tilde{t}' \geq \tau$ then $|e((k+1)\tau^-)| \leq \overline{\eta} < \Lambda\overline{\eta}$. On the other hand, if $\tilde{t}' < \tau$ then, by (17), $|e((k+1)\tau^-)| \leq \overline{\eta}e^{(\tau-\tilde{t}')\mathcal{L}(\hat{W})} < \Lambda\overline{\eta}$, which establishes (20).

Trajectory boundness: Fix any $\Lambda > 1$ and let

$$\overline{E} = \Lambda\overline{\eta}. \tag{21}$$

By equation (20) and as long as $x(t) + d_e \in B(0,\hat{W})$, this implies that, in the eventuality of an overflow at time $k_0\tau$ (i.e. $q_{k_0} = 0$),

$$x(k_0\tau), x(k_0\tau) + d_e \in B(\hat{x}(k_0\tau), \overline{E} + E), \tag{22}$$

In view of (19) and (7), this implies that $x((k_0+1)\tau) \in Q((k_0+1)\tau)$, that is $q_{k_0+1} > 0$. This means that it is not possible to have two successive overflows.

Let $\Omega_c := \{x \in \mathbb{R}^n : V(x) \leq c\}$, where $c = \overline{\alpha}\Big(\chi(\sup_{d' \in \mathcal{D}} |d'|) + \Gamma(\sup_{\mu',\nu' \in \mathcal{P}} |\mu' - \nu'|) + \gamma\big(\max\{\Delta + E + \sigma_\infty E, \Lambda(\overline{E}+E)+E\}\big)\Big)$, where σ_∞ is defined as

$$\sigma_\infty := (\Lambda+1)N/(N-\Lambda). \tag{23}$$

Let

$$\hat{W} := W + \max\{\Delta + E + \sigma_\infty E, \Lambda(\overline{E}+E)+E\}, \tag{24}$$

$$W := \max\{\Delta, \sup_{x,z \in \Omega_c} |x - z|\}, \tag{25}$$

and pick the sampling period τ as $\tau = \frac{\ln(\Lambda)}{\mathcal{L}(\hat{W})}$. We prove the following in Sections 8.2 and 8.3 respectively.

Claim 1. *The solutions of the closed-loop system satisfy* $|x(t)| \leq W, \quad |\hat{x}(t)| \leq \hat{W}, \quad \forall t \geq 0.$

Claim 2. *As long as no overflow occurs, it holds that* $|e(t)| \leq e^{-\lambda t} L(0) + \sigma_\infty E$, *where* $\lambda = -\frac{1}{\tau}\ln\left(\frac{\Lambda}{N}\right)$

Conclusion: From the proof Claim 1, it results that, for all $t \geq 0$, $|e(t)| \leq \max\{(\Delta+E)e^{-\lambda t} + \sigma_\infty E, \Lambda(\overline{E}+E)+E\} \leq \Delta e^{-\lambda t} + \Lambda(\overline{E}+E) + (2+\sigma_\infty)E$, $\forall t \geq 0$. From Assumption 1, this implies that the trajectories of the closed loop system (9), with parameters $\{N, \tau, \Lambda, \overline{E}, \nu, E\}$, satisfy

$$|x(t)| \leq \underline{\alpha}^{-1}\Bigg(\overline{\alpha}\Big(\gamma(\Delta e^{-\lambda t}) + \chi(\|d\|) + \Gamma(\|\mu - \nu\|) + \gamma(\Lambda(\overline{E}+E) + (2+\sigma_\infty)E)\Big)\Bigg),$$

for all $x(0) \in B(0,\Delta)$, all $\nu \in \mathcal{P}$, all $\mu : \mathbb{R}_{\geq 0} \to \mathcal{P}$, all $d : \mathbb{R}_{\geq 0} \to \mathcal{D}$ and all $d_e : \mathbb{R}_{\geq 0} \to B(0,E)$. From this and from the fact that $\sigma(a+b) \leq \sigma(2a) + \sigma(2b)$ for all nondecreasing function σ and all $a, b \geq 0$, the theorem is proved with

$$\overline{\beta}(\cdot,t) = \underline{\alpha}^{-1}(\overline{\alpha}(2\gamma(\cdot)\gamma(e^{-\lambda t}))) \tag{26a}$$

$$\overline{\chi}(\cdot) = \underline{\alpha}^{-1}(\overline{\alpha}(4\chi(\cdot))) \tag{26b}$$

$$\overline{\Gamma}(\cdot) = \underline{\alpha}^{-1}(\overline{\alpha}(8\Gamma(\cdot))) \tag{26c}$$

$$\overline{\gamma}(\cdot) = \underline{\alpha}^{-1}(\overline{\alpha}(8\gamma(\cdot))). \tag{26d}$$

8.2 Proof of Claim 1

Let $\Theta := \inf\{t \in \mathbb{R}_{>0} : |x(t)| > W \quad \text{or} \quad |\hat{x}(t)| > \hat{W}\}$. This time is well defined as $|x(0)| \leq \Delta \leq W$ and, from the fact that $q_0 > 0$ by construction, $\hat{x}(0) \in B(0,\Delta+E) \subset B(0,\hat{W})$. For all $t \in [0,\Theta)$, $\mathcal{L}(\hat{W})$ can be correctly interprated as an upper bound on the expansion of the system. In particular, Claim 2, (19) and (20) hold for all $t \in [0,\Theta)$.

Define $k_0\tau$ as the time of the first overflow. Then, by Claim 2, it holds that, for all $t \in [0, \min(\Theta, (k_0-1)\tau))$, $|e(t)| < L(0)e^{-\lambda t} + \sigma_\infty E \leq \Delta + E + \sigma_\infty E$, where $\lambda = \frac{|\ln(\frac{\Lambda}{N})|}{\tau}$.
If $\Theta < (k_0-1)\tau$, then, from Assumption 1 and (A.Isidori, 1999, Section 10.4), it results that the set $\Omega_{\tilde{c}} = \{x \in \mathbb{R}^n : V(x) \leq \tilde{c}\}$, where $\tilde{c} := \overline{\alpha}(\chi(\sup_{d' \in \mathcal{D}} |d'|) + \Gamma(\sup_{\mu',\nu' \in \mathcal{P}} |\mu' - \nu'|) + \gamma(\Delta + E + \sigma_\infty E))$, is an invariant attractive set, and, noting that $\tilde{c} \leq c$, it follows that $\Omega_{\tilde{c}} \subseteq \Omega_c$, which, by the definition of W in (25), implies $|x(t)| \leq W$ for all $t \in [0,\Theta]$. This in turn ensures that $\sup_{t \in [0,\Theta]} |\hat{x}(t)| \leq W + \sup_{t \in [0,\Theta]} |e(t)| \leq W + \Delta + E + \sigma_\infty E \leq \hat{W}$ (cf. (24)). This contradicts the definition of Θ and hence we conclude that $\Theta \geq (k_0-1)\tau$.

If $\Theta \in [(k_0-1)\tau, k_0\tau)$, then, by (19) and (20) and the definition of $\overline{E}$ in (21), it results that $|e(t)| < \overline{E}$ for all $t \in [t_0 - \tau, \Theta]$. With the same arguments as before, this contradicts the definition of Θ and hence we conclude $\Theta \geq k_0\tau$.

If $\Theta \in [k_0\tau, (k_0+1)\tau)$, by $\overline{E} > \overline{\eta}$ and (22), we get that $|e(t)| \leq \Lambda(\overline{E}+E)$ for all $t \in [k_0\tau, \Theta]$, again contradicting the definition of Θ. Hence we can conclude $\Theta \geq (k_0+1)\tau$.

If $\Theta = (k_0+1)\tau$, then by construction $q_{k_0+1} > 0$ and $|e((k_0+1)\tau)| \leq L((k_0+1)\tau) = \Lambda(\overline{E}+E) + E$, again contradicting the definition of Θ. Hence $\Theta > (k_0+1)\tau$.

The system properties established along the whole proof being uniform in time, we can set $t' = t - (k_0+1)\tau$ and apply the same arguments with new "initial" condition $L(0) = \Lambda(\overline{E}+E) + E$ until the next overflow. By reiterating for successive overflows, we conclude $\Theta = \infty$, which is enough to prove the claim.

8.3 Proof of Claim 2

The first line in (5), can be rewritten as

$$L(k\tau) = R^k L(0) + E(\Lambda+1)\sum_{i=0}^{k} R^i < \tilde{L}(k\tau), \tag{27}$$

where $\tilde{L} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is defined as

$$\tilde{L}(t) = R^{t/\tau} L(0) + E(\Lambda+1)\sum_{i=0}^{\infty} R^i, \tag{28}$$

and $R := \Lambda/N < 1$ (cf. (15)).
If Λ is chosen appropriately to compensate for error divergences between sampling times, if no overflow occurs, then, recalling the definition of the maximum quantization error (6), forall $t \in [k\tau, (k+1)\tau)$, $|e(t)| \leq \Lambda^{\frac{t-k\tau}{\tau}}\left(\frac{L(k\tau)}{N} + E\right)$. Substituting (27) in the this equation, we obtain that, for all $t \in [k\tau, (k+1)\tau)$,

$|e(t)| \leq \left(\frac{\Lambda}{N}\right)^{t/\tau} L(0) + \frac{\Lambda^{\frac{t-k\tau}{\tau}}}{N} E(\Lambda+1)\sum_{i=0}^{k}\left(\frac{\Lambda}{N}\right)^{i} + \Lambda E < \tilde{L}(t)$. Recalling finally that the geometric series in the definition of $\tilde{L}$ (28) converges, $\sum_{i=0}^{\infty}\left(\frac{\Lambda}{N}\right)^{i} = \frac{N}{N-\Lambda}$, this finishes to establish Claim 2 by recalling the definition of σ_∞ in (23) and by noticing that the error can only decrease at the sampling times.

8.4 Proof of Theorem 3

The proof follows along the same lines as that of Theorem 1. The difference stands in the fact that Λ can no longer be chosen arbitrarily. Instead, given any $\tau > 0$, we define $\Lambda = e^{\mathcal{L}(\hat{W})\tau}$, where $\hat{W}$ is defined as before (cf. (24)). Note that Λ enters the definition of $\hat{W}$, in particular $\hat{W}$ and $\mathcal{L}(\hat{W})$ are both continuous increasing functions of Λ. Hence, Λ is required to satisfy the following two equations $\Lambda = e^{\mathcal{L}(\hat{W})\tau}$ and $\hat{W} = \hat{W}(\Lambda)$. This set of equations admits other solutions than the trivial one ($\Lambda = 1$, $\tau = 0$) provided that

$$\tau e^{\mathcal{L}(\hat{W}(\Lambda))\tau} \frac{d}{d\Lambda}\mathcal{L}(\hat{W}(\Lambda))\Big|_{\Lambda=1} < 1\,.$$

By continuity of the equations in τ and Λ, we conclude that there exists $\tau_{\max} > 0$ such that a solution exists for all $\tau \in (0, \tau_{\max})$. The rest of the proof follows that of Theorem 1.

REFERENCES

A.Isidori (1999). *Nonlinear control system II*. Springer Verlag.

Freeman, R. A. and Kokotovich, P. V. (1993). Global robustness of nonlinear systems to state measurment disturbances. *IEEE 32nd Conference on Decision and Control*, pages 1507–1512.

Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95:138–162.

Jaglin, J., de Wit, C. C., and Siclet, C. (2008). Delta modulation for multivariable centralized linear networked controlled systems.

Jaglin, J., de Wit, C. C., and Siclet, C. (2009). Adaptive quantization for linear systems. *Submitted to IEEE Conf. on Decision and Control 2009*.

Jiang, Z. P., Teel, A., and Praly, L. (1994). Small gain theorems for ISS systems and applications. *Math. of Cont. Sign. and Syst.*, 7:95–120.

Kameneva, T. and Nešić , D. (2008). Input-to-state stabilization of nonlinear systems with quantized feedback. In *Proc. 17th. IFAC World Congress*, pages 12480–12485.

Liberzon, D. (2003). On stabilization of linear systems with limited information. *IEEE Trans. on Automat. Contr.*, 48(2):304–307.

Liberzon, D. (2009). Nonlinear control with limited information. Roger Brockett legacy special issue. *Communications in Information Systems*, 9:44–58. Available at: http://decision.csl.uiuc.edu/ liberzon/research/legacy.pdf.

Liberzon, D. and Hespanha, J. P. (2005). Stabilization of nonlinear systems with limited information feedback. *IEEE Trans. on Automat. Contr.*, 50:910–915.

Liberzon, D. and Nešić , D. (2007). Input-to-state stabilization of linear systems with quantized state measurements. *IEEE Trans. on Automatic Control*, 52:767–781.

Montestruque, L. A. and Antsaklis, P. (2004). Stability of model-based networked control systems with time-varying transmission times. *IEEE Trans. on Automat. Contr.*, 49(9):1562–1572.

Nair, G. N. and Evans, R. J. (2004). Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM J. Control Optim.*, 43(2):413–436.

Nair, G. N., Fagnani, F., Zampieri, S., and Evans, R. J. (2007). Feedback control under data rate constraints: An overview. *Proc. of the IEEE*, 95(8):108–137.

Persis, C. D. and Isidori, A. (2004). Stabilizability by state feedback implies stabilizability by encoded state feedback. *System & Control Letters*, 53:249–258.

Petersen, I. R. and Savkin, A. V. (2001). Multi-rate stabilization of multivariable discrete-time linear systems via a limited capacity communication channel. *Proc. 40th IEEE Conf. on Decision and Control*.

Praly, L. and Wang, Y. (1996). Stabilization in spite of matched unmodelled dynamics and an equivalent definition of input-to-state stability. *Math. of Cont. Sign. and Syst.*, 9:1–33.

Sharon, Y. and Liberzon, D. (2007). Input-to-State Stabilization with minimum number of quantization regions. pages 20–25.

Sharon, Y. and Liberzon, D. (2008). Input-to-State Stabilization with quantized output feedback. pages 500–513.

Sontag, E. and Wang, Y. (1995). On characterizations of the Input-to-State Stability property. *Syst. & Contr. Letters*, 24:351–359.

Teel, A. and Praly, L. (1998). On assigning the derivative of a disturbance attenuation clf. *Proc. 37th. IEEE Conf. Decision Contr.*, 3:2497–2502.

DISTRIBUTED KALMAN FILTER-BASED TARGET TRACKING IN WIRELESS SENSOR NETWORKS

Phuong Pham and Sesh Commuri
School of Electrical and Computer Engineering, The University of Oklahoma
110 W. Boyd St., Devon Energy Hall 150, Norman, Oklahoma 73019-1102, U.S.A.
{Phuongpham, scommuri}@ou.edu

Keywords: Distributed Kalman Filter, Wireless Sensor Networks, and Target Tracking.

Abstract: The tracking of mobile targets using Distributed Kalman Filters in a Wireless Sensor Network (WSN) is addressed in this paper. In contrast to the Kalman Filter implementations reported in the literature, our approach has the Kalman Filter running on only one network node at any given time. The knowledge learned by this node, i.e. the system state and the covariance matrix, is passed on to the subsequent node running the filter. Since a finite subset of the sensor nodes is active at any given time, target tracking can be accomplished using lower power compared to centralized implementations of the Kalman Filter. Numerical simulations demonstrate that the proposed algorithm is robust to measurement noise and changes in the velocity of the target. The results in this paper show that the proposed technique for target tracking will result in significant savings in power consumption and will extend the useful life of the WSN.

1 INTRODUCTION

Surveillance of remote inaccessible areas and the detection and tracking of intruders are some of the important applications of Wireless Sensor Networks (WSNs). Research in WSNs has addressed several important issues in optimal deployment, coverage, routing, and energy efficiency of the WSNs (Akyildiz, Su, Sankarasubramaniam, and Cayirci, 2002; Al-Karaki and Kamal, 2004; Cardei, Thai, Li, and Wu, 2005; Chiang, Wu, Liu, and Gerla, 1997; Watfa and Commuri, 2006a, 2006b) Diffusion and directed diffusion approaches have been proposed to address coverage, routing, discovering, and sensing fusion issues in WSNs (Intanagonwiwat, Govindan, and Estrin, 2000). The application of WSNs in surveillance and monitoring of target areas have also been widely researched (Chen, Gonzalez, and Leung, 2007). While the results presented in these papers are encouraging, their applicability in low cost WSNs with large measurement noise and faulty measurements is fraught with problems. In recent years, Kalman Filters have been proposed to address the uncertainty and the noise in the measurements (Rao and Durrant-Whyte, 1991; Olfati-Saber, 2007; Alriksson and Rantzer, 2007; Olfati-Saber and Shamma, 2005; Cattivelli, Lopes, and Sayed, 2008; Uhlmann, 1996; Kim, West, Scholte, and Narayanan, 2008; Mutambara, 1998; Hashemipour, Roy, and Laub, 1998). Both centralized and distributed implementation of the Kalman Filter was proposed to make their use suitable to WSN applications. However, these techniques are still power intensive and require significant amounts of onboard power for communication and computation.

Two classes of Kalman filtering approaches have been implemented in WSNs. The first approach is centralized Kalman Filters (Rao, et al., 1991) where every sensor node takes measurements and communicates with the other nodes while simultaneously performing its own version of Kalman Filter. In this approach, the sensor nodes' power will be depleted quickly because of excessive measurements and inter-node communication. Moreover, it is sometimes impractical for a sensor node to communicate with all the other nodes due to limitation of communication ranges. The second method is distributed Kalman Filters (Olfati-Saber, 2007; Olfati-Saber, et al., 2005; Cattivelli, et al., 2008) where every neighbor node runs its own version of the Kalman Filter and shares the information with all other neighbors to reach the consensus of the system. The approaches above are distributed in processing. The number of neighbor nodes determines how expensive the algorithms are

in terms of power consumption and communication complexity. Consequently, these approaches are not efficient because they require extensive inter-communication among neighbor nodes. In comparison with the distributed version of Kalman Filter in literature (Rao, et al., 1991; Olfati-Saber, 2007; Alriksson, et al., 2007; Olfati-Saber, et al., 2005; Cattivelli, et al., 2008; Hashemipour, et al., 1998), our version of the distributed Kalman Filter simplifies computational burden and reduces inter-node communication. Thus, the total power consumption in the entire sensor network is lower than that reported elsewhere in the literature.

Our approach is different from the above work in the sense that the Kalman Filter is implemented in a distributed fashion across the WSNs. At a given instant, only one master node runs the Kalman Filter using the measurement inputs from its neighbors and shares the estimated knowledge with the subsequent master node. The neighbors within a certain distance from the target measure the distance to the target, and transmit measurements to the master node. On one hand, the procedure significantly reduces the communication costs among the neighbor nodes in comparison with the algorithms proposed in (Rao, et al., 1991; Olfati-Saber, 2007; Alriksson, et al., 2007; Olfati-Saber, et al., 2005; Cattivelli, et al., 2008; Hashemipour, et al., 1998). On the other hand, since the master node alone executes the Kalman Filter and the neighbor nodes only perform measurement functions, the complexity of the WSN is greatly reduced.

Another contribution of this paper is that the master node determines the direction and velocity of the intruder and wakes up appropriate sensor nodes in the direction of the target travel. As the target moves into the sensing range of a sensor node, it is already activated and is ready to take measurements. Whereas the other nodes that are far away from the target are automatically turned off to save energy. The master node also decides to wake up sufficient nodes to take measurements. By knowing the maximum target's velocity, the boundary nodes of the sensor field are activated in round robin fashion discussed in (Watfa, et al., 2006b) to save energy.

Unlike other approaches mentioned above, we do not make an assumption about the linear movement of the target. In this paper, the distributed Kalman Filter is proposed to estimate the position of the target. This approach is validated through simulation examples and the results are compared with those represented in literature. We show the main contribution, the approach, validations, and comparison between our method and the previous work on distributed Kalman filtering. The algorithm was also able to track the target with random directions with acceptable estimated results. The estimation results showed that the model is robust to measurement noise and the change in velocity. The estimated knowledge of the Kalman Filter including system state and covariance matrix is passed directly to the subsequent master node where the Kalman Filter is run. Consequently, the performance of the distributed Kalman Filter is as good as that of the centralized Kalman Filter.

The rest of the paper is organized as follows: Section 2 discusses the algorithm in details. In section 3, we show the numerical simulation. Section 4 and 5 are discussion and conclusion.

2 ALGORITHM

2.1 Problems and Assumptions

A sensor field is densely deployed with sensor nodes. It is assumed that each node has omnidirectional sensing capability to measure the distance between the target and itself. Moreover, every node knows its coordinates in the sensor field, and all nodes are stationary. Initially, all the nodes except those at the boundary of the monitored area are assumed to be in sleep mode. Assuming that there is an intruder entering the sensor field with an unknown nonlinear trajectory and a known maximum velocity, the problem is to track the position of the intruder accurately. When a target moves in the sensor field, the nodes close to the target will automatically activate and sense the target.

All sensing nodes are within one communication hop from the master node. The trilateration algorithm requires that every point in the field is covered by at least three sensor nodes.

A node can be either the master node or a measurement node. Nodes take measurements and sends data to the master node if they are actively in the sensing region. Concurrently, the master node collects data from its neighbors, running estimation algorithms and broadcasting the information of the target to its neighbors, including the target's current coordinates and direction. Depending on the information from the master node, the neighbor nodes around the target automatically turn off when they are not in the region of activation $\boldsymbol{R}$ around which is defined as the following.

The target, represented by $\star$ symbol shown in Figure 1, is moving in horizontal direction. The

region ***R*** is defined by the circle radius R_1, the radius of R_2 and angle 2α – the region limited by the bold line. R_2, R_1, and R_a ($R_2 > R_1 > R_a$) are activation radius, sensing radius, and measurement radius respectively. All the sensor nodes inside the region of activation ***R*** are activated, while the nodes outside the region are in sleep mode to save power. All the nodes inside the circle (O, R_1) can sense the target while no node outside can detect the target. However, only nodes inside the circle (O, R_a) are actively taking measurements and reporting the data to the master node. This is done to account the imprecision in the location information of a given sensor node. For example, if there is 20% uncertainty in measurement accuracy then the solution $R_1=1.2R_a$ can ensure that there are no sensor nodes outside the circle (O, R_1) that can detect the target ⋆. Assuming that the maximum target velocity is known, and the direction of the target does not change sharply. The selection of $R_2=1.8R_a$ and $2\alpha = 60^0$ can guarantee the sensors in the moving direction of the target are activated in advance. Thus, the WSN can track the target continuously without any interruption.

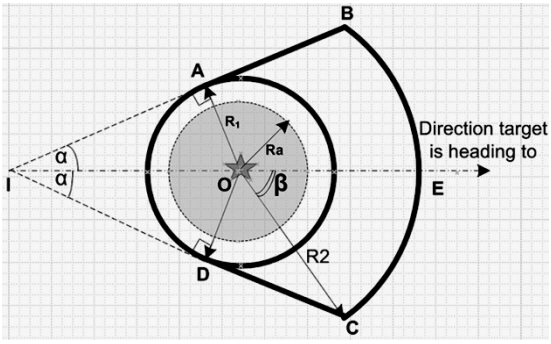


Figure 1: The target represented by ⋆ at point O. The boundary of the region of activation ***R*** is limited by line AB, curve BC, line CD and curve CA (the bold line above). The curve BC is formed by part of the circle (O, R_2). No nodes outside circle (O, R_1) can sense the target. All the nodes inside ***R*** are activated. However, only the sensors inside the circle (O, R_a) are actively taking measurement.

2.2 Settings

Initially, the sensor nodes in the boundary of the field are on to detect intruders while the all other sensors are off. If the maximum velocity of a target is known, then the boundary nodes can turn on and off periodically without losing the ability to track the incoming target according to (Watfa, et al., 2006b). When the boundary nodes detect an intruder, the region ***R*** is formed and the nodes inside are activated.

A master node is selected depending on two criteria: the distance to the target and power residual. The sensors inside the circle with radius R_a take measurements and transfer the measured data to the master node. The master node runs the Kalman Filter and obtains the estimated position and the direction of the target. The master node broadcasts the learned knowledge of the target to its neighbors. After receiving the information, a node will turn on or off depending on whether it is inside or outside region ***R***.

2.3 Position Calculation

After receiving the measurement from the target's neighbor sensor nodes, the master node uses the trilateration and the least square algorithm to calculate the position of the target.

Suppose there are k sensor nodes that are actively taking measurements whose coordinates are (x_1, y_1); (x_2, y_2); … (x_k, y_k), and measured distances from each nodes to the target are d_1, d_2 , … d_k respectively.

The least square solution of the target's coordinate (x_t, y_t) is:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = (A^T A)^{-1} Ab \tag{1}$$

where A and b are in the following form:

$$A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) \\ \dots & \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) \\ 2(x_1 - x_k) & 2(y_1 - y_k) \end{bmatrix} \tag{2}$$

$$b = \begin{bmatrix} (d_1^2 - d_2^2) + (x_2^2 + y_2^2) - (x_1^2 + y_1^2) \\ (d_2^2 - d_3^2) + (x_3^2 + y_3^2) - (x_2^2 + y_2^2) \\ \dots \\ (d_{k-1}^2 - d_k^2) + (x_k^2 + y_k^2) - (x_{k-1}^2 + y_{k-1}^2) \\ (d_k^2 - d_1^2) + (x_1^2 + y_1^2) - (x_k^2 + y_k^2) \end{bmatrix} \tag{3}$$

2.4 Power Consumption

The transmitted power P_{Tx}, received power P_{Rx}, idle power P_i and sleeping power P_s are *1400 mW, 1000 mW, 830 mW,* and *130 mW* respectively based on the power consumption analysis in (Watfa, et al.,

2006b). From region **R**, the number of sensor nodes inside the circle radius R_a is N_a. N_i is the number of sensor nodes outside the circle with radius of R_a, but inside the region ***R***. The number of sensor nodes in the sensor field and number of active sensor nodes in the boundary are N and N_b respectively. The total power consumption of the sensor field in one sampling cycle is calculated as following.

The N_a neighbors make N_a transmissions and the master node receives N_a times.

$$P_{meas} = N_a(P_{Tx} + P_{Rx}) \tag{4}$$

The master node broadcasts the target position and its directions, and it makes one transmission. Each of $(N_a + N_i)$ neighbors in the cone area receives the information of the target once.

$$P_{broadcast} = (N_a + N_i)P_{Rx} + P_{Tx} \tag{5}$$

Each active node, except measurement nodes, consumes an amount of the idle energy

$$P_{idle} = (N_b + N_i)P_i \tag{6}$$

The other nodes are sleeping, and the total power consumed by these nodes is

$$P_{sleep} = (N - N_a - N_i - N_b)P_s \tag{7}$$

Then total consumed power is

$$P_w = P_{meas} + P_{broadcast} + P_{idle} + P_{sleep} \tag{8}$$

2.5 Distributed Kalman Filter

Local prediction (see (Rao, et al., 1991))

$$\begin{aligned} \hat{x}(k+1\,|\,k) &= F(k)\times\hat{x}(k\,|\,k) \\ P(k+1\,|\,k) &= F(k)\times P(k\,|\,k)\times F^T(k) + Q \end{aligned} \tag{9}$$

Local update

$$\begin{aligned} P^{-1}(k+1|k) &= P^{-1}(k|k) + H^T(k+1)\times R^{-1}(k+1)\times H(k+1) \\ W(k+1) &= P(k+1)\times H^T(k+1)\times R^{-1}(k+1) \\ \breve{x}(k+1\,|\,k+1) &= \hat{x}(k+1\,|\,k) + W(k+1) \\ &\quad \times\left[z(k+1) - H(k+1)\times\hat{x}(k+1\,|\,k)\right] \end{aligned} \tag{10}$$

Where $z(k+1)$ is the target position calculated in (1). The knowledge passed to the subsequent master node $\breve{x}(k+1\,|\,k+1)$ *and* $P(k+1\,|\,k+1)$

3 NUMERICAL EXAMPLES

We will consider two scenarios to demonstrate the distributed Kalman Filter for target tracking. In the first, it is assumed that sensor nodes are uniformly distributed. This requirement is relaxed in the second scenario where the nodes are randomly deployed. It is assumed that there is no hole in coverage within the regions to be monitored, and every point is covered by at least three sensors.

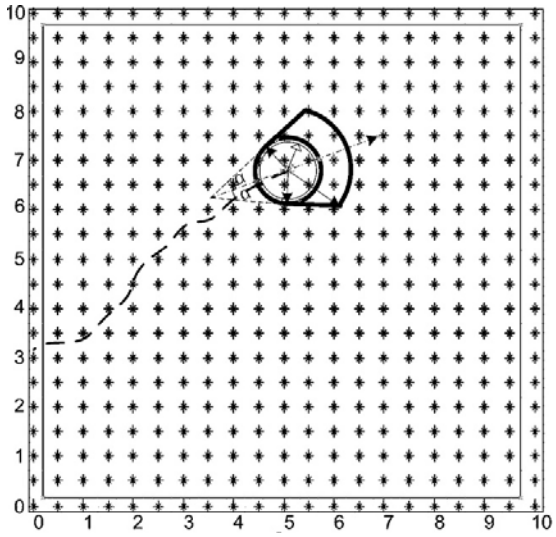


Figure 2: Example of sensor field and the trajectory of the target. The sensor nodes in the boundary of the field are always active. In the figure, all the nodes in the cone area around the target are activated.

The sensor field is assumed to be a square of the dimension 10×10 units as seen in Figure 2. By choosing the distance of any two closest nodes is 0.5 units, the total number of uniformly distributed sensor nodes is 441. The target is assumed to move along the horizontal trajectory with the sinusoid velocity profile while the vertical coordinate remains at y = 5. In 10 seconds, the target travels between the coordinates (0, 5) and (10, 5). The sampling frequency is 200Hz and the simulation time is 10 seconds. The following difference equations are used to model the dynamic behaviors of the moving target.

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ z_k &= Hx_k + v_k \end{aligned} \tag{11}$$

Where $F = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix}$, $x_k = \begin{bmatrix} v_k \\ p_k \end{bmatrix}$, $H = [0\ \ 1]$

x_k is the target velocity and and p_k is target position in x the direction at time k. Δt is the sampling time.

Moreover, w_k and v_k are Gaussian distributed with zero mean state noise and measurement noise. From scenario 1 to scenario 4, the initial condition for the Kalman Filter is the same as the true value while it is nonzero in scenario 5. The sensor nodes are uniformly deployed in scenario 1 to scenario 5 while randomly deployed in scenario 6.

Scenario 1: Without using the Kalman Filter, more sensors used in measurement results in better estimated tracking. As seen in Table 1, when the average measured sensor nodes increased from 4.5 to 17.5, the noise variance decreased from 21.71×10^{-3} to13.49×10^{-3}. However, the trade off is the total power consumption of the network increases from 1.38×10^5 to 2.09×10^5 (mW). The power consumption analysis is shown in Figure 3.

Table 1: Performance analysis.

Average measured sensors	Average active sensors	Error variance without Kalman Filter ($\times 10^{-3}$)	Error variance with Kalman Filter ($\times 10^{-3}$)	Average total power consumption ($mW \times 10^5$)
4.5	9.3	24.71	3.63	1.38
17.5	39.2	13.49	1.57	2.09
60.4	139.9	7.03	0.98	4.48
130.8	275.5	4.62	0.31	7.88
279.1	416.2	5.43	0.10	12.60

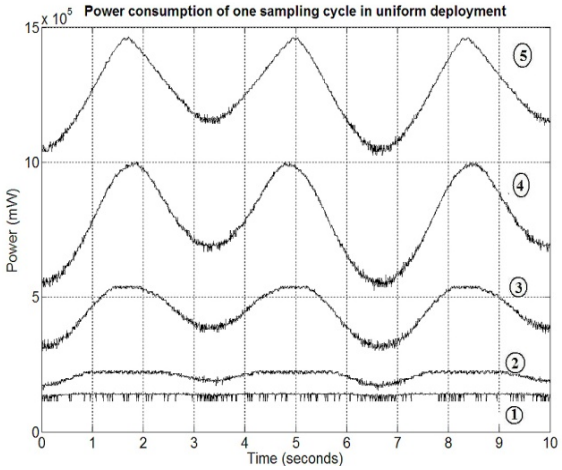


Figure 3: Without the Kalman Filter, the line number 1, 2, 3, 4, and 5 have average measured sensor nodes of 4.5, 17.5, 60.4, 130.8, and 279 respectively. For the line number 3 to 5, the total power consumption is fluctuated because when the target moves close to the boundary the number of active sensors is reduced. Then the total power consumption reduces. Line #1 and #2 are quite flat because in these scenarios the relatively small cone regions result in small difference in the number of active sensors when the target in the middle of the field and when it is close to the boundary.

Scenario 2: When the Kalman Filter is used, the variance of the estimated error is smaller and Figure 4 shows the smoother tracking performance compared to scenario 1. As shown in Table 1, by using the Kalman Filter, only an average of 4.5 measured sensors is sufficient to achieve the error variance of 3.63×10^{-3} which is smaller than 5.43×10^{-3}resulted by an average of 279.1 measured sensors without using Kalman filtering.

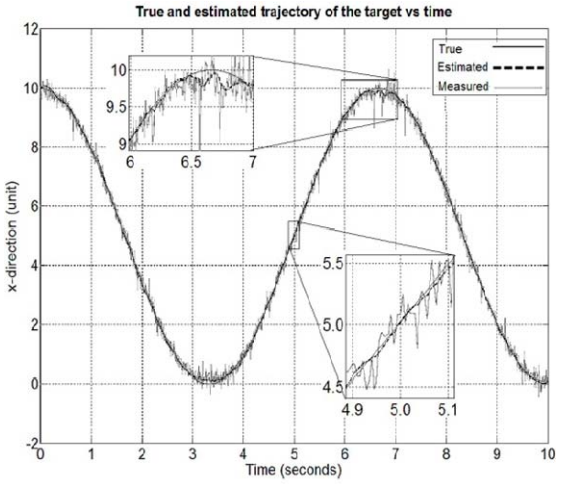


Figure 4: Target's true trajectory is the solid black line, and its estimations using trilateration with the Kalman Filter and without the Kalman Filter are the solid gray line and the dashed black line respectively. The average number of measured sensors is 4.5, and the standard deviation of state noise and measurement noise are 0.01 and 0.2 respectively. The Kalman Filter yielded both a smaller error variance and smoother estimated trajectory. As we zoom in two small sub figures, the estimated position is close to the true position when the target moves in a linear part of the sinusoid trajectory. Without using the Kalman Filter, the estimated trajectory is noisy.

Scenario 3: When the number of average measured sensors and the sampling frequency are fixed, slower average velocity results in smaller estimated tracking error as shown in Figure 5. In this scenario, the sampling frequency is 200Hz, the standard deviation of state noise and measurement noise are 0.01 and 0.2 respectively, and the average number of measured sensors is 6.3.

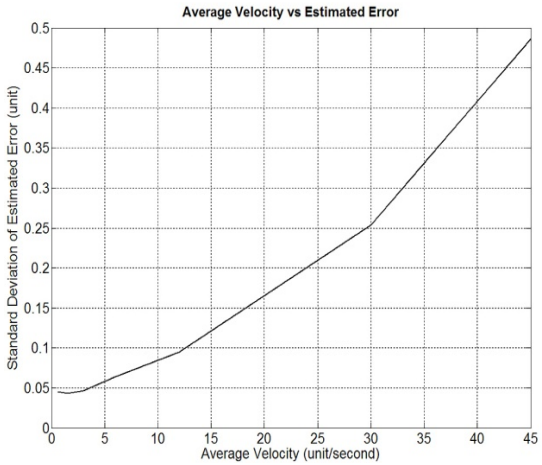


Figure 5: Average velocity increases as the estimated error has a larger standard deviation.

Scenario 4: In this scenario, the sampling frequency is kept at 200Hz, average target velocity is three units per second and the average number of measured sensors is 6.5. In Figure 6, the standard deviation of state noise is fixed at 0.01 while the measurement noise has a standard deviation varying from 0.01 to 0.5. The variance of estimated error increases with the increase in measurement noise. In addition, with the same number of average measured sensors of 6.5, the smaller measurement noise leads to the better tracking performance. The tracking performance, shown in Figure 7, is better when the measurement noise is smaller.

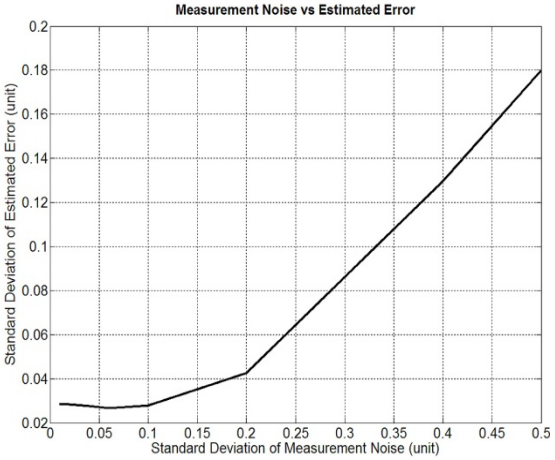


Figure 6: When the distance measurement is subjected to a larger noise, the variance of estimated tracking error becomes bigger.

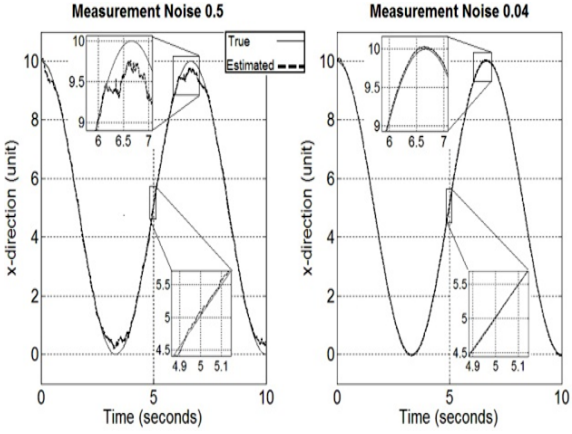


Figure 7: The true and the estimated trajectory with different measurement noise levels. The standard deviation of measurement noise is 0.5 in the left side while it is 0.04 on the right side.

Scenario 5: When the master node does not share the knowledge of the target including the target state and the covariance matrix with the subsequent one, the subsequent master node has to run the Kalman Filter with the default initial conditions. Assuming that the difference between the initial position and the actual target position is the measurement error, the change in master nodes is indicated by the abrupt jumps in estimated error as shown in Figure 8. When there is a change in the master node, the Kalman Filter requires some extra time steps to converge.

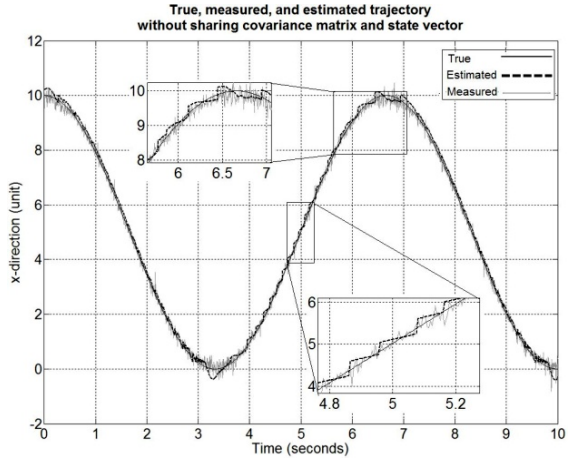


Figure 8: Without sharing the state vector and covariance matrix to the subsequently master node, each master node has to start the Kalman Filter from scratch. The measurement noise standard deviation is 0.2, while the number of average measured sensor nodes is 7.6.

Scenario 6: As shown in Figure 9, when the sensor nodes are randomly distributed, we get similar results in comparison with the uniform scenario

shown in Figure 3. However, the power consumption is not as smooth as it is in the uniform scenario. Due to the random nature, there are more sensor nodes covering a specific point while fewer sensor nodes are covering other points. In order for our algorithm to work effectively, at least three sensor nodes must cover each point in the sensor field

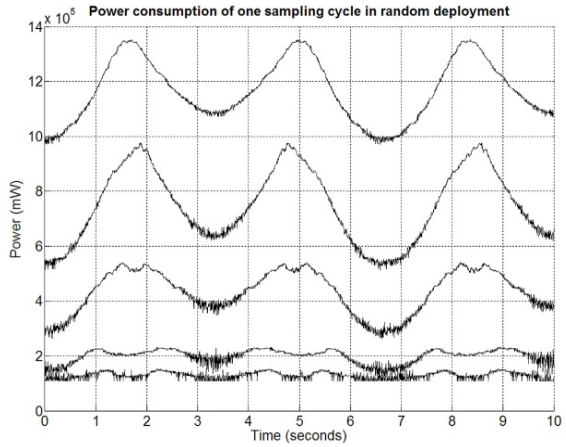


Figure 9: Power consumption of one sampling cycle in random deployment. There are 441 sensor nodes deployed in the sensor field of 10×10. The line number 1, 2, 3, 4 and 5 have average measured sensors of 3.4, 15.7 59.5, 127.2, and 259.7 respectively.

4 DISCUSSIONS

The above results show that the distributed Kalman Filter implementation in a WSN is successful in tracking moving targets. The tracking error is small when the target follows a linear trajectory while nonlinear trajectories with high target velocities result in higher tracking errors. However, in all these scenarios, the tracking error is 12.5% smaller than that obtained in the absence of the Kalman Filter. In addition to the improved tracking performance, the distributed filter requires fewer nodes to be active at any given instant, thereby reducing the overall power consumption of the WSN. This is significant because the lowered power consumption increases the useful life of the WSN.

The choice of the cluster head is determined by the residual power ($P_{residual}$) of each node and its distance to the target. At each instant, every active node in the proximity of the target computes the weighted sum of its residual power and its distance to the target (D) as following $W_{node} = \alpha D + \beta P_{residual}$ with constants α and β in the interval $[0, 1]$. A node will become the new master node if its weighted sum is smaller than that of the current master node. Consequently, the knowledge of the Kalman filtering is transferred from the current master node to the new one.

5 CONCLUSIONS

In this paper, a method for the target tracking problem using distributed Kalman Filter in WSNs is demonstrated. The algorithm is robust to changes in the velocity of the target and measurement noises. The algorithm reduces the total power consumption in the network in comparison with distributed Kalman Filter algorithms elsewhere in literature. Another contribution of the proposed algorithm is the activation of a reduced set of sensor nodes for target tracking. Thus, sensor nodes further away from the target are inactive and thereby conserve power. Fewer active nodes also mean reduced communication among nodes. These two factors together increase the useful life of the WSN while provide accurate tracking in the presence of measurement noise and target uncertainty.

The results presented in this paper assume that each sensor node knows its position accurately and share a common system clock with other nodes. This is not a detriment as results in time synchronization and localization already exist in the literature. Proof of the convergence of the tracking error and the stability of the overall system will be presented in an extended version of the paper.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor network, *IEEE Communications Magazine* (Vol. 40 pp. 102-114).

Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE, 11*(6), 6-28.

Alriksson, P., & Rantzer, A. (2007, December 12-14). *Experimental Evaluation of a Distributed Kalman Filter Algorithm.* Paper presented at the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA

Cardei, M., Thai, M. T., Li, Y., & Wu, W. (2005, March 13-17). *Energy-efficient target coverage in wireless sensor networks.* Paper presented at the INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, Florida.

Cattivelli, F. S., Lopes, C. G., & Sayed, A. H. (2008, June). *Diffusion strategies for distributed kalman filtering: Formulation and performance analysis.* Paper presented at the Proc. 2008 IAPR Workshop on Cognitive Information Processing, Santorini, Greece.

Chen, M., Gonzalez, S., & Leung, V. C. M. (2007). Applications and design issues for mobile agents in wireless sensor networks. *IEEE Wireless Communications, 14*(6), 20-26.

Chiang, C., Wu, H., Liu, W., & Gerla, M. (1997, April). *Routing In Clustered Multihop, Mobile Wireless Networks With Fading Channel.* Paper presented at the In Proc. IEEE SICON'97.

Hashemipour, H. R., Roy, S., & Laub, A. J. (1998). Decentralized structures for parallel Kalman filtering. *IEEE Transaction on Automatic Control, 33*(1), 88-94.

Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000, August). *Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks.* Paper presented at the In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00).

Kim, J.-H., West, M., Scholte, E., & Narayanan, S. (2008, June 11-13). *Multiscale consensus for decentralized estimation and its application to building systems.* Paper presented at the 2008 American Control Conference, Seattle, WA

Mutambara, G. O. (1998). *Decentralized estimation and control for multisensor systems*: CRC Press.

Olfati-Saber, R. (2007, December 12-14). *Distributed Kalman filtering for sensor networks.* Paper presented at the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA

Olfati-Saber, R., & Shamma, J. S. (2005, December 12-15). *Consensus Filters for Sensor Networks and Distributed Sensor Fusion.* Paper presented at the 44th IEEE Conference on Decision and Control, CDC-ECC'05.

Rao, B. S., & Durrant-Whyte, H. F. (1991). Fully decentralized algorithm for multisensor Kalman filtering. *IEE Proceedings-D Control Theory & Application, 138*(5), 413 - 420.

Uhlmann, J. K. (1996). General data fusion for estimates with unknown cross covariances. *Proceedings of SPIE, 2755*, 536-547.

Watfa, M. K., & Commuri, S. (2006a, August 14). *The 3-Dimensional Wireless Sensor Network Coverage Problem.* Paper presented at the 2006 IEEE International Conference on Networking, Sensing and Control. ICNSC '06, Ft. Lauderdale, FL.

Watfa, M. K., & Commuri, S. (2006b). *Optimal sensor placement for Border Perambulation.* Paper presented at the 2006 IEEE International Conference on Control Applications, Munich, Germany.

ESTIMATION AND COMPENSATION OF DEAD-ZONE INHERENT TO THE ACTUATORS OF INDUSTRIAL PROCESSES

Auciomar C. T. de Cequeira, Marcelo R. B. G. Vale, Daniel G. V. da Fonseca
Laboratory of Automation in Petroleum, Federal University of Rio Grande do Norte, Campus Universitário, Natal, Brazil
auciomar@gmail.com, marceloguerra@dca.ufrn.br, dangvf@gmail.com

Fábio M. U. de Araújo, André L. Maitelli
Department of Computing and Automation, Federal University of Rio Grande do Norte, Natal, Brazil
{meneghet, maitelli}@dca.ufrn.br

Keywords: Parameters estimation, Nonlinearity, Inverse compensation, Dead-zone, Hammerstein model.

Abstract: The oscillations present in control loops can cause damages in industry. Canceling, or even preventing such oscillations, would save up to large amount of dollars. Studies have identified that one of the causes of these oscillations are the nonlinearities present on industrial processes actuators. This paper has the objective to develop a methodology for removal of the harmful effects of nonlinearities. Will be proposed a parameters estimation method to the Hammerstein model, whose nonlinearity is represented by dead-zone. The estimated parameters will be used to construct the inverse model of compensation. A simulated level system was used as test platform. The valve that controls inflow has a dead-zone. Results analysis shows an improvement on system response.

1 INTRODUCTION

Inside industrial process there are hundreds of control loops, which are mainly composed by sensors, actuators, Programmable Logic Control (PLC) and Supervisory Control and Data Acquisition (SCADA). The control efficiency is, therefore, important to ensure a high quality product and low cost production. So, finding and solving control loop problems of a process implies in reject reduction, better product homogeneity, lower production costs and higher rates of production. Even an 1% energy or control efficiency improvement means a huge economy in industrial process, of millions of dollars (Desborough and Miller, 2002).

Several studies related to control loop performance indicate that the majority present deficient behavior, showing oscillations at process output. One of those researches (Desborough and Miller, 2002) evaluated 26 thousand control loops and classified them this way:

- 16% as excellent;
- 16% as acceptable;
- 22% as fair;
- 10% as poor;
- 36% as open loop.

Among the causes for this deficient performance are included bad tune of controllers, wrong process project, the incoming oscillatory perturbations and the nonlinearities of the actuators. And those nonlinearities cause dead-band in actuators as well.

An audit made by a big producer of valves has shown that 30% of the products presented about 4% or more of dead-band and approximately 65% of the valves had a dead-band higher than 2% (FISCHER, 2005). As most of the actions of regulatory control consist of small variations in the order of 1% or less, the control loops would not act effectively in the process for responding to these small variations. For a good performance, it is recommended that the control valve dead-band is about 1% or less (Campos and Teixeira, 2007).

A point to mention is that 20 to 30% of the oscillations in control loops are caused by nonlinearities of the valves (Ulaganathan and Rengaswamy, 2008), among which we can point out the static friction, hysteresis, backlash and dead-zone as the best known. The compensation of the effects of such nonlinearities would help in solving the problem of poor perfor-

mance of about a quarter of the controllers present in the industry.

The aim of this study is therefore to minimize or cancel the oscillations observed in the outputs of industrial processes, which are caused by dead-zone inherent to the actuators of control loops.

The industrial processes were represented by the Hammerstein model. Inverse models of nonlinearity will be built based on dead-zone parameter estimation. The intention is to make these inverse models capable to compensate the nonlinearity, reducing the oscillations and its harmful effects. It will be proposed a method of parameter estimation for a Hammerstein model that contains as the non-linear part a dead-zone.

2 MATHEMATIC MODELS

This section describes the mathematic models utilized in dead-zone estimation and compensation methodology. This methodology uses the Hammerstein model to represent the industrial processes containing dead-zone. Thereby, the linear part of Hammerstein model is represented by Output Error model and the non-linear part is represented by dead-zone. Besides the Hammerstein model, this section also describes the inverse model for dead-zone compensation. This one will reduce prejudicial effects of nonlinearity.

It should be clear that the mathematic models described in this section are simplified descriptions of real physical phenomena.

2.1 Hammerstein Model

The nonlinear Hammerstein model is composed by a static nonlinearity preceding a linear dynamic (Aguirre, 2007). This model is called block-oriented or block-structured model (Chen, 1995). Thus, both the non-linearity and the dynamics are represented by blocks, as shown in Figure 1. Here, the NL block represents the static nonlinearity function and the L block represents the linear dynamic of modeled process. The signs $u(k)$, $y(k)$ and $e(k)$ are the nonlinearity input, the output and the noise of the system, respectively. The signal $x(k)$ is called internal variable of the Hammerstein model (nonlinearity output and linear dynamic input), and, in general, it cannot be measured, making it difficult to estimate the parameters in the same models.

Although very simple, this structure may represent several actual physical processes, such as industrial processes with variable gain and control systems

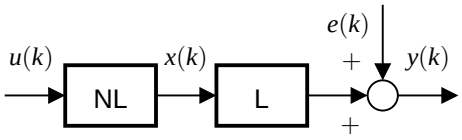


Figure 1: Hammerstein model.

with linear processes and nonlinear actuators (the latter falls within the subject matter in this work). Therefore Hammerstein models are popular in control engineering.

2.2 Output Error Model

There are some mathematical representations that are especially suitable for system identification, using classic algorithms to the estimation of its parameters. Along with the ARX and ARMAX models, the Output Error model is one of the most used structures. In this study, this model represents the linear dynamic of the Hammerstein system (block L of Figure 1) and it is represented in Figure 2. In the same model, it is assumed that the noise disturbs the output in an additive manner, as equations below.

$$y(k) = q^{-d}\frac{B(q)}{A(q)}x(k) + e(k) \tag{1}$$

$$A(q)y(k) = q^{-d}B(q)x(k) + A(q)e(k) \tag{2}$$

$A(q)$ and $B(q)$ are polynomials of order n_a and n_b, respectively, and are defined below. d represents the pure delay system and q^{-1} is the shift operator, so $x(k)q^{-d} = x(k-d)$.

$$A(q) = 1 + a_1q^{-1} + \ldots + a_nq^{-n_a}$$
$$B(q) = b_0 + b_1q^{-1} + \ldots + b_mq^{-n_b}$$

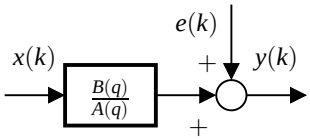


Figure 2: Output Error model.

The Output Error model is much more realistic than the ARX and ARMAX because the modeling of noise does not include the dynamics of the process $1/A(q)$ (Nelles, 2000). So, the parameter estimation task becomes more difficult. As shown in Equation (2), the noise is not white but colored due to the presence of the polynomial $A(q)$. For this reason, the least squares method cannot be used. A non-polarized algorithm should be used so that the estimation is not biased.

The Equation (2) can be rewritten in the form of summations, already introducing the delay in the input signal.

$$y(k) = \sum_{i=0}^{n_b} b_i x(k-d-i) - \sum_{j=1}^{n_a} a_j y(k-j) + \sum_{j=1}^{n_a} a_j e(k-j) + e(k) \quad (3)$$

The signals $y(k)$, $x(k)$ and $e(k)$ are the same as the Hammerstein model (Figure 1), and have been defined previously.

2.3 Dead-zone

The dead-zone is a static nonlinearity with no memory that describes the insensitivity of components for small signals. It can be seen as a static relationship between input and output signals, in which, for a range of input values, there is no answer. Once the output appears, the relationship between input and output is linear.

Figure 3 shows a graphical representation of the dead-zone, where $u(k)$ is the input and $x(k)$ is the output. The limits b_r and b_l represent the range where the output signal remains unchanged, and m_r and m_l indicate the slope of the lines. By definition $b_r > 0$, $b_l < 0$, $m_r > 0$ and $m_l > 0$, and in general, neither the limits nor the slopes are equal.

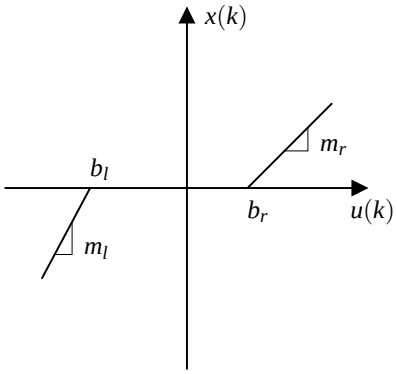


Figure 3: Dead-zone graphic.

Analytically, the dead-zone can be written as follows:

$$x(k) = \begin{cases} m_r\,[u(k)-b_r], & \text{if } u(k) \geq b_r \\ 0, & \text{if } b_l < u(k) < b_r \\ m_l\,[u(k)-b_l], & \text{if } u(k) \leq b_l \end{cases} \quad (4)$$

One way to write the behavior of the dead-zone so that it is linear in the parameters is:

$$x(k) = X_r(k) m_r\,[u(k)-b_r] + X_l(k) m_l\,[u(k)-b_l] \quad (5)$$

where $X_r(k)$ and $X_l(k)$ are auxiliary functions that take the value 0 (zero) or 1 (one) according to the following conditions:

$$X_r(k) = \begin{cases} 1, & \text{if } u(k) \geq b_r \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$X_l(k) = \begin{cases} 1, & \text{if } u(k) \leq b_l \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

2.4 Inverse Model for Dead-zone Compensation

It is known that the nonlinearities are among the key factors that limit the static and dynamic performance of control systems, preventing high precisions when using linear controllers. In order to cancel the harmful effects generated by the dead-zone, it is proposed to implement its inverse model.

The Figure 4 shows the structure used in this work for the cancellation of this nonlinearity. The inverse nonlinearity (INL block) was allocated before the nonlinearity (NL block) to cancel out its effects. When implemented with the real parameters, such compensation cancels completely the effects of dead-zone. Therefore, if the dead-zone is fully compensated, the input signal $u_c(k)$ must be equal to the signal $x(k)$.

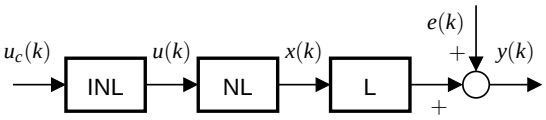


Figure 4: Block diagram of nonlinearity compensation.

The graphical relationship between the input signal $u_c(k)$ and output signal $u(k)$ is shown in Figure 5.

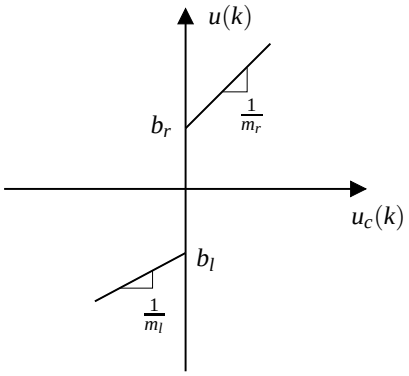


Figure 5: Graphic of dead-zone inverse compensation.

The dead-zone inverse model is represented by Equation 8. The parameters b_r, b_l, m_r and m_l are the same used in modeling of dead-zone.

$$u(k) = \begin{cases} \frac{1}{m_r}\left[u_c(k) + m_r b_r\right], & \text{if } u_c(k) > 0 \\ 0, & \text{if } u_c(k) = 0 \\ \frac{1}{m_l}\left[u_c(k) + m_l b_l\right], & \text{if } u_c(k) < 0 \end{cases} \quad (8)$$

For a linear parameterization of inverse compensation, we have:

$$\begin{aligned} u(k) = \chi_r(k)\frac{1}{m_r}\left[u_c(k) + m_r b_r\right] \\ +\chi_l(k)\frac{1}{m_l}\left[u_c(k) + m_l b_l\right] \end{aligned} \quad (9)$$

where $\chi_r(k)$ and $\chi_l(k)$ are auxiliary functions defined as:

$$\chi_r(k) = \begin{cases} 1, & \text{if } u_c(k) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$\chi_l(k) = \begin{cases} 1, & \text{if } u_c(k) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The inverse model equation is similar to the dead-zone model. The variables m_r, m_l, b_r and b_l have the same meaning of Equation (5). The difference lies in the definition of auxiliary functions $\chi_r(k)$ and $\chi_l(k)$.

To check the accuracy of the inverse model, in other words, to conclude that $x(k) = u_c(k)$, three situations will be analyzed: $u_c(k) > 0$, $u_c(k) < 0$ and $u_c(k) = 0$. For this proof, the function of the inverse of the dead-zone will be called $ZI(\cdot)$.

Lemma 1. *(Dead-zone Inverse) when implemented with real parameters m_r, m_l, b_l and b_r, the dead-zone inverse* (8) *cancels the effect of dead-zone* (4)*, that is*

$$u(k) = ZI(u_c(k)) \Rightarrow x(k) = u_c(k), \forall k \geq 0.$$

Proof. Suppose $u_c(k) > 0$. For $u_c(k) > 0$, the auxiliary function $\chi_r(k)$ (10) will be equal to 1 and $\chi_l(k)$ (11) will take value 0. Therefore, $u(k)$ (9) will be:

$$u(k) = \frac{1}{m_r}\left[u_c(k) + m_r b_r\right] = \frac{u_c(k)}{m_r} + b_r \quad (12)$$

As it was admitted that $u_c(k) > 0$, and by definition $m_r > 0$, the portion $\frac{u_c(k)}{m_r}$ is also positive. So, $u(k) > b_r$. The auxiliary function $X_r(k)$ (6) will take value 1, while $X_l(k)$ (7) will be 0. Substituting (12) in (5) with the appropriate values of the auxiliary functions we have:

$$x(k) = m_r\left(\frac{u_c(k)}{m_r} + b_r - b_r\right) \quad (13)$$

Making the simplifications, we conclude that $x(k) = u_c(k)$.

Suppose that $u_c(k) < 0$. For $u_c(k) < 0$, the auxiliary function $\chi_r(k)$ (10) will be equal to 0 and $\chi_l(k)$ (11) will take value 1. As a result, $u(k)$ (9) will be:

$$u(k) = \frac{u_c(k) + m_l b_l}{m_l} = \frac{u_c(k)}{m_l} + b_l \quad (14)$$

As it was admitted that $u_c(k) < 0$, and by definition $m_l > 0$, the portion $\frac{u_c(k)}{m_r}$ will be negative. So, $u(k) < b_l$. The auxiliary function $X_r(k)$ (6) will take value 0 while $X_l(k)$ (7) will be equal to 1. Substituting (14) in (5) with the appropriate values of the auxiliary functions we have:

$$x(k) = m_l\left(\frac{u_c(k)}{m_l} + b_l - b_l\right) \quad (15)$$

Making the simplifications, we conclude that $x(k) = u_c(k)$.

Suppose that $u_c(k) = 0$. For $u_c(k) = 0$, the auxiliary functions $\chi_r(k)$ (10) and $\chi_l(k)$ (11) are equal to 0 and the signal $u(k)$ will take value 0 too. Since, by definition, $b_r > 0$ and $b_l < 0$, the signal $u(k)$ will be $b_l < u(k) < b_r$, and according to Equation (4) the signal $x(k) = 0$. Therefore, $x(k) = u_c(k)$. □

3 PARAMETER ESTIMATION METHODOLOGY

There is a lot of work in literature regarding the identification of the Hammerstein model. Many works require that the nonlinearity is approximated by a static and continuous function, usually a polynomial. The convergence is guaranteed. However, in the case of this paper, the nonlinearity is represented by discontinuous models.

The methodology proposed here is based on (Vörös, 1997, 2003). The author developed an iterative (Vörös, 1997) and recursive (Vörös, 2003) method to estimate parameters of the Hammerstein model with discontinuous nonlinearities. He relates the problem of identification because of the impossibility of measuring the internal variable of the Hammerstein model. Instead of measuring this variable, its estimate is used based on the estimated parameters in the previous step of the recursion. There is no proof of convergence for this identification method of Hammerstein with internal variable estimation. However, it is satisfactory for most practical applications (Vörös, 2006).

There are certain situations that the least squares method is polarized or tendentious. One of these situations occur when the noise or error in the regression equation is not white, which is the case of the

Output Error models. To solve the problem of polarization, non-polarized estimators must be used, like: extended least squares, generalized least squares, instrumental variables estimator (Aguirre, 2007). The method chosen for this study was the recursive instrumental variables estimation (RIV) with forgetting factor. The equations that are utilized in this estimation method are written below (Ljung, 1987):

$$K(k+1) = \frac{P(k)z(k+1)}{\lambda + \phi^T(k+1)P(k)z(k+1)} \tag{16}$$

$$P(k+1) = \frac{1}{\lambda}\left[P(k) - K(k+1)\phi^T(k+1)P(k)\right] \tag{17}$$

$$\hat{y}(k+1) = \phi^T(k+1)\hat{\theta}(k) \tag{18}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)\left[y(k+1) - \hat{y}(k+1)\right] \tag{19}$$

where K is the estimator gain calculated from the covariance matrix P, $\hat{y}$ is the estimated value of system output y, $\hat{\theta}$ is the vector of estimated parameters, ϕ is the vector of regressors, z is the vector of instrumental variables and λ is the forgetting factor.

3.1 Equations Development

The vector of instrumental variables was chosen so that the estimated system output $\hat{y}$ and the system input u were utilized. The equation can be seen below.

$$\begin{aligned} z(k) = \Big[&\hat{y}(k-1) \ \cdots \ \hat{y}(k-n_a), \\ &u(k-d) \ \cdots \ u(k-d-n_b)\Big] \end{aligned} \tag{20}$$

Substituting Equation (5) in Equation (3) we have equation (21), which describes the total behavior of the system with its both linear and non-linear characteristics.

$$\begin{aligned} y(k) = \sum_{i=0}^{n_b} b_i \Big\{ & X_r(k-d-i)m_r\left[u(k-d-i) - b_r\right] \\ & + X_l(k-d-i)m_l\left[u(k-d-i) - b_l\right] \Big\} \\ & - \sum_{j=1}^{n_a} a_j y(k-j) + \sum_{j=1}^{n_a} a_j e(k-j) + e(k) \end{aligned} \tag{21}$$

It is observed that, if we multiply the coefficients b_i by the term in braces, there will be a number of parameters like $n_a + 4(n_b+1)$ to be estimated, besides, they are connected to each other ($b_i m_r$, $b_i m_r b_r$, for example). To avoid this large amount of parameters, the key term separation principle was used (Vörös, 1995). In this new formulation, the internal variable $b_0 x(k-d)$ is separated from the others, which generated the following equation, with the number of parameters equals to $n_a + n_b + 4$:

$$\begin{aligned} y(k) = b_0 \Big\{ & X_r(k-d)m_r\left[u(k-d) - b_r\right] \\ & + X_l(k-d)m_l\left[u(k-d) - b_l\right] \Big\} \\ & + \sum_{i=1}^{n_b} b_i x(k-d-i) - \sum_{j=1}^{n_a} a_j y(k-j) \\ & + \sum_{j=1}^{n_a} a_j e(k-j) + e(k) \end{aligned} \tag{22}$$

For Equation (22), the vector of regressors and the vector of parameters can be respectively defined such as:

$$\begin{aligned} \phi^T(k) = \Big[& -y(k-1), ..., -y(k-n_a), \\ & X_r(k-d)u(k-d), -X_r(k-d), \\ & X_l(k-d)u(k-d), -X_l(k-d), \\ & x(k-d-1), ..., x(k-d-n_b)\Big] \end{aligned} \tag{23}$$

$$\begin{aligned} \theta^T = \Big[& a_1, ..., a_{n_a}, b_0 m_r, b_0 m_r b_r, \\ & b_0 m_l, b_0 m_l b_l, b_1, ..., b_{n_b}\Big] \end{aligned} \tag{24}$$

The internal variables $x(k-d-1), ..., x(k-d-n_b)$ cannot be measured directly. Estimates of their values will be used, based on the parameters of the previous step of the recursive estimation. In other words, the estimated values of m_r , b_r, m_l and b_l will be used in Equation (5) for the construction of the regressors $x(k-d-1), ..., x(k-d-n_b)$.

The dead-zone parameters are estimated with b_0. To obtain the separated values, it is necessary to know the parameter b_0. For this, it was admitted that the plant gain is known. By the final value theorem (Nelles, 2000):

$$\frac{\sum_{i=0}^{n_b} b_i}{1 + \sum_{j=1}^{n_a} a_j} = K_p \tag{25}$$

where K_p is the plant gain. So, b_0 is:

$$b_0 = -\sum_{i=1}^{n_b} b_i + K_p\left(1 + \sum_{j=1}^{n_a} a_j\right) \tag{26}$$

We can conclude that, in order to discover the separated value of each dead-zone parameter, simply perform the following divisions:

$$\begin{aligned} m_r &= b_0 m_r / b_0 \\ b_r &= b_0 m_r b_r / b_0 m_r \\ m_l &= b_0 m_l / b_0 \\ b_l &= b_0 m_l b_l / b_0 m_l \end{aligned}$$

Although all parameters are estimated, these last four parameters are the ones used to construct the inverse model of compensation.

4 TEST PLATFORM

The testing process is a level system in which we want to control tank height and it can be seen in Figure 6. It consists of an incompressible fluid reservoir having a flow input q_{in} controlled by a pneumatic valve that has an associated nonlinearity, and a flow output q_{out} dependent on the height.

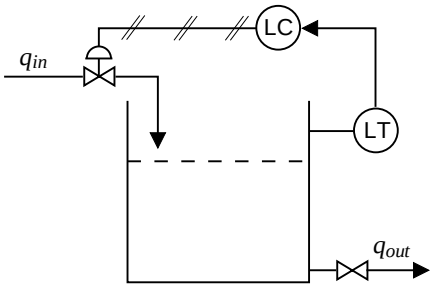


Figure 6: Level system.

The valve is a pneumatic actuator of fluid flow control and has associated dynamics (Wigren, 1993). It was assumed that it has linear opening characteristics and the model can be seen as follows:

$$G_v(s) = \frac{25}{s^2 + 5s + 25} \tag{27}$$

The reservoir contains incompressible fluid and it is classically found in literature. The model used here is a linearization of a more complex model (Ikonen and Najim, 2002), and its transfer function is:

$$G_t(s) = \frac{2}{s + 0,9} \tag{28}$$

The continuous model of the entire system, considering the transport delay of $d = 3$, is:

$$G(s) = \frac{50e^{-0,3}}{s^3 + 5,9s^2 + 29,5s + 22,5} \tag{29}$$

The discrete model of the level system using a zero-order hold and a discretization time of 0,1s is:

$$G(z) = z^{-3}\,\frac{0,00713z^2 + 0,02441z + 0,0053}{z^3 - 2,328z^2 + 1,899z - 0,5543} \tag{30}$$

5 SIMULATION AND RESULTS

References generated in the process and the measurements of tank height are expressed in percentage. As a excitation sign the PRS (pseudo random signal) was used within a range of values averaging 50% and varying uniformly from 45 to 55% being the chosen values kept constant in a minimum of 10 sampling periods. The forgetting factor was kept constant during the first 2000 sampling periods having a value of 0,995, and after this time, it changed exponentially to 1, according to Equation (31), with $\lambda_0 = 0,995$.

$$\lambda(k) = \lambda_0\lambda(k-1) + (1-\lambda_0) \tag{31}$$

The noise was considered as a white additive one, with average zero and Gaussian variance of 0, 03. The initial values of the parameter vector θ were 10^{-3} and covariance matrix P was initialized as a diagonal matrix whose elements were equal to 10^6.

In order to quantify the efficiency of controls with and without compensation, two metrics of performance evaluation were implemented (Goodhart et al., 1991). The first one considers the variance of the control signal,

$$\varepsilon_1 = \frac{\sum\left(u(k) - \frac{\sum u(k)}{N}\right)^2}{N} \tag{32}$$

and the second metric evaluates the deviation of the process output regarding the reference according to the integral absolute error (IAE),

$$\varepsilon_2 = \frac{\sum |r(k) - y(k)|}{N} \tag{33}$$

N being the number of samples.

The evaluations were divided into 3 tracks. In the first one, the reference is kept constant at a value of 50% from 1 to 60s. At the 10s instant, a -10 amplitude disturbance occurs and ceases to exist at the 40s instant. In track 2, the reference is changed to 49% at 60s, and at 80s, in the last track, it is changed to 51%.

The block diagram for the estimation process can be seen in Figure 7. Block E represents the generator of excitation signal PRS, NL block represents the dead-zone nonlinearity, blocks A and T are respectively the dynamics of the valve and tank and represent the linear part of the Hammerstein model. The RIV estimation method with the presence of the forgetting factor is represented by block M.

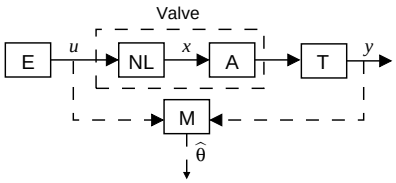


Figure 7: Block diagram of the level system estimation.

The block diagram of the compensation is shown in Figure 8. Block C represents a PI controller, which was tuned empirically so that, for the level system without the presence of nonlinearities, the plant response would behave without a large overshoot and with no regime error (less than 2%). Mathematical manipulations were made so that a control signal equals to zero would correspond to a level of 50%. Block INL represents the inverse nonlinearity, and was allocated before its respective nonlinearity. The others blocks have the same meanings described above.

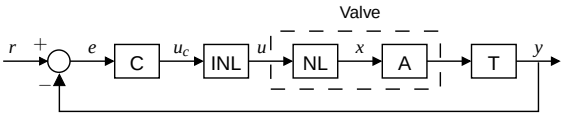


Figure 8: Block diagram of the level system with nonlinearity compensation.

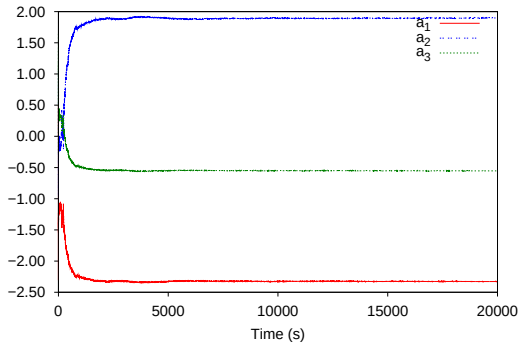
The dead-zone in the actuator of this process was built with the following parameters: $m_r = 3$, $m_l = 3$, $b_r = 1$ and $b_l = -1$. The graphics containing the parameters estimation results of linear dynamics and dead-zone can be seen in Figure 9. The parameter values obtained at the end of the recursive estimation process are shown in Table 1. The actual values of each one are also in Table 1 for comparison.

Analyzing the estimation graphics, it is observed that all the parameters have converged up to 7500s, the last ones being the coefficients of the polynomial $B(q)$. The values obtained in the estimation process are shown in Table 1, and have small errors (the biggest errors are in the order of 10^{-2}) in relation to the real values. The algorithm showed good convergence for the noise presence.

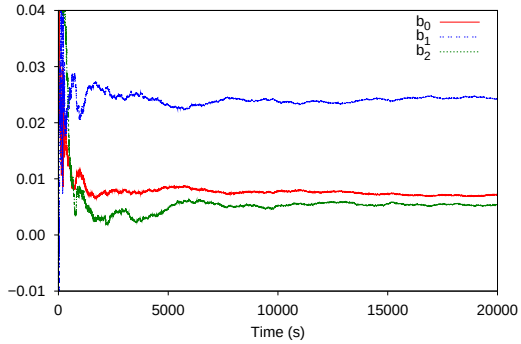
Table 1: Parameters of level system with dead-zone.

Parameters	Estimated Value	Real Value
a_1	-2.3271	-2.328
a_2	1.8968	1.899
a_3	-0.55309	-0.5543
b_0	0.00718	0.00713
b_1	0.02424	0.02441
b_2	0.00542	0.0053
m_r	3.0256	3
m_l	2.993	3
b_r	1.0214	1
b_l	-0.99561	-1

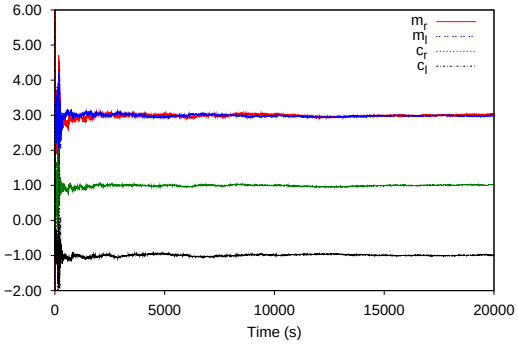
The controller was empirically tuned to $k_p = 0.5$ and $k_i = 0.7$. Figure 10 contains the graphics of plant level output for the linear case, that is, without the presence of dead-zone. Figures 11 and 12 represent,



(a) Polynomial $A(q)$.



(b) Polynomial $B(q)$.



(c) Dead-zone.

Figure 9: Parameter estimation of level system with dead-zone.

respectively, the cases of plant output with the presence of dead-zone with and without the compensator block. The control signals for these last two cases are shown in Figure 13.

For track 1, the control without compensation was rather oscillatory during the existence of the disturbance (10 to 40s) with the tank level ranging approximately from 45 to 55%. After the disturbance, the level returned to the reference and remained without oscillations. This was due to mathematical manipulations that keep the reservoir level by 50% for a valve input signal equal to zero. In tracks 2 and 3 the reference is changed respectively to 49 and 51%. In these two tracks, it is observed the existence of oscillations

maintained in the output with amplitude around the reference of $\pm 1\%$. As for the control with compensation, the behavior of the plant output is very similar to the case where there is not the dead-zone presence. Thereby, and according to Table 2, the control with compensation had better IAE indices (ε_2) for the 3 tracks compared to the control without compensation.

Table 2 also shows the index ε_1 of control signal variance evaluation for the two cases with the presence of dead-zone. Note that, in the case with compensation, the inverse nonlinearity (INL block) output was considered as control signal, and not the output of PI controller. Based on the graphic of Figure 13, the control signal with the inverse nonlinearity is more aggressive in relation to the sign of pure PI control for the dead-zone region. This is caused by the discontinuity present in the graphic of the inverse dead-zone (see Figure 5) around de zero point. Whenever the PI output inverts its sign (from positive to negative or vice versa), a jump in the output compensation occurs. Even with this discontinuity, the control with compensation had a smaller variance in its signal to the 3 tracks of the evaluation.

Table 2: Metrics for performance evaluation of nonlinear system with and without compensation.

Track	With Comp.		Without Comp.	
	ε_1	ε_2	ε_1	ε_2
1	3.0574	1.7683	2.4786	0.3696
2	0.1242	0.3722	0.0492	0.2028
3	0.1201	0.4188	0.0547	0.2439

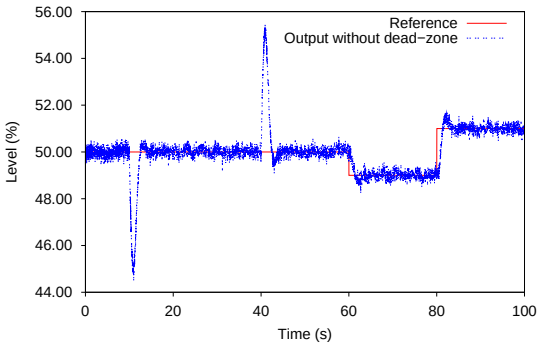


Figure 10: Plant output without dead-zone.

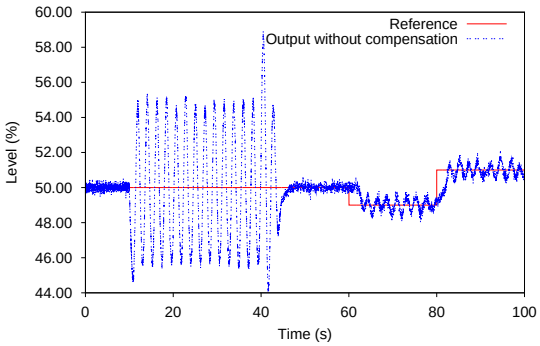


Figure 11: Plant output with dead-zone and without compensation.

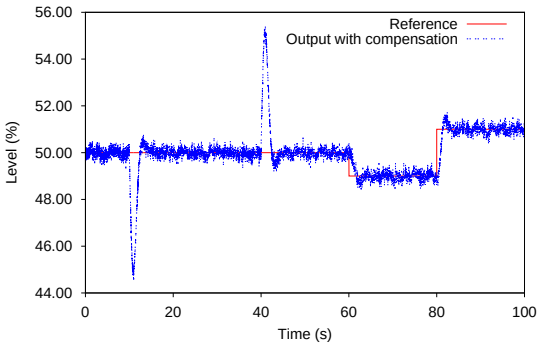


Figure 12: Plant output with dead-zone and with compensation.

6 CONCLUSIONS

In this work, it was developed a method of estimation and compensation of dead-zone that is present in the actuators of various industrial processes. It was used, as a testing process, a simulation of a level tank, which has a valve with a dead-zone to control the input flow.

First it was developed an estimation method of parameters for a Hammerstein model, in which the nonlinear part is represented by dead-zone. As linear dynamic, the Output Error model was used, which is a more complex model and the estimation task becomes more difficult when compared to the estimation of ARX and ARMAX models, because the noise is much more influential in the process. The method used the key term separation principle, reducing the number of parameters to be estimated.

In practice, the process operator defines the duration and type of measures that can be collected from the process. In fact, large variations in the excitation signal are very useful in identifying systems. However, they are not often allowed by the operators. Then, the identification should be made using normal operation data.

After being estimated, the parameters that make up the dead-zone were used to construct the inverse model of compensation. In general, the controller with compensation was more aggressive than the control without compensation during the dead-zone region. However, the plant output is much less oscillating in the compensated case. Performance metrics

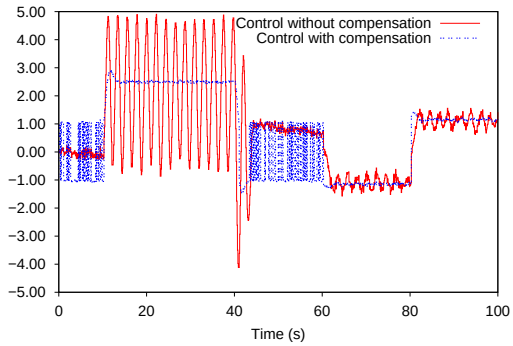


Figure 13: Control signals for the nonlinear system.

quantify the control actions and the error at the plant response. Therefore, the purpose of minimizing or even canceling the oscillations was achieved.

The estimation and compensation techniques developed here can be applied to any industrial plant that is represented according to Hammerstein model, which has the dead-zone as nonlinearity.

It is intended, as future work, to estimate and compensate the backlash nonlinearity. In addition, we intend to implement the proposal in a Programmable Logic Controller (PLC). This will bring this work to a possible application in the real world.

ACKNOWLEDGEMENTS

Especial thanks for ANP PRH-14, CNPq and Petrobras.

REFERENCES

Aguirre, L. A. (2007). *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 3 edition.

Campos, M. C. M. M. and Teixeira, H. C. G. (2007). *Controles Típicos de Equipamentos e Processos Industriais*. Editora Edgar Blücher.

Chen, H.-W. (1995). Modeling and identification of parallel nonlinear systems: structural classification and parameter estimation methods. *Proceedings of the IEEE*, 83(1):39–66.

Desborough, L. and Miller, R. (2002). Increasing customer value of industrial control performance monitoring - honeywell's experience. In *AIChE Symposium Series 2001*, number 326, pages 172–192, Arizona.

FISCHER (2005). *Control Valve Handbook*. Emerson Process Management, Iowa, USA.

Goodhart, S. G., Burnham, K. J., and James, D. J. G. (1991). A bilinear self-tuning controller for industrial heating plant. *IEEE Conference Publication*, 2(332):779–783.

Ikonen, E. and Najim, K. (2002). *Advanced Process Identification and Control*. Marcel Dekker, New York.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall.

Nelles, O. (2000). *Nonlinear System Identification*. Springer, Berlin.

Ulaganathan, N. and Rengaswamy, R. (2008). Blind identification of stiction in nonlinear process control loops. In *American Control Conference*, pages 3380–3384.

Vörös, J. (1995). Identification of nonlinear dynamic systems using extended hammerstein and wiener models. *Control Theory and Advanced Technology*, 10(4):1203–1212.

Vörös, J. (1997). Parameter identification of discontinuos hammerstein systems. *Automatica*, 33(6):1141–1146.

Vörös, J. (2003). Recursive identification of hammerstein systems with discontinuous nonlinearities containing dead-zones. *IEEE Transactions on Automic Control*, 48(12):2203–2206.

Vörös, J. (2006). Recursive identification of hammerstein systems with polynomial nonlinearities. *Journal of Electrical Engineering*, 57(1):42–46.

Wigren, T. (1993). Recursive prediction error identification using the nonlinear wiener model. *Automatica*, 29(4):1011–1025.

SHORT PAPERS

IDENTIFICATION OF DISCRETE EVENT SYSTEMS
*Implementation Issues and Model Completeness**

Matthias Roth[1,2], Lothar Litz[1] and Jean-Jacques Lesage[2]
[1]*Institute of Automatic Control, University of Kaiserslautern, Germany*
[2]*LURPA, Ecole Normal Supérieur de Cachan, France*
{*mroth, litz*}*@eit.uni-kl.de, jean-jacques.lesage@lurpa.ens-cachan.fr*

Keywords: Discrete event systems, Identification, Implementation.

Abstract: This paper presents some practical issues for the identification of discrete event systems (DES). The considered class of systems consists of a plant and a controller running is a closed-loop. Special emphasis is given to a data collection procedure using industrial controllers and its impact on the external DES-behavior of the considered systems. For models identified on the basis of observed external DES behavior using the algorithm from (Klein, 2005) it is shown that under some conditions, the identified model language simulates the *complete* original system language even if only a subset of this language is available for identification. This model characteristic is crucial for many model-based techniques like diagnosis or verification. Analyzing the observed data of a laboratory facility it is shown how it can be decided if the conditions for a *complete* model hold for an existing application.

1 INTRODUCTION

Model-based techniques play a key role in many modern control applications. For systems that can be modeled as Discrete Event Systems (DES) various approaches to improve system dependability using model-based methods have been proposed in the last two decades. Examples for these methods are diagnosis (Sampath et al., 1996) and formal verification with model checking (Machado et al., 2006). A bottleneck for the application of model-based techniques is the process of model-building which is usually expensive due to high costs for the necessary specialists. A promising way to facilitate the use of model-based methods is to offer efficient identification methods in order to decrease the cost of model-building.

First approaches for the identification of DES have been proposed in the sixties and seventies of the last century in the field of computer science (Biermann and Feldman, 1972). The identification of physical systems which is a typical interest in many engineering domains is not the aim of these works. More recent works especially on identification of Petri nets are summarized in (Fanti and Seatzu, 2008). In the last years two main directions of constructing a Petri net from samples of its language have been followed: In the first class of approaches specific rules about interdependencies of observed events are used to identify a Petri net on the basis of observed firing and marking sequences (Meda-Campana and Lopez-Mellado, 2005). The second class of approaches uses optimization techniques like integer programming to derive a Petri net structure according to given constraints (Giua and Seatzu, 2005), (Dotoli et al., 2006).

The main obstacle for the application of these methods to real world systems is their relatively high degree of abstraction. The work is usually not focused on questions like how to represent data that can be captured from a real system and how to cope with inadequacies inherent to the data collection process. In (Dotoli et al., 2006) a case study shows that there is a considerable potential of identification methods to obtain meaningful DES models of physical systems. Since the identification data base in this work has been obtained by *simulating* a three tank system, important issues of working with data captured from a *real* system have not been addressed.

In (Klein et al., 2005) an algorithm for the identification of closed-loop DES is presented. The algorithm has been designed to work with data obtained from real systems and yields a monolithic automaton. In this paper some implementation issues concerning the application of this algorithm to a real system are

*This works was partially supported by a grant from Région Île-de-France

presented. In section 2 the class of closed-loop DES is presented and it is outlined that this system class is an appropriate modeling formalism for many industrial systems. Section 3 summarizes some practical implications of the data collection procedure and defines an appropriate data format for the identification algorithm. The identification algorithm is compactly presented in section 4. An important property of models identified with this algorithm is proofed in this section: Under some well-defined conditions concerning the observed system language, the identified model is able to simulate the *complete* original system language of arbitrary length[1]. In section 5 the data collection procedure and the identification algorithm are applied to a laboratory facility in order to show the relevance of the approach for real systems. It is shown how it can be decided of if the preconditions presented in the former section hold only using measured system data.

2 CLOSED-LOOP DES

A typical configuration of industrial systems is a closed-loop of controller and plant. In the plant, a set of sensors measures certain process values and delivers them to the controller using the controller inputs. The controller executes a control algorithm and determines appropriate actuator settings for the plant. Commands to actuators in the plant are transfered via controller output signals. Figure 1 shows this principle.

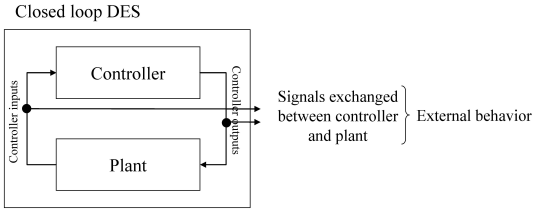


Figure 1: Closed-loop Discrete Event system.

The external behavior of such systems can be obtained by an analysis of the signals exchanged between controller and plant. In the considered class of systems these signals are binary. From an external point of view, a signal changes its value asynchronously which can be considered as the occurrence of an event. The closed-loop system can be characterized as non-deterministic since it consists of the combination of a deterministic subsystem (controller) and a non-deterministic subsystem (plant). Since the closed-loop system does not have any inputs, it must be considered as an event generator (note that the controller inputs also belong to the *external* behavior of the closed-loop DES). This system characteristic shows that the application of test functions for identification purposes like it is often done in continuous systems is not possible for industrial closed-loop DES. Only passive identification approaches working on observed system evolutions are suitable for the considered system class.

The aim of identification is to deliver a model to reproduce the external behavior of the closed-loop system. Hence, it is necessary to capture the signals exchanged between controller and plant in order to get samples of the external system behavior. In case of existing industrial facilities this process must be non-invasive to avoid disturbances of the process. In the next section, a method to capture these signals in an efficient way is presented.

3 DATA COLLECTION

3.1 Technical Implementation

The implementation of a data collection procedure for closed-loop DES necessitates capturing the signals exchanged between plant and controller. The most accurate approach to get the according signal values is to connect the wires between sensors or actuators and the controller with a special data collection hardware like described in (de Smet et al., 2001). Although such an approach is possible within a laboratory environment, for existing industrial facilities the necessary cabling effort would be too important. Since one of the main reasons to use identification methods is to save costs, the effort to apply the method must not exceed the costs of manually model building. A slightly less accurate data collection approach that can be implemented with less effort is to collect the signals after they have been captured by the controller.

Figure 2 shows the functional principle of a programmable logic controller (PLC) which is a widely used class of controllers in industry. The controller cyclically performs the steps 'input reading' where it reads the signals from the sensors, 'program execution' to determine new output values for the actuators, and 'output writing' where the newly determined commands are sent to the plant actuators. Modern PLCs are equipped with a communication processor which makes it possible to send the values of the input and output signals to a standard PC where they can be stored in a data base. The implementation of such a connection is relatively easy (it is mainly a software problem) and does not require any special hardware.

[1]Language L_A simulates language L_B if $L_A \supseteq L_B$ holds

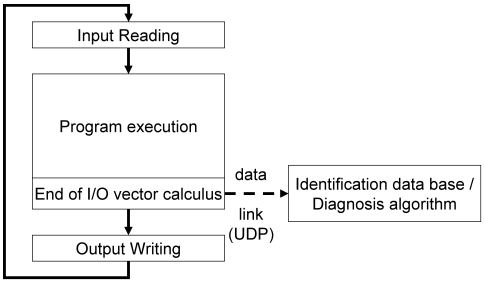


Figure 2: PLC cycle and data collection.

As implementation of the data link between controller and identification data base, a UDP (User Datagram Protocol) connection is used. At the end of the 'program execution' phase, the communication processor of the controller sends a UDP-datagram which is received by the PC with the identification data base. In order to validate that this connection is fast enough to be used for data collection, tests have been performed using a Siemens PLC (CPU 315-2 DP) equipped with a program leading to a PLC-cycle time of 25 to 30 ms. The PLC as a communication processor (CP 343-1 IT) which sends the data to a standard PC (identification data base). Figure 3 shows that the time between the reception of two packages alters between 25 ms and 30 ms according to the slightly varying PLC-cycle. It could also be observed that no data packets got lost during the transmission. This shows that the UDP connection is adapted for our purposes.

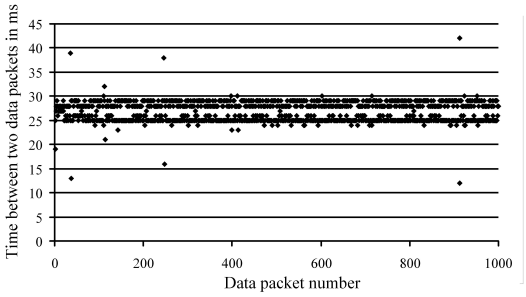


Figure 3: Validation of the UDP connection.

When the signal values are sent to the identification data base, they are grouped in the controller I/O (input/output) vector defined as follows:

Definition 1 (Controller I/O vector)**.** *Given r different controller inputs I_1, ..., I_r and s different controller outputs O_1, ..., O_s, the controller I/O vector $u = (IO_1, ..., IO_m)$ with $m = r + s$ is given by $IO_i = I_i \ \forall \ i = 1,..,r$ and $IO_{r+1} = O_i \ \forall \ i = 1,..,s$. $m = |u|$ denotes the length of the vector (number of controller I/Os).*

The controller I/O vectors which are sent to the identification data base differ slightly from the real values of the signals. Three scenarios are considered to describe how the controller I/O vector captured in the data base is affected by the data collection process at the end of the 'program execution' step in the PLC.

Figure 4 shows the evolution of two controller inputs (sensor values) in the plant. It can be seen that the two signals change their values at different times since the according sensors are not triggered simultaneously. In the middle of the figure, the PLC-cycle is shown over time. The dotted line indicates that the input values of the plant are read by the PLC during the step 'input reading'. After the program execution phase, the input values are sent to the identification data base. The evolution of the data received in the data base (the *sampled* data) is shown below the PLC cycle. Since the real values of the two inputs have been captured *simultaneously* during the 'input reading' phase they also change their value simultaneously in the identification data base. As a consequence for the identification algorithm it is important to use an appropriate definition of the notion *event*. Since in DES theory events cannot occur simultaneously it is not possible to define the change in value of *one* signal as an event like it would probably be the most intuitive way. Instead we use the following definition:

Definition 2 (Event)**.** *The appearance of an event leads to a new I/O vector $u(j)$ with $u(j) \neq u(j-1)$. Only I/O vectors generated by events are stored in the data base.*

As a consequence of the this definition, two successive I/O vectors $u(j)$ and $u(j+1)$ always differ in at least one (but possibly more than one) I/O value.

In figure 5 a scenario with a controller input and a controller output is shown. In the example it is assumed that there is a logical condition in the control algorithm relating these two signals: if the input changes its value, the controller changes the value of the output as a consequence. It can be seen that the cause (change in value of the input) and the according effect (change in value of the output) are sent simultaneously to the data base. Hence, in the data base cause and effect cannot not directly be seen. Additionally, the figure shows that using the described data collection procedure it is possible to receive I/O vectors in the data base before the according output values are valid for the plant. The output values are sent to the plant during the step 'output writing' which takes place after the transfer of the I/O vector.

The third scenario in figure 6 shows the case of an actuator influencing a sensor in the plant. When the according output is set in the controller it is transfered to the data base and with a short delay written to the the plant. The actuator controlled by the output can only then start influencing the sensor connected with

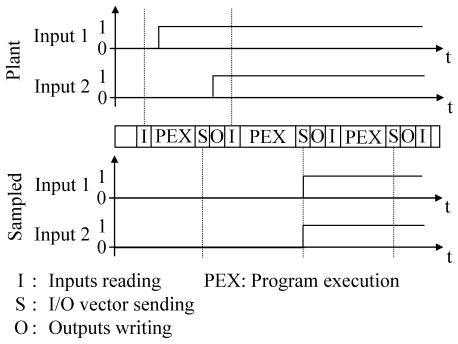


Figure 4: Sampling scenario 1.

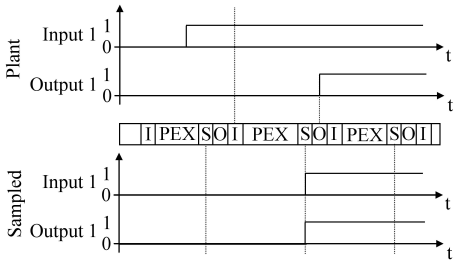


Figure 5: Sampling scenario 2.

input 1. Hence, in this case cause and effect cannot be captured simultaneously since the change in value of input 1 occurs some time after the vector with the change in value of output 1 has been sent to the identification data base. The delay of cause and effect is often not considered in manually built models like in (Sampath et al., 1996). A second issue of the data collection process can also be seen in the figure: although input 1 and output 1 change their value relatively fast one after the other (faster than the duration of a PLC-cycle), it can take up to one PLC cycle until this change in value is sent to the identification data base.

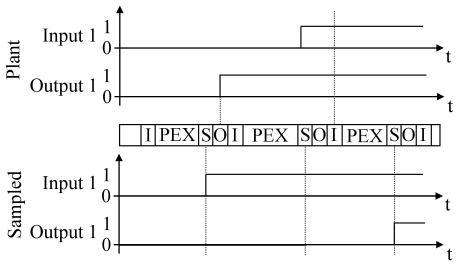


Figure 6: Sampling scenario 3.

The three scenarios show that the selected data collection procedure introduces some inadequacies that will also be part of the identified model. It is thus important to clearly indicate when the data has been captured in the PLC-cycle to precisely describe the data in the identification data base.

3.2 Definition of the Observed Language

For identification, the captured system data can be interpreted as the language of the considered closed-loop DES. The identification is based on the observation of I/O vector sequences during l_h different system evolutions:

Definition 3 (I/O vector sequence)**.** *If during the h-th system evolution l_h I/O vectors u_h have been observed, the sequence is denoted as* $\sigma(h) = (u_h(1), u_h(2), \ldots, u_h(l_h))$.

The term 'system evolution' refers to a system run of a certain length. In manufacturing systems such an evolution can be a production cycle. Based on the I/O vector sequences it is possible to define the observed word set (I/O vector sequences of a given length) and the observed language:

Definition 4 (Observed word set and language)**.** *The observed words of length q captured during p different system evolutions are denoted as*

$$W^q_{Obs} = \bigcup_{i=1}^{p} \Big(\bigcup_{j=1}^{l_i-q+1} (u_i(j), u_i(j+1), \ldots, u_i(j+q-1)) \Big).$$

With the observed word set we can define the observed language of length n of the system starting from any reachable state as

$$L^n_{Obs} = \bigcup_{i=1}^{n} W^i_{Obs}$$

In most practical applications the *observed* system language is only a subset of the *possible* system language L^n_{Orig}. The longer a closed loop system is observed, the more likely the cardinality of L^n_{Obs} converges to a certain value. If new system evolutions do not lead to new words in L^n_{Obs}, the system language L^n_{Orig} can reasonably be considered as completely observed ($L^n_{Obs} \approx L^n_{Orig}$). Figure 7 shows typical evolutions of the observed language in case of convergence and in case of continued growth. In practical application it is often the case that L^n_{Obs} converges for smaller values of n but still grows for larger values. As an example consider the case when each possible single system output has been observed (L^1_{Obs} converges) but there still occur new combinations (sequences) of already known single system outputs ($L^{n>1}_{Obs}$ continues to grow).

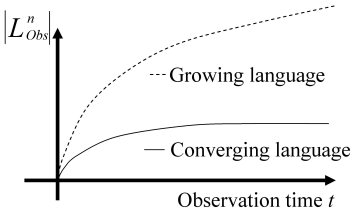


Figure 7: Principle of a converging language.

4 IDENTIFICATION

The aim of identification is to build a model that approximates the original system language L^n_{Orig}. In (Klein, 2005) a non-deterministic autonomous automaton with output is chosen as an appropriate model to reproduce the observed language of closed-loop discrete event systems:

Definition 5 (Non Deterministic Autonomous Automaton with Output)**.** *NDAAO* $= (X, \Omega, r, \lambda, x_0)$ *with X finite set of states,* Ω *output alphabet,* $r : X \rightarrow 2^X$ *non-deterministic transition relation,* $\lambda : X \rightarrow \Omega$ *output function and* x_0 *the initial state.*

If the output alphabet Ω consists of the captured I/O vectors it is possible to approximate the observed language L^n_{Obs} by performing state trajectories in the automaton:

Definition 6 (Words and Language of the NDAAO)**.** *The set of words of length n generated from a state* $x(i)$ *is defined as:*

$$W^{n=1}_{x(i)} = \{w \in \Omega^1 : w = \lambda(x(i))\}$$

and

$$W^{n>1}_{x(i)} = \{w \in \Omega^n : w = (\lambda(x(i)), \lambda(x(i+1)), \ldots, \lambda(x(i+n-1))) : x(j+1) \in r(x(j)) \; \forall i \leq j \leq i+n-2\}$$

The language of length n generated by the NDAAO is given by

$$L^n_{Ident} = \bigcup_{i=1}^{n} \bigcup_{x \in X} W^i_x$$

In (Klein, 2005) an algorithm to identify an NDAAO based on an observed language is given. The algorithm delivers a model that is $k+1$-complete which means that $L^{k+1}_{Ident} = L^{k+1}_{Obs}$ (proof can be found in (Klein, 2005)). This property excludes that the identified model contains any non-observed word of length $k+1$. This makes the model suitable for fault detection purposes (Roth et al., 2009). It is also the basis for the completeness of the model which will be shown at the end of the section. The algorithm uses words of the parametric length k to construct the NDAAO. The observed I/O vector sequences in the data base have to be modified according to the following equation. It duplicates the first vector of each sequence $k-1$ times:

$$\sigma^k_h(i) = \begin{cases} \sigma_h(1) \text{ for } 1 \leq i \leq k \\ \sigma_h(i-k+1) \text{ for } k < i \leq k + |\sigma_h| - 1 \end{cases} \quad (1)$$

On the basis of $\Sigma^k = \{\sigma^k_1, \ldots, \sigma^k_p\}$ we determine the observed word sets of length k and $k+1$ for the identification:

$$W^k_{Obs,\Sigma^k} = \bigcup_{\sigma^k_h \in \Sigma^k} \left(\bigcup_{i=1}^{|\sigma^k_h|-k+1} (u_h(i), u_h(i+1), \ldots, u_h(i+k-1)) \right) \quad (2)$$

$$W^{k+1}_{Obs,\Sigma^k} = \bigcup_{\sigma^k_h \in \Sigma^k} \left(\bigcup_{i=1}^{|\sigma^k_h|-k} (u_h(i), u_h(i+1), \ldots, u_h(i+k)) \right)$$

The identification procedure is given in algorithm 1. It is a condensed version of the algorithm given in (Klein, 2005). For the algorithm, we define an operator $w[a..b]$ to deliver the substring from position a to position b in word w. In the first step, for each word of length k a state is created. If two words of length k have been observed successively, they build a word of length $k+1$. Hence, the states representing the two words of length k are connected in step 2 of the algorithm. In step 3, the output function of each state is redefined. The new state output is the I/O vector at the end of the word of length k representing the state output so far. In the last step, states with equal output and equal following states are merged. A detailed description of this procedure is given in (Klein, 2005).

An interesting characteristic of a model identified with algorithm 1 is that under certain conditions the identified language does not only reproduce the system behavior observed so far ($L^n_{Ident} \supseteq L^n_{Obs}$) but the *complete* original system language ($L^n_{Ident} \supseteq L^n_{Orig}$). This property (which is not proved in (Klein, 2005)) is very important for many model-based techniques relying on a complete system description. In the following, the necessary conditions to proof the characteristic are given. The first step is to show that each state trajectory producing a word of length k ends in the same state. In the following, w^k denotes a word of length k.

Lemma 1. *In an NDAAO identified with parameter k each state trajectory* $\lambda(x(i), \ldots, x(i+k-1)) = w^k \in L^k_{Ident} | x(j+1) \in r(x(j)) \forall i \leq j < i+k-1$ *ends in the same state.*

Algorithm 1: Identification algorithm.

Require: Parameter k, observed word sets W^k_{Obs,Σ^k} and W^{k+1}_{Obs,Σ^k}

1: $X = \{x | \forall w \in W^k_{Obs,\Sigma^k} : \exists! x | \lambda(x) := w, r(x) := \{\}\}$
2: $\forall (x,x',w) \in X \times X \times W^{k+1}_{Obs,\Sigma^k} | \lambda(x) = w[1,\ldots,k] \wedge \lambda(x') = w[2,..,k+1] : \quad r(x) := r(x) \cup x'$
3: $x_0 = x \in X | \lambda(x) = w^k$ and $w^k[i] = \sigma_1(1) \forall 1 \leq i \leq k$
4: $\forall x \in X : \lambda(x) := \lambda(x)[|\lambda(x)|]$
5: Merge $x_1, x_2 \in X$ with $\lambda(x_1) = \lambda(x_2)$ and $r(x_1) = r(x_2)$

We define a function $\widetilde{\lambda}(x)$ delivering w^k used in step 1 for each state of the algorithm. It represents the state output before it has been replaced by the I/O vector at the end of the word of length k representing the state output until step 3.

Proof of lemma 1. From equations 1 and 2 it follows that $\forall w^k \in L^k_{Obs} \exists v^k_1, \ldots, v^k_k \in W^k_{Obs,\Sigma^k} | v^k_1[k] = w^k[1], v^k_2[k-1 \ldots k] = w^k[1 \ldots 2], \ldots, v^k_k[1 \ldots k] = w^k[1 \ldots k]$. From steps 1 and 2 of algorithm 1 it follows that states representing $v^k_1, \ldots, v^k_k$ are connected since $\forall v^k_1, v^k_2 \exists w^{k+1} \in W^{k+1}_{Obs,\Sigma^k} | w^{k+1} = v^k_1[1 \ldots k] v^k_2[k]$. Since step 1 assures that $\forall w^k \in L^k_{Ident} \exists! x | \widetilde{\lambda}(x) = w^k$ it follows that each state trajectory with $\lambda(x(i), \ldots, x(i+k-1)) = w^k \in L^k_{Ident} | x(j+1) \in r(x(j)) \forall i \leq j < i+k-1$ ends in the same state. □

In the next theorem, it is stated that the identified language simulates the original system language of arbitrary length if $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ holds for a given value of the identification parameter k.

Theorem 1. *If $L^{k+1}_{Orig} = L^{k+1}_{Obs}$, then $L^{k+n}_{Ident} \supseteq L^{k+n}_{Orig}$ for an NDAAO identified with parameter k.*

Proof of theorem 1. $L^{k+1}_{Ident} \supseteq L^{k+1}_{Orig}$ since the identified NDAAO is $k+1$ complete. For $k+2$ it holds: $\forall w^{k+2} \in L^{k+2}_{Orig} \exists a^k b^1 c^1 = d^1 e^1 f^k = s^1 u^k v^1 = w^{k+2} | a^k b^1, e^1 f^k, u^k v^1 \in L^{k+1}_{Orig} = L^{k+1}_{Obs}$. Each state trajectory producing a^k ends in the same state x_1 (lemma 1). In step 2 of the algorithm, this state is connected with $x_2 | \widetilde{\lambda}(x_2) = u^k$. Each state trajectory producing u^k ends in the same state x_2 which gets connected to $x_3 | \widetilde{\lambda}(x_3) = f^k$. Since there is a trajectory leading to state x_1 and x_1, x_2 and x_3 are in one trajectory, it follows that $\forall w^{k+2} \in L^{k+2}_{Orig}$ there exists a trajectory of states producing this word. For larger values than $k+2$ $\forall w^{k+n} \in L^{k+n}_{Orig}$ there is always an appropriate decomposition into already observed substrings of $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ to find a trajectory of connected states like presented above. Hence, it follows that $L^{k+n}_{Ident} \supseteq L^{k+n}_{Orig}$ if $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ holds. □

Theorem 1 shows that it is crucial to state $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ for a precise k in order to deliver a model which is able to simulate the complete original system behavior. In section 3.2 it is shown how it can be decided if $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ holds for a given value of k.

5 APPLICATION

In order to show that the identification algorithm of section 4 is capable of delivering a model of existing systems that can be interpreted as closed-loop DES, one of the case studies we have treated will be presented in this section. The system depicted in figure 8 has 30 digital I/Os and is controlled by a Siemens PLC equipped with a communication processor. The system treats work pieces stored in the left most part of the facility. Each work piece is successively treated by the three tools. For identification a system evolution is defined as the run of two work pieces through the machine. Hence, there are at most two work pieces treated concurrently in the whole system.

Figure 8: Laboratory facility.

50 system evolutions (treating 50 times two work pieces) have been performed and the according data has been collected using the procedure described in section 3. The observed word sets of different length are shown in figure 9. It can be seen that for small values of n like $n = 2$ or $n = 3$ the according observed word set and thus the observed language converges to a stable level which implies that the observed language converges to the original system language ($L^{n=3}_{Obs} \approx L^{n=3}_{Orig}$). Although for L^3_{Obs} there is a new word observed in one of the last evolutions, it can still reasonably be considered as completely observed. Hence, the precondition of theorem 1 is fulfilled. Since $L^{n=3}_{Obs} \approx L^{n=3}_{Orig}$, it is possible to identify an NDAAO with $k+1 = n = 3$ ($\rightarrow k = 2$) to simulate the original system behavior.

For the identification of an NDAAO a software tool has been developed. Like depicted in figure 10, it takes the data base consisting of the observed system evolutions and the identification parameter k as input and applies algorithm 1. The software allows an analysis of the model structure (number of states, number of transitions etc.) and a behavioral analysis (see below). The identified model can be exported to

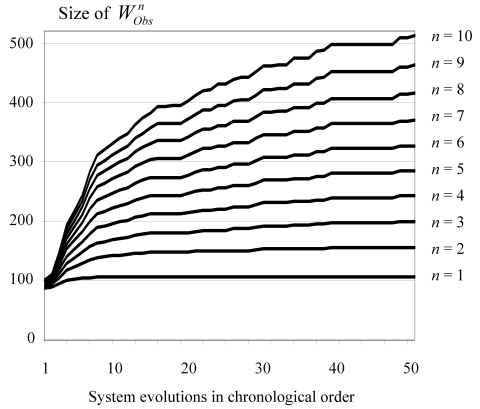


Figure 9: Observed word set for the laboratory system.

an XML-file to use it in other tools such as diagnosis software (Roth et al., 2009). To visualize the resulting automaton, an interface to GRAPHVIZ is provided. The identification of the NDAAO with $k = 2$ on the basis of 50 system evolutions took 170 ms on a standard PC with a 1.79 Ghz CPU and 1.96 GB RAM. The identified model has 121 states and 162 transitions.

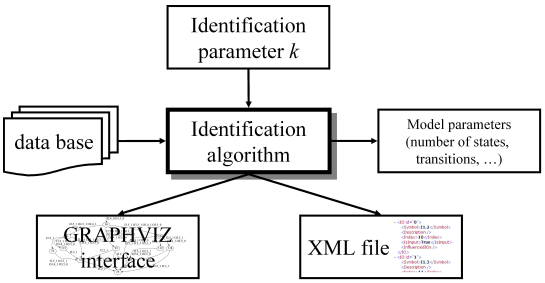


Figure 10: Features of the implemented software.

A part of the identified NDAAO can be seen in figure 11. Due to space limitations, instead of giving the complete I/O vector in each state output, only I/Os changing their value from one state to another are given (e.g. $I2.4\_1$ $O2.5\_1$ to indicate a rising edge ($\_1$) of controller *I*nput $I2.4$ and a rising edge of controller *O*utput $O2.5$ from state 48 to state 49). Several examples for the different I/O vector sampling scenarios presented in section 3.1 can be seen in the model. For the scenario with two controller inputs changing their value synchronously due to the data collection process (figure 4), the transition between states 93 and 54 is an example. Both inputs $I2.2$ and $I3.2$ change their value when taking this transition. The transition from state 48 to 49 is an example for the second sampling scenario (figure 5) where the change in value of an input triggers the change in value of an output. This transition represents the situation when the second work arrives at the entrance of the second station (position sensor connected to $I2.4$ changes its value) and the conveyor of this station is started (controller output $O2.5$ is set to 1). Both changes in value appear at one transition due to the data collection process as explained in section 3.1. An example for the third scenario form section 3.1 (figure 6) can be seen in the state trajectory $x_{48} \rightarrow x_{49} \rightarrow x_{50}$. From state 48 to state 49 the conveyor is started ($O2.5\_1$) to transport the work piece away from the entrance position ($I2.5$). Since it takes at least until the next PLC cycle to transport the work piece away from the sensor ($I2.5\_0$ for the falling edge) cause and effect cannot be seen at once but at some successive transitions. The non-deterministic nature of the identified NDAAO can also be seen in figure 11: there are several ways to go from state 50 to state 55. Being in state 50 the choice of the trajectory is not determined but taken at random like in the closed-loop DES where unpredictable physical conditions in the plant lead to non-deterministic behavior.

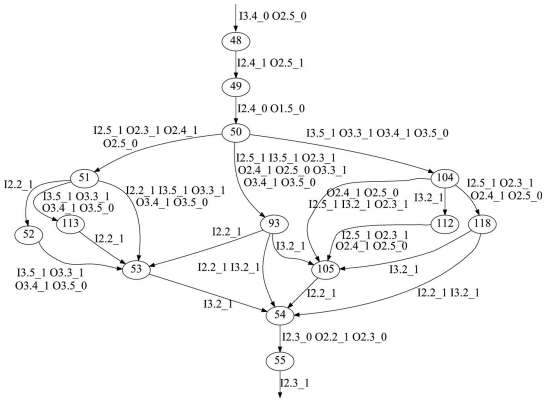


Figure 11: Part of the identified NDAAO.

One of the important characteristics of the identified model is its capability to simulate the *complete* original system language (theorem 1). Even if only a subset of possible I/O vector sequences connecting the closed-loop system states represented by NDAAO states 50 and 55 has been observed, the model contains each possible trajectory that can occur when going from one state to the other as long as no new word of length $k+1$ is produced. If $L^{k+1}_{Orig} = L^{k+1}_{Obs}$, the model is capable of exhibiting words of length $k+n$ although these words have not been seen before. This capability comes at cost of also exhibiting words that have not and probably will not be observed ($w^{k+n} \notin L^{k+n}_{Orig}$). To get some information of the amount of words which are created without having been observed, the coefficient

$$C^n_B = \frac{|L^n_{Ident}|}{|L^n_{Obs}|}$$

is a useful indicator. The presentation of the coeffi-

cient in figure 12 describes the relation between the identified and the observed language of an automaton identified with $k = 2$. For $n = k+1$, the model strictly creates the observed language L_{Obs}^{k+1}. It can be seen that for larger values of n the automaton generates a larger language than L_{Obs}^{n}. A certain part of the additionally created words is probably not part of the original system language which may lead to a need for specific precautions in some model based techniques like diagnosis. However, from theorem 1 it is clear that each word with length $n \geq k+1$ of the original language not observed so far is part of the identified language. In the case of model based diagnosis for example, this allows stating that there will be no false alerts using the identified automaton as fault-free reference model (Roth et al., 2009).

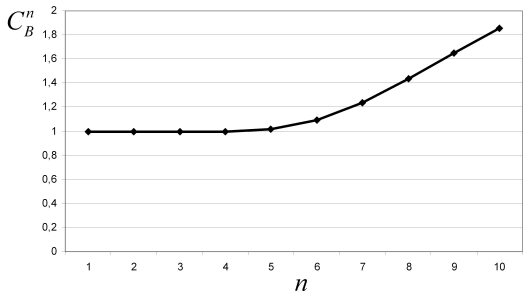


Figure 12: Coefficient of identified and observed language.

6 CONCLUSIONS

In this paper practical implications of identification of closed-loop discrete event systems have been addressed. It has been shown how the necessary data can be obtained in the case of industrial closed-loop systems. For many model-based techniques it is crucial to have a model of the complete system behavior. For the identification algorithm of (Klein, 2005) it has been proved that the identified automaton is simulates the original system language of arbitrary length if some conditions concerning the observed system language hold.

REFERENCES

Biermann, A. and Feldman, J. (1972). On the synthesis of finite-state machines from the sample of their behavior. *IEEE transactions on computers*, 21:592–597.

de Smet, O., Denis, B., Lesage, J.-J., and Roussel, J.-M. (2001). Dispositif et procd d'analyse de performances et d'identification comportementale d'un systme industriel en tant qu'automate vnements discrets et finis. Technical report, French Patent 01 110 933.

Dotoli, M., Fanti, M. P., and Mangini, A. M. (2006). Online identification of discrete event systems: a case study. In *2006 IEEE international conference on automation science and engineering*, pages 405–410.

Fanti, M. P. and Seatzu, C. (2008). Fault diagnosis and identification of discrete event systems using petri nets. In *Proceedings of the 9th International Workshop on Discrete Event Systems, Gtebor, Sweden*, pages 432–435.

Giua, A. and Seatzu, C. (2005). Identification of free-labeled petri nets via integer programming. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005*, pages 7639–7644.

Klein, S. (2005). *Identification of Discrete Event Systems for Fault Detection Purposes*. Shaker Verlag.

Klein, S., Litz, L., and Lesage, J.-J. (2005). Fault detection of discrete event systems using an identification approach. In *Proceedings of the 16th IFAC World Congress*, pages CDROM paper n02643, 6 pages.

Machado, J., Denis, B., and Lesage, J. J. (2006). A generic approach to build plant models for DES verification purposes. In *Proceedings of the 8th international workshop on discrete event systems*, pages 407–412.

Meda-Campana, M. and Lopez-Mellado (2005). Identification of concurrent discrete event systems using petri nets. *2005 IMACS: Mathematical Computer, Modelling and Simulation Conference.*

Roth, M., Lesage, J.-J., and Litz, L. (2009). An FDI method for manufacturing systems based on an identified model. In *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'09*, pages 1389 – 1394, Moscow, Russia. IFAC.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1996). Failure diagnosis using discrete-event models. *IEEE transactions on control systems technology*, 4(2):105–124.

AN INVERSE SENSOR MODEL FOR EARTHQUAKE DETECTION USING MOBILE DEVICES

Thomas Collins and John P. T. Moore
Thames Valley University, London, U.K.
{*Thomas.Collins, moorejo*}*@tvu.ac.uk*

Keywords: Environmental monitoring and control, Earthquake detection, Nonlinear signals and systems.

Abstract: We describe a sensory framework to be used for the purposes of earthquake detection using minimal cost, accelerometer equipped, hardware units. Combining techniques from mobile robotics this model is intended to address the current issue in the field whereby high fidelity hardware units tuned to detect specific characteristics such as wave features and/or high fidelity event models derived from data analysis are required for such detection. In this paper we present and contextualise the architecture under construction in addition to outlining the salient elements of the problem we are addressing.

1 INTRODUCTION

At the onset of an earthquake, dedicated detection systems are capable of issuing alerts thereby providing valuable time for people further from the event epicentre to take action to protect themselves. This is possible as the means employed to facilitate the transmission of alert information is generally faster than the speed of seismic waves. However this capability obviously relies on the existence of a monitoring network. Within the field of earthquake engineering two broad approaches have emerged toward the creation of such networks. One approach consists of creating banks of accurate units capable of detecting seismic characteristics because they have been tuned to detect specific characteristics such as fluctuating ambient seismic noise level. Countries such as Japan have successfully developed early warning systems based on such techniques. Unfortunately it is not always possible to deploy such technology in earthquake prone areas around the world for varying degrees of political, technical and financial considerations. Where such situations manifest themselves the use of commodity hardware as the foundation of a seismic detection network is a viable alternative. This highlights the second domain approach which consists of employing the distributed computing paradigm where the individual physical nodes are typically Laptop or Desktop computers equipped with sensors such as accelerators. For example the Network for Earthquake Engineering Simulation Cyberinfrastructure Centre (NEESit) has utilised the accelerometers in Apple Macintosh laptops to develop an educational and research platform for measurement and recording of vibrations and dynamic responses. Likewise the Quake-Catcher Network (QCN)2 links existing laptop and desktop computers with the aim of forming a large earthquake monitoring system.

While the availability of Laptop/Desktop based systems does remove a number of obstacles barring the realisation of detection systems the installations in themselves suffer from a number of manifest problems. For example the existence of networking infrastructure capable of linking the individual machines, the potential availability of technically skilled people (or appropriate training programmes) to operate the machines, the ability to quickly disseminate alerts is a required component of any such system.

Our aim is to directly address this problem by supplementing traditional Laptop/Desktop based approaches using mobile phone technology. The core design goal is to develop a portable system which is capable of running on constrained and resource limited hardware thereby allowing earthquake detection in sparsely seismically-instrumented regions. Therefore a key facet of the system is the development of a signal processing model which does not have the distributed quantitative analysis requirements of the Laptop/Desktop techniques mentioned above and is capable of operation in the context of low entropy sensory information. In this paper we present such a model, illustrating its core features and operational characteristics, and presenting initial results illustrating the competencies of the model and finally highlight areas

of future work.

2 SEISMIC EVENT SIGNAL DETECTION/HANDLING

There are two broad approaches to the detection and/or handling of seismic event signals. One approach consists of creating dedicated accurate sensory units tuned to detect specific signal characteristics therefore making the units capable of detecting emergent seismic characteristics e.g. fluctuating ambient seismic noise level. The other approach consists of employing knowledge based reasoning and established signal processing techniques to derive signal processing techniques formed from the feature analysis of historical data.

Hardware based detectors use signal averaging techniques in an attempt to achieve an optimum signal to noise ratio which is capable of determining a true seismic event from a false positive. The ability of such sensors is directly related to the noise model that has been pre-determined and incorporated into the units. Such noise models are generally formed through traditional signal manipulation techniques i.e. the statistical analysis of an appropriate domain characteristic function. However to be useful in a practical sense this noise model must be determined for every new installation of such the units and limits the detection threshold thereby reducing the overall effectiveness of such units (Newmark and Rosenblueth,).

The direct application of Machine Learning and/or statistical techniques, typically realised in software, in the form of knowledge-based reasoning is an alternate approach which provides for a level of flexibility in the detecting of seismic activities. Such systems are exemplified in (Hewitt, 1992; Zareian and Krawinkler, 2009). In this case the detection threshold associated with the system is not directly dependent on a physical characteristic such as seismic noise level. Rather relevant characteristics are determined through the analysis of domain expertise in the form of historical data and knowledge acquired from human experts. The Laptop/Desktop based systems outlined previously typically employ such techniques. The gathered information is employed to construct an operational model which is used to evaluate the sensory information received from the Laptop/Desktop sensor(s). The overall success of such approaches however is largely dependent on characteristics such as the selection of appropriate domain classifications and refinement/training of the derived model.

Within the context of the domain we are addressing the realisation of an efficient and expressive signal processing mechanism is paramount to the overall performance of detection system. Unfortunately neither of the existing techniques outlined above are directly usable for what we need to achieve. Techniques associated with tuned hardware units are not usable because the hardware units we are concerned with are standard mobile phone handsets meaning that the modification of same would require specific technical expertise and the availability of specialised hardware which is not a feasible goal for the intended deployment locations. In addition the application of existing 'knowledge based' techniques is not directly possible because of the data requirements both in terms of constructing an initial model and subsequent data propagation throughout the network.

From an operational perspective, any signal processing technique must consider real time operation as being paramount. In addition there should be no requirement for historical knowledge. However any such information, if available, should be easily incorporated into the model developed using the processing technique. Finally the technique must accommodate low entropy sensory information.

In evaluating these requirements techniques from a number of varied domains such as speech recognition e.g. (Vargas et al., 2001), and telecoms e.g. (Murooka et al., 2001) and mobile robotics e.g. (Ehlers et al.,) were considered. After domain evaluation we determined that the problem that is closest to the problem we are addressing in developing our signal handling model is the field of Occupancy Grid based robotic mapping.

3 MOBILE ROBOTIC MAPPING

Within the field of mobile robotics a key concern is providing the robot with the ability to acquire a model of its operating environment as this model is required for the safe and productive operation of the robot. The actual performance of the robot in acquiring a meaningful spatial model of its operating environment depends greatly on its capability to quickly evaluate the potentially erroneous information received from its sensors. As it operates in the environment, the robot gathers sensory information and subsequently incorporates this into a representation of the environment. Occupancy Grids have become the dominant paradigm for environmental modelling in mobile robotics because of their operational characteristics (Kortenkamp et al., 1998) . The creation of these Occupancy Grid maps is a non trivial process as the robot has to interpret the findings of its sensors in order to make deductions regarding the state of its en-

vironment. This is facilitated by the use of a sensor model which is a means of interpreting received signals through perceptual channels. In occupancy grid based robotic mapping there are largely two types of sensory model; the Inverse Model and the Forward Model(Collins et al., 2007). In the context of our requirements the inverse approach is currently most applicable. This is because it facilitates iterative real time operation without any requirement for historical knowledge and facilitates operation with low entropy sensory information.

4 PHYSICAL ARCHITECTURE

A key design goal is to produce a portable system which is able to run on constrained hardware. Although the target device for the prototype is a mobile phone it is envisaged that the software could run on other more limited embedded devices. The overall design of the prototype can be split into specific problem domains involving obtaining the data, communication of the data, encoding of the data and processing of the data on the device itself.

4.1 Obtaining the Data

While movement will be detected by the accelerometers contained within the device, an initial decision is how often to sample this movement and how many samples are needed before we process the data. Once we have accumulated sufficient samples this data needs to be analysed to decide whether or not we think an adequate amount of shaking or movement is taking place. Studies of accelerometer data include calculating and using the covariance of the values obtained (Ravi et al., 2005). For the prototype currently in development we take the covariance of our X,Y sample data using equation 1.

$$covar = (1/(n-1))\sum_{i=1}^{n}(x_i - \hat{x})(y_i - \hat{y}) \qquad (1)$$

We then compare this *covar* result with a predetermined threshold value. If it exceeds this threshold we must then communicate our findings to other clients.

4.2 Communicating the Data

The first challenge to overcome is deciding how to broadcast or share data between multiple connected devices in a scalable way. The Spread Toolkit offers an open source solution based on a shared message bus. It has been optimised to provide efficient message exchange with the ability to guarantee delivery

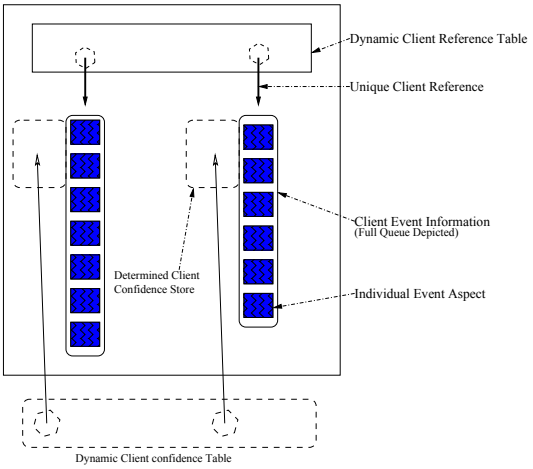


Figure 1: Device architecture.

and ordering of messages if required. To improve performance we will use unreliable communication. Each mobile client connects to an individual Spread daemon using a uniquely generated id. In addition, Spread daemons can be connected together to form a larger shared single communication bus where Spread daemon 1 connects to Spread daemon n.

4.3 Processing the Data on the Device

All messages received by a single client will be queued for a specific period of time. Thus, the number of independent queues created reflects the number of unique clients who have transmitted messages within the sampling time period. Processing the queues involves examining the number of queues at time t as well examining their queue length. If the number of queues is below a certain threshold or the mean queue length is below a certain threshold we can reset all queues and wait for the next sampling period before repeating the process. Otherwise, we need to process the data in the queues. For the prototype each queue contains data representing a covariance value v obtained from the accelerometer data. Each queue will have an independent scalar value representing a confidence level k. Applying $k(v_1, v_2, \ldots, v_n)$ yields $(kv_1, kv_2, \ldots, kv_n)$ for each queue. Summation of these queue vectors will provide a simplistic overview of whether or not we suspect an earthquake is taking place. Each time the queues are processed they are cleared ready for the next sample. A key challenge will be deriving an accurate confidence scalar value for each queue. This will ultimately need to take into account historical data between sampling cycles.

All software used or written needs to be portable and able to run on ARM and MIPS based hardware. The software must also operate within the constraints

of the target device. The hardware used for the prototype is the Openmoko Freerunner[1]. Significant features of the device include its accelerometers, WiFi and GPRS. The ability to test over a GPRS connection will be important as it may not be possible to access a wireless access point or there may not be a 3G network available. Therefore being able to concisely encode data for communication across Spread will be essential. Regardless of the connection we also want network communication to be light-weight in terms of processing load. This eliminates standard approaches such as structuring packets with XML data (Moore, 2007).

5 SEISMIC DETECTION MODEL

From an operational perspective the architecture we are in the process of realising operates as follows. Each device begins an operational cycle by populating its client event queue through taking in data propagated from the various other devices in the network. This per queue information must then be used as the basis for determining the whether or not an event may be happening. This problem is far from trivial as each queue is subject to a potentially different and non deterministic sampling rate meaning that it is the indirect information contained within the queues that must be used. In addition each device will have an independent view of the problem meaning that it is not possible to directly rely on device interdependency characteristics.

5.1 Client Queue Information

As a device can only determine information about the operating environment indirectly through its sensor(s) and the information propagated from its peer units the determination of a world model is an applied example of an estimation theory problem (Thrun, 2002). Therefore to facilitate the interpretation of the data provided from a client event a probabilistic sensor model of the form $p(r|z)$ is used. This model facilitates the derivation of the individual client event confidence values v, mentioned previously in section 4. Therefore:

$$v_i = p(r_i|z_i)$$

where the model we use in this prototype is based upon the characteristics outlined previously in section 4.3. This model relates the client event reading r to the true event state z. This density function is subsequently used in a Bayesian estimation procedure to determine the event state probabilities. Finally a deterministic world model is employed to facilitate the derivation of a optimal world estimator which can be propagated between the individual units that form the world state.

A classical Bayesian approach is used for the determination of the per queue confidence score. Given the current estimate of the state of client C_i, $p[s(C_i) = SE|\{r\}_t]$ based on the observations $r_i = r_1, \ldots, r_t$ and given a new client observation r_{t+1} the new state estimate is provided by

$$k = p[s(C_i) = SE|\{r\}_t + 1] = \frac{p[r_{t+1}|S(C_i) = SE]p[S(C_i) = SE|\{r\}_t]}{\sum_{s(C_i)} p[r_{t+1}|s(C_i)]p[s(C_i)|\{r\}_t]} \quad (2)$$

In the above the previous estimated value of the client state $p[S(C_i) = SE|\{r\}_t]$ serves as the prior and is obtained directly from a localised representation of the global state. The new state of a particular client. determined through the above, is subsequently stored in this representation and propagated to the world.

To facilitate prior estimation for client state a simplified one dimensional Gaussian estimator model is employed.

$$p(r|z) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(\frac{-(r-z)^2}{2\sigma^2}\right) \quad (3)$$

5.2 Inter Device Confidence Regions

As presented above the sensor model is a one dimensional construct associated with a determined or evaluated distance between client devices. Therefore the model can be considered a client information axis from the one dimensional viewpoint. While useful for determining information relating to the 1-1 spatial mapping directly between the devices the model cannot consider areas outside of this conceptual spatial line. It is conceivable that the spatial area between a host device and its client will is also an area of interest. In particular it would be beneficial to have the ability to model a region of confidence emanating directly from the host device to the immediate vicinity of the client device.The basic premise of this concept is outlined in figure 2. When extended in this manner the probabilistic model approaches more closely the type of robotic mapping inverse sensor model highlighted previously. The extended model can be specified as equation 4 where Q is the angle associated with the created confidence region.

$$p(r|z,Q) = \frac{1}{2\pi\sigma_r\sigma_Q} exp\left[-\frac{1}{2}\left(\frac{(r-z)^2}{2{\sigma_r}^2} + \frac{Q^2}{\sigma_Q^2}\right)\right] \quad (4)$$

[1] http://openmoko.com

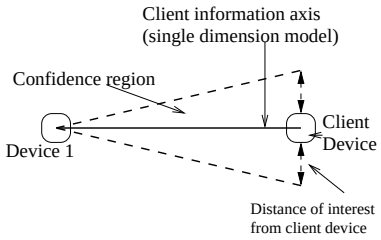


Figure 2: Creating a confidence region between devices.

The availability of these inter device confidence regions will provide for a more information rich profile of the event to be computed. In addition the overlapping of such regions will provide for the ability of assessing and verifying the information coming from individual clients thus adding a novel dimension to the client confidence estimation.

6 MODEL CONSIDERATIONS

As the architecture evolves the consideration of sensory units or other sources of relevant information is necessary. These considerations are highlighted here.

6.1 Client Event Information

At its core the actual sensor model is a statistical estimation formulation which interprets relative range information received from peer devices. Upon the activation of a devices sensors an event signal is propagated through the network. The determination of realistic events on a device versus false positives or false negatives is a separate problem to the data signal handling the sensor model is designed to consume and hence an exposition of same is outside the scope of this context. When an event signal is received the sensor model calculates a probabilistic profile for the event. To illustrate, consider the ideal scenario where a device receives notification of an event from a peer device at what is determined to be at distances of 60km and 100 km respectively from the device. The associated probabilistic profiles determined through the model are outlined in figure 3 where it can be seen that the model is Gaussian in nature. In terms of the device architecture each profile corresponds to a single component of an event queue. The preceding example presented the model in the ideal scenario of there being a 1-1 correspondence between the physical devices distance and the actual distance the information has been determined to travel. In real world settings such an assumption cannot be guaranteed. The model takes cognisance of this fact by its nature as illustrated in figure 4.

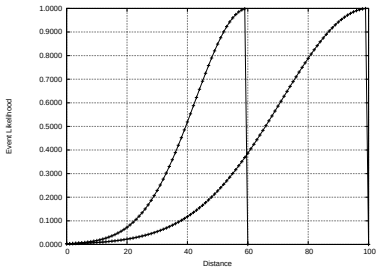


Figure 3: Event profiles for hypothetical distances of 60km and 100km respectively.

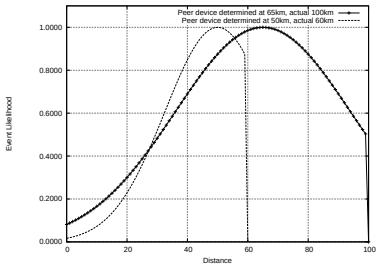


Figure 4: Event profiles in the context of non ideal peer device distances. Hypothetical actual distances are 50km and 65km respectively.

6.2 Information Source Integration

To increase the capability of any such system in general requires that multiple sources of information can be incorporated into a single, useful, information source. This is known as the data fusion problem. Fusion processes are frequently categorised as low, intermediate or high, depending on the processing stage at which fusion takes place(Klien,). Low level fusion, (Data fusion) combines several sources of raw data to produce new raw data. The expectation is that fused data is more informative and synthetic than the original inputs. Within the context of multi-device seismic detection this integration can be performed using a formulation such as that outlined in equation 3 to combine the estimates provided by the independent clients. For two clients C_1 and C_2 this means using the associated client data models $p_1(r|z)$ and $p_2(r|z)$ as the basis for determining the associated combined probability and subsequently applying an appropriate normalisation across the state encapsulated in the client confidence table illustrated in figure 1.

7 CONCLUSIONS AND FUTURE WORK

In this paper we have detailed a sensor modelling framework for earthquake detection using mobile de-

vices which is used within the context of a novel seismic event detection architecture. The sensor model outlined is a probabilistic one Gaussian in nature and similar to the inverse sensor models prevalent in the robotic mapping field. As such it is capable of incrementally and efficiently interpreting event signals propagated throughout the network without the need for predetermined models or sensor associated segmentation decisions. For example the characterisation highlighted in section 6 illustrated that meaningful client event evaluation is possible with a minimal of information i.e. an event notification and a client distance estimate.

In terms of future work regarding the model and its usage a number of areas are prevalent. The choice of an inverse sensor model has some specific implications. Because of its theoretical basis the disambiguation and analysis of client event data is achieved primarily through the use of additional sensing. This has performance implications which need to be addressed. Another area of future work is determining appropriate characteristics for the extension of the one dimensional sensor model to two dimensions. The attribute of interest here is determining a meaningful distance of interest from a client device. To address this problem we initially propose to employ simple heuristic values determined from operational experience. Our long term aim however, is to facilitate the automated derivation of the distance of interest, using triangulation between clients. The evaluation of received client events to determine the true likelihood of an actual earthquake event as opposed to user directed movement is another area of future research. Benchmarking the detection ability of our technique and subsequent model refinement is also an obvious area of future work. Toward this end we intend to correlate our detection results with actual real earthquake data obtained from national earthquake centres and the Stanford Quake-Catcher Network. Finally within the context of the project as a whole another important area of future work will be the specification of a meaningful benchmarking technique, applicable to the domain, to facilitate direct quantitative comparison between techniques such as ours and natural language centric techniques such as the U.S. Geological Surveys Twitter Earthquake Detector (TED)[2].

REFERENCES

Collins, T., Collins, J., and Ryan, C. (2007). Occupancy grid mapping: An empirical evaluation. In *Proceedings of Mediterranean Conference on Control and Automation*.

Ehlers, F., Gustafsson, F., and Spaan, M. Signal processing advances in robots and autonomy. *EURASIP J. Adv. Signal Process*, 2009.

Hewitt, C. (1992). Open information systems semantics for distributed artificial intelligence. *Foundations of artificial intelligence Special Issue of 'Artificial Intelligence' Series*, pages 79–106.

Klien, L. *Sensor and data fusion: A tool for information assessment and decision making*. SPIE Press.

Kortenkamp, D., Bonasso, R., and Murphy, R. (1998). AI-based Mobile Robots: Case studies of successful robot systems.

Moore, J. P. T. (2007). Thumbtribes: Low bandwidth, location-aware communication. In Obaidat, M. S., Lecha, V. P., and Caldeirinha, R. F. S., editors, *WINSYS*, pages 197–202. INSTICC Press.

Murooka, T., Takahara, A., and Miyazaki, T. (2001). A novel network node architecture for high performance and function flexibility. In *ASP-DAC*, pages 551–557.

Newmark, N. and Rosenblueth, E. *Fundamentals of earthquake engineering*. Prentice-Hall.

Ravi, N., Dandekar, N., Mysore, P., and Littman, M. L. (2005). Activity recognition from accelerometer data. In *IAAI'05: Proceedings of the 17th conference on Innovative applications of artificial intelligence*, pages 1541–1546. AAAI Press.

Thrun, S. (2002). Robotic mapping: A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.

Vargas, F., Fagundes, R. D., and D. Barros, J. (2001). Summarizing a new approach to design speech recognition systems: A reliable noise-immune hw-sw version. *Integrated Circuit Design and System Design, Symposium on*, 0:0109.

Zareian, F. and Krawinkler, H. (2009). Simplified performance based earthquake engineering. Technical report, Stanford University.

[2]http://recovery.doi.gov/press/us-geological-survey-twitter-earthquake-detector-t ed/

NONLINEAR INTO STATE AND INPUT DEPENDENT FORM MODEL DECOMPOSITION

Applications to Discrete-time Model Predictive Control with Succesive Time-varying Linearization along Predicted Trajectories

Przemyslaw Orlowski
Institute of Control Engineering, West Pomeranian University of Technology, Szczecin, Poland
orzel@zut.edu.pl

Keywords: Non-linear systems, Successive linearization, Predictive control, Optimal control, Discrete time systems.

Abstract: Linearization techniques are well known tools that can transform nonlinear models into linear models. In the paper we employ a successive model linearization along predicted state and input trajectories resulting in linear time-varying model. The nonlinear behaviour is represented in each time sample by recurrent set of linear time-varying models. Solution of the optimal non-linear model predictive control problem is obtained in an iterative way where the most important step is the linearization along predicted trajectory. The main aim of this paper is to analyse how the nonlinear system should be transformed into linear one to ensure possibly fast solution of the model predictive control problem based on the successive linearization method.

1 INTRODUCTION

Model predictive control (MPC) is attractive control strategy, which have 3 common properties (Camacho et. al. 2004): explicit use of a model to predict the output at future time instants, calculation of a control trajectory minimizing an objective function and receding horizon (moving horizon) strategy. MPC issues for linear systems including stability are well known (Camacho et. al., 2004), (Morari et. al. 1999), (Tatjewski, 2007), (Mayne et. al., 2000) also (Qin et. al. 2003), (Magni et. al. 1999), including fast algorithms (Blachuta, 1999) and discrete-time system with delays (Kowalczuk et. al. 2005). Many real systems are inherently nonlinear. Due to higher product quality specifications, some important environmental and economical reasons linear models are often inadequate to describe the system properties. Computing the optimal control trajectory directly for nonlinear model is difficult, non-convex optimization problem. Generally there is no guarantee that the computed solution is global optimal solution. Moreover it is difficult to prove global stability of the system using directly the nonlinear model for control synthesis. In practise some transformations and simplifications are applied to the nonlinear model in order to prove stability, and also to take advantages of theory for linear systems.

Among some existing approaches in nonlinear model predictive control in the paper we consider successive model linearization along predicted state and input trajectories with recurrent linear time-varying (LTV) model. A large class of these methods uses a common algorithm, i.e. (Kouvartiakis et. al., 1999) employ an optimal control trajectory calculated at the previous time instant of the control algorithm for NMPC. (Lee et. al., 2002) use a similar methodology and employ a linearization at points of the seed trajectory for the discrete-time model of the system. Also the technique presented in (Dutka et. al., 2004), (Ordys et. al., 2001), (Mracek et. al., 1998), (Grimble et. al., 2001), (Dutka et. al., 2003) uses similar idea to (Kouvartiakis et. al., 1999), (Lee et. al., 2002) but with a different model representation and an optimisation technique. Similar approach for the construction of an explicit nonlinear control law approximating nonlinear constrained finite-time optimal control using approximate mapping of a general nonlinear system into a set of piecewise affine systems is presented in (Ulbig et. al., 2007).

The main aim of this paper is to analyse how to linearize (decompose) nonlinear system into linear one for using with the successive model linearization method along predicted state and input trajectories.

The main difficulty is to find proper transformation method, which ensure fast computation of stable and optimal solution for nonlinear control problem.

2 SYSTEM DESCRIPTION

Let us assume general discrete-time, time-varying nonlinear model in the following form:

$$\mathbf{x}(k+1)=\mathbf{f}\left(\mathbf{x}(k),\mathbf{u}(k),k\right) \tag{1}$$

The nonlinear system can be transformed into following discrete-time, time-varying state-dependent form:

$$\mathbf{x}(k+1)= \mathbf{A}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)\mathbf{x}(k)+\mathbf{B}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)\mathbf{u}(k) \tag{2}$$

where $\mathbf{A}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$, $\mathbf{B}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$ are state and input dependent matrices calculated for given initial condition $\mathbf{x}_0$ and control trajectory $\mathbf{u}(k)$ at each time instant.

Then, using the past input and state trajectories, matrices

$$\mathbf{A}(k)=\mathbf{A}\left(\mathbf{x}(k),\mathbf{u}(k),k\right),\ \mathbf{B}(k)=\mathbf{B}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$$

may be calculated for the subsequent points of the trajectories and the nonlinear system (1) is approximated by the LTV model with matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$. Discrete-time LTV system is given in the state space form:

$$\mathbf{x}(k+1)=\mathbf{A}(k)\mathbf{x}(k)+\mathbf{B}(k)\mathbf{u}(k) \tag{3}$$

where

$\mathbf{A}(k)\in\mathbb{R}^{n\times n}, \mathbf{B}(k)\in\mathbb{R}^{n\times m}$, $k=k_0,...,k_0+N-1$ and N is the prediction horizon.

Linear time-varying discrete-time system can be equivalently defined using evolution operators or in the finite horizon case, also by following block matrix operators $\hat{\mathbf{L}},\hat{\mathbf{N}},\hat{\mathbf{B}}$:

$$\hat{\mathbf{L}}=\begin{bmatrix}\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\\ \phi_{k_0+1}^{k_0+1} & \mathbf{I} & \mathbf{0} & \vdots\\ \vdots & \ddots & \mathbf{I} & \mathbf{0}\\ \phi_{k_0+1}^{k_0+N-1} & \cdots & \phi_{k_0+N-1}^{k_0+N-1} & \mathbf{I}\end{bmatrix},\ \hat{\mathbf{N}}=\begin{bmatrix}\phi_{k_0}^{k_0}\\ \vdots\\ \phi_{k_0}^{k_0+N-1}\end{bmatrix} \tag{4}$$

$$\hat{\mathbf{B}}=\begin{bmatrix}\mathbf{B}(k_0) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{B}(k_0+N-1)\end{bmatrix} \tag{5}$$

where $\phi_i^k=\mathbf{A}(k)\mathbf{A}(k-1)\ldots\mathbf{A}(i)$. For state and input trajectories $\hat{\mathbf{x}}$, $\hat{\mathbf{u}}$ we use the following block column vector notation, i.e.

$$\hat{\mathbf{x}}=\left[\mathbf{x}^T(k_0+1)\cdots\mathbf{x}^T(k_0+N)\right]^T \tag{6}$$

$$\hat{\mathbf{u}}=\left[\mathbf{u}^T(k_0)\cdots\mathbf{u}^T(k_0+N-1)\right]^T \tag{7}$$

It follows that the mathematical model can be rewritten in the final form as

$$\hat{\mathbf{x}}=\hat{\mathbf{L}}\hat{\mathbf{B}}\hat{\mathbf{u}}+\hat{\mathbf{N}}\mathbf{x}_0 \tag{8}$$

We assume that at each time instant the system can be analyzed as starting from time sample equal to zero with a current initial condition $\mathbf{x}_0=\mathbf{x}(k_0)$ up to N steps into the future (prediction horizon).

The operator $\hat{\mathbf{L}}\hat{\mathbf{B}}$ is a compact and Hilbert-Schmidt one from l_2 into l_2 and boundedly maps signals $\mathbf{u}(k)\in\mathcal{L}=l_2[k_0,k_0+N-1]$ into signals $x\in\mathcal{X}$.

For simulation purposes we employ cost function in the following form:

$$J=\left(\hat{\mathbf{x}}-\hat{\mathbf{x}}_{ref}\right)^T\hat{\mathbf{P}}\left(\hat{\mathbf{x}}-\hat{\mathbf{x}}_{ref}\right)+\hat{\mathbf{u}}^T\hat{\mathbf{Q}}\hat{\mathbf{u}} \tag{9}$$

where $\hat{\mathbf{P}}\in\mathbb{R}^{(nN)\times(nN)}, \hat{\mathbf{Q}}\in\mathbb{R}^{(mN)\times(mN)}$ are weighting operators, constructed with weighting matrices $\mathbf{P}(k)\in\mathbb{R}^{n\times n}$, $k=1...N$, $\mathbf{Q}(k)\in\mathbb{R}^{m\times m}$, $k=0...N-1$, respectively usually given in following block matrix form:

$$\hat{\mathbf{P}}=\begin{bmatrix}\mathbf{P}(1) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{P}(N)\end{bmatrix},\ \hat{\mathbf{Q}}=\begin{bmatrix}\mathbf{Q}(0) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{Q}(N-1)\end{bmatrix}$$

Usually weighting matrices are time-invariant with the exception of $\mathbf{P}(N)$ which represents the terminal cost. Equivalently the cost function can be rewritten in the following form:

$$J=\sum_{k=1}^{N}\begin{pmatrix}\mathbf{x}(k+k_0)\\ -\mathbf{x}_{ref}(k+k_0)\end{pmatrix}^T\mathbf{P}(k)\begin{pmatrix}\mathbf{x}(k+k_0)\\ -\mathbf{x}_{ref}(k+k_0)\end{pmatrix} +\sum_{k=0}^{N-1}\mathbf{u}^T(k+k_0)\mathbf{Q}(k)\mathbf{u}(k+k_0) \tag{10}$$

where the term

$$\begin{pmatrix}\mathbf{x}(N+k_0)\\ -\mathbf{x}_{ref}(N+k_0)\end{pmatrix}^T\mathbf{P}(N)\begin{pmatrix}\mathbf{x}(N+k_0)\\ -\mathbf{x}_{ref}(N+k_0)\end{pmatrix}$$

for $k=N$ in the first sum of (10) is the terminal cost.

3 PROBLEM DESCRIPTION

The nonlinear system described by the discrete-time nonlinear state space model can be rearranged into the so-called state and control dependent linear form (Mracek et. al., 1998), (Huang et. al., 1996). The

non-linear behaviour of the system is included in the state and control dependent matrices. If the trajectory prediction for the system may be obtained within the algorithm then one can pretend that the future behaviour is known during the prediction horizon (Dutka et. al., 2004). Such a system can be treated as a linear time-varying (LTV) one. Most often the algorithm, shown on fig. 1 has common steps (Kouvartiakis et. al., 1999), (Orlowski, 2005).

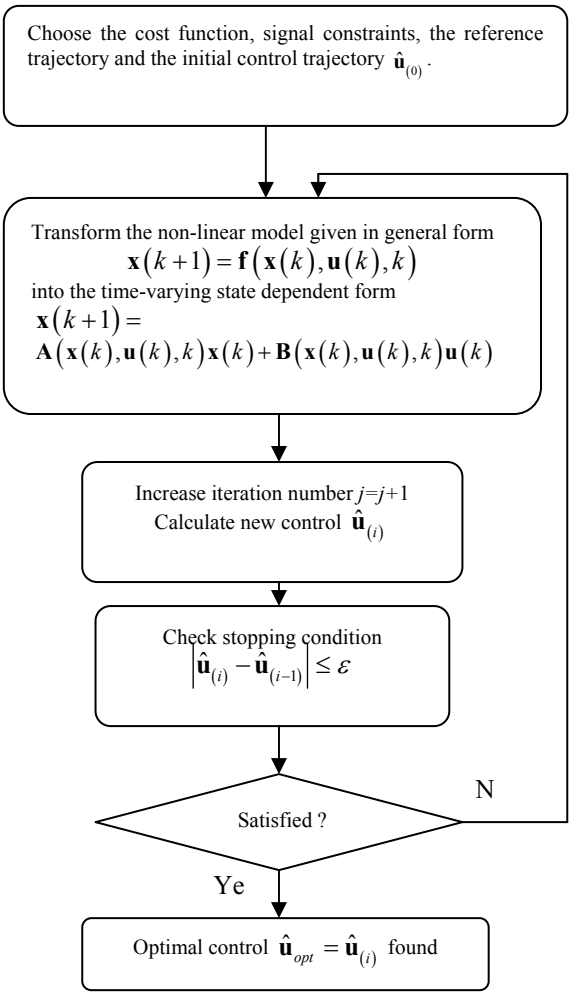


Figure1: Algorithm of the time-varying linearization along predicted trajectory.

In general there no restrictions to the cost function. For simulation purposes we employ cost function given by eq. (9). However in practise the method can be also used with different frequently used in MPC cost functions and stabilizing conditions, e.g.: terminal cost function, terminal equality constraint, terminal constraint set. It is only required to define an MPC problem for the LTV system.

The second important problem is choosing initial control trajectory. The simplest choice could be step control signal with amplitude from normal operating range for the control. Another possibility is to use at the beginning a few initial control trajectories and choose the one which results in the smallest cost function. The trajectory is required *only for linearization purposes* and *only in the first iteration* of the algorithm *for the first time step*. For the consecutive time steps on receding horizon it may be assumed from previous control predictions.

Definition 1. The algorithm from fig. 1 is convergent if there exists a limiting control sequence $\hat{\mathbf{u}}_{opt}$ such that for any arbitrarily small positive number $\varepsilon>0$, there is a large integer I such that for all $i\ge I$, $\left|\hat{\mathbf{u}}_{(i)}-\hat{\mathbf{u}}_{opt}\right|\le\varepsilon$. The algorithm that is not convergent is said to be divergent.

The algorithm converges both for local or global optimal solutions. Divergent algorithm cannot satisfy a stopping condition usually given by following absolute tolerance condition:

$$\left|\hat{\mathbf{u}}_{(i)}-\hat{\mathbf{u}}_{(i-1)}\right|\le\varepsilon \tag{11}$$

for arbitrarily small ε.

The control can be computed using arbitrary method for LTV systems, including algorithms with signal constraints. The algorithm from fig. 1 refer only to one time step computation. Usually it is employed with receding horizon. The algorithm must be repeated for successive time steps $k_0=k_0+1$.

4 NONLINEAR SYSTEM DECOMPOSITION

To transform of the non-linear model (1) into the time-varying state dependent form given by eq. (2) one needs to decompose nonlinear function $\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)$ into 2 factors corresponding to state and input matrices such that: $\mathbf{A}(k)\mathbf{x}(k)+\mathbf{B}(k)\mathbf{u}(k)=\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),k)$.

For example, let us assume nonlinear function:

$f(x(k),u(k))=x(k)\sin(x(k))+u(k)\arctan(u(k))$

Transformation into state and input dependent form can be easily done by simple expansion terms dependent on state and input only, i.e.:

$$\mathbf{A}(k)=\sin(x(k)),\quad \mathbf{B}(k)=\arctan(u(k))$$

More difficult problem is decomposition of a system consisting coupled input-state terms. Assume for example function $f(x(k),u(k))=x(k)u(k)$. One

of possible decompositions is to divide the function into following 2 additive terms:

$$f\left(x(k),u(k)\right)=\alpha x(k)u(k)+(1-\alpha)x(k)u(k)$$

where:

$$\mathbf{A}(k)=\alpha u(k),\ \ \mathbf{B}(k)=(1-\alpha)x(k)$$

In general we propose following method which allow to decompose arbitrary nonlinear function $\mathbf{f}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$ into series of M additive components. Using the simplified notation $\mathbf{f}_i=\mathbf{f}_i\left(\mathbf{x}(k),\mathbf{u}(k),k\right)$ for a fixed input trajectory and initial conditions we have

$$\mathbf{f}\left(\mathbf{x}(k),\mathbf{u}(k),k\right)=\sum_{i=1}^{M}\mathbf{f}_i\left(\mathbf{x}(k),\mathbf{u}(k),k\right)=\sum_{i=1}^{M}\mathbf{f}_i \quad (12)$$

Every system (1) can be decomposed into the state dependent form (2). In general, this decomposition takes the following form:

$$\mathbf{x}(k+1)=\sum_{i=1}^{M}\mathbf{f}_i=\sum_{j=1}^{n}\left(\sum_{i=1}^{M}\alpha_{i,j}\mathbf{f}_i\right)+\sum_{j=1}^{m}\left(\sum_{i=1}^{M}\beta_{i,j}\mathbf{f}_i\right) \quad (13)$$

$$\mathbf{x}(k+1)= \sum_{i=1}^{M}\alpha_{i,1}\mathbf{f}_i+\ldots+\sum_{i=1}^{M}\alpha_{i,n}\mathbf{f}_i+\sum_{i=1}^{M}\beta_{i,1}\mathbf{f}_i+\ldots+\sum_{i=1}^{M}\beta_{i,m}\mathbf{f}_i \quad (14)$$

What can be arranged into following vector-matrix state and input dependent form:

$$\begin{aligned}\mathbf{x}(k+1)&=\mathbf{a}_1x_1+\ldots+\mathbf{a}_nx_n+\mathbf{b}_1u_1+\ldots+\mathbf{b}_mu_m\\&=\left[\mathbf{a}_1\cdots\mathbf{a}_n\right]\left[x_1\cdots x_n\right]^T+\left[\mathbf{b}_1\cdots\mathbf{b}_m\right]\left[u_1\cdots u_m\right]^T\\&=\mathbf{Ax}+\mathbf{Bu}\end{aligned} \quad (15)$$

where

$$\mathbf{a}_jx_j=\sum_{i=1}^{M}\alpha_{i,j}\mathbf{f}_i, \quad (16)$$

$$\mathbf{b}_ju_j=\sum_{i=1}^{M}\beta_{i,j}\mathbf{f}_i, \quad (17)$$

$$\forall_i\ \sum_{j=1}^{n}\alpha_{i,j}+\sum_{j=1}^{m}\beta_{i,j}=1 \quad (18)$$

The component column vectors of matrices $\mathbf{A}(k)$ and $\mathbf{B}(k)$ can be determined under assumption that the following limits $\forall_j \lim_{x_j\to 0}\frac{\mathbf{a}_jx_j}{x_j}, \forall_j \lim_{u_j\to 0}\frac{\mathbf{b}_ju_j}{u_j}$ exist and are finite. These vectors are given by expressions

$$\mathbf{a}_j=\begin{cases}\dfrac{\mathbf{a}_jx_j}{x_j} & x_j\neq 0\\ \lim\limits_{x_j\to 0}\dfrac{\mathbf{a}_jx_j}{x_j} & x_j=0\end{cases} \quad (19)$$

$$\mathbf{b}_j=\begin{cases}\dfrac{\mathbf{b}_ju_j}{u_j} & \mathrm{u}_j\neq 0\\ \lim\limits_{u_j\to 0}\dfrac{\mathbf{b}_ju_j}{u_j} & \mathrm{u}_j=0\end{cases} \quad (20)$$

where

$\mathbf{A}(k)=\left[\mathbf{a}_1\cdots\ \mathbf{a}_n\right], \mathbf{B}(k)=\left[\mathbf{b}_1\cdots\mathbf{b}_m\right]$, n – order, m – number of inputs, $\mathbf{a}_j, \mathbf{b}_j$ - column vectors with n rows

Let us assume that function $\mathbf{f}(\mathbf{x},\mathbf{u},k)$ can be decomposed into the following four additive terms:

$$\mathbf{f}\left(\mathbf{x},\mathbf{u},k\right)= \mathbf{f}_1\left(\mathbf{x},k\right)+\mathbf{f}_2\left(\mathbf{x},\mathbf{u},k\right)+\mathbf{f}_3\left(\mathbf{u},k\right)+\mathbf{f}_4\left(k\right) \quad (21)$$

The vector functions $\mathbf{f}$ must be continuous and the following limits calculated in respect to all coordinates of $\mathbf{f}$ and $\mathbf{x}/\mathbf{u}$ must be finite:

$$\lim_{\mathbf{x}\to 0}\frac{\mathbf{f}_1}{\mathbf{x}}, \lim_{\mathbf{u}\to 0}\frac{\mathbf{f}_3}{\mathbf{u}} \quad (22)$$

and either

$$\lim_{\mathbf{x}\to 0}\frac{\mathbf{f}_2}{\mathbf{x}},\ \text{or/and}\ \lim_{\mathbf{u}\to 0}\frac{\mathbf{f}_2}{\mathbf{u}} \quad (23)$$

where

$$\mathbf{f}_1=\begin{bmatrix}f_{1[1]}\\ \vdots\\ f_{1[R]}\end{bmatrix},\ \lim_{\mathbf{x}\to 0}\frac{\mathbf{f}_1}{\mathbf{x}}\sim\begin{bmatrix}\lim\limits_{x_1\to 0}\dfrac{f_{1[1]}}{x_1} & \cdots & \lim\limits_{x_R\to 0}\dfrac{f_{1[1]}}{x_R}\\ \vdots & \ddots & \vdots\\ \lim\limits_{x_1\to 0}\dfrac{f_{1[R]}}{x_1} & \cdots & \lim\limits_{x_R\to 0}\dfrac{f_{1[R]}}{x_R}\end{bmatrix}$$

And the limit is finite if and only if all elements in above matrix are finite.

Norms of matrices **A, B** should approach neither zero nor infinity. The best performance is achieved if the norms of matrices **A, B** have similar order of magnitudes.

Although the convergence of the algorithm from fig. 1 for a given decomposition cannot be proved for general nonlinear systems stability for linearized ones follows directly from the applied computation method for control. The conversion from a nonlinear into LTV system can be successfully applied to all systems for which the optimal nonlinear control lies in the neighbourhood of the optimal control for the linearized LTV system.

5 NUMERICAL EXAMPLE

In the example algorithm from fig. 1 is combined with formula (24), where $\mathbf{x}_0$ is current initial

condition $\mathbf{x}_0 = \mathbf{x}(k_0)$ and $\hat{\mathbf{P}}$, $\hat{\mathbf{Q}}$ are weighting matrices. Control is calculated iteratively using cost function (9) with $\hat{\mathbf{x}}_{ref} = \mathbf{0}$, from following formula:

$$\hat{\mathbf{u}}_{(i+1)} = -\left(\left(\hat{\mathbf{L}}_{(i)}\hat{\mathbf{B}}_{(i)}\right)^T \hat{\mathbf{P}}\hat{\mathbf{L}}_{(i)}\hat{\mathbf{B}}_{(i)} + \hat{\mathbf{Q}}\right)^{-1}\left(\hat{\mathbf{L}}_{(i)}\hat{\mathbf{B}}_{(i)}\right)^T \hat{\mathbf{P}}\hat{\mathbf{N}}_{(i)}\mathbf{x}_0 \quad (24)$$

We assume following model for the nonlinear system:

$$x_{k+1} = x_k^2 + 0.5x_k^2 u_k + u_k^3 \quad (25)$$

The initial control trajectory is equal to $\hat{\mathbf{u}}_{(0)} = -0.5[1,1,1]$, the absolute tolerance, defined by (11) $\varepsilon = 0.001$ and the weighting matrices are unitary $\hat{\mathbf{P}} = \hat{\mathbf{I}}_P$, $\hat{\mathbf{Q}} = \hat{\mathbf{I}}_Q$. The system (25) can be decomposed into two following state and input dependent parts:

$$x_{k+1} = \underbrace{\left(x_k + \alpha x_k u_k\right)}_{A(x_k,u_k)} \cdot x_k + \underbrace{\left(\left(0.5-\alpha\right)x_k^2 + u_k^2\right)}_{B(x_k,u_k)} \cdot u_k \quad (26)$$

The decomposition is dependent on parameter α. Equation (26) is equivalent to (25) for arbitrary values of α, although convergence of the algorithm from fig. 1 is analysed for $\alpha \in [-5, 0.5]$.

Figure 2 shows number if iterations η required to converge to optimal control solution for given initial state $x_0 \in (0, 8]$ and decomposition parameter $\alpha \in [-5, 0.5]$. To improve readability of the figure 2 it is also assumed that $\eta \leq 100$. Value $\eta = 100$ corresponds to a divergent solutions or solutions with that require more than 100 iterations. It may be concluded from fig. 2 that convergence of the algorithm from fig. 1 is dependent both on the initial state and the decomposition. Usually it is required for the algorithm to be convergent and possibly fast for all initial conditions from given range. To ensure fast convergence (the minimal number if iterations) for e.g. x_0=8 parameter α should be chosen in the range $\alpha \in [-0.5, 0]$, whereas for x_0=1.4 the smallest number if iterations is for $\alpha \in [-3, -1.5]$. For $x_0 < 1$ the algorithm is fast convergent for all α.

It should be underlined that the convergence/divergence is a property of: the system, the initial condition, the decomposition and the initial control trajectory. First of all it is assumed that the system is controllable and observable and the state is reachable from arbitrary initial state x_0. Although changes in each of three above factors may be effective to achieve convergence of the algorithm, the easiest way to improve the method or fasten the algorithm is to change the decomposition. Convergence of the algorithm is strongly connected with the conditional number r_{cond} of the inverse of

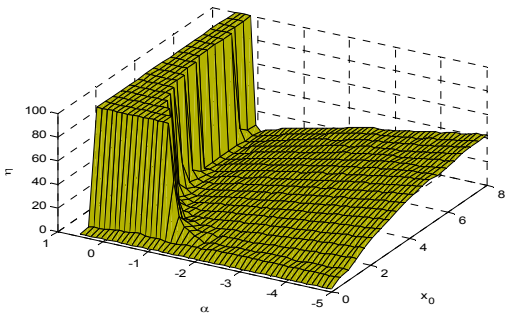


Figure 2: Number of iterations η required to converge optimal control solution for given initial state x_0 and the decomposition parameter α for unitary weighting operators without terminal cost and time horizon N=3.

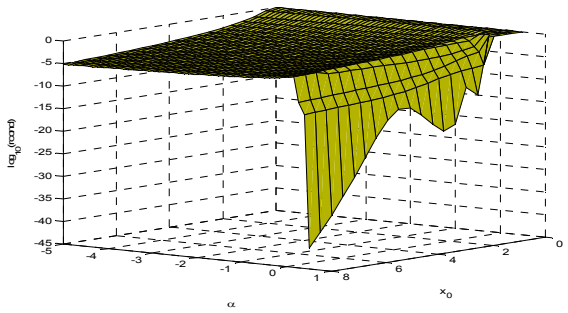


Figure 3: Logarithm base 10 of reciprocal condition number estimate vs. initial state x_0 and the decomposition parameter α for unitary weighting operators without terminal cost and time horizon N=3.

matrix $\left(\left(\hat{\mathbf{L}}_{(i)}\hat{\mathbf{B}}_{(i)}\right)^T \hat{\mathbf{P}}\hat{\mathbf{L}}_{(i)}\hat{\mathbf{B}}_{(i)} + \hat{\mathbf{Q}}\right)$. Logarithm base 10 of the conditional number is shown in figure 3.

6 CONCLUSIONS

The paper discuss selected problems concerned to successive model linearization along predicted state and input trajectories with linear time varying model.

The paper mainly focus on the transformation method from a general nonlinear form into the state space dependent form. We formulate the problem and introduce the generalised form of the algorithm.

Nonlinearities are decomposed into two additive terms – state and input dependent matrices of the state space dependent form and then model predictive control can be calculated using methods for linear systems.

An important consequence of the chosen decomposition is reachability of the optimal solution

and required computation time – number of iterations. In many cases the number of iterations can be cut down. The optimal decomposition, for which the algorithm is convergent with minimal number of iterations depends on the initial condition – for receding horizon problems the initial condition is the current state in each time sample. The selection of the decomposition parameters α, β should be always connected with current value of the state to ensure suitable value of conditional number corresponding to the inverse of matrix in formula (24).

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Science and Higher Education in Poland under the grant N N514 298535.

REFERENCES

Błachuta, M. J. 1999. On Fast State-Space Algorithms for Predictive Control. *Int. J. Appl. Math. Comput. Sci*., Vol. 9, No. 1, 149-160.

Camacho EF., Bordons C. 2004. *Model Predictive Control*. Springer.

Chen H., Allgower F. 1998. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1218.

De Nicolao G., Magni L., Scattolini R.. 2000. Stability and robustness of nonlinear receding horizon control. In F. Allgower and A. Zheng, editors, *Nonlinear Predictive Control*, pp. 3–23. Birkhauser.

Dutka A. S., Ordys A. W., Grimble M. J. 2003. Nonlinear Predictive Control of 2 dof helicopter model, *IEEE CDC proceedings*.

Dutka, A., Ordys A. 2004. The Optimal Non-linear Generalised Predictive Control by the Time-Varying Approximation, *Proc. of 10th IEEE Int. Conf. MMAR.* Miedzyzdroje. Poland. pp. 299-304.

Fontes F. A. 2000. A general framework to design stabilizing nonlinear model predictive controllers. *Syst. Contr. Lett.*, 42(2):127–143.

Grimble M. J., Ordys A. W. 2001. Non-linear Predictive Control for Manufacturing and Robotic Applications, *Proc. of 7th IEEE Int. Conf. MMAR.* Miedzyzdroje.

Huang Y., Lu W.M. 1996. Nonlinear Optimal Control: Alternatives to Hamilton-Jacobi Equation", *Proc. of the 35th IEEE Conference on Decision and Control*, pp. 3942-3947.

Kouvaritakis B., Cannon M., Rossiter J. A. 1999. Non-linear model based predictive control, *Int. J. Control*, Vol. 72, No. 10, pp. 919-928.

Kowalczuk Z, Suchomski P. 2005. Discrete-Time Predictive Control With Overparameterized Delay-Plant Models And An Identified Cancellation Order. *Int. J. Appl. Math. Comput. Sci*, Vol. 15, No. 1, 5–34.

Lee Y. I., Kouvaritakis B., Cannon M. 2002. Constrained receding horizon predictive control for nonlinear systems, *Automatica*, Vol. 38, No. 12, pp. 2093-2102.

Magni, L., De Nicolao, G., Scattolini, R. 1999. Some Issues in the Design of Predictive Controllers. *Int. J. Appl. Math. Comput. Sci.*, Vol. 9, No. 1, 9-24.

Mayne D.Q., Rawlings J.B., Rao C.V., Scokaert P.O.M. 2000. Constrained model predictive control: stability and optimality. *Automatica*, 26(6):789–814.

Morari M. and Lee J. 1999. Model predictive control: Past, present and future. *Comput. Chem. Engi.*, Vol. 23, No. 4/5, pp. 667–682.

Morari M., de Oliveira Kothare S. 2000. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Aut. Contr*., 45(6):1053–1071.

Mracek C. P., Cloutier J. R. 1998. Control Designs for the nonlinear benchmark problem via the State-Dependent Riccati Equation method, *International Journal of Robust and Nonlinear Control*, 8, pp. 401-433.

Ordys A. W., Grimble M. J. 2001. Predictive control design for systems with the state dependent non-linearities, *SIAM Conference on Control and its Applications*, San Diego, California.

Orlowski, P. 2005. Convergence of the optimal non-linear GPC method with iterative state-dependent, linear time-varying approximation, *Proc. of Int. Workshop on Assessment and Future Directions of NMPC*, Freudenstadt-Lauterbad, Germany, pp. 491-497.

Primbs J., Nevistic V., Doyle J. 1999. Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24.

Qin S. J. and Badgwell T. 2003. A survey of industrial model predictive control technology. *Contr. Eng. Pract.*, Vol. 11, No. 7, pp. 733–764.

Scokaert P.O.M., Mayne D.Q., Rawlings J.B. 1999. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Automat. Contr*., 44(3):648–654.

Tatjewski P. 2007. *Advanced Control of Industrial Processes, Structures and Algorithms.* London: Springer.

Ulbig A., Olaru S., Dumur D., Boucher P. 2007. Explicit solutions for nonlinear model predictive control : a linear mapping approach, *European Control Conference ECC 2007*, Kos, Grece.

A SUBOPTIMAL FAULT-TOLERANT DUAL CONTROLLER IN MULTIPLE MODEL FRAMEWORK

Ivo Punčochář and Miroslav Šimandl
Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, Plzeň, Czech Republic
{*ivop, simandl*}*@kky.zcu.cz*

Keywords: Fault-tolerant control, Fault detection, Optimal control, Dual control, Stochastic systems.

Abstract: The paper focuses on the design of a suboptimal fault-tolerant dual controller for stochastic discrete-time systems. Firstly a general formulation of the active fault detection and control problem that covers several special cases is presented. One of the special cases, a dual control problem, is then considered throughout the rest of the paper. It is stressed that the designed dual controller can be regarded as a fault-tolerant dual controller in the context of fault detection. Due to infeasibility of the optimal fault-tolerant dual controller for general non-linear system, a suboptimal fault-tolerant dual controller based on rolling horizon technique for jump Markov linear Gaussian system is proposed and illustrated by means of a numerical example.

1 INTRODUCTION

Fault detection is an important part of many automatic control systems and it has attracted a lot of attention during recent years because of increasing requirements on safety, reliability and low maintenance costs. An elementary aim of fault detection is early recognition of faults, e.i. undesirable behaviors of an observed system.

The very earliest fault detection methods use additional sensors for detecting faults. These methods are simple and still used in safety-critical systems. A slightly better fault detection methods utilize some basic assumptions on measured signals and therefore they are usually called signal based methods (Isermann, 2005). To further improve fault detection, more complex methods called model based were developed (Basseville and Nikiforov, 1993).

Except for a few situations were the primary objective is the fault detection itself, it usually complements a control system where the quality of control is of main concern. This fact has stimulated research in area of so called fault-tolerant control. Fault-tolerant control methods can be divided into two basic group: passive fault-tolerant control and active fault-tolerant control methods (Blanke et al., 2003). Passive fault-tolerant control methods design a controller that is robust with respect to considered faults and thus an acceptable deterioration of control quality is caused by the considered faults. On the other hand, active fault-tolerant control methods try to estimate faults and reconfigure a controller in order to retain desired closed loop behavior of a system.

The mentioned fault detection methods and fault-tolerant approaches usually use available measurements passively as shown at the top of Fig. 1, where a passive detector uses inputs $\mathbf{u}_k$ and measurements $\mathbf{y}_k$ for generating decisions $\mathbf{d}_k$. In the case of stochastic systems further improvement can be obtained by applying a suitable input signal, see e.g. (Mehra, 1974) for application in parameter estimation problem. This idea leads to so-called active fault detection which is depicted at the bottom of Fig. 1. The active detector and controller generates, in addition to a decision $\mathbf{d}_k$, an input signal $\mathbf{u}_k$ that controls and simultaneously excites the system and thus improves fault detection and control quality. Note, that the terms passive and active have different meaning than in the fault-tolerant control literature.

The active fault detection is a developing area. The first attempt to formulate and solve the active fault detection problem can be found in (Zhang, 1989), where the sequential probability ratio test was used for determining a valid model and an auxiliary input signal was designed to minimize average number of samples. More general formulation of active fault detection was proposed in (Kerestecioğlu, 1993). An active fault detection for systems with deterministic bounded disturbances was introduced in (Campbell and Nikoukhah, 2004). A unified formulation of active fault detection and control for stochastic systems that covers several special cases was proposed

in (Šimandl and Punčochář, 2009). One of these special cases is the optimal dual control problem that has not been elaborated in the context of that general formulation, yet.

Therefore, the aim of this paper is to examine the dual control problem in the context of fault detection problem. The general formulation for the optimal dual control problem is adopted from (Šimandl and Punčochář, 2009) and an optimal fault-tolerant controller that uses idea of active probing for improving the quality of control is designed. Because of infeasibility of the optimal fault-tolerant dual controller for a general nonlinear stochastic system, the systems that can be described using jump linear Gaussian multiple models are considered and the rolling horizon technique is used for obtaining an approximate solution.

The paper is organized as follows. A general formulation of active fault detection and control is given in Section 2 and the design of a fault-tolerant dual controller is introduced as a special case of the general formulation. The optimal fault-tolerant dual controller obtained using the closed loop information processing strategy is presented in Section 3. Section 4 is devoted to the description of a system using multiple models and the relations for state estimation are given. Finally, a suboptimal fault-tolerant dual controller based on rolling horizon technique is presented in Section 5.

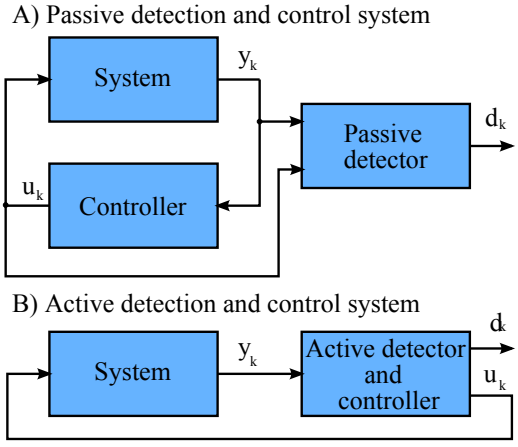


Figure 1: Block diagrams of the passive detection and control system and the active detection and control system.

2 PROBLEM STATEMENT

In this section a general formulation of the active fault detection and control problem is adopted and then a fault-tolerant dual control problem is specified as a special case of the general formulation.

2.1 System

The problem is considered on the finite horizon F. Let an observed system be described at each time $k \in \mathcal{T} = \{0,\ldots,F\}$ by the state space discrete-time nonlinear stochastic model

$$\mathbf{x}_{k+1} = \mathbf{f}_k\left(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{w}_k\right), \tag{1}$$

$$\mu_{k+1} = \mathbf{g}_k\left(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{e}_k\right), \tag{2}$$

$$\mathbf{y}_k = \mathbf{h}_k\left(\mathbf{x}_k, \mu_k, \mathbf{v}_k\right), \tag{3}$$

where nonlinear vector functions $\mathbf{f}_k\left(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{w}_k\right)$, $\mathbf{g}_k\left(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{e}_k\right)$ and $\mathbf{h}_k\left(\mathbf{x}_k, \mu_k, \mathbf{v}_k\right)$ are known. The input and output of the system are denoted as $\mathbf{u}_k \in \mathcal{U}_k \subseteq \mathbb{R}^{n_u}$ and $\mathbf{y}_k \in \mathbb{R}^{n_y}$, respectively. The subset $\mathcal{U}_k$ can be continuous or discrete and it determines admissible values of the input $\mathbf{u}_k$. The unmeasured state $\bar{\mathbf{x}}_k = \left[\mathbf{x}_k^T, \mu_k^T\right]^T$ consists of variables $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and $\mu_k \in \mathcal{M} \subseteq \mathbb{R}^{n_\mu}$. The variable $\mathbf{x}_k$ is the part of the state that should be driven by the input $\mathbf{u}_k$ to a desirable value or region. The variable μ_k carries information about faults. The variable μ_k can be a vector representing fault signals or a scalar that determines the mode of system behavior. The initial state $\bar{\mathbf{x}}_0$ is described by the known probability density function (pdf) $p\left(\bar{\mathbf{x}}_0\right) = p\left(\mathbf{x}_0\right) p\left(\mu_0\right)$. The pdfs $p\left(\mathbf{w}_k\right)$, $p\left(\mathbf{e}_k\right)$ and $p\left(\mathbf{v}_k\right)$ of the white noise sequences $\{\mathbf{w}_k\}$, $\{\mathbf{e}_k\}$ and $\{\mathbf{v}_k\}$ are known. The initial state $\bar{\mathbf{x}}_0$ and the noise sequences $\{\mathbf{w}_k\}$, $\{\mathbf{e}_k\}$, $\{\mathbf{v}_k\}$ are mutually independent.

2.2 Active Fault Detector and Controller

In the general formulation, the goal is to design a dynamic causal deterministic system that uses complete available information to generate a decision about faults and an input to the observed system. Such a system can be described at each time step $k \in \mathcal{T}$ by the following relation

$$\begin{bmatrix} \mathbf{d}_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \sigma_k\left(\mathbf{I}_0^k\right) \\ \gamma_k\left(\mathbf{I}_0^k\right) \end{bmatrix} = \rho_k\left(\mathbf{I}_0^k\right), \tag{4}$$

where $\sigma_k\left(\mathbf{I}_0^k\right)$ and $\gamma_k\left(\mathbf{I}_0^k\right)$ are some unknown vector functions which should be designed to obtain an active fault detector and controller. The complete available information, which has been received up to the time k, is stored in the information vector $\mathbf{I}_0^k = \left[\mathbf{y}_0^{k\,T}, \mathbf{u}_0^{k-1\,T}, \mathbf{d}_0^{k-1\,T}\right]^T$. The notation $\mathbf{y}_i^j$ represents a sequence of the variables $\mathbf{y}_k$ from the time step i up to the time step j. If $i > j$ then the sequence $\mathbf{y}_i^j$ is empty and the corresponding variable is simply left out from an expression. According to this rule, the information vector for time $k = 0$ is defined as $\mathbf{I}_0^0 = \mathbf{I}_0 = \mathbf{y}_0$.

2.3 Criterion

Analogously to the optimal stochastic control problem (Bar-Shalom and Tse, 1974), the design of the optimal active detector and controller is based on minimization of a criterion. A general criterion that penalizes wrong decisions $\mathbf{d}_k$ and deviations of variables $\mathbf{x}_k$ and $\mathbf{u}_k$ from desired values over the finite horizon is the following

$$J\left(\rho_0^F\right) = \mathrm{E}\left\{L\left(\mathbf{x}_0^F, \mu_0^F, \mathbf{u}_0^F, \mathbf{d}_0^F\right)\right\}, \tag{5}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator with respect to all included random variables and $L\left(\mathbf{x}_0^F, \mu_0^F, \mathbf{u}_0^F, \mathbf{d}_0^F\right)$ is a non-negative real-valued cost function. Due to practical reasons, the cost function is considered in the following additive form

$$L\left(\mathbf{x}_0^F, \mu_0^F, \mathbf{u}_0^F, \mathbf{d}_0^F\right) = \sum_{k=0}^{F} \alpha_k L_k^{\mathrm{d}}\left(\mathbf{d}_k, \mu_k\right) + \left(1-\alpha_k\right) L_k^{\mathrm{c}}\left(\mathbf{x}_k, \mathbf{u}_k\right), \tag{6}$$

where $L_k^{\mathrm{d}}(\mu_k, \mathbf{d}_k)$ is a non-negative real-valued cost function representing the detection aim, the non-negative real-valued cost function $L_k^{\mathrm{c}}(\mathbf{x}_k, \mathbf{u}_k)$ expresses the control aim, and the coefficient α_k belonging to the closed interval $[0,1]$ weights between these two aims. In order to regard the function $L_k^{\mathrm{d}}(\mu_k, \mathbf{d}_k)$ as a meaningful cost function, it should satisfy the inequality $L_k^{\mathrm{d}}(\mu_k, \mu_k) \leq L_k^{\mathrm{d}}(\mu_k, \mathbf{d}_k)$ for all $\mu_k \in \mathcal{M}$, $\mathbf{d}_k \in \mathcal{M}$, $\mathbf{d}_k \neq \mu_k$ at each time step $k \in \mathcal{T}$, and the strict inequality has to hold at least at one time step. The sequence of the functions $\rho_0^{F*} = [\rho_0^*, \rho_1^*, \ldots, \rho_F^*]$ given by minimization of (5) specifies the optimal active detector and controller. The minimization of the criterion (5) can be solved by using three different information processing strategies (IPS's) (Šimandl and Punčochář, 2009), but only the closed loop (CL) IPS is considered in this paper because of its superiority.

2.4 Fault-tolerant Dual Controller

The introduced general formulation covers several special cases that can be simply derived by choosing a particular weighting coefficient α_k and fixing the function $\sigma_k\left(\mathbf{I}_0^k\right)$ or the function $\gamma_k\left(\mathbf{I}_0^k\right)$ in advance. This paper is focused on the special case where only control aim is considered, i.e. the coefficient α_k is set to zero for all $k \in \mathcal{T}$ and none of the functions $\sigma_k\left(\mathbf{I}_0^k\right)$ and $\gamma_k\left(\mathbf{I}_0^k\right)$ are specified in advance. The cost function $L_k^{\mathrm{c}}(\mathbf{x}_k, \mathbf{u}_k)$ is considered to be a quadratic cost function

$$L_k^{\mathrm{c}}\left(\mathbf{x}_k, \mathbf{u}_k\right) = \left[\mathbf{x}_k - \mathbf{r}_k\right]^T \mathbf{Q}_k \left[\mathbf{x}_k - \mathbf{r}_k\right] + \mathbf{u}_k^T \mathbf{R}_k \mathbf{u}_k, \tag{7}$$

where $\mathbf{Q}_k$ is a symmetric positive semidefinite matrix, $\mathbf{R}_k$ is a symmetric positive definite matrix, and $\mathbf{r}_k$ is a reference signal. It is considered that the reference signal $\mathbf{r}_k$ is known for the whole horizon in advance.

Since decisions are no longer penalized in the criterion, the function $\sigma_k\left(\mathbf{I}_0^k\right)$ can not be determined by the minimization and the aim is to find only functions $\gamma_k\left(\mathbf{I}_0^k\right)$ for all k. The resulting controller will steer the system in such a way that the criterion is minimized regardless the faults μ_k. Moreover the controller can exhibit the dual property because the CL IPS is used. Due to these two facts the controller can be denoted as the fault-tolerant dual controller.

3 DESIGN OF FAULT-TOLERANT DUAL CONTROLLER

This section is devoted to the optimal fault-tolerant dual controller design. The minimization of the criterion (5) using the CL IPS can be solved by the dynamic programming where the minimization is solved backward in time (Bertsekas, 1995).

The optimal fault-tolerant dual controller is obtained by solving the following backward recursive equation for time steps $k = F, F-1, \ldots, 0$

$$V_k^*\left(\mathbf{y}_0^k, \mathbf{u}_0^{k-1}\right) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \mathrm{E}\left\{L_k^{\mathrm{c}}\left(\mathbf{x}_k, \mathbf{u}_k\right) + V_{k+1}^*\left(\mathbf{y}_0^{k+1}, \mathbf{u}_0^k\right) \middle| \mathbf{y}_0^k, \mathbf{u}_0^k\right\}, \tag{8}$$

where $\mathrm{E}\{\cdot|\cdot\}$ stands for the conditional expectation operator and the Bellman function $V_k^*\left(\mathbf{y}_0^k, \mathbf{u}_0^{k-1}\right)$ is the estimate of the minimal cost incurred from time step k up to the final time step F given the input-output data $\left[\mathbf{y}_0^k, \mathbf{u}_0^k\right]$. The initial condition for the backward recursive equation (8) is $V_{F+1}^* = 0$ and it can be shown that the optimal value of the criterion (5) is $J^* = J\left(\rho_0^{F*}\right) = \mathrm{E}\left\{V_0^*\left(\mathbf{y}_0\right)\right\}$. Obviously, the optimal input signal $\mathbf{u}_k^*$ is given as

$$\mathbf{u}_k^* = \gamma_k^*\left(\mathbf{y}_0^k, \mathbf{u}_0^{k-1}\right) = \arg \min_{\mathbf{u}_k \in \mathcal{U}_k} \mathrm{E}\left\{L_k^{\mathrm{c}}\left(\mathbf{x}_k, \mathbf{u}_k\right) + V_{k+1}^*\left(\mathbf{y}_0^{k+1}, \mathbf{u}_0^k\right) \middle| \mathbf{y}_0^k, \mathbf{u}_0^k\right\}, \tag{9}$$

where the function $\gamma_k^*(\mathbf{I}_0^k)$ represents the optimal fault-tolerant dual controller. The pdf's $p(\bar{\mathbf{x}}_k|\mathbf{I}_0^k, \mathbf{u}_k, \mathbf{d}_k)$ and $p(\mathbf{y}_{k+1}|\mathbf{I}_0^k, \mathbf{u}_k, \mathbf{d}_k)$ needed for the evaluation of the conditional expectation can be obtained using nonlinear filtering methods. Note that there isn't any closed form solution to equations (8) and (9). Therefore approximate techniques have to be used to get at least a suboptimal solution. The selection of a suitable approximation depends on a particular system description and estimation method.

4 MULTIMODEL APPROACH

In the case of a general nonlinear system the state estimation pose a complex functional problem that has to be solved using approximate techniques. One of the attractive method is based on the assumption that the system exhibits distinct modes of behavior. Such systems can be encountered in various field of interests including maneuvering target tracking (Bar-Shalom et al., 2001), abrupt fault detection (Zhang, 1989) and adaptive control (Athans et al., 2006). In this paper, the multimodel approach is used as one step towards the design of feasible fault-tolerant dual controller.

Henceforth, it is assumed that the variable μ_k is a scalar index from the finite discrete set $\mathcal{M} = \{1,2,\ldots,N\}$ that determines the model valid at time step k. If the exact behavior modes of the system are not known, the set $\mathcal{M}$ can be determined by using existing techniques, see e.g. (Athans et al., 2006).

It is considered that the system can be described at each time $k \in \mathcal{T}$ as

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_{\mu_k}\mathbf{x}_k + \mathbf{B}_{\mu_k}\mathbf{u}_k + \mathbf{G}_{\mu_k}\mathbf{w}_k, \\ \mathbf{y}_k &= \mathbf{C}_{\mu_k}\mathbf{x}_k + \mathbf{H}_{\mu_k}\mathbf{v}_k \end{aligned} \tag{10}$$

where the meaning of the variables $\mathbf{x}_k$, $\mathbf{y}_k$, $\mathbf{u}_k$, $\mathbf{w}_k$ and $\mathbf{v}_k$ is the same as in (1) to (3). The set $\mathcal{U}_k$ is considered to be discrete. The pdf's of the noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are Gaussian with zero-mean and unit variance. The scalar random variable $\mu_k \in \mathcal{M}$ denotes the index of the correct model at time k. Random model switching from model i to model j is described by the known conditional transition probability $P(\mu_{k+1} = j|\mu_k = i) = P_{ij}$. Obviously, the decision $d_k \in \mathcal{M}$ is now scalar too. Known matrices $\mathbf{A}_{\mu_k}$, $\mathbf{B}_{\mu_k}$, $\mathbf{G}_{\mu_k}$, $\mathbf{C}_{\mu_k}$, and $\mathbf{H}_{\mu_k}$ have appropriate dimensions.

The conditional pdf of the state $\mathbf{x}_k$ is a weighted sum of Gaussian distributions

$$\begin{aligned} p\left(\mathbf{x}_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right) = \\ \sum_{\mu_0^k} p\left(\mathbf{x}_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1},\mu_0^k\right) P\left(\mu_0^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right), \end{aligned} \tag{11}$$

where Gaussian conditional pdf $p(\mathbf{x}_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1},\mu_0^k)$ can be computed using a Kalman filter that corresponds to the model sequence μ_0^k. The pdf $P(\mu_0^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1})$ can be obtained recursively as

$$\begin{aligned} P\left(\mu_0^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right) = \frac{p\left(\mathbf{y}_k|\mathbf{y}_0^{k-1},\mathbf{u}_0^{k-1},\mu_0^k\right)}{c} \\ \times P\left(\mu_k|\mu_{k-1}\right) P\left(\mu_0^{k-1}|\mathbf{y}_0^{k-1},\mathbf{u}_0^{k-2}\right), \end{aligned} \tag{12}$$

where c is a normalization constant. The computation of probability of the terminal model $P(\mu_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1})$ and the predictive conditional pdf $p(\mathbf{y}_{k+1}|\mathbf{y}_0^k,\mathbf{u}_0^k)$ is straightforward.

Unfortunately, as the number of model sequences exponentially increases with time, memory and computational demands become unmanageable. To overcome this problem several techniques based on pruning or merging of Gaussian sum have been proposed. A technique that merges model sequences with the same terminal sequence μ_{k-l}^k is used here. The probability of the terminal sequence of models μ_{k-l}^k is

$$P\left(\mu_{k-l}^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right) = \sum_{\mu_0^{k-l-1}} P\left(\mu_0^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right) \tag{13}$$

and the filtering density that has the form of a Gaussian sum

$$\begin{aligned} p\left(\mathbf{x}_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1},\mu_{k-l}^k\right) = \sum_{\mu_0^{k-l-1}} \frac{P\left(\mu_0^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right)}{P\left(\mu_{k-l}^k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1}\right)} \\ \times p\left(\mathbf{x}_k|\mathbf{y}_0^k,\mathbf{u}_0^{k-1},\mu_0^k\right) \end{aligned} \tag{14}$$

is replaced by a Gaussian distribution in such a way that the first two moments, i.e. mean value and covariance matrix, of the variable $\mathbf{x}_k$ remain unchanged.

5 FEASIBLE ALGORITHM BASED ON ROLLING HORIZON

Even if the state and output pdfs are known, the backward recursive relation (8) can not be solved analytically because of intractable integrals. A systematic approach to forward solution of the backward recursive relation (8) based on the stochastic approximation method is presented e.g. in (Bayard, 1991). A simple alternative approach is represented by the rolling horizon technique, where the optimization horizon is truncated and terminal cost-to-go of such truncated optimization horizon is replaced by zero. The length $F_o > 0$ of truncated horizon should be as short as possible to save computational demands but on other hand it has to preserve dependence of value of the minimized criterion on the input signal $\mathbf{u}_k$. In this paper the optimization horizon $F_o = 3$ will be considered to simplify computations. The cost-to-go function $V_{k+3}^*\left(\mathbf{y}_0^{k+3},\mathbf{u}_0^{k+2}\right)$ is replaced by zero value. Then the input $\mathbf{u}_{k+2}^a = 0$ and the cost-to-go function $V_{k+2}^a\left(\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right)$ is

$$\begin{aligned} V_{k+2}^a\left(\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right) = \\ \mathrm{E}\left\{\left[\mathbf{x}_{k+2}-\mathbf{r}_{k+2}\right]^T\mathbf{Q}_{k+2}\left[\mathbf{x}_{k+2}-\mathbf{r}_{k+2}\right]|\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right\}. \end{aligned} \tag{15}$$

The input $\mathbf{u}_{k+2}$ is zero because it can not influence the value of the criterion on the optimization horizon and the matrix $\mathbf{R}_{k+2}$ is positive definite. Note, that the value of the cost-to-go function $V^{\mathrm{a}}_{k+2}\left(\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right)$ can be computed analytically based on the first two moments of the state $\mathbf{x}_{k+2}$ given by the pdf $p\left(\mathbf{x}_{k+2}|\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right)$. The input $\mathbf{u}^{\mathrm{a}}_{k+1} = -\mathbf{W}^{-1}\mathbf{D}$ and the cost-to-go function at time step $k+1$ is

$$V^{\mathrm{a}}_{k+1}\left(\mathbf{y}_0^{k+1},\mathbf{u}_0^{k+1}\right) = \mathrm{E}\left\{\left[\mathbf{x}_{k+1}-\mathbf{r}_{k+1}\right]^T\mathbf{Q}_{k+1}\left[\mathbf{x}_{k+1}-\mathbf{r}_{k+1}\right]|\mathbf{y}_0^{k+2},\mathbf{u}_0^{k+1}\right\} + K - \mathbf{D}^T\mathbf{W}^{-1}\mathbf{D}, \quad (16)$$

where

$$\mathbf{W} = \mathbf{R}_{k+1} + \sum_{\mu_{k+1}}\mathbf{B}^T_{\mu_{k+1}}\mathbf{Q}_{k+2}\mathbf{B}_{\mu_{k+1}}P\left(\mu_{k+1}|\mathbf{y}_0^{k+1},\mathbf{u}_0^{k}\right), \quad (17)$$

$$\mathbf{D} = \sum_{\mu_{k+1}}\mathbf{B}^T_{\mu_{k+1}}\mathbf{Q}_{k+2}\left[\mathbf{A}_{\mu_{k+1}}\hat{\mathbf{x}}_{k+1}(\mu_{k+1})-\mathbf{r}_{k+2}\right]\times P\left(\mu_{k+1}|\mathbf{y}_0^{k+1},\mathbf{u}_0^{k}\right), \quad (18)$$

$$K = \sum_{\mu_{k+1}}\left\{\left[\mathbf{A}_{\mu_{k+1}}\hat{\mathbf{x}}_{k+1}(\mu_{k+1})-\mathbf{r}_{k+2}\right]^T\mathbf{Q}_{k+2}\times\left[\mathbf{A}_{\mu_{k+1}}\hat{\mathbf{x}}_{k+1}(\mu_{k+1})-\mathbf{r}_{k+2}\right] + \mathrm{Tr}\left(\mathbf{Q}_{k+2}(\mathbf{A}_{\mu_{k+1}}\mathbf{P}_{k+1}(\mu_{k+1})\mathbf{A}^T_{\mu_{k+1}}+\mathbf{G}_{\mu_{k+1}}\mathbf{G}^T_{\mu_{k+1}})\right)\right\}\times P\left(\mu_{k+1}|\mathbf{y}_0^{k+1},\mathbf{u}_0^{k}\right). \quad (19)$$

The mean $\hat{\mathbf{x}}_{k+1}(\mu_{k+1}) = \mathrm{E}\left\{\mathbf{x}_{k+1}|\mathbf{y}_0^{k+1},\mathbf{u}_0^{k},\mu_{k+1}\right\}$ and the corresponding covariance matrix $\mathbf{P}_{k+1}(\mu_{k+1})$ can be obtained from estimation algorithm. If the input $\mathbf{u}_{k+1}$ was used at time step $k+1$ the resulting controller would be cautious because it would respect uncertainty. The input at time step k is given as

$$\mathbf{u}^{\mathrm{a}}_k = \min_{\mathbf{u}_k\in\mathcal{U}_k}\mathrm{E}\left\{L^{\mathrm{c}}_k(\mathbf{x}_k,\mathbf{u}_k)+V^{\mathrm{a}}_{k+1}\left(\mathbf{y}_0^{k+1},\mathbf{u}_0^{k}\right)\mathbf{y}_0^{k},\mathbf{u}_0^{k}\right\}.$$

The expectation of the cost function $V^{\mathrm{a}}_{k+1}\left(\mathbf{y}_0^{k+1},\mathbf{u}_0^{k}\right)$ with respect to $\mathbf{y}_{k+1}$ seems to be computationally intractable. Therefore the expectation and subsequent minimization over discrete set $\mathcal{U}_k$ are performed numerically.

6 NUMERICAL EXAMPLE

The proposed fault-tolerant dual controller is compared with a cautious (CA) controller and a heuristic certainty equivalence (HCE) controller. The CA controller is obtained when just one-step look ahead policy is used and it takes uncertainties into account but lacks probing. The HCE controller is based on the assumption that the certainty equivalence principle holds even it is not true and inputs are determined as solutions to the problem where all uncertain quantities were fixed at some typical values.

Although the relative performance of three suboptimal controllers can differ in dependence on a particular system, the dual controller should outperform HCE and CA controllers in problems where uncertainty plays a major role. This numerical example illustrates a well known issue of pure CA controllers called 'turn-off' phenomenon, where the CA controller refuses to control a system because of large uncertainty. The initial uncertainty is quite high, but once it is reduced through measurements the problem becomes almost certainty equivalent. It is the reason why the HCE controller performs quite well in this particular example.

The quality of control is evaluated by M Monte Carlo runs. The value of the cost L for particular Monte Carlo simulation is denoted L_i and the value of the criterion J is estimated as $\hat{J} = 1/M\sum_{i=1}^{M}L_i$. Variability among Monte Carlo simulations is expressed by $\mathrm{var}\{L\} = 1/(M-1)\sum_{i=1}(L_i-\hat{J})^2$ and the quality of the criterion estimate $\hat{J}$ is expressed by $\mathrm{var}\{\hat{J}\}$ which is computed using bootstrap technique.

The detection horizon $F = 30$ is considered and the parameters of a single input single output scalar system are given in Table 1. The initial probabilities are $P(\mu_0 = 1) = P(\mu_0 = 2) = 0.5$, the transition probabilities are $P_{1,1} = P_{2,2} = 0.9$, $P_{1,2} = P_{2,1} = 0.1$, and parameters of Gaussian distribution are $\hat{x}'_0 = 1$ and $\mathbf{P}'_{x,0} = 0.01$. The discrete set of admissible values of input u_k is chosen to be $\mathcal{U}_k = \{-3,-2.9,\ldots,2.9,3\}$ for all $k\in\mathcal{T}$. The reference signal is the square wave with peaks of ± 0.4 and the period 13 steps and the weighting matrices in the cost function are chosen to be $Q_k = 1$ and $R_k = 0.001$ for all time steps.

Table 1: Parameters of the controlled system.

μ_k	$a_k(\mu_k)$	$b_k(\mu_k)$	$g_k(\mu_k)$	$c_k(\mu_k)$	$h_k(\mu_k)$
1	0.9	0.1	0.01	1	0.05
2	0.9	-0.098	0.01	1	0.05

An example of the typical state trajectories for all three controllers is given in Fig. 2. It can be seen that the CA controller does not control the system at the beginning of the control horizon at all. The criterion value estimates $\hat{J}$, the accuracies of these estimates $\mathrm{var}\{\hat{J}\}$, and the variability of Monte Carlo simulations $\mathrm{var}\{L\}$, that were computed using $M = 200$ Monte Carlo simulations, are given in Table 2. In

comparison with the CA controller, the quality of control is improved by 55% in the case of the HCE controller and by 68% in the case of the dual controller.

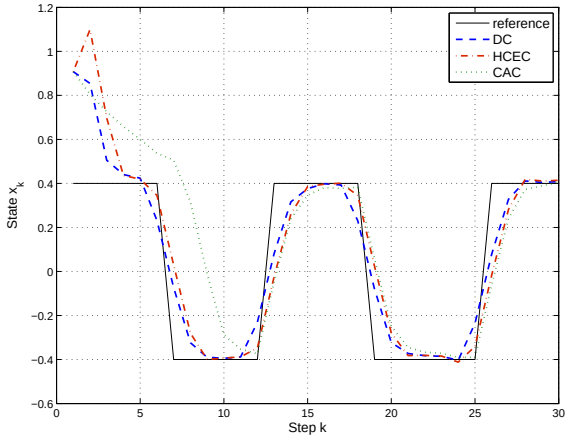


Figure 2: State trajectories for dual controller (DC), heuristic certainty equivalence controller (HCEC) and cautious controller (CAC).

Table 2: Criterion value estimates for particular controllers.

Controller	$\hat{J}$	var$\{\hat{J}\}$	var$\{L\}$
HCEC	3.2126	0.0164	3.2885
CAC	7.2186	0.0068	1.3194
DC	2.3131	0.0109	2.0889

7 CONCLUSIONS

The optimal fault-tolerant dual controller has been obtained as a special case of the general formulation. Since the optimal fault-tolerant controller is computationally infeasible the multimodel approach and rolling horizon techniques were used to obtain a suboptimal fault-tolerant dual controller. The performance of the proposed controller was compared with a heuristic certainty equivalence controller and cautious controller in a numerical example. Although all controllers were able to control the system even a fault occurred, the fault-tolerant dual controller exhibits the best performance.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic, project No. 1M0572, and by the Czech Science Foundation, project No. GA102/08/0442.

REFERENCES

Athans, M., Fekri, S., and Pascoal, A. (2006). Issues on robust adaptive feedback control. In *Proceedings of the* 16^{th} *IFAC World Congress*, Oxford, UK.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, New York, USA.

Bar-Shalom, Y. and Tse, E. (1974). Dual effects, certainty equivalence and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19:494–500.

Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes – Theory and application*. Prentice Hall, New Jersey, USA.

Bayard, D. S. (1991). A forward method for optimal stochastic nonlinear and adaptive control. *IEEE Transactions on Automatic Control*, 36(9):1046–1053.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control: Volume I*. Athena Scientific, Massachusetts, USA.

Blanke, M., Kinnaert, M., Staroswiecki, M., and Lunze, J. (2003). *Diagnosis and Fault-tolerant Control*. Springer Verlag, Berlin, Germany.

Campbell, S. L. and Nikoukhah, R. (2004). *Auxiliary signal design for failure detection*. Princeton University Press, New Jersey, USA.

Isermann, R. (2005). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer.

Kerestecioğlu, F. (1993). *Change Detection and Input Design in Dynamical Systems*. Research Studies Press, Taunton, England.

Mehra, R. K. (1974). Optimal input signals for parameter estimation in dynamic systems – Survey and new results. *IEEE Transactions on Automatic Control*, 19(6):753–768.

Šimandl, M. and Punčochář, I. (2009). Active fault detection and control: Unified formulation and optimal design. *Automatica*, 45(9):2052–2059.

Zhang, X. J. (1989). *Auxiliary Signal Design in Fault Detection and Diagnosis*. Springer-Verlag, Berlin, Germany.

ASYMPTOTIC ANALYSIS OF PHASE CONTROL SYSTEM FOR CLOCKS IN MULTIPROCESSOR ARRAYS

G. A. Leonov, S. M. Seledzhi
Saint-Petersburg State University, Universitetski pr. 28, Saint-Petersburg, 198504, Russia
leonov@math.spbu.ru

N. V. Kuznetsov, P. Neittaanmäki
University of Jyväskylä, P.O. Box 35 (Agora), FIN-40014, Finland

Keywords: Array processors, Clock generator, Phase-locked loop, Stability.

Abstract: New method for the rigorous mathematical analysis of electronic synchronization systems is suggested. This method allows to calculate the characteristics of phase detectors and carry out a rigorous mathematical analysis of transient process and stability of the system.

1 INTRODUCTION

In recent years, it has actively produced and used array processors systems, which face the problem of generation of synchronous signals and the mutual synchronization of processors.

In realizing parallel algorithms, the processors must perform a certain sequence of operations simultaneously. These operations are to be started at the moments of arrival of clock pulses at processors. Since the paths along which the pulses run from the clock to every processor are of different length, a mistiming in the work of processors arises. This phenomenon is called a clock skew.

The elimination of the clock skew is one of the most important problems in parallel computing and information processing (as well as in the design of array processors).

Several approaches to solving the problem of eliminating the clock skew have been devised for the last thirty years.

In developing the design of multiprocessor systems, a way was suggested Kung, 1988 for joining the processors in the form of an H-tree, in which (Fig. 1) the lengths of the paths from the clock to every processor are the same. However, in this case the clock skew is not eliminated completely because of heterogeneity of the wires (Kung, 1988). Moreover, for a great number of processors, the configuration of communication wires is very complicated. This leads to difficult technological problems.

Among the disadvantages we note the deceleration of performance of parallel algorithms. In addition to the problem of eliminating the clock skew, another important problem arose. The increase in the number of processors in multiprocessor systems required an increase in the power of the clock. But the powerful clock came to produce significant electromagnetic noise. Not so long ago a new method for eliminating the clock skew and reducing the generator's power was suggested. It consists of introducing a special distributed system of clocks controlled by phase-locked loops (Fig. 2). This approach enables one to reduce significantly the power of clocks.

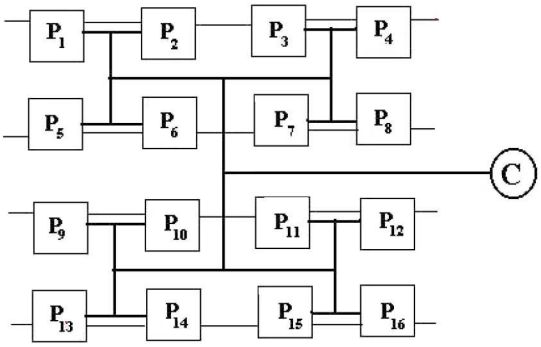


Figure 1: H-tree.

Phase-locked loops (PLLs) are widely used in telecommunication and computer architectures. They were invented in the 1930s-1940s (De Bellescize, 1932; Wendt & Fredentall, 1943) and then the theory and practice of PLLs were intensively studied (Viterbi, 1966; Lindsey, 1972; Gardner, 1979).

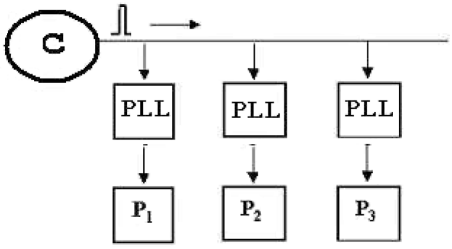


Figure 2: Distributed system of clocks controlled by PLLs.

Various methods for analysis of phase-locked loops are well developed by engineers and considered in many publications (see, e.g., (Best, 2003; Kroupa, 2003; Egan, 2007)) but the problems of construction of adequate nonlinear models and nonlinear analysis of such models are still far from being resolved and require using special methods of qualitative theory of differential, difference, integral, and integro-differential equations (Leonov et al., 1992; Leonov et al., 1996; Abramovitch, 2002; Margaris, 2004; Kudrewicz & Wasowicz, 2007; Kuznetsov, 2008; Leonov, 2006).

In this paper new method for the rigorous mathematical analysis of electronic synchronization systems is suggested. This method consists in considering a phase synchronization system on three levels:

1) a level of electronic realizations;

2) a level of phase and frequency relations between inputs and outputs in block diagrams;

3) a level of differential and integro-differential equations, and performing the asymptotic analysis of high-frequency periodic oscillations.

This method allows one to calculate the characteristics of phase detectors and make a rigorous mathematical analysis of transient and stability of the system (Leonov, 2006; Kuznetsov et al., 2008; Kuznetsov et al., 2009; Leonov et al., 2009).

2 ASYMPTOTIC ANALYSIS AND PHASE DETECTORS CHARACTERISTICS CALCULATION

Consider a differentiable 2π-periodic function $g(x)$, having two and only two extremums on $[0,2\pi]$: $g^- < g^+$, and the following properties.

For any number $\alpha \in (g^-, g^+)$ there exist two and only two roots of the equation $g(x) = -\alpha$:

$$0 < \beta_1(\alpha) < \beta_2(\alpha) < 2\pi.$$

Consider the function

$$F(\alpha) = 1 - \frac{\beta_2(\alpha) - \beta_1(\alpha)}{\pi}$$

if $g(x) < -\alpha$ on $(\beta_1(\alpha), \beta_2(\alpha))$ and the function

$$F(\alpha) = -\left(1 - \frac{\beta_2(\alpha) - \beta_1(\alpha)}{\pi}\right)$$

if $g(x) > -\alpha$ on $(\beta_1(\alpha), \beta_2(\alpha))$ and $a < b$, ω.

Suppose, ω is sufficiently large relative to the numbers a, b, α, π.

Lemma 1. The following relation

$$\int_a^b \text{sign}\,[\alpha + g(\omega t)]dt = F(\alpha)(b-a) + O(\frac{1}{\omega}) \quad (1)$$

is satisfied.

Lemma 1 results from the formula for definitions of $F(\alpha)$.

Consider now the propagation of pulse high-frequency oscillations through linear filter (Fig. 3)

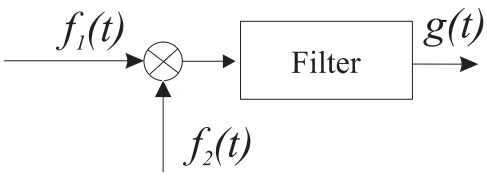


Figure 3: Multiplicator and filter.

Here

$$f_j(t) = A_j \text{sign}\,\sin(\omega_j(t)t + \psi_j), \quad (2)$$

$$g(t) = \alpha(t) + \int_0^t \gamma(t-\tau) f_1(\tau) f_2(\tau) d\tau, \quad (3)$$

$\otimes$ is a multiplicator, $A_j > 0$, ψ_j are certain constants, $j = 1, 2$, $\gamma(t)$ is a impulse response of linear filter and $\alpha(t)$ is an exponentially damped function, linearly depending on initial state of filter at moment $t = 0$.

A high-frequency property of generators can be reformulated as the following condition.

Consider a large fixed time interval $[0,T]$, which can be partitioned into small intervals of the form

$$[\tau, \tau + \delta],\ (\tau \in [0,T]),$$

where the following relations

$$|\gamma(t) - \gamma(\tau)| \le C\delta, \quad |\omega_j(t) - \omega_j(\tau)| \le C\delta, \quad \forall t \in [\tau, \tau+\delta],\ \forall \tau \in [0,T], \quad (4)$$

$$|\omega_1(\tau) - \omega_2(\tau)| \le C_1,\ \forall \tau \in [0,T], \quad (5)$$

$$\omega_j(\tau) \ge R,\ \forall \tau \in [0,T], \quad (6)$$

are satisfied.

We shall assume that δ is small enough relative to the fixed numbers T, C, C_1 and R is sufficiently large relative to the number $\delta : R^{-1} = O(\delta^2)$.

The latter means that on small intervals $[\tau, \tau+\delta]$ the functions $\gamma(t)$ and $\omega_j(t)$ are "almost constant" and the functions $f_j(t)$ on them are rapidly oscillating. Obviously, such a condition occurs for high-frequency oscillations.

Consider now 2π-periodic function $\varphi(\theta)$ of the form

$$\varphi(\theta) = A_1 A_2 \left(1 - \frac{2|\theta|}{\pi}\right), \quad \theta \in [-\pi, \pi] \qquad (7)$$

and a block-scheme in Fig. 4

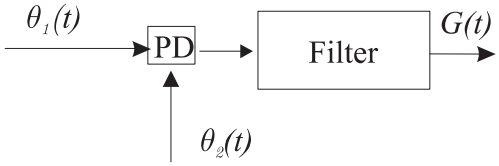


Figure 4: Phase detector and filter.

Here $\theta_j(t) = \omega_j(t)t + \psi_j$ are phases of the oscillations $f_j(t)$, PD is a nonlinear block with the characteristic $\varphi(\theta)$ (being called a phase detector or discriminator) with the output

$$G(t) = \alpha(t) + \int_0^t \gamma(t-\tau)\varphi(\theta_1(\tau) - \theta_2(\tau))d\tau. \qquad (8)$$

Theorem 1. *If conditions (4)–(6) are satisfied, then for the same initial states of filter we have*

$$|G(t) - g(t)| \le D\delta, \ \forall t \in [0, T]. \qquad (9)$$

Here D is a certain not depending on δ number.

Proof. It is readily seen that

$$g(t) - \alpha(t) = \int_0^t \gamma(t-s)A_1A_2 \operatorname{sign}[\cos((\omega_1(s) -$$
$$-\omega_2(s))s + \psi_1 - \psi_2) - \cos((\omega_1(s) + \omega_2(s))s +$$
$$+\psi_1 + \psi_2)]ds =$$
$$= A_1A_2 \sum_{k=0}^{m} \gamma(t-k\delta) \left[\int_{k\delta}^{(k+1)\delta} \operatorname{sign}[\cos((\omega_1(k\delta) - \right.$$
$$-\omega_2(k\delta))k\delta + \psi_1 - \psi_2) - cos((\omega_1(k\delta) + \omega_2(k\delta))s +$$
$$\left. +\psi_1 + \psi_2)]ds + O(\delta^2) \right], \ t \in [0, T].$$

Here the number m is such that

$$t \in [m\delta, (m+1)\delta].$$

By Lemma 1 this implies the estimate

$$g(t) = \alpha(t) + A_1A_2 \left(\sum_{k=0}^{m} \gamma(t-k\delta)\varphi(\theta_1(k\delta) - \right.$$
$$\left. - \theta_2(k\delta))\delta \right) + O(\delta) = G(t) + O(\delta).$$

This relation proves the assertion of Theorem 1.

Consider now a block-scheme of typical phase-locked loop [1–6] (Fig. 5)

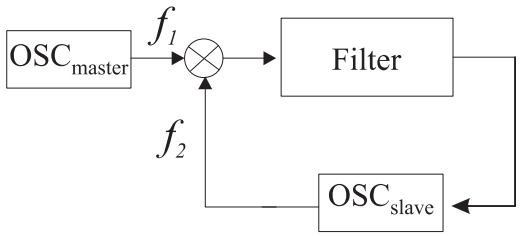


Figure 5: Phase-locked loop with multiplicator.

Here $\mathrm{OSC}_{\mathrm{master}}$ is a master oscillator, $\mathrm{OSC}_{\mathrm{slave}}$ is a slave (tunable) oscillator and block $\otimes$ is a multiplier of oscillations of $f_1(t)$ and $f_2(t)$.

From Theorem 1 it follows that for pulse generators, at the outputs of which there are produced signals (2), this block-scheme can be asymptotically changed (for high-frequency generators) to a block-scheme on the level of frequency and phase relations (Fig. 6)

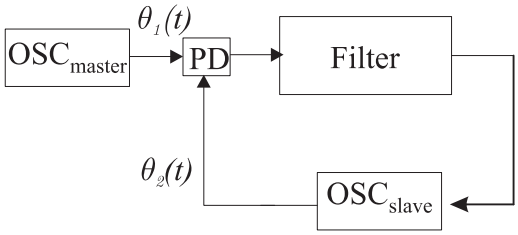


Figure 6: Phase-locked loop with phase detector.

Here PD is a phase detector with characteristic (7).

Thus, here on basis of asymptotical analysis of high-frequency pulse oscillations (Lemma 1 and Theorem 1) a characteristic of phase detector (7) is computed.

We give now a scheme for computing characteristics of phase detector for PLL with squarer. Consider a block-scheme in Fig. 1 with

$$f_1(t) = A_1^2(1 + \operatorname{sign}\sin(\omega_1(t)t + \psi_1))^2$$
$$f_2(t) = A_2 \operatorname{sign}\sin(\omega_2(t)t + \psi_2)$$

Consider then a block-scheme in Fig. 2, where PD is a block with characteristic $F(\theta) = 2A_1\varphi(\theta)$.

Theorem 2. *If conditions (4)–(6) are satisfied, then for the same initial states of filter the relation*

$$|G(t) - g(t)| \le D\delta, \quad t \in [0, T]$$

holds true. Here D is a certain independent of δ *number.*

Proof of Theorem 2 is similar to that of Theorem 1.

Finally it may be remarked that for modern processors a transient process time in PLL is less than or equal to 10 sec. and a frequency of clock oscillators attains 10Ghz . Given $\delta = 10^{-4}$ (i.e. partitioning each second into thousand time intervals), we obtain an expedient condition for the proposed here asymptotical computation of phase detectors characteristics:

$$\omega^{-1} = 10^{-10} = 10^{-2}(\delta^2) = O(\delta^2).$$

3 CONCLUSIONS

Thus consideration of phase synchronization system at three levels (electronic realizations; phase and frequency relations differential and integro-differential equations) make it possible to calculate the characteristics of the phase detector and perform rigorous mathematical analysis of the stability of the system.

ACKNOWLEDGEMENTS

This work was supported by projects of Federal Program "Scientific and scientific-pedagogical cadres Innovative Russia" in 2009 - 2013 years.

REFERENCES

Kung, S. (1988). *VLSI Array Processors*, Prentice Hall

Lindsey, W. (1972). *Synchronization systems in communication and control*, Prentice-Hall. New Jersey.

Viterbi, A. (1966). *Principles of coherent communications*, McGraw-Hill. New York.

Gardner, F. (2005). *Phase–lock techniques*, John Wiley & Sons, New York, 2ed.

Best Ronald, E. (2003). *Phase-Lock Loops: Design, Simulation and Application*, McGraw Hill, 5ed.

Egan, W. F. (2000). *Frequency Synthesis by Phase Lock*, (2nd ed.), John Wiley and Sons, 2ed.

Kroupa, V. (2003). *Phase Lock Loops and Frequency Synthesis*, John Wiley & Sons.

Leonov, G., Reitmann, V., Smirnova, V. (1992). *Nonlocal Methods for Pendulum-Like Feedback Systems*, Teubner Verlagsgesselschaft. Stuttgart; Leipzig.

Leonov, G., Ponomarenko, D., Smirnova, V. (1996). *Frequency-Domain Methods for Nonlinear Analysis. Theory and Applications*, World Scientific. Singapore.

Abramovitch, D. (2002). *Phase-Locked Loope A control Centric. Tutorial, Proceedings of the American Control Conference 2002*. vol. 1, pp. 1–15

Margaris, N. I. (2004). *Theory of the Non-Linear Analog Phase Locked Loop*, Springer Verlag

Kudrewicz, J. and Wasowicz S. (2007). *Equations of Phase-Locked Loops: Dynamics on theCircle, Torus and Cylinder*, World Scientific

Kuznetsov, N. V., (2008). *Stability and Oscillations of Dynamical Systems: Theory and Applications* Jyväskylä Univ. Printing House.

Leonov, G. A., (2006). *Phase synchronization. Theory and Applications* Automation and remote control, N 10, pp. 47–85. (survey)

Kuznetsov, N. V., Leonov, G.A., Seledzhi, S.M. (2008). *Phase Locked Loops Design And Analysis*, International Conference on Informatics in Control, Automation and Robotics, Madeira, Portugal, pp. 114–118.

Kuznetsov, N. V., Leonov, G.A., Seledzhi, S.M., Neittaanmaki, P. (2009). *Analysis and design of computer architecture circuits with controllable delay line*, Informatics in Control, Automation and Robotics, 2009, Proceedings, Vol. 3 - Signal processing, Systems Modeling and Control, pp. 222–224.

Leonov, G., Kuznetsov, N., Seledzhi S. (2010). *Nonlinear Analysis and Design of Phase-Locked Loops*, Chapter in the book *Automation and Control - Theory And Practice*, In-Tech, 2010 [in print]

A MINIMUM RELATIVE ENTROPY PRINCIPLE FOR ADAPTIVE CONTROL IN LINEAR QUADRATIC REGULATORS

Daniel A. Braun and Pedro A. Ortega
University of Cambridge, Dept. of Engineering, CB2 1PZ Cambridge, U.K.
dab54@cam.ac.uk, peortega@dcc.uchile.cl

Keywords: Minimum relative entropy principle, Adaptive control, Bayesian control rule, Linear quadratic regulator.

Abstract: The design of optimal adaptive controllers is usually based on heuristics, because solving Bellman's equations over information states is notoriously intractable. Approximate adaptive controllers often rely on the principle of certainty-equivalence where the control process deals with parameter point estimates as if they represented "true" parameter values. Here we present a stochastic control rule instead where controls are sampled from a posterior distribution over a set of probabilistic input-output models and the true model is identified by Bayesian inference. This allows reformulating the adaptive control problem as an inference and sampling problem derived from a minimum relative entropy principle. Importantly, inference and action sampling both work forward in time and hence such a Bayesian adaptive controller is applicable on-line. We demonstrate the improved performance that can be achieved by such an approach for linear quadratic regulator examples.

1 INTRODUCTION

Learning how to act in an unknown environment poses the problem of adaptive control (Åström and Wittenmark, 1995). Solving adaptive control problems optimally is a notoriously hard problem because it requires the solution of Bellman's optimality equations over large trees of information states, which becomes quickly intractable. Therefore, a number of approximate adaptive control methods have been devised in the literature (Åström and Wittenmark, 1995). Most heuristics for adaptive control are based on the certainty-equivalence principle, i.e. when they estimate the unknown plant parameters, the uncertainty of these estimates has no impact on the pertinent control strategies. Instead, a point estimate of the system parameters is treated as if it represented the "true" system parameters.

It is well known in optimal control theory that the certainty-equivalence principle holds exactly for linear quadratic systems with known dynamics (Åström and Wittenmark, 1995). In case of adaptive control, however, the certainty-equivalence principle breaks down in general and is only used as a heuristic. In fact, previous studies have shown that even for the linear quadratic controller correct closed-loop system identification cannot be guaranteed under certainty-equivalence, which has led to the proposal of cost-biased estimators (Campi and Kumar, 1996). Non-certainty-equivalent controllers are usually designed as extensions of a certainty-equivalent solution, such as *cautious* or *dual* controllers that reduce the control gain in the face of high parameter uncertainty or actively probe the environment by random excitation (Wittenmark, 1975). Here we propose a non-certainty equivalent approach to adaptive control based on a Bayesian control rule derived from a minimum relative entropy principle. We demonstrate how such an approach can be employed to solve adaptive control problems with linear dynamics and quadratic cost.

2 A BAYESIAN RULE FOR ADAPTIVE CONTROL

In the following we assume that the observations of our controller are given by a state variable x_t and the possible actions of our controller are u_t. The controller can then be defined as an input-output system that is characterized by the conditional probabilities

$$P(x_{t+1}|x_{\leq t}, u_{\leq t}) \quad \text{and} \quad P(u_{t+1}|x_{\leq t+1}, u_{\leq t})$$

where $x_{\leq t} = x_1, x_2, \ldots, x_t$ and $u_{\leq t} = u_1, u_2, \ldots, u_t$ denote concatenations of past states and actions respectively. Analogous to the controller, the plant can be thought of as an input-output system with conditional probabilities

$$Q(u_{t+1}|x_{\leq t+1}, u_{\leq t}) \quad \text{and} \quad Q(x_{t+1}|x_{\leq t}, u_{\leq t}).$$

If the controller can perfectly predict the plant for all histories $x_{\leq t}, u_{\leq t}$ then

$$P(x_{t+1}|x_{\leq t}, u_{\leq t}) = Q(x_{t+1}|x_{\leq t}, u_{\leq t}).$$

In this case the plant equation is perfectly known and the controller P can be tailored to the particular plant Q. Especially, the control law $P(u_{t+1}|x_{\leq t+1}, u_{\leq t})$ can be chosen in such a way that it maximizes some optimality criterion given full knowledge of the plant Q.

Consider now the case when the controller does not know the plant dynamics, but assume we know that the plant has dynamics Q_m drawn randomly from a set $\mathcal{Q}$ of possible dynamics indexed by m. Assume further we have available a set of tailored controllers P_m, where each P_m is tailor-made for one of the possible plants Q_m. The set of possible plant dynamics and tailored controllers can then be expressed as conditional probabilities given by the following likelihood and intervention models

$$P(x_{t+1}|m, x_{\leq t}, u_{\leq t}) \quad \text{and} \quad P(u_{t+1}|m, x_{\leq t+1}, u_{\leq t})$$

with $m \in \mathcal{M}$ indexing the different plant dynamics Q_m and the different tailored controllers P_m. How can we now construct a controller P such that its behavior is as close as possible to the tailored controller P_m under any realization of $Q_m \in \mathcal{Q}$?

A convenient measure of how much P deviates from P_m is given by the relative entropy. In particular, we can quantify the average deviation of a control law $P(u_{t+1}|x_{\leq t+1}, \bar{u}_{\leq t})$ from the tailored control law $P(u_{t+1}|m, x_{\leq t+1}, \bar{u}_{\leq t})$ of P_m by computing

$$\left\langle D_{KL}\left(P(u_{t+1}|m, x_{\leq t+1}, \bar{u}_{\leq t}) || P(u_{t+1}|x_{\leq t+1}, \bar{u}_{\leq t})\right)\right\rangle$$

where the average is taken with respect to a prior $P(m)$ and all possible input-output sequences with probabilities $P(x_{\leq t+1}\bar{u}_{\leq t}|m)$. The bar symbol $\bar{u}_{\leq t}$ indicates that past actions have been set by the controller and therefore have to be formalized as interventions (Pearl, 2000; Ortega and Braun, 2010). One can then show that the above quantity is minimized by the following control rule.

Theorem 1 (**Bayesian Control Rule**).

$$P(u_{t+1}|x_{\leq t+1}, \bar{u}_{\leq t}) = \sum_m P(u_{t+1}|m, x_{\leq t+1}, u_{\leq t}) P(m|x_{\leq t+1}, \bar{u}_{\leq t})$$

where $P(m|x_{\leq t+1}, \bar{u}_{\leq t})$ *is given by the recursive expression*

$$P(m|x_{\leq t+1}, \bar{u}_{\leq t}) = \frac{P(x_{t+1}|m, x_{\leq t}, u_{\leq t}) P(m|x_{\leq t}, \bar{u}_{<t})}{\sum_{m'} P(x_t|m', x_{\leq t}, u_{\leq t}) P(m'|x_{\leq t}, \bar{u}_{<t})} \quad (1)$$

The proof can be found in (Ortega and Braun, 2010). Here we apply the Bayesian control rule to adaptive control. It describes a mixture distribution over different tailored controllers indexed by m, each of them suggesting the next control signal u_{t+1} with probability $P(u_{t+1}|m, x_{\leq t+1}, u_{\leq t})$. The mixture weights are given by the posterior probability $P(m|x_{\leq t+1}, \bar{u}_{\leq t})$. It resembles Bayesian inference in that it starts out with a prior distribution over input-output models index by m and computes a posterior distribution after experiencing an interaction. Actions can then be sampled from this posterior distribution.

3 LINEAR QUADRATIC REGULATOR

A linear quadratic regulator is characterized by a linear dynamical system and a quadratic cost function. In the following we will deal with the time-discrete case. Formally, let $\mathbf{x}_t \in \mathbb{R}^N$ be the state vector of the plant at time t, $\mathbf{u}_t \in \mathbb{R}^M$ be the action of the controller, and $\mathbf{F} \in \mathbb{R}^{N\times N}$ and $\mathbf{G} \in \mathbb{R}^{N\times M}$ the time-invariant system matrices describing the dynamics of the plant such that

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{u}_t + \xi_t$$

where $\xi_t \in \mathbb{R}^N$ is a Gaussian random variable with known covariance matrix Ω_ξ. Furthermore, let c_t be the scalar instantaneous cost

$$c_t(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^T \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^T \mathbf{R}\mathbf{u}_t$$

where $\mathbf{R} \in \mathbb{R}^{M\times M}$ is positive definite and $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is positive semi-definite. Thus, the time-average cost J is given by

$$J(\mathbf{x}_t, \mathbf{u}_t) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} c_t(\mathbf{x}_t, \mathbf{u}_t).$$

If the matrices $\mathbf{F}, \mathbf{G}, \mathbf{Q}$ and $\mathbf{R}$ are all known the optimal controller has a well-known solution that is a simple state-feedback law

$$\mathbf{u}_t^* = -\mathbf{L}^* \mathbf{x}_t$$

where $\mathbf{L}^*$ can be computed from the algebraic Riccati equation (Stengel, 1993).

3.1 Indirect Adaptive Bayesian Control

In this section we will assume that we know the cost matrices $\mathbf{Q}$ and $\mathbf{R}$, but have to estimate $\mathbf{F}$ and $\mathbf{G}$ during the control process. Since we have to estimate them explicitly in order to compute the optimal policy

$\mathbf{L}^*$ this is often called *model-based* or *indirect* adaptive control. This means we have to deal with an inference problem—estimating $\mathbf{F}$ and $\mathbf{G}$—and an optimal control problem—generating control commands given the estimates $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$.

In order to solve the estimation problem we use an Unscented Kalman Filter (UKF) in our simulation experiments because it can estimate Gaussian random variables both under linear and nonlinear circumstances (Julier and Durrant-Whyte, 1995; Haykin, 2001). The parameter vector we want to estimate is given by the vectorized system matrices $\hat{\mathbf{w}} = \mathrm{vec}([\hat{\mathbf{F}}; \hat{\mathbf{G}}])$. Initially, we assume a Gaussian prior over $\hat{\mathbf{w}}_0$. We model the evolution of the parameter estimate as a Brownian diffusion process given by

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t + \omega_t \tag{2}$$

where $\omega \in \mathbb{R}^{N(N+M)}$ is a Gaussian random variable with covariance matrix Ω_ω. The covariance matrix determines the step size of the adaptation process. The likelihood model needed for the inference process is provided by

$$P(\mathbf{x}_{t+1}|\hat{\mathbf{w}}, \mathbf{x}_t, \mathbf{u}_t) \propto e^{-\frac{1}{2}\left(\mathbf{x}_{t+1}-\hat{\mathbf{F}}\mathbf{x}_t-\hat{\mathbf{G}}\mathbf{u}_t\right)^T \Omega_\xi^{-1}\left(\mathbf{x}_{t+1}-\hat{\mathbf{F}}\mathbf{x}_t-\hat{\mathbf{G}}\mathbf{u}_t\right)} \tag{3}$$

The adaptation rate Ω_ω can be adjusted dynamically depending on how well the current parameter estimates fit the observations. In case of poor predictions this should lead to high variability and fast adaptation in big steps, in case of very good predictions this should implies only small adaptation steps. This can be implemented using a Robbins-Monroe innovation update

$$\begin{aligned} \Omega_\omega^{(t+1)} &= (1-\alpha)\Omega_\omega^{(t)} + \alpha \mathbf{I}_t \\ \mathbf{I}_t &= \mathbf{K}_t^{\hat{\mathbf{w}}}\left[\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}\right]\left[\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}\right]^T (\mathbf{K}_t^{\hat{\mathbf{w}}})^T \end{aligned}$$

where $\mathbf{K}_t^{\hat{\mathbf{w}}}$ is the Kalman gain as used in the UKF and $\hat{\mathbf{x}}_{t+1}$ stems from the prediction step of the UKF (Haykin, 2001).

In order to solve the control problem we have to use the current estimate $\hat{\mathbf{w}} = \mathrm{vec}([\hat{\mathbf{F}}; \hat{\mathbf{G}}])$ to compute the optimal control commands. A certainty-equivalent self-tuning regulator would simply take the mean estimate $\mathbb{E}[\hat{\mathbf{w}}]$ and use this estimate in the algebraic Riccati equation at every point in time as if it was the true parameter vector. While this often works fine if only a few parameters of the matrix are unknown, in general this can lead to suboptimal solutions. Instead, we propose to use the Bayesian control rule as laid down in equation (1). This means we have to specify a likelihood and an intervention model. The likelihood model $P(\mathbf{x}_{t+1}|\hat{\mathbf{w}}, x_{\leq t}, u_{\leq t})$ is given by equation (3). The intervention model is deterministic and given by

$$P(\mathbf{u}_{t+1}|\hat{\mathbf{w}}, \mathbf{x}_{\leq t+1}, \mathbf{u}_{\leq t}) \propto \delta\left(\mathbf{u}_{t+1} + \mathbf{L}_{\hat{\mathbf{w}}}\mathbf{x}_{t+1}\right)$$

It might seem that this would imply taking the entire probability distribution over $\hat{\mathbf{w}}$ and propagating it through the Riccati equation. Then we would sample an $\mathbf{L}$ at each point in time to determine $\mathbf{u}_{t+1}$. Fortunately, an explicit computation of the posterior is not necessary. We can simply sample from the distribution over $\hat{\mathbf{w}}$, propagate this sampled value through the Riccati equation and obtain a sampled policy $\mathbf{L}$. The more precise the estimates over $\hat{\mathbf{w}}$ are going to be, the more precise the sampled policies $\mathbf{L}$ will get.

Example. In many motor control studies the hand is modeled as a point mass, where the state vector $\mathbf{x}_t$ comprises position and velocity in the plane (Todorov and Jordan, 2002). In a discrete state space this yields the following equation:

$$\mathbf{x}_{t+1} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_t + \begin{pmatrix} 0 & 0 \\ \Delta t/m & 0 \\ 0 & 0 \\ 0 & \Delta t/m \end{pmatrix} \mathbf{u}_t + \xi_t$$

where we chose ξ_t to be distributed according to

$$\xi_t \propto \mathcal{N}\left[\mathbf{0}, \sqrt{\Delta t}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}\right]$$

The noise ξ_t models uncertainty in the force production when controlling the mass point. In our simulation of a reaching task with unknown system dynamics the controller had to learn to bring the mass point from the periphery to the center of the coordinate system trying to find the optimal feedback gains. This requires estimating a 24-dimensional parameter vector $\mathbf{w}$ and sampling a 2×8-dimensional feedback gain. We chose the following parameter settings: $\Delta t = 0.01$, $m = 1$, $R = [[0.001, 0]; [0, 0.001]]$, $Q = [[1,0,0,0]; [0,0.01,0,0]; [0,0,1,0]; [0,0,0,0.01]]$ and $\alpha = 0.05$ for the UKF. The results can be seen in figure 1. The first entry of the parameter vector $\hat{\mathbf{w}}$ is depicted in figure 1a, the first entry of the correspondingly sampled $\mathbf{L}$ is depicted in figure 1b. After an initial exploration phase in which $\mathbf{L}$ is sampled from a broad distribution the controller settles down and only samples from a very narrow distribution centered at the optimal value. Figure 1c,d shows initial and final trajectories and speed profiles: initially amorphous, a straight-line movement is learned with a bell-shaped speed profile. Importantly, the Bayesian controller converges much faster to the correct feedback gain than the certainty-equivalent controller which never

fully reached the optimal value in our simulation—compare figure 1e,f. In the following table the mean absolute feedback gain error—the difference between optimal feedback gain and actually executed feedback gain—is shown averaged over the last 3000 time steps of 100 runs. We have also averaged over all 2×8 feedback gains.

	Abs. Error
Certainty Equivalent Controller	8.26 ± 0.01
Bayesian Control Rule	2.085 ± 0.002

The results show that the Bayesian control rule incurs approximately 4 times less error on average than the certainty-equivalent controller in this example. To ensure that this result does not depend on the particular system we chose we ran the same simulation but all the entries of the true F and G were drawn randomly from a uniform distribution $[0;1]$ in each run, with 100 runs in total. However, these random draws were "frozen" such that both controllers faced the same random variables and differences cannot be attributed to different random draws. Each run of this simulation had 500 time steps and we compared the feedback gain error in the last 100 time steps.

	Abs. Error
Certainty Equivalent Controller	0.536 ± 0.002
Bayesian Control Rule	0.111 ± 0.001

On average the Bayesian control rule incurred approximately 5 times less error than the certainty-equivalent controller.

3.2 Direct Adaptive Bayesian Control

The adaptive linear quadratic control problem can be reformulated in a way that does not require estimating the system matrices $\mathbf{F}$ and $\mathbf{G}$ explicitly (Bradtke, 1993). Instead we can work directly on the policy space and assign a Q *value* to each policy such that the Q value of policy $\mathbf{L}$ is given by

$$Q_{\mathbf{L}}(\mathbf{x}_t, \mathbf{u}_t) = c_t(\mathbf{x}_t, \mathbf{u}_t) + \big(\mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{u}_t\big)^T \mathbf{V}_{\mathbf{L}} \big(\mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{u}_t\big)$$

where $\mathbf{V}_{\mathbf{L}}$ corresponds to the cost-to-go function. Thus, $Q_{\mathbf{L}}(\mathbf{x}_t, \mathbf{u}_t)$ can be expressed as a quadratic form

$$\begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix}^T \underbrace{\begin{pmatrix} \mathbf{Q} + \mathbf{F}^T \mathbf{V}_{\mathbf{L}} \mathbf{F} & \mathbf{F}^T \mathbf{V}_{\mathbf{L}} \mathbf{G} \\ \mathbf{G}^T \mathbf{V}_{\mathbf{L}} \mathbf{F} & \mathbf{R} + \mathbf{G}^T \mathbf{V}_{\mathbf{L}} \mathbf{G} \end{pmatrix}}_{= \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix}$$

The matrix $\mathbf{M} \in \mathbb{R}^{(N+M)(N+M)}$ is positive definite and represents the Q value of policy $\mathbf{L}$. The relationship between $\mathbf{M}$ and $\mathbf{L}$ is given by

$$\mathbf{L} = -\mathbf{M}_{22}^{-1} \mathbf{M}_{21} \tag{4}$$

as can be readily seen when computing $\partial_{\mathbf{u}_t} Q_{\mathbf{L}}(\mathbf{x}_t, \mathbf{u}_t) = 0$. Previous studies have applied Q-learning to solve this *direct* adaptive control problem by reinforcement learning methods (Bradtke, 1993). Here we want to transform it into an inference problem. To this end, we need to relate $\mathbf{M}$ to an observable quantity in a way that is independent of the policy that is currently executed by the controller. We can achieve this by noting that Bellman's optimality equation imposes a recurrent relationship between consecutive Q values, namely

$$Q_{\mathbf{L}}(\mathbf{x}_t, \mathbf{u}_t) = c_t(\mathbf{x}_t, \mathbf{u}_t) + Q_{\mathbf{L}}(\mathbf{x}_{t+1}, -\mathbf{L}\mathbf{x}_{t+1}) \tag{5}$$

Since c_t is an observable quantity we can take it on one side of the equation and put all Q-quantities of equation (5) on the other side. Only the "true" Q-function can predict all c_t for all data points $\{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}\}$. Thus, we can use this relationship to do inference over $\mathbf{M}$ where

$$\begin{aligned} \hat{c}_t \;=\; & \begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix}^T \mathbf{M} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix} \\ & - \begin{pmatrix} \mathbf{x}_{t+1} \\ -\mathbf{M}_{22}^{-1}\mathbf{M}_{21}\mathbf{x}_{t+1} \end{pmatrix}^T \mathbf{M} \begin{pmatrix} \mathbf{x}_{t+1} \\ -\mathbf{M}_{22}^{-1}\mathbf{M}_{21}\mathbf{x}_{t+1} \end{pmatrix} \end{aligned}$$

Assuming Gaussian noise with known variance σ^2 for the cost observations we obtain the following likelihood model for our Bayesian controller:

$$P(c_t | \mathbf{M}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{u}_t) \propto \exp\left[-\frac{1}{2\sigma^2} \Big(\hat{c}_t - c_t \Big)^2 \right]$$

The intervention model is again deterministic:

$$P(\mathbf{u}_{t+1} | \mathbf{M}, \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{u}_t) \propto \delta\Big(\mathbf{u}_{t+1} + \mathbf{M}_{22}^{-1}\mathbf{M}_{21}\mathbf{x}_{t+1} \Big)$$

Doing inference over $\mathbf{M}$ is complicated by three facts: (i) the likelihood model is highly nonlinear in the parameters, (ii) $\mathbf{M}$ must be constrained to the set of positive definite matrices and (iii) $\mathbf{M}$ will be ill-conditioned in many examples because the different parts of the matrix differ usually by various orders of magnitude, as for example the unknown cost matrices $\mathbf{Q}$ and $\mathbf{R}$ are often of different orders of magnitude. Here we can only address problem (i) and (ii), i.e. the examples to demonstrate the Bayesian controller have to be well-conditioned—which is, for instance not true for the previous simulation. With regard to (i) we found that for this inference process the UKF only works robustly when the propagated means are simply computed as an un-weighted average over sigma points instead of the more common weighted average. With regard to (ii) we note that any positive definite matrix can be expressed as a product of its unique Cholesky factors: $\mathbf{M} = \mathbf{m}^T\mathbf{m}$ where $\mathbf{m}$ is upper triangular with diagonal elements strictly positive. Then we can do inference over $\mathbf{m}$ with the simpler

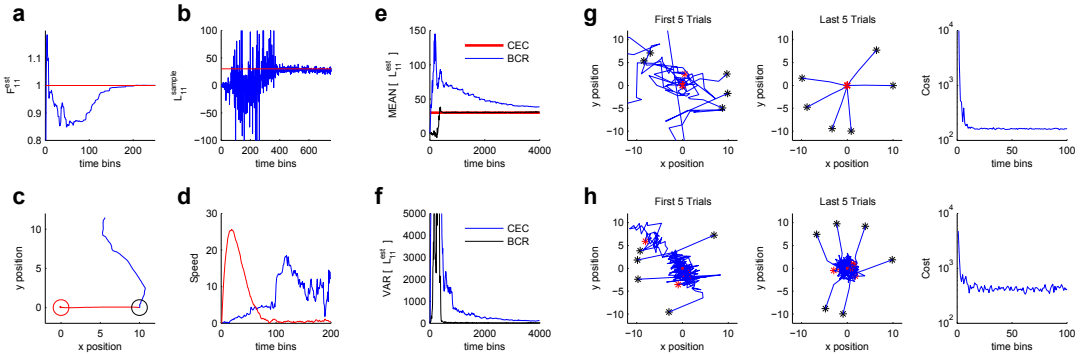


Figure 1: Results. (a-d) Learning to move a mass point when the system dynamics matrices **F** and **G** are unknown. A single run of the control process is shown. (a) Temporal evolution of estimate of the first entry of $\hat{\mathbf{F}}$ and the respective uncertainty as represented by the Kalman filter. (b) Sampled feedback gain—only the first entry of **L** is shown. The initial exploration phase is followed by a stable performance after 400 time steps. The thin red line indicates the optimal feedback gain. (c,d) Trajectories and speed profiles. Initially, the trajectory takes a random direction with an amorphous speed profile (blue curves). Later movement trajectories are straight and speed profiles bell-shaped (black curves). Panels (e-f) show sampled feedback gains over 100 runs. (e) Mean executed feedback gain. The certainty-equivalent controller (CEC) slowly converges to the region of optimal feedback gains. The exact optimal value was not reached in this simulation. The Bayesian control rule (BCR) converges very fast to the optimal feedback gain. (f) Variance of executed feedback gain. The Bayesian controller that used sampled feedback gains converges much faster than the certainty-equivalent controller. (g,h) Learning to move a mass-less point when both the system dynamics matrices **F** and **G** and the cost matrices **Q** and **R** are unknown. (g) Bayesian Control Rule. Trajectories of the first and last 5 trials. Initially, movements are undirected but later converge to straight line movements. The pertinent cost converges to the optimum. (h) Policy Iteration. Trajectories of the first and last 5 trials. The trajectories are wiggly because noise has to be added to the controller for exploration. Due to this extra noise the controller cannot converge to the optimal cost.

constraint that the diagonal elements must be positive. In our simulation we implemented this constraint by simply discarding any Kalman filter updates that would violate it. In general, such constraints can be easily implemented using particle filters.

Example. A simple well-conditioned example is a mass-less particle that moves around in the plane. The system dynamics can be formalized as:

$$\mathbf{x}_{t+1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x}_t + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}_t$$

The observations are noisy observations of the cost

$$c_t = \mathbf{x}_t^T \mathbf{Q} \mathbf{x}_t + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t + \xi_t$$

where ξ_t is a normally distributed scalar variable with variance $\sigma_{obs} = 0.1$. Both Q and R were assumed to be identity matrices and $\alpha = 0.5$ as previously. This is a 10-dimensional estimation problem. Figure 1g shows that the Bayesian controller managed to find the optimal control solution only relying on inference and sampling. We compared against a policy iteration algorithm for linear quadratic controllers as proposed in (Bradtke, 1993) – compare Figure 1h. In the latter exploration can only be achieved by adding extra noise to the control command. Note that the Bayesian control rule incurs this noise automatically by sampling from the posterior. We simulated 100 trials with 50 time steps each.

To ensure again that this result does not depend on the particular system we chose we ran another simulation where each entry of **F** and **G** were drawn from the uniform distribution $[0;1]$ and **Q** and **R** were drawn from an inverse Wishart distribution with identity covariance matrix and degree of freedom 2. The noise was again "frozen" for comparison between the two algorithms. We compared the absolute error between the optimal and the actually executed feedback gain over the last 20 trials. The Bayesian control rule outperformed the policy iteration algorithm roughly by factor 5.

	Abs. Error
Policy Iteration (Bradtke, 1993)	2.5 ± 0.1
Bayesian Control Rule	0.55 ± 0.01

4 CONCLUSIONS

In this paper we suggest a minimum relative entropy formulation of adaptive control problems when the plant dynamics are unknown but known to belong to a pre-defined set of possible dynamics. This formulation has an explicit solution given by the Bayesian control rule, a stochastic rule for adaptive control. We have presented two example classes that show how adaptive linear quadratic control problems can

be tackled using this problem formulation. Usually, adaptive controllers rely on the certainty equivalence principle and ignore parameter uncertainty in the control process (Åström and Wittenmark, 1995). In contrast, a controller based on the Bayesian control rule considers this uncertainty for balancing exploration and exploitation in a way that minimizes the expected relative entropy with regard to the true control law.

In particular, indirect control methods provide an interesting perspective here, because they allow solving the adaptive control problem purely based on inference and sampling methods that can be recruited from a rich arsenal in machine learning. Both inference and action sampling work forward in time and are therefore applicable online. Also they do not require different phases of policy evaluation and policy improvement as some of the previous reinforcement learning methods. Inference can be done online independent of the sampled policy. Several other studies have previously proposed to solve adaptive control problems based on inference methods (Toussaint et al., 2006; Engel et al., 2005; Haruno et al., 2001). Crucially, however, these studies have concentrated on the observation part of the learning problem with no principled solution for the action selection problem. Usually, exploration noise has to be introduced in an *ad hoc* fashion in order to avoid suboptimal performance. In contrast, the minimum relative entropy cost function naturally leads to stochastic policies.

The main contribution of this study is to illustrate how a relative entropy formulation can be applied to solve an adaptive control problem. This is done by deriving a stochastic controller based on the Bayesian control rule for the LQR problem with unknown system and cost matrices. Similar minimum relative entropy formulations have recently also been proposed to solve optimal control problems with known system dynamics (Todorov, 2009; Kappen et al., 2009). How these two approaches for adaptive and optimal control relate is an interesting question for future research. Also, the Bayesian control rule suggested here could in principle be employed to solve more general adaptive control problems with possibly nonlinear dynamics. However, finding optimal tailored controllers for complex sub-environments can in general be highly non-trivial. Therefore, finding inference and sampling methods that work for more general classes of adaptive control problems poses a future challenge.

REFERENCES

Åström, K. and Wittenmark, B. (1995). *Adaptive Control*. Prentice Hall, 2nd edition.

Bradtke, S. (1993). Reinforcement learning applied to linear quadratic control. *Advances in Neural Information Processing Systems 5*.

Campi, M. and Kumar, P. (1996). Optimal adaptive control of an lqg system. *Proc. 35th Conf. on Decision and Control*, pages 349–353.

Engel, Y., Mannor, S., and Meir, R. (2005). Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208.

Haruno, M., Wolpert, D., and Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220.

Haykin, S. (2001). *Kalman filtering and neural networks*. John Wiley and Sons.

Julier, S.J., U. J. and Durrant-Whyte, H. (1995). A new approach for filtering nonlinear systems. *Proc. Am. Control Conference*, pages 1628–1632.

Kappen, B., Gomez, V., and Opper, M. (2009). Optimal control as a graphical model inference problem. *arXiv:0901.0633*.

Ortega, P. and Braun, D. (2010). A bayesian rule for adaptive control based on causal interventions. In *Proceedings of the third conference on artificial general intelligence*, pages 121–126. Atlantis Press.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK.

Stengel, R. (1993). *Optimal control and estimation*. Dover Publications.

Todorov, E. (2009). Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences U.S.A.*, 106:11478–11483.

Todorov, E. . and Jordan, M. (2002). Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.*, 5:1226–1235.

Toussaint, M., Harmeling, S., and Storkey, A. (2006). Probabilistic inference for solving (po)mdps. Technical report, EDI-INF-RR-0934, University of Edinburgh, School of Informatics.

Wittenmark, B. (1975). Stochastic adaptive control methods: a survey. *International Journal of Control*, 21:705–730.

DESIGN OF A MULTIOBJECTIVE PREDICTIVE CONTROLLER FOR MULTIVARIABLE SYSTEMS

F. Ben Aicha, F. Bouani and M. Ksouri
Laboratory of Analysis and Control of Systems, National Engineering School of Tunis
BP 37, le Belvedere 1002, Tunis, Tunisia
{faten.benaicha, bouani.faouzi, mekkiksouri}@yahoo.fr

Keywords: Generalized predictive control, Multiobjective optimization, Multivariable systems, Decentralized control, Decouplers, Genetic algorithm.

Abstract: In this paper, a strategy for automatic tuning of decentralized predictive controller synthesis parameters based on multiobjective optimization for multivariable systems is proposed. This strategy integrates the genetic algorithm to generate the synthesis parameters (the prediction horizon, the control horizon and the cost weighting factor) making a compromise between closed loop performances (the overshoot, the variance of the control and the settling time). A simulation example is presented to illustrate the performance of this strategy in the on-line adjustment of generalized predictive control parameters.

1 INTRODUCTION

Processes with only one output being controlled by a single manipulated variable are classified as single-input single output (SISO) systems. Many processes, however, do not conform to such simple control configuration. These systems are known as multi-input multi-output (MIMO) or multivariable systems. As most of the multivariable systems present interactions, the interaction problem between control loops has long been recognised as an area for concern and many approaches to deal with this problem were proposed. The method used in this work is to design non-interacting or decoupling controllers to eliminate completely the effects of loop interactions. This is achieved via decouplers (Albertos and Sala, 2004). As a control technique**,** we have used the Generalized Predictive Control (GPC) which has achieved great success in practical applications in recent decades. This strategy of control requires the determination of synthesis parameters: prediction horizon, control horizon and cost weighting factor which give acceptable closed loop performances. But, there is not exact rules giving the values of required parameters. Some works deal with the automatic tuning of GPC such as (Ben Abdennour, Ksouri and Favier, 1998) in which, an on-line adjustment of GPC's synthesis parameters using the fuzzy logic is presented. But, this method does not give exact values of synthesis parameters but allows a fuzzy description of each parameter (small, average, big). On the other hand, in (Ben Abdennour, Ksouri and Favier, 1998) to determine the GPC parameters, each performance criterion is minimized without considering the others criteria, so the problem is considered as a single-objective one. In practice, the optimization problems are rarely single-objective; where from the interest of multiobjective optimization (MOO) based on the minimization of all performance criteria at every sample time. The MOO leads to a set of optimal solutions, i.e. the Pareto optimal solutions or the non dominated solutions (Collette and Siarry, 2002). In this context, many works such as (Popov, Farag and Werner, 2005), (Yang and Pedersen, 2006), (Bemporada and Muñoz de la Peñab, 2009) and (Muldera, Tiwari and Kothare, 2009) were interested in the synthesis of controllers based on multiobjective optimisation which has more and more interest. In this paper, we propose a new method allowing the on-line adjustment of synthesis parameters of predictive controller using the genetic algorithm and that for the multivariable systems. The performances' criteria to be simultaneously minimized are the settling time, the overshoot and the variance of the control. This paper is organized as follows. The problem is formulated in section two where the multivariable decoupling control and the predictive control principle are given. The proposed

method allowing the tuning of synthesis parameters and the design of the multiobjective predictive controller are described in section three. The obtained simulation results are presented in section four. Conclusions are given in the last section.

2 PROBLEM FORMULATION

2.1 Multivariable System Representation

We consider a multivariable linear system with m inputs $u_i(k)$: $i=1,\ldots,m$ and n outputs $y_j(k)$: $j=1,\ldots,n$. The system equation is given by:

$$Y(k) = G(z^{-1})U(k) \tag{1}$$

with: $U(k) = [u_1(k), u_2(k), \ldots, u_m(k)]^T$ is the control vector, $Y(k) = [y_1(k), y_2(k), \ldots, y_n(k)]^T$ is the output vector and $G(z^{-1})$ is the transfer function matrix having as dimension $m \times n$ given by:

$$G(z^{-1}) = \begin{pmatrix} g_{11}(z^{-1}) & \cdots & g_{1m}(z^{-1}) \\ \vdots & \ddots & \vdots \\ g_{n1}(z^{-1}) & \cdots & g_{nm}(z^{-1}) \end{pmatrix} \tag{2}$$

For the P canonical structure (Albertos and Sala, 2004)., in the case of a system with two inputs and two outputs, the outputs are related to the inputs according to:

$$y_1(k) = g_{11}(z^{-1})u_1(k) + g_{12}(z^{-1})u_2(k) \tag{3}$$

$$y_2(k) = g_{22}(z^{-1})u_2(k) + g_{21}(z^{-1})u_1(k) \tag{4}$$

2.2 Multivariable Decoupling Control

Generally, in the industry the distributed control is the most favorable and the most used thanks to its structure simplicity. During the decentralized control design for a two inputs two outputs (TITO) process, the input-output pairing is essential and determining for the obtained performances as well as for the stability of the system (Moaveni and Khaki-Sedigh, 2006). Several methods were proposed to solve the interaction problem (Bristol, 1966), (Khelassi, Wilson and Bendib, 2004). The method which will be applied in this work is the one using decouplers having as role to decompose a multivariable process into a series of independent single-loop sub-systems, and the multivariable process can be controlled using independent loop controllers. As well as the input-output representation of multivariable processes, different structures are possible, like P or V decouplers. Judging by the literature, the P-decoupler seems to be the most popular. In this work, we choose to use the decoupling network of Zalkind given in (Zalkind, 1967). The structure of the obtained decoupled process having as auxilliary inputs $v_1(k)$ and $v_2(k)$ is presented in the figure below.

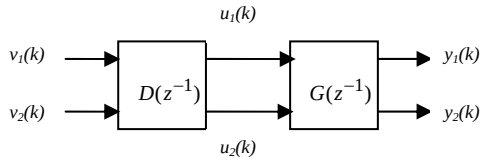


Figure 1: The structure of the decoupled process.

The control signals are given by:

$$u_1(k) = D_{11}(z^{-1})v_1(k) + D_{12}(z^{-1})v_2(k) \tag{5}$$

$$u_2(k) = D_{21}(z^{-1})v_1(k) + D_{22}(z^{-1})v_2(k) \tag{6}$$

where $D_{ij}(z^{-1})$, $i = 1,2$ *and* $j = 1,2$ are the elements of the transfer function $D(z^{-1})$.

In taking into account equations (3), (4), (5) and (6), we shall have:

$$\begin{aligned} y_1(k) = &\left[D_{11}(z^{-1})g_{11}(z^{-1}) + D_{21}(z^{-1})g_{12}(z^{-1}) \right] v_1(k) \\ &+ \left[D_{22}(z^{-1})g_{12}(z^{-1}) + D_{12}(z^{-1})g_{11}(z^{-1}) \right] v_2(k) \end{aligned} \tag{7}$$

$$\begin{aligned} y_2(k) = &\left[D_{11}(z^{-1})g_{21}(z^{-1}) + D_{21}(z^{-1})g_{22}(z^{-1}) \right] v_1(k) \\ &+ \left[D_{22}(z^{-1})g_{22}(z^{-1}) + D_{12}(z^{-1})g_{21}(z^{-1}) \right] v_2(k) \end{aligned} \tag{8}$$

To have $y_2(k)$ independent of $v_1(k)$ and $y_1(k)$ independent of $v_2(k)$, we introduce the decouplers between the process and the controller such as :

$$D_{12}(z) = \frac{-g_{12}(z)D_{22}(z)}{g_{11}(z)} \tag{9}$$

$$D_{21}(z) = \frac{-g_{21}(z)D_{11}(z)}{g_{22}(z)} \tag{10}$$

Generally we take $D_{11}(z)=1$ and $D_{22}(z)=1$ except in case the delays are more important in the direct branches than in the crossed branches (Albertos and Sala, 2004).
By using (9) and (10) in (7) and (8), we obtain:

$$y_1(k) = \left(g_{11}(z^{-1}) - \frac{g_{12}(z^{-1})\, g_{21}(z^{-1})}{g_{22}(z^{-1})} \right) v_1(k) \tag{11}$$

$$y_2(k) = \left(g_{22}(z^{-1}) - \frac{g_{12}(z^{-1})\, g_{21}(z^{-1})}{g_{11}(z^{-1})} \right) v_2(k) \qquad (12)$$

The use of (9) and (10) leads to the following control signals:

$$u_1(k) = \frac{-g_{12}(z^{-1})}{g_{11}(z^{-1})} v_2(k) + v_1(k) \qquad (13)$$

$$u_2(k) = \frac{-g_{21}(z^{-1})}{g_{22}(z^{-1})} v_1(k) + v_2(k) \qquad (14)$$

The ($m \times n$) multivariable process is treated as a set of n SISO processes. Each SISO process is characterized by a CARIMA (Controlled Auto Regressive Integrated Moving Average) dynamic model. This model is given by the following relation:

$$A(z^{-1})\, y(k) = z^{-d} B(z^{-1})\, \mathrm{v}(k-1) + \frac{C(z^{-1})}{\Delta(z^{-1})}\, e(k) \qquad (15)$$

where

- $y(k)$ and $v(k)$ are respectively the output and the input of the system.
- $e(k)$ is a sequence of white noise with zero mean average and a finite variance.
-The polynomials $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ and $\Delta(z^{-1})$ are given by:

$$A(z^{-1}) = 1 + a_1 z^{-1} + ... + a_{nA} z^{-nA} \qquad (16)$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + ... + b_{nB} z^{-nB} \qquad (17)$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + ... + c_{nC} z^{-nC} \qquad (18)$$

$$\Delta(z^{-1}) = 1 - z^{-1} \qquad (19)$$

-The roots in z of $C(z^{-1})$ must be strictly inside the unit circle.
- d represents the time delay of the system.

2.3 The GPC Optimal Control

The generalized predictive control is based on the minimization of a quadratic criterion given by the following expression (Richalet, Lavielle and Mallet, 2005), (Clarke, Mohtadi and Tuffs, 1987):

$$J_{GPC} = \sum_{j=1+d}^{Hp+d} (r_c(k+j) - \hat{y}(k+j/k))^2 + \rho \sum_{j=0}^{Hc-1} (\Delta v(k+j))^2 \qquad (20)$$

where H_p is the prediction horizon, H_c is the control horizon, ρ is the cost weighting factor, $r_c(k)$ is the set point, $\hat{y}(k+j/k)$ is the predicted output and $\Delta v(k+j)$ is the future increments of the control given by:

$$\Delta v(k+j) = v(k+j) - v(k+j-1) \qquad (21)$$

By minimizing the criterion J_{GPC}, we can determine the expression of the optimal vector $\Delta V(k) = \left[\Delta v(k), ..., \Delta v(k + H_c - 1) \right]^T$ as follows:

$$\Delta V(k) = K_{GPC} \left[R_c(k) - \left[\frac{1}{C(z^{-1})} [Gy(k) + R\Delta(z^{-1}) v(k-1)] \right] \right] \qquad (21)$$

where

$$K_{GPC} = [N_1^T N_1 + \rho I_{H_c}]^{-1} N_1^T \qquad (23)$$

$$R_c(k) = [r_c(k+1+d), ..., r_c(k+H_p+d)]^T \ . \qquad (24)$$

N_1 is a (H_p, H_c) matrix, G and R are obtained by the resolution of Diophantine equations (Clarke, Mohtadi, and Tuffs, 1987). The optimal control to be applied to the process is defined from the vector given by (22) using the receding horizon principle. This optimal control $v(k)$ is computed from the first element $\Delta v(1)$ of the vector $\Delta V(k)$:

$$v(k) = v(k-1) + \Delta v(1) \qquad (25)$$

It is evident that the optimal predictive control depends on synthesis parameters (H_p, H_c, ρ). So, in this paper, we present a new method allowing the automatic determination of required GPC's synthesis parameters in the case of multivariable systems.

3 MULTIOBJECTIVE GENERALIZED PREDICTIVE CONTROL

Multi-objective optimization (MOO) can be defined as the problem of finding a vector of parameters $X = \left[x_1, ..., x_l \right]^T$, which optimizes a vector of objective functions $(J_1, ..., J_n)$ (Gambier, 2008). In general, the MOO problem can be formulated as follows:

$$\min_X \ (J_1(X),\ J_2(X), ..., J_n(X)) \qquad (26)$$

At present, a very huge number of methods to solve MOO problems can be found in literature (Collette and Siarry, 2002), (Gambier, 2008). The method applied in this work is the weighted sum method that belongs to the family of aggregative methods.

3.1 Weighted Sum Method

This method allows the transformation of the objective functions vector in a single-objective function. It is known for its efficiency and suitability to generate a strongly non dominated solution that can be used as an initial solution for other techniques. The single criterion is obtained by the sum of the weighted criteria as follows (Gambier, 2008):

$$J = \sum_{i=1}^{n} w_i \; J_i \tag{27}$$

where the weights are chosen such that:

$$\sum_{i=1}^{n} w_i = 1 \;\; and \;\; 0 \le w_i \le 1 \tag{28}$$

The MOO leads to a set of solutions known as a Pareto set. This set is also called non-dominated solutions. When the non dominated solutions are collectively plotted in the criterion space, they constitute the Pareto front (Gambier, 2008). All points of the Pareto front are equally acceptable solution for the problem. However, it is necessary to obtain only one point in order to be able to implement the controller (Gambier, 2008).To choose one solution from the Pareto front, we can compute the following norm for each solution which gives a compromise between all criteria (Bouani, Laabidi, and Ksouri, 2006):

$$d_i = \sqrt{J_1^2 + J_2^2 + ... + J_n^2} \tag{29}$$

The quality of a control applied to a process is generally estimated by the closed loop performances of the system. Among these performances we choose as objective functions to optimize:

- The overshoot $D_{\%}$

$$D_{\%} = 100 \frac{|y_{\max} - r_c|}{r_c} \tag{30}$$

$y_{\max}$ is the maximum value of the output and r_c is the set point value.

- The variance of the control V_v

$$V_v = \frac{\sum_{N_1}^{N_2} v(k)^2}{N_2 - N_1} \tag{31}$$

N_1 is the first measure iteration and N_2 is the last one.

- The settling time T_s: It is the first instant after which, the system output doesn't exceed $\pm 5\%$ of the set point value.

So, to estimate the synthesis parameters for GPC, the following criterion will be minimized.

$$J = w_1 \; D_{\%} + w_2 \; V_v + w_3 \; T_s \;\; . \tag{32}$$

such that:

$$w_1 + w_2 + w_3 = 1 \text{ and } 0 \le w_i \le 1; \; i = 1,...,3 \; .$$

3.2 Generating Optimal Solutions Using Genetic Algorithms

In genetic algorithms, each parameter is represented by a string structure. This is similar to the chromosome structure in natural genes (Goldberg, 1991). A group of strings are called population. It should be notice that GAs evaluate a set of solutions in the population at each iteration step. Every solution is formed by GPC's synthesis parameters. A number of genetic operators (selection, crossover and mutation) are available to generate new individuals in next generation.

In this paper, we propose an on-line supervisor for each classic predictive controller based on genetic algorithms. In figure 2, we present the structure of this supervisor. Each supervisor permits the on-line adjustment of the GPC algorithm parameters in order to optimize simultaneously closed loop performances.

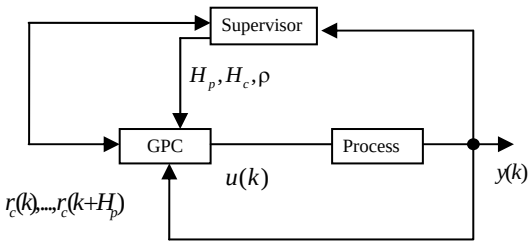


Figure 2: The Supervisor of the Classic Predictive Controller.

In our work, the GA population is formed by the synthesis parameters (H_p, H_c, ρ). The initial population is formed by arbitrary values, such as: $1 \le H_p \le 20$; $1 \le H_c \le 3$ and $0 < \rho \le 10$. For each individual of the population, we use the process model and the generalized predictive controller in order to compute, for a given set point, the output sequence along two hundreds sample times. Then, we evaluate the performance indices ($D_{\%}, V_v, T_s$)

and the fitness. To obtain the new population, we use the roulette wheel as a selection operator. To acquire more information in the new population, the crossover and the mutation operators are needed. This procedure will be repeated until a stop criterion (e.g. max number of generation) is reached. Then, we obtain the best individual (optimal values of H_p, H_c and ρ) that minimizes the performances indices. The steps used to compute the best synthesis parameters are given in algorithm 1. In this algorithm, we design by max_gen the maximum number of generations and by max_pop the maximum number of population.

Algorithm 1: The principal steps to design multi-objective predictive controller.

Form the initial population
For j=1 To max_gen
  For i=1 To max_pop
    - Take the ith individual of the population,
    - Use the GPC with the process model,
    - Compute the model output,
    - Evaluate the criteria: $D_{\%}, V_v, T_s$
    - Evaluate the fitness using (32)
  End
  Use the GA operators (selection, crossover and mutation) to form the new population.
End
Take the best individual (H_p, H_c, ρ).

Once the non dominated solutions are computed, the problem is which solution can be used with the GPC to handle the real process. To choose one solution from the Pareto front, we compute the following norm for each solution:

$$d_i = \sqrt{D_{\%}^{\,2} + V_v^{\,2} + T_s^{\,2}} \quad . \tag{33}$$

The steps allowing to find the synthesis parameters which minimize the performance criteria, given by the proposed algorithm is executed twice because the TITO system is decomposed into two monovariable systems controlled each by multiobjective predictive controller.

4 SIMULATION RESULTS

To estimate the closed loop performances obtained by applying the approach presented in this paper, we consider the TITO process given in (Miskovic, Karimi, Bonvin and Gevers, 2007) characterized by the next transfer functions matrix:

$$G(z^{-1}) = \begin{pmatrix} \dfrac{0.09516\ z^{-1}}{1-0.9048\ z^{-1}} & \dfrac{0.03807\ z^{-1}}{1-0.9048\ z^{-1}} \\ \dfrac{-0.02974\ z^{-1}}{1-0.9048\ z^{-1}} & \dfrac{0.04758\ z^{-1}}{1-0.9048\ z^{-1}} \end{pmatrix} \tag{34}$$

4.1 Generating Optimal Solutions

To apply the genetic algorithm, we choose a population of 20 individuals and a maximum number of generations equals to 150. The crossover probability and the mutation probability are fixed respectively to $c_p = 0.7$ and $m_p = 0.3$. We vary w_1 between 0 and 0.9, and w_2 and w_3 are computed by:

$$w_2 = w_3 = \frac{1-w_1}{2} \quad . \tag{35}$$

For every set of (w_1, w_2, w_3), the genetic algorithm evaluates the criterion given by (32) and generates the best individual (H_p, H_c, ρ).

In tables 1 and 2, we have, respectively reported the values of the best individuals corresponding to every set of weights for the first and the second SISO systems.

Table 1: The values of best individuals corresponding to every set of weights for the first SISO system.

	Weights			Best individuals		
i	w_1	w_2	w_3	H_p	H_c	ρ
1	0	0.5	0.5	2	1	5.75
2	0.1	0.45	0.45	3	2	6.71
3	0.2	0.4	0.4	3	2	6.71
4	0.3	0.35	0.35	2	2	7.98
5	0.4	0.3	0.3	3	2	6.72
6	0.5	0.25	0.25	3	2	6.77
7	0.6	0.2	0.2	3	1	9.40
8	0.7	0.15	0.15	2	2	9.99
9	0.8	0.1	0.1	3	1	9.42
10	0.9	0.05	0.05	2	2	5.62

Table 2: The values of best individuals corresponding to every set of weights for the second SISO system.

	Weights			Best individuals		
i	w_1	w_2	w_3	H_p	H_c	ρ
1	0	0.5	0.5	6	3	7.51
2	0.1	0.45	0.45	5	3	7.43
3	0.2	0.4	0.4	7	2	8.36
4	0.3	0.35	0.35	4	2	6.43
5	0.4	0.3	0.3	5	3	7.41
6	0.5	0.25	0.25	4	2	6.47
7	0.6	0.2	0.2	2	1	9.78
8	0.7	0.15	0.15	2	1	9.76
9	0.8	0.1	0.1	6	3	7.43
10	0.9	0.05	0.05	7	2	8.31

Figures 3 and 4, describe respectively the non dominated solutions which constitute the Pareto front for the first and the second SISO systems.

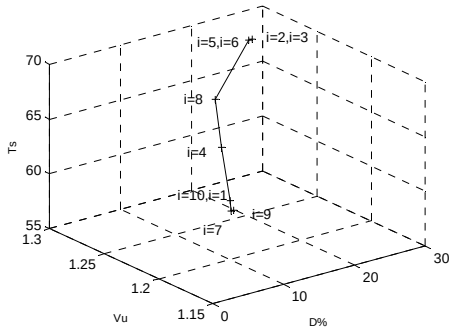


Figure 3: The Pareto front for the first SISO system.

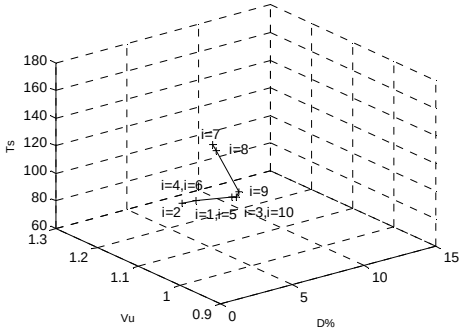


Figure 4: The Pareto front for the second SISO system.

4.2 Multiobjective Predictive Controller

To implement the controller, it is necessary to choose a single solution among all non dominated solutions. This choice is made by the user, if he decides to give the priority to the minimization of overshoot, he will choose the solution giving the overshoot minimum value. If the most important criterion to be minimized for the user is the settling time, he will choose the solution giving the minimum settling time. In this paper we choose to make a compromise between the three closed loop performances. For that, the step to be followed is to calculate the norm given by (33) for every set of w_i and to choose the synthesis parameters corresponding to the smallest value of d_i .

For the first SISO system, the synthesis parameters giving a minimal value of the norm d_i are given in Table 3. For the second SISO system the synthesis parameters chosen by the supervisor are presented in table 4. So we can notice that this proposed method allows automatic adjusting of synthesis parameters.

Table 3: The Synthesis Parameters Chosen by the Supervisor for the first SISO system.

H_{p1}	H_{c1}	ρ_1
2	2	7.98

Table 4: The Synthesis Parameters Chosen by the Supervisor for the second SISO system.

H_{p2}	H_{c2}	ρ_2
5	3	7.43

The obtained synthesis parameters, given in Table 3 and Table 4 are used with the two predictive controllers to control the multivariable process.
The obtained results are shown in Figure 5 and Figure 6 which respectively present the evolution of the system outputs and the set points and the evolution of the control signals. From these figures, we can notice that this proposed method allows automatic adjusting of synthesis parameters permitting a compromise between closed loop performances.

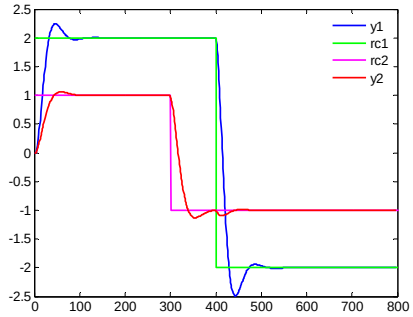


Figure 5: Evolution of the outputs and the set points.

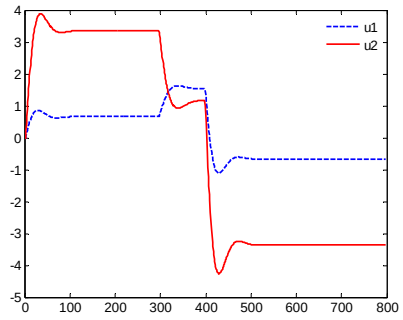


Figure 6: Evolution of the control signals.

The tables 5 and 6 recapitulate respectively the overshoots, the settling times values and the variances of the controls found for the first and the second SISO system.

Table 5: Closed loop performances Values Obtained for the First SISO System.

	Overshoot ($D_{\%}$)	Settling time (T_s)	Variance of the control (V_v)
k ∈ [0;400]	12	64s	0.69
k ∈ [401;800]	24	66s	

Table 6: Closed loop Performances values for the Second SISO System.

	Overshoot ($D_{\%}$)	Settling time (T_s)	Variance of the control (V_v)
k ∈ [0;300]	05.8	71s	10.06
k ∈ [301;800]	12.8	77s	

5 CONCLUSIONS

In this paper, a new method allowing the on line adjustment of the predictive controller synthesis parameters for multivariable systems has been presented. The decentralized control using the decoupling network is applied to decouple the different subsystems and to control the MIMO system using multiple SISO controllers. Genetic algorithms and the weighted sum method are exploited to find the synthesis parameters by minimizing simultaneously three criteria which are the overshoot, the settling time and the variance of the control. The obtained simulation results have shown that the proposed method can lead to acceptable closed loop performances.

REFERENCES

Albertos, P., Sala, A., 2004. *Multivariable control systems: an engineering approach*, Springer.

Bemporada, A., Muñoz de la Peñab, D., 2009. *Multiobjective model predictive control*, Automatica, vol. 45, issue 12, p. 2823-2830.

Ben Abdennour, R., Ksouri, M., Favier, G., 1998. *Application of fuzzy logic to the on-line adjustement of the parameters of a generalized predictive controller.* Intelligent Automation and soft computing, vol.4, No 3, p.197-214.

Bouani, F., Laabidi, K., Ksouri, M., 2006. *Constrained Nonlinear Multi-objective Predictive Control,* IMACS Multiconference on "Computational Engineering in Systems Applications"(CESA), Beijing, China, p. 1558-1565.

Bristol, E.H., 1966. On a new measure of interaction for multivariable process control. IEEE Transactions on control, p133-134.

Clarke, W., Mohtadi, C., Tuffs, P. S., 1987. *Generalised Predictive Control- parts I & II.* Automatica, vol.23, N° 2, p. 137-160.

Collette, Y., Siarry, P., 2002. *Optimisation multiobjectif*, Editions Eyrolles.

Gambier, A., 2008. *MPC and PID Control Based on Multi-Objective Optimization*. American Control Conference, Washington, USA, p. 4727-4732.

Goldberg, D. E., 1991. *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley, Massachusetts.

Khelassi, A., Wilson, J.A., Bendib,R., 2004. *Assessment of Interaction in Process*, Control Systems Dynamical Systems and Applications Proceedings, Antalya, Turkey, p. 463-471.

Miskovic, L., Karimi, A., Bonvin, D., Gevers, M., 2007. *Correlation-based tuning of decoupling multivariable controllers*, Automatica, vol. 43, p. 1481-1494.

Moaveni, C., Khaki-Sedigh, A., 2006. *Input-Output Pairing based on Cross-Gramian Matrix*, SICE-ICASE International joint conference, Korea, p. 2378-2380.

Muldera, E. F., Tiwari, P. Y., Kothare, M. V., 2009. *Simultaneous linear and anti-windup controller synthesis using multiobjective convex optimization*, Automatica, vol.45, issue 3, p.805-811.

Popov, A., Farag, A., Werner, H., 2005. *Tuning of a PID controller Using a Multi-objective Optimization Technique Applied to a Neutralization Plant*, 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain, p. 7139-7143.

Richalet, J., Lavielle, G., Mallet, J., 2005. *La commande prédictive : mise en œuvre et applications industrielles*, Editions Eyrolles.

Yang, Z., Pedersen, G., 2006. *Automatic Tuning of PID Controller for a 1-D Levitation System Using a Genetic Algorithm - A Real Case Study*, IEEE International Symposium on Intelligent Control, Munich, Germany, p. 3098-3103.

Zalkind, C.S., 1967. *Practical approach to non-interacting control parts I and II*. Instruments and control systems, vol.40, No.3 and No.4.

EFFICIENT IMPLEMENTATION OF CONSTRAINED ROBUST MODEL PREDICTIVE CONTROL USING A STATE SPACE MODEL

Amira Kheriji, Faouzi Bouani and Mekki Ksouri
Laboratory of Analysis and Control of Systems, National Engineering School of Tunis, B.P 37, 1002 Tunis, Tunisia
amirakheriji@gmail.com, {bouani.faouzi, mekkiksouri}@yahoo.fr

Keywords: Predictive control, Parametric uncertainty, State space model, Generalized geometric programming, Constrained control, Set-point tracking, Disturbance rejection.

Abstract: The goal of this paper is to evaluate the closed loop performances of a new approach in constrained state space Robust Model Predictive Control (RMPC) in the presence of parametric uncertainties. The control law is obtained by the resolution of a min-max optimization problem, initially non convex, under input and input deviation constraints, using worst case strategy. The technique used is the Generalized Geometric Programming (GGP) which is a global optimization method for non convex functions constrained in a specific domain. The key idea of the proposed approach is the convexification of the optimization problem allowing to compute the optimal control law using standard optimization technique. The proposed method is efficient since it guarantees set-point tracking different from the origin and non zero disturbances rejection. The efficiency of this approach is illustrated with two examples and compared with a recent state space RMPC algorithm.

1 INTRODUCTION

The MPC algorithms present a series of selling points over other methods amongst which stand out: its ability to handle non linear systems, multi input mutlti output systems as well as systems having input and/or state constraints. The model quality plays a vital role in MPC, but in reality there always exist model uncertainties, which may significantly degrade the system performances (Fukushima et al., 2007). Uncertainties can be represented in different forms reflecting in certain ways the knowledge of the physical mechanisms which cause the discrepancy between the model and the process (Camacho and Bordons, 2004). To describe the dynamic of the system, structured uncertainty was used by several Robust MPC (RMPC) works. A number of RMPC methods have been developed to cope with the presence of the uncertainties in the system model. A representative list of RMPC methods includes: (Campo and Morari, 1987), (Cordon and Boucher, 1994), (Kothare et al., 1996), (Rossiter and Kouvaritakis, 1998), (Huaizhong et al., 1998), (Lee and Kouvaritakis, 2000), (Ramirez et al., 2002), (Pannochia, 2004), (Fukushima et al., 2007), (Alamo et al., 2004), (Bouzouita et al., 2007), (Mayne et al., 2009), (Qian et al., 2010).

Most existing state-space RMPC algorithms are unable to control uncertain systems when the set-point is different from the origin or when it is changed such as LMI method introduced by (Kothare et al., 1996). Another limitation of this method consists on returning local optimum in some cases.

In the present work, we evaluate the closed loop performances of the proposed state space RMPC approach. This approach uses the state space output deviation method presented by (Watanabet et al., 1991) to compute the j step ahead output predictor with a finite prediction horizon since this method gives robust adaptive controlled results against the unknown plant parameters. Thus, the optimal control actions are determined by a min-max optimization problem. However, the criterion to be optimized is initially non convex relatively to the uncertain parameters and the control action. Hence, it can't be solved by a standard optimization technique. To overcome this difficulty, the GGP method, which is a global optimization technique, is adopted to convexify the criterion by means of variable transformations.
The main features of the proposed algorithm are:

- guarantee non zero set-point tracking,
- move the system with time-varying model uncertainty form set-point to another without offset,
- satisfy process constraints,
- reject non zeros disturbances,

- the on-line optimization algorithm is computed with a reasonable amount of time.

The efficiency of this algorithm is illustrated through two examples and compared with the method proposed by (Pannochia, 2004).

2 GENERALIZED PREDICTIVE CONTROL ALGORITHM

In this section, we will be based on the output deviation method introduced by (Watanabet et al., 1991) to compute the j step ahead output predictor value as well as the cost function. It is already proved that this method gives robust adaptive controlled result against the unknown plant parameters compared with the direct output method. The model considered at first for uncertain system is a linear discrete time single-input/single-output described by the following CARIMA model of the plant results performing an effective integral action:

$$A(q^{-1})\Delta y(k) = B(q^{-1})\Delta u(k) \tag{1}$$

where: - $\Delta y(k)$ and $\Delta u(k)$ are respectively the output and the input deviation system.

- Δ is the integral action which ensures offset-free steady-state response in the presence of variable set point.

- $A(q^{-1})$, $B(q^{-1})$ and $\Delta(q^{-1})$ are polynomials on q^{-1} with bounded coefficients:

$$A(q^{-1}) = 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a} \tag{2}$$

$$a_i \in [\underline{a_i}, \overline{a_i}], 1 \leq i \leq n_a$$

$$B(q^{-1}) = b_0 q^{-1} + b_1 q^{-2} + ... + b_{n_b} q^{-(n_b+1)} \tag{3}$$

$$b_j \in [\underline{b_j}, \overline{b_j}], 0 \leq j \leq n_b$$

$$\Delta(q^{-1}) = 1 - q^{-1} \tag{4}$$

Then, equation 1 can be transformed using the observer canonical form into a state space model as follows:

$$\Delta x(k+1) = F\Delta x(k) + G\Delta u(k) \tag{5a}$$

$$\Delta y(k) = H\Delta x(k) \tag{5b}$$

where $\Delta x(k)$ is an n_a dimensional vector and F, G and H are represented by the following matrices:

$$F = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -a_{n_a-1} & 0 & 0 & \cdots & 1 \\ -a_{n_a} & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n_a-2} \\ b_{n_a-1} \end{bmatrix} \tag{6a}$$

$$H = \overbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}}^{n_a} \tag{6b}$$

where $b_i = 0$ for $i > n_b$. Consequently, we can obtain using equation 5 the following state deviation at $k+j$ time:

$$\Delta x(k+j|k) = F^j \Delta x(k) + \sum_{i=1}^{j} F^{j-i} G\Delta u(k+i-1) \tag{7}$$

Then, it follows from equations 5 and 7, that the j-step ahead output predicted value is given by:

$$\begin{aligned} y(k+j|k) &= y(k) + \sum_{i=1}^{j} HF^i \Delta x(k) \\ &+ \sum_{i=1}^{j} \sum_{l=0}^{j-i} HF^l G\Delta u(k+i-1) \end{aligned} \tag{8}$$

Moreover, the cost function is defined by the following equation:

$$J = \sum_{i=1}^{H_p} (y(k+i|k) - w(k+i))^2 + \lambda \sum_{i=1}^{H_c} \Delta u(k+i-1)^2 \tag{9}$$

The output sequence on H_p prediction horizon can be written as follows:

$$Y = L_u \Delta U + f \tag{10}$$

where:

$$Y = [y(k+1|k), y(k+2|k), \ldots, y(k+H_p|k)]^T$$

$$\Delta U = [\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+H_c-1)]^T$$

The L_u with the (H_p, H_c) dimension and f which is an (H_p) dimensional vector are given by:

$$L_u = \begin{pmatrix} HG & 0 & \ldots & & 0 \\ HG+HFG & HG & 0 & \ldots & 0 \\ HG+HFG+HF^2G & HG+HFG & \ddots & & 0 \\ & & & & HG \\ \vdots & & \ddots & & \\ \sum_{j=1}^{H_c+1} HF^{j-1}G & & \ldots & & HG+HFG \end{pmatrix}$$

$$f = \begin{pmatrix} y(k) \\ y(k) \\ y(k) \\ \vdots \\ y(k) \end{pmatrix} + \begin{pmatrix} HF \\ HF + HF^2 \\ HF + HF^2 + HF^3 \\ \vdots \\ \sum_{i=1}^{H_p} HF^i \end{pmatrix} \Delta x(k)$$

Hence, the cost function of equation 9 is equivalent to:

$$J = (Y - W)^T (Y - W) + \lambda \Delta U^T \Delta U \qquad (11)$$

where Y is given by equation 10, λ is the weighting factor and W is the sequence of set-points on H_p prediction horizon:

$$W = [w(k+1), \ldots, w(k+H_p)]^T$$

3 PROBLEM STATEMENT

The strategy used to find the optimal control law is the minimization of the worst case objective function. The min-max problem is the following:

$$\min_{\Delta U(k) \in M} \max_{\substack{a_i \in [\underline{a_i}, \overline{a_i}] \\ b_j \in [\underline{b_j}, \overline{b_j}]}} J(\Delta U, a_i, b_j) \qquad (12)$$

where J is given by equation 11 and the set M represents the set of constraints on input and input deviation signals which can be described by: $M = \{\forall \Delta U : C\Delta U \leq D\}$ (Ramirez et al., 2002).

The maximization is over the bounds of A and B polynomial coefficients. This maximization would lead to a worst case value of J over all the values of a_i and b_j belonging respectively to $[\underline{a_i}, \overline{a_i}]$ and $[\underline{b_j}, \overline{b_j}]$ (Bouzouita et al., 2007). Therefore, it is deduced from equations 10 and 11, that the objective function J is non convex relatively to F, G and ΔU (see section 5 for more details). Hence, it is non convex relatively to the uncertain parameters a_i and b_j. Effective algorithm is proposed in the present paper to solve this maximization problem and obtain the global optimality within a good precision. The main idea of the GGP is to convexify the objective function and the constraints by applying different variable transformation techniques. Furthermore, this worst case value is minimized over present and future control moves $\triangle U = [\triangle u(k), ..., \triangle u(k + H_c - 1)]$. We present now the global optimization method (GGP) which allows us to solve the maximization problem of equation 12. This optimization problem can be converted to the given one:

$$\min_{\substack{a_i \in [\underline{a_i}, \overline{a_i}] \\ b_j \in [\underline{b_j}, \overline{b_j}]}} -J(\Delta U, a_i, b_j) \qquad (13)$$

Generalized geometric programming is an optimization technique for solving a class of non convex non linear programming problems (Tsai et al., 2007). The GGP problems occur frequently in engineering design, chemical process industry and management (Tsai, 2009), (Nand, 1995), (Chul and Dennis, 1996), (Maranas and Floudas, 1997) and (Porn et al., 2007). This class concerns the optimization problems with the objective function and constraints are in polynomial forms. Several specialized approaches have been proposed to locate the global optimum based mainly on variable transformations. Hence, the strategy of this technique is to replace all non convex signomials of the objective function with specific features into convex terms according to some specific transformation rules which will be formulated in next section.

4 CONVEXIFICATION STRATEGY OF THE GGP APPROACH

The mathematical formulation of a GGP problem is expressed as follows (Tsai, 2009):

$$\min_X Z(X) = \sum_{j=1}^{T_0} c_j z_j \qquad (14)$$

subject to:

$$\sum_{q=1}^{T_k} h_{k_q} z_{k_q} \leq l_k, k = 1, \ldots, K \qquad (15a)$$

$$z_p = x_1^{\alpha_{p_1}} x_2^{\alpha_{p_2}} \ldots x_n^{\alpha_{p_n}}, p = 1, \ldots, T_0, \qquad (15b)$$

$$z_{k_q} = x_1^{\beta_{k_{q_1}}} x_2^{\beta_{k_{q_2}}} \ldots x_n^{\beta_{k_{q_n}}}, k = 1, \ldots, K, q = 1, \ldots, T_k \qquad (15c)$$

$$X = (x_1, \ldots, x_n) \qquad (15d)$$

$$x_i > 0 \text{ for } 1 \leq i \leq n \qquad (15e)$$

$$\underline{x_i} \leq x_i \leq \overline{x_i} \qquad (15f)$$

Following the GGP formulation, the proposed method can be solved with only positive variables due to the logarithmic/exponential transformation used in the convexification strategy. Therefore, this transformation requires to replace x_i by e^{y_i}. Hence, x_i must be strictly positive. However, in several problems the polynomial variables can be negative. To overcame this limitation, a simple variable translation allows taking into account negative variables. Consequently, following the negative translation variable technique, the definition set of the polynomial variable of the objective function of equation 14 is $\Re_+^n$. Using equations 14 and 15 , the polynomial can be written as follows:

$$\min \sum_{j=1}^{T_0} c_j x_1^{\alpha_{p_1}} x_2^{\alpha_{p_2}} \ldots x_n^{\alpha_{p_n}}, p = 1, \ldots, T_0 \qquad (16)$$

In fact, the signomial function Z(X) is a sum of monomial terms $f_j(X)$ given by the following equation:

$$f_j(X) = c_j x_1^{\alpha_{p_1}} x_2^{\alpha_{p_2}} \ldots x_n^{\alpha_{p_n}}, j = 1, \ldots, T_0 \qquad (17)$$

Based on the given three propositions (Tsai et al., 2007), we can judge either each monomial term of the polynomial is convex or not.

Proposition 1. The function $f(X) = c\Pi_{i=1}^{n} x_i^{\alpha_i}$ is convex in $\Re_+^n$ if $c \geq 0$, $x_i \geq 0$ and $\alpha_{p_i} \leq 0$ (for all $i = 1, \ldots, n$).

Proposition 2. The function $f(X) = c\Pi_{i=1}^{n} x_i^{\alpha_i}$ is convex in $\Re_+^n$ if $c \leq 0$, $x_i \geq 0$, $\alpha_{p_i} \geq 0$ (for all $i = 1, \ldots, n$) and $(1 - \sum_{i=1}^{n} \alpha_i) \geq 0$.

Proposition 3. The function $f(X) = cexp(r_1 x_1 + r_2 x_2 + \ldots + r_n x_n)$ is convex in $\Re_+^n$ if $c \geq 0$ and $r_i \in \Re$. Hence, if one of the three above propositions is not satisfied for a signomial, by applying the following transformation rules we can convexify it:

Rule 1. If $c > 0$ and $\alpha_i > 0$, then $cx_1^{\alpha_{p_1}} x_2^{\alpha_{p_2}} \ldots x_n^{\alpha_{p_n}} = cexp(r_1 y_1 + r_2 y_2 + \ldots + r_n y_n)$ where $y_i = log(x_i)$, $i = 1, \ldots, n$.

Rule 2. If $c < 0$, $\alpha_i > 0$ and $\sum_{i=1}^{n} \alpha_i > 1$, then $cx_1^{\alpha_{p_1}} x_2^{\alpha_{p_2}} \ldots x_n^{\alpha_{p_n}} = cX_1^{\alpha_1/R} \ldots X_m^{\alpha_m/R}$ where $x_i = X_i^{1/R}$, $i = 1, \ldots, n$ and $R = \sum_{i=1}^{n} \alpha_i$.

5 SUMMARY OF THE STATE SPACE RMPC ALGORITHM

In this section, we provide a summary of the needed steps to find the optimal control law using the new proposed RMPC method in the state space model:

1. Fix the upper and lower bounds of a_i and b_j which are $\underline{a_i}$, $\overline{a_i}$ $(i = 1, \ldots, n_a)$, $\underline{b_j}$ and $\overline{b_j}$ $(j = 0, \ldots, n_b)$. Several works have been published addressing facets of finding model uncertainty bounds (Messaoud and Akoum, 2000), (Messaoud and Favier, 1994).
2. Find the optimum values of a_i and b_j by solving the minimization optimization problem of equation 13. This problem is initially non convex. By applying the transformation techniques (exponential and power transformations) of the GGP method, the transformed problem (objective function and constraints) becomes convex. The GGP technique is applied with a polynomial form.
3. Find ΔU, the solution of the minimization problem of equation 12 with the optimal values of a_i and b_j found in step 2.
4. Inject the control action in the plant to find the state and the output actions of the future sequences.
5. Go to step 2 and repeat with the optimal value of the control signal found in step 3.

To explain more step 2, we consider a simple example where the state matrix is $F = -a_1$, the input matrix is $G = b_0$ and the output matrix is $H = 1$. The controller parameters are: $H_p = 1$, $H_c = 1$ and $\lambda = 1$. Then using equations 8 and 9, the criterion J is written as following:

$$J = (y(k) - a_1 \Delta x(k) + b_0 \Delta u(k) - w(k+1))^2 + \Delta u(k)^2 \qquad (18)$$

Consequently, after expanding equation 18, we observe that the J criterion is non convex relatively to x_1, x_2 and x_3 (according to proposition 1 and proposition 2).

6 SIMULATION EXAMPLES

In this section, the new RMPC method using state space description and based on GGP will be illustrated through two examples.

6.1 Example 1

The first example is a simple system described by the discrete state model given by equation 5, where the state matrices are:

$$F = -a_1 \ , G = 0.11 \text{ and } H = 1$$

The uncertain variable bounds are:

$$-1.6 \leq a_1 \leq -1.2$$

This system is unstable for all values of a_1. The initial state points is fixed at $x(0) = x(1) = 0$. We consider the following control parameters: $H_p = 3$, $H_c = 1$ and $\lambda = 0.02$. The set-point is changed between 1 and -1. Moreover, constraints on control and control moves signals process have been taken into account. Their values are: $-3.5 \leq u(k) \leq 2.3$ and $-1.5 \leq \Delta u(k) \leq 1.5$. For model 1, $a_1 = -1$ and for model 2, $a_1 = -1.2$.

Fig. 1 shows the closed loop response of the system for the two models using the proposed state space RMPC approach based on the GGP technique. A load disturbance is added to the model output. This disturbance takes 0.2 for $15 \leq k \leq 25$ and $45 \leq k \leq 55$

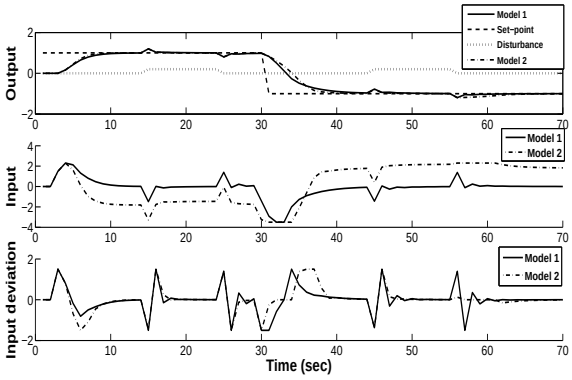


Figure 1: Closed-loop simulation results for model 1 and model 2.

and 0 else. The simulation results show good performances of the proposed approach. This approach successfully controls the above system. It achieves variable set point tracking and non zero disturbance rejection with respect to input and input deviation constraints. Moreover, the on-line optimization algorithm takes about $0.17s$ per sample time. Consequently, the proposed technique is accomplished in a reasonable amount of time.

6.2 Example 2: Comparison with Pannochia Method

In this example, we consider a jacketed continuous stirred tank reactor (CSTR) presented by Henson and Seborg (Henson and Seborg, 1997). After linearization around the middle-conversion open-loop unstable steady-state and discretization with a sampling time of $5s$ (Pannochia, 2004), we obtain the following state space model matrices:

$$F = \begin{bmatrix} -a_1 & 1 \\ -a_2 & 0 \end{bmatrix}, \; G = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

where the uncertainty variables are bounded as follows: $-2.3006 \leq a_1 \leq -2.1617$, $1.1555 \leq a_2 \leq 1.2863$, $0.2022 \leq b_0 \leq 0.2153$ $-0.1804 \leq b_1 \leq -0.1718$. Model 1 is described by the following state matrices:

$$F_1 = \begin{bmatrix} 2.1617 & 1 \\ -1.1555 & 0 \end{bmatrix}, \; G_1 = \begin{bmatrix} 0.2022 \\ -0.1718 \end{bmatrix} \quad H_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

However, for model 2 we consider the following state matrices:

$$F_2 = \begin{bmatrix} 2.3006 & 1 \\ -1.2863 & 0 \end{bmatrix}, \; G_2 = \begin{bmatrix} 0.2153 \\ -0.1804 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Fig.2 compares the closed loop performances of the proposed optimization algorithm using GGP technique and the RMPC method presented by (Pannochia, 2004) using the above system for the two models. The control parameters are the following: $H_c = 2$, $\lambda = 0.002$. In the proposed approach $H_p = 3$. The initial state is x=$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Inputs constraints are fixed as follows: $-10 \leq u(k) \leq 10$. In the proposed approach, we suppose that the future set-points are unknown. Moreover, the two models are considered:

for $1 \leq i < 40$ the true system is model 1

for $i \geq 40$ the true system is model 2

Fig.2 shows a slightly difference between the two outputs. The response time of the Pannochia RMPC method is about $t_r = 4.87s$, however the one of the proposed RMPC method is $t_r = 6.1s$. Concerning the control signal, the proposed RMPC method shows less oscillations at the set point variations than the Pannochia RMPC method since it presents less of peaks.

(Pannochia, 2004) uses two algorithms: an off-line algorithm and an on-line one. The off-line algorithm computes a nominal system and a feedback gain design which guarantees the closed loop system stability. This algorithm solves a non convex min-max optimization problem. Hence, both the minimization and the maximization problem give local solutions. In fact, this limitation can affect the closed loop performance responses. However, The proposed approach uses only one on-line algorithm based on the GGP method which is a global optimization technique for non convex polynomial functions.

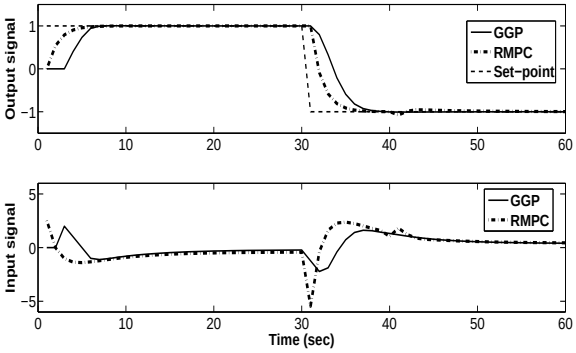


Figure 2: Closed-loop simulation results for models 1 and 2.

7 CONCLUSIONS

An examination of the closed loop performances of a new approach in constrained state space Robust Model Predictive Control (RMPC) in the presence of parametric uncertainties is presented. Based on simulation example results, we have shown that the proposed method is able to guarantee variable set-point

tracking respecting to the input and input deviation constraints and to reject non zero disturbance. Moreover, our method features good performances in the on-line algorithm time computation and a simplicity of implementation. These features make this method particulary attractive for industrial applications. A comparison with a recent state space RMPC method is also given.

ACKNOWLEDGEMENTS

I warmly thank my colleague Baddreddine Bouzouita for his helpful comments.

REFERENCES

Alamo, T., Ramirez, D., and Camacho, E. (2004). Efficient implementation of constrained min-max model predictive control with bounded uncertainties: a vertex rejection approach. *Journal of Process Control*, 15 (2005):149–158.

Bouzouita, B., Bouani, F., and Ksouri, M. (2007). Solving non convex min-max predictive controller. In *Conference Proceedings of 2007 Information, Decision and Control, Adelaide*.

Camacho, E. and Bordons, C. (2004). *Model Predictive Control*. Springer, London.

Campo, P. and Morari, M. (1987). Robust model predictive control. In *American control conference*, pages 1021–1026.

Chul, C. and Dennis, L. (1996). Effectiveness of a geometric programming algorithm for optimization of machining economics models. *Computers and operations research*, 23:957–961.

Cordon, P. and Boucher, P. (1994). Multivariable generalized predictive control with new multiple reference model: a robust stability analysis. *Mathematics and computers in simulation*, 37:207–219.

Fukushima, H., Kim, T., and Sugie, T. (2007). Adaptive model predictive control for a class of constrained linear systems based on comparison model. *Automatica*, 43 (2):301–308.

Henson, M. and Seborg, D. (1997). *Non linear Process Control*. Prentice Hall.

Huaizhong, L., Niculescu, S., Dugard, L., and Dion, J. (1998). Robust guaranteed cost control of uncertain linear time-delay systems using dynamic output feedback. *Mathematics and computers in simulation*, 45:3–4.

Kothare, M., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32 (10):1361–1379.

Lee, Y. and Kouvaritakis, B. (2000). A linear programming approach to constrained robust predictive control. *IEEE Trans. Auto. Contr.*, 45:1765–1770.

Maranas, C. and Floudas, C. (1997). Global optimization in generalized geometric programming. *Computers and chemical engineering*, 21:351–369.

Mayne, D., Rakovic, S., Findeisen, R., and Allgower, F. (2009). Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, 45:2082–2087.

Messaoud, H. and Akoum, Z. (2000). An algorithm for computing parameter bounds using prior information on physical parameter bounds. In *7th conference on Electronics, Circuits and Systems (ICECS)*, pages 218–221.

Messaoud, H. and Favier, G. (1994). Recursive determination of parameter uncertainty intervals for linear models with unknown but bounded errors. In *10th IFAC Symp. on SYSID, Copenhagen, Denmark*, pages 365–370.

Nand, K. (1995). Geometric programming based robot control design. *Computers and industrial engineering*, 29:631–635.

Pannochia, G. (2004). Robust model predictive control with guaranteed set point tracking. *Journal of process control*, 14 (2004):927–937.

Porn, R., Bjork, K., and Westerlund, T. (2007). Global solution of optimization problems with signomial parts. *Discrete optimization*, 5:108–120.

Qian, W., Liu, J., Sun, Y., and Fei, S. (2010). A less conservative robust stability criteria for uncertain neutral systems with mixed delays. *Mathematics and computers in simulation*, 80:1007–1017.

Ramirez, D., Alamo, T., , and Camacho, E. (2002). Effecient implementation of constrained min-max model predictive control with bounded uncertainties. In *Conference on decision and control*.

Rossiter, J. and Kouvaritakis, B. (1998). Youla parameter and robust predictive control with constraints handling. In *Workshop on Non linear Predictive Control ,Ascona, Switzerland*.

Tsai, J. (2009). Treating free variables in generalized geometric programming problems. *Computers and chemical engineering*, 33:239–243.

Tsai, J., Lin, M., and Hu, Y. (2007). On generalized geometric programming problems with non positive variables. *European journal of operational research*, 178:10–19.

Watanabet, K., Ikeda, K., Fukuda, T., and Tzafestas, S. (1991). Adaptive generalized predictive control using a state space approach. In *International workshop on intelligent robots and systems IROS, Osaka, Japan*.

MIXED COLOR/LEVEL LINES AND THEIR STEREO-MATCHING WITH A MODIFIED HAUSDORFF DISTANCE

Noppon Lertchuwongsa, Michèle Gouiffès and Bertrand Zavidovique
IEF, Institut d' Electronique Fondamentale, CNRS 8622, Université Paris Sud 11, Paris, France
Noppon.lertchuwongsa@u-psud.fr

Keywords: Computer Vision, Stereovision, Color Lines, Hausdorff, Shape Matching.

Abstract: Level lines and sets are competitive features to support recognition. Color is assumed more informative than intensity, so color-lines are preferred to exhibit the set basis. In the paper, they are defined after level lines, and then extracted and characterize. Greater information is kept by color lines resulting into more efficient grouping towards objects. A novel Hausdorff-inspired disparity finder is introduced fed in by color lines with respect to epipolar constraints. The efficient disparity map resulting from pixel wise line-matching between left and right images justifies our technical choices.

1 INTRODUCTION

Usual segmentation appears sensitive in practice to the view point and shadows (contrast changes). In this paper, morphologically stable sets are extracted and a Hausdorff distance provides for global and local pattern matching all in once. Level sets make a basis - the topographic map - easy to compute.

According to psychologists (Koschan a Abidi 2008), in most contexts color prevails on shape and texture: whence using color to make lines more distinctive seems sensible.

Defining color sets and lines is not straightforward because of the intrinsic tri-dimensional nature of color. Usually, color data are transformed from a 3D color space to a 1D Level Space, by combining the three components, either to specify a total order of colors or towards some optimal function of those. Here, our of the commonly used HSV space, the 1D level space is provided by a mixture of H and V weighted by S. After mixture line features have been extracted from two images to compare, matching is carried out with a coarse to fine strategy involving a modified Hausdorff distance (Huttenlocher 1993), to pair portions of lines.

As the disparity in stereoscopic images is computed from the distance of corresponding points, results of stereo shape matching and their comparison with the ground truth will assert the efficiency of our matching process, founding the algorithm evaluation.

The paper is organized as follows. Section 2 is a brief reminder on lines, color and matching for notations and basic algorithms. Section 3 details our procedure to enhance intensity lines into color lines. Then, Section 4 deals with pattern and point selections for matching towards disparity from line pairing. Finally, the validity and efficiency of the proposed procedure are evaluated through comparing our depth map with the ground truth.

2 BIBLIOGRAPHY

Level sets and lines. Level sets (Caselles 1999), the topographic map, prove invariant to contrast changes and naturally robust to occlusions. Converting images into sets and back is straightforward. Projections follow equation (1) or (2)

$$X_\lambda = \{x \in \Re^2, u(x) \le \lambda\} \quad (1)$$

$$X^\lambda = \{x \in \Re^2, u(x) \ge \lambda\} \quad (2)$$

where $u(x)$ is the gray level at pixel x in the image and λ is the parameter – threshold – defining the lower (resp. upper) set X_λ (resp. X^λ). reconstruction follows equations (3)

$$u(x) = \inf_\lambda \{\lambda, u \in X^\lambda\} = \sup_\lambda \{\lambda, u \in X_\lambda\} \quad (3)$$

A level line is the border of a level set, therefore parameterized by the same λ. In practice it still depends on the threshold's step: as it is usually low

valued in hope of an exhaustive topographic map, images generate more lines than necessary to matching. Color is likely to improve a priori the line separability then lowering their number.

Color Edges and Lines. Color was first proved to extend the notion of topographic map by (Coll & Froment 2000) who designs a total order in the HSV space. Gouiffès (2008) proposed to extract color sets from color bodies in the RGB 3-D histogram. Moreover the Hue in HSV proves ultimately discriminative and invariant to shadow, however it is ill-defined at low saturation. Compared to gray level, color provides more edge or line information.

Feature matching. Set-correspondence finding between two images can be classified into three principal algorithmic lines: Point matching, based on correlation windows on raw intensity data, but suffer on homogeneous areas. Geometrical Feature matching are the corners or curvature points or line segments with attributes. Region and shape matching strategies. The hypothesis is made here that corresponding patterns maintain the shape between images (Loncaric 1998).

The method we detail in the present paper exploits the shape stability granted by the invariance to contrast through color sets.

3 COLOR SETS AND LINES

3.1 Color Sets

Color image data can be represented in the RGB cube. The pixel intensity – i.e. level – amounts to projecting the given RGB point onto the principal diagonal of the cube (the gray level scale). The question then arises to find a transformation more adaptive to the image content. Gouiffes & Zavidovique (2008) proposed to use the dichromatic model to find body colors: vectors pointing to principal body colors are used separately instead of the sole cube diagonal. Related data – i.e. close enough in the RGB space – is projected onto that vector exhibiting associated level sets.

Our leading idea to build the transformation of the HSV color space into a 1-D level space takes after the vanishing of Hue, independently at both low light intensity and low saturation.

3.2 H,V Mixtures vs. S

Considering the above-mentioned drawbacks of the HSV space, the following formulation of S is preferred:

$$S = Max(R,G,B) - Min(R,G,B) \quad (4)$$

Second, we propose to use hue when it is relevant(high saturation) and intensity otherwise (low saturation). To ensure color sets homogeneous enough respective to what is expected from regions in image segmentation, we design a smooth transition with a sigmoid function Sig(s,k):

$$O_F(p) = Sig(s,k)H(p) + (1 - Sig(s,k))I(p) \quad (5)$$

Sig(s,k) is thus parameterized by its slope s and inflection point k to be adapted from former α and β.

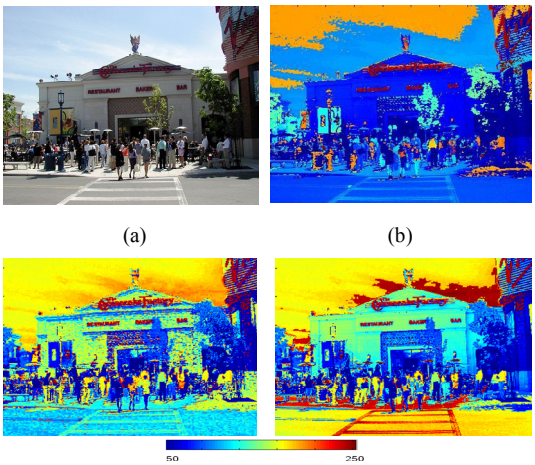


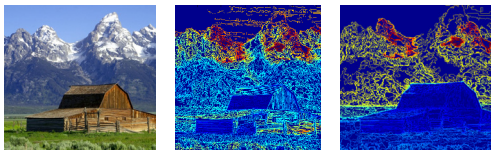
Figure 1: (a) Original image, (b) pseudo-color image in the data space from the step function O_F and threshold on intensity, (c) pseudo-color image in the data space from the sigmoid mixture O_F and a threshold on intensity, (d) pseudo-color image from the same data space and a threshold on magnitude of the saturation, both (b) (c) and (d) use same scale of pseudo color.

Fig 1 compares the use of a sigmoid (Fig. 1(c)) is compared with the use of a Heaviside step function (see Fig 1 (b)). On top of regular noise, the artifacts of the step function are likely due to the frequency of switches between the intensity and hue scales when saturation lies in around the threshold. The overall consequence is a loss of some details when trying to lower the effect. The drawback of the sigmoid function, compared to the step function, is conversely its smoothness. When saturation is low, although hue does not keep much of an effect, a change of the RGB vector even to a close one might result into significant modification of the final result O_F due to the small signal situation. Pepper noise can occur on the image during the transition of the Sig function around point k. Fig.1(c) finally illustrates that issue: for example, on the road area, the wall of the restaurant, or among the crowd, the

sigmoid combination outputs some sparse noise.

Sharpening the sigmoid to make it closer to a step function will reduce details. Since, again, the sigmoid is more biased by the hue and is exactly used to take advantage from it, the inflection point k is set to a low value. Figure 1(d) shows better results in that respect thanks to replacing the usual formula of the saturation – a ratio – by the difference version given in equation (4). Finally, Fig. 2 compares our lines with the classical gray level lines.

Our line extraction method derives from the one proposed by Bouchafa (2006) to direct close curve extraction.



(a) (b) (c)

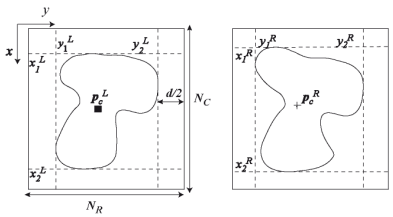
Figure 2: (a) Original color image, (b) gray level lines, (c) color lines. Note that (b) and (c) result from a same artificial look-up table to make lines more distinct.

4 COLOR LINE MATCHING

To speed the match up, our method begins to sort patterns by global features for a coarse stage. Then, at the fine stage, point matching is performed.

Coarse Scale Matching. Techniques of comparing border of sets between 2 shapes, such as, length, level of set, standard deviation and position shape, which is exploited from stereo vision knowledge.

Fine Matching. Published techniques relying on a reference point, e.g. Chang (1991), strongly depend on this point stability.



(a) (b)

Figure 3: Example of corresponding sets A and B and their centroids and notations.

The Hausdorff distance founds an interesting technique to get the process free from the center. Let $\boldsymbol{A} = \{a_1,...,a_n\}$ and $\boldsymbol{B} = \{b_1,...,b_m\}$ be two finite sets. Their Hausdorff distance is defined as

$$H(A,B) = Max\left(h(A,B), h(B,A)\right) \quad (6)$$

where

$$h(A,B) = \underset{a\in A}{Max}\ \underset{b\in B}{Min}\left\|a_i - b_j\right\| \quad (7)$$

and $\|\ \|$ is a norm in the affine space of the image.

In previous works, e.g. (Huttenlocher 1993) Hausdorff was used to locate shapes in a scene by using a template.

Here, a topographic map contains abundant lines which are close together, hampering the accuracy of the distance map in our application. Therefore a progressive scheme is rather tried: every tentative couple of left and right lines is extracted and superimposed within a common window W with center $\boldsymbol{p_c^W}$ of coordinates (x_c^W, y_c^W) and of size N_R N_C. However, using the center of W, $p^W{}_c = (N_C/2, N_R/2)$ as the reference point for mapping, two problems occur. First, direct mapping of the centroid of W may make some part to exceed the window. Second, right and left patterns to be matched can be dissimilar since they were filtered roughly, therefore large enough space d is needed to compensate.

Thus, a slightly more elaborated shifting strategy into W has to be designed (see Fig. 3). x_1^L, x_2^L, y_1^L, y_1^L are the coordinates of the bounding rectangle of the left pattern, and $p_c^L = (x_c^L, y_c^L)^T$ is the centroid (small square) of the line to be mapped in the comparison. It is likely different from the center of the window, illustrating the first problem.

Finally, both each point $\boldsymbol{p^L} = (x^L, y^L)$ of L_L and each point $\boldsymbol{p^R} = (x^R, y^R)$ of L_R are translated into W, of reference point $p^W = (xc^W, yc^W)$ and of size N_R x N_C respectively with vectors v_L^W and v_R^W :

$$\boldsymbol{p'}_L = \boldsymbol{p}_L + \boldsymbol{v'}_{LW} \quad with \quad \boldsymbol{v'}_{LW} = (x_c^W - x_c^L, y_c^W - y_c^L)^T \quad (8)$$

$$\boldsymbol{p'}_R = \boldsymbol{p}_R + \boldsymbol{v'}_{RW} \quad with \quad \boldsymbol{v'}_{RW} = (x_c^W - x_c^R, y_c^W - y_c^R)^T \quad (9)$$

where The new centroid $p^W{}_c$ is then defined as:

$$\boldsymbol{p_c^W} = \begin{pmatrix} x_c^W \\ y_c^W \end{pmatrix} = \begin{pmatrix} N_R \\ N_C \end{pmatrix}^T \begin{pmatrix} (x_c^L - x_1^L)/(x_2^L - x_1^L) \\ (y_c^L - y_1^L)/(y_2^L - y_1^L) \end{pmatrix} \quad (10)\ (11)$$

and comparison window is the rectangle N_R x N_C:

$$N_R = x_2^L - x_1^L + d \quad (12)$$

$$N_C = y_2^L - y_1^L + d \quad (13)$$

where d is the extension of the window from the size of the line

Note that, according to the stereovision application targeted in our paper, we assume that the epipolar constraint holds, Therefore, the translation on row x is similar in $v'_L{}^W$ and $v'_R{}^W$. This assumption reduces significantly the complexity of the Hausdorff matching so made one-dimensional. After shifting the selected left line L_L to W (equation (8))

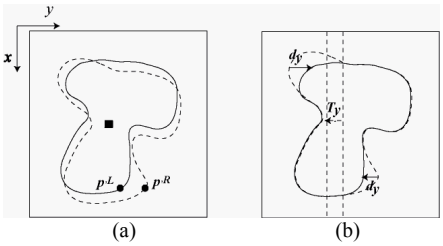
the distance map is computed with a city block distance. Then, candidate samples of right lines are mapped to W – equations (8,9) – and the Hausdorff distance is computed for all right line candidates in C_R as resulting from the coarse scale matching. The final homologous line is the line which provides the minimum Hausdorff distance - equation (6) .

Indeed, when finding the point-to-point or line-to-point distances, we use their minimum. Pattern-to-pattern distances, i.e. set of points to set of points, result from the furthest of those closest points. When a pattern is selected by the Hausdorff's condition, all points in the set will find their corresponding part.

$\boldsymbol{T}_y$ is the translation vector bound to the minimum Hausdorff distance (see Fig. 4), and $\boldsymbol{d}_y$ is the local Hausdorff vector between corresponding columns of points $\boldsymbol{p'_L}$ and $\boldsymbol{p'_R}$ (respectively y'_R and y'_L). The disparity D of corresponding points $\boldsymbol{p^L}$ and $\boldsymbol{p^R}$ is obtained from the stereo pair, following:

$$D(\boldsymbol{p}^L) = \left| y^L - y^R \right| = \left| y_c^L - y_c^R \right| - \left\| \boldsymbol{T}_y \right\| - \left\| \boldsymbol{d}_y \right\| \qquad (14)$$

Note that the disparity value is computed at all sample points of a line.



(a) (b)

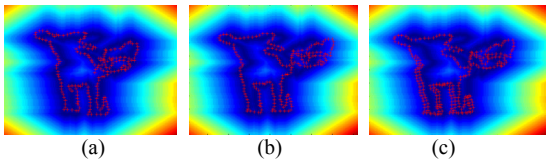
Figure 4: (a) Right line L_R (dotted line), superimposed on L_L (continuous line) when L_R was firstly projected into W which already had L_L as distance map, centroid of L_R is marked as $\mathbf{p_c^R}$. (b) The Hausdorff method makes L_R translate to new centroid $\mathbf{q_c^R = p_c^R + T_y}$. Finally, the vector $\mathbf{d_y}$ goes from $\mathbf{p'^R + T_y}$ to $\mathbf{p'^L}$.

Decomposing Lines. A single global distance to all points corresponds to a simple linear transformation between homologous points. Indeed, one region or line can relate to several depths. Also, one line likely refers to several objects at different depths. Rather than a complicate set of equations, the line can be decomposed into portions where the rigid motion applies well enough.

In figure 5, the line points extracted from the right image of a stereoscopic pair (red crosses) superimpose with the left line from the distance map. Blue means close and the redder the further.

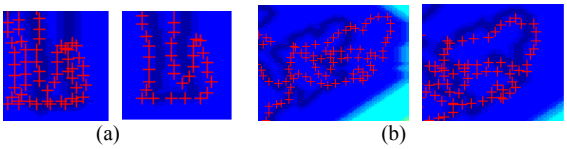
Let us call "centroid match" the process where the two centroids are superposed first and then the result for every pair of corresponding points is computed.

In fig.5 (a) the centroid match leads to large distances between corresponding points, up to missing corresponding points on the right border of the right leg (reader's right) that are paired with the left border of the right leg. In Fig. 5(b) Hausdorff makes both lines have correct co-points, generally better than before. The distance of corresponding points is reduced; however there are still some misfits in the tail (right area of the deer). Finally, Fig. 5(c) illustrates our technique. Obviously, the number of mismatched points was efficiently reduced. Fig. 6 shows some enlarged details of the points matching.



(a) (b) (c)

Figure 5: Illustration of the matching procedure. Right lines' (Red Cross symbol) superimposed on left lines' distance map (dark). (a) lines based on the same centroid at the center window, (b) lines after optimal Hausdorff translation, (c) result of the proposed method.



(a) (b)

Figure 6: (a) Enlarged version of the leg part in figure 17: left, centroid matching; right, modified Hausdorff. (b) close up on the tail part of figure 17: left, general Hausdorff, to compare with right, modified Hausdorff.

5 EXPERIMENT AND RESULTS

Color lines are evaluated first in comparing with gray level lines. They are then used to evaluate the efficiency of the modified Hausdorff distance. This distance is compared to classical line matching methods through the stereo matching quality for both color and gray lines. The database, 2003-2006, in Scharstein (2003, 2007) and Hirschmüller (2007) is our test bench.

5.1 Contribution of the Color Lines

The relevance of our color mixture lines wrt. gray level lines, is indicated by two criteria: 1) *the number of lines*, compactness of the topographic map. 2) The *average PSNR values* and number of lines computed on the image data base are collected in the table 1. Three different data are considered:

gray, abrupt mixture, and sigmoid. λ is the quantization step, both PSNR and "line number" are decreasing functions of λ. k sets the mix (section 3).

From the analysis of the table 1, we can note that the PSNR for the sigmoid mixture is lower than with Gray lines except when the saturation is greater than or equal to 0,4. The number of lines is appreciably lower in same conditions. That means images reconstructed from a topographic map after smooth-mixture are closer to the input image, while the topographic map is more compact. Fig. 7 shows examples of level and color sets.

Table1: Comparison results between images after color mixture data and gray level ones: average PSNR of each kind of data and the average number of lines.

	λ	k	PSNR	Nb. lines.
Gray	1	-	42.33	36474.30
	2	-	37.97	29234.31
	5	-	31.19	18666.96
		0.1	36.46	29378.13
		0.2	38.15	30617.52
	1	0.4	42.53	33416.48
		0.1	32.63	20784.17
Sigmoid method		0.2	34.41	22332.59
	2	0.4	38.68	25575.09
		0.1	27.97	11498.09
		0.2	29.77	12779.41
	5	0.4	33.51	15507.00

Figure 7: Examples of level and color sets: (a) Original color image, (b) Pseudo color image of gray level sets, (c) Pseudo color image of sigmoid color set: for all pseudo color images the step parameter is set equal to 5, the amplitude is coded on 8 bits and k = 0,2 for the mixture.

Line images are shown in Fig. 8. As expected, many lines appear on colorimetrically homogeneous objects. Shadows produce lots of irrelevant lines, unstable for matching since they do not correspond to real objects. With the sigmoid mixture, the topographic maps are more compact and lines correspond to salient physical items at sight.

This preliminary experimental evaluation suggests that our HSV mixture produces lines more appropriate for matching, i.e. more distinctive, quicker to compute, and more related to object-boundaries.

5.2 Matching Results

Figure 9 shows an example of results of our disparity computation method and figure 13 displays the error from the ground truth in every pixel. Most matching errors occur for large color lines related to several objects at different depths.

In Table 2 , N_T refers to the average number of line points computed in the whole data base. N_C is the number of correct points, those for which the disparity error is less than 5. E_T stands for the mean disparity error computed on the N_T points and E_C same on the N_C points.

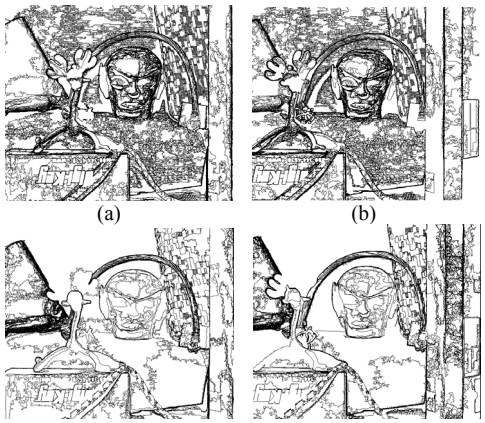


Figure 8: Examples of lines from the stereo pair of Fig.7. Step λ is set to 2, and the inflection point k is 0.2. (a), (b) are the gray level lines in left and right images respectively, (c) (d): Lines from sigmoid mixture.

The parameter $\%D_{E>5}$ (resp.$\%D_{E>1}$) is the percentage of points with a disparity error $\frac{\sum_x \sum_y |D_n(x,y) - D_r(x,y)|}{n}$ greater than 5 pixels (resp. 1 pixel), $D_n(x,y)$ being the disparity after our method at pixel (x,y) and $D_r(x,y)$ the truth after the data base; n is the number of pixels.

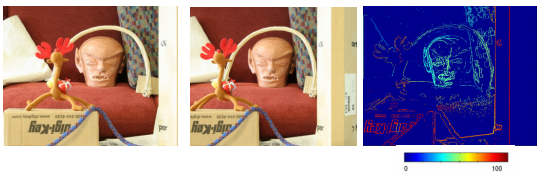


Figure 9: Example of matching: (a), (b) left and right stereoscopic images, (c) the disparity-line image and its color map value.

These criteria are computed for the Hausdorff distance and the centroid method. The Hausdorff method is used with gray and Sigmoid Mixture data (the parameter that we use is T_S = 0.2). Out of table 2, gray level data yields a larger number of points than the mixture in same technical conditions.

Most added points are not salient and those from shadow lines are unstable. Consequently, 19,3% of the gray lines have not been correctly matched,

compared to only 13 % for the mixture lines. Too large a number of lines is also problematic in terms of computation times and resources. Eventually, disparity errors are always lower from mixture lines.

Table 2: Comparison of the results provided by the Hausdorff matching and classical centroid line matching.

	N_T	N_C	E_T	E_C	$\%D_{E>5}$	$\%D_{E>1}$
Hausdor						
Gray	6092	497	4,9	0,7	18	54
Mixture	3585	313	4,0	0,6	13,51	50,63
Centroid	3892	289	6,9	1,0	28,07	67,73

For comparison purposes, Table 2 collects also results from the Centroid technique on mixture data. N_T is normalized to find a number of lines comparable to Hausdorff's by controlling the T_a threshold of acceptable data as defined in section 4. Even if T_a is adjusted in the Hausdorff case, it could only worsen results since the matching condition states that a tighter threshold means more similar patterns. T_a depends on the line length not on the type of input data. Then on the same image sub-part, 26,6 % of the lines have not been correctly matched. Moreover, the errors E_T and E_C are significantly higher than Hausdorff 's (E_T : 69% and E_C : 59%).

The same N_T value is kept for gray vs. mixture lines not to bias results (same technique and parameters).

Contribution of the Modified Hausdorff Distance. Table 3 compares the classical and modified Hausdorff distances for the color mixture. In the latter case, lines are divided if their length is higher than an experimentally set threshold T_l, the value of which depends on the image size through natural stretching and shrinking in stereo (T_l=500 in our experiments). Same measures as before are collected

Table 3: Comparison of classical Hausdorff techniques and modified techniques.

	N_T	N_C	E_T	E_C	$\%D_{E>5}$	$\%D_{E>1}$
Classical	23185	18621	3,5744	1,0840	21,01	65,53
Modified	24585	21357	3,1402	0,8753	14,22	60

Because these techniques are based on a different Hausdorff matching, the threshold of acceptable data *Ta* is separately chosen to reach the same level of details in the disparity image.

According to table 3, the classical Hausdorff method yields a smaller number of points which means less detail compared to the modified Hausdorff. It produces even a smaller number of points, N_C, for which the disparity error is less than 5. Nevertheless, the average error E_C (Column 4) is 3,57 pixels for the global Hausdorff and only 3,14 pixels for the modified version. Likewise, the rate $\%D_{E>5}$ in column 5 is 21,01% vs.14,22%, meaning that 79% of the lines are correctly matched by the classical approach, while 85,8% are correctly matched with the modified version. And finally, Table 3 also shows that the disparity errors are always lower with the modified Hausdorff distance.

6 CONCLUSIONS

Our work studies color line matching and evaluates its relevance in a stereo matching application. A novel color topographic map is proposed with less irrelevant lines, more related to objects and more distinctive. Direct close curve extraction based on the color map reduces the memory and CPU greed. Color sets finally prove more stable in practice than usual results of region segmentation. The proposed modified Hausdorff shows its efficiency in finding more accurate correspondences for image registration.

REFERENCES

Koschan & Abidi (2008). *Digital color image processing.* Hoboken, N.J.: Wiley-Interscience.

Huttenlocher, Klanderman & Rucklidge (1993). *Comparing images using the Hausdorff distance.* IEEE Trans. on PAMI, Vol. 15. N° 9. pp. 850–863.

Caselles, Coll & Morel (1999).*Topographics maps and local contrast invariance in natural images.* IJCV, pp. 5-27.

Coll & Froment (2000), *Topographic Maps of Color Images.* In ICPR Vol. 3. p 3613. 2000.

Gouiffès & Zavidovique (2008), *A Color Topographic Map Based on the Dichromatic Reflectance Model.* Eurasip JIVC, n.17.

Loncaric (1998). *A survey of shape analysis techniques.* Pattern Recognition Vol. 31, pp 983-1001.

Bouchafa & Zavidovique (2006) *Efficient cumulative matching for image registration.* IVC, Elsevier Vol. 24, pp.70-79.

Chang, Hwang & Buehrer (1991) *A shape recognition scheme based on relative distances of feature points from the centroid,* Pattern Recognition, Vol. 24, N°11, pp. 1053-1063.

Scharstein & Szeliski (2003). *High-accuracy stereo depth maps using structured light.* In IEEE CVPR Vol. 1, pp. 195-202.

Scharstein & Pal (2007). *Learning conditional random fields for stereo.* In IEEE CVPR.

Hirschmüller & Scharstein (2007). *Evaluation of cost functions for stereo matching.* In IEEE CVPR.

FEATURE EXTRACTION AND SELECTION FOR AUTOMATIC SLEEP STAGING USING EEG

Hugo Simões, Gabriel Pires
Institute of Systems and Robotics, University of Coimbra, Coimbra, Portugal
{hugo, gpires}@isr.uc.pt

Urbano Nunes, Vitor Silva
Department of Electrical Engineering, University of Coimbra – Polo II, Coimbra, Portugal

Keywords: Feature Extraction, Feature Selection, EEG Sleep Staging, Bayesian Classifier.

Abstract: Sleep disorders affect a great percentage of the population. The diagnostic of these disorders is usually made by a polysomnography, requiring patient's hospitalization. Low cost ambulatory diagnostic devices can in certain cases be used, especially when there is no need of a full or rigorous sleep staging. In this paper, several methods to extract features from 6 EEG channels are described in order to evaluate their performance. The features are selected using the R-square Pearson correlation coefficient (Guyon and Elisseeff, 2003), providing this way a Bayesian classifier with the most discriminative features. The results demonstrate the effectiveness of the methods to discriminate several sleep stages, and ranks the several feature extraction methods. The best discrimination was achieved for relative spectral power, slow wave index, harmonic parameters and Hjorth parameters.

1 INTRODUCTION

About a third of the population suffers from sleep disorders, including the obstructive sleep apnea syndrome (Doroshenkov *et al*, 2007). The diagnosis of such diseases is performed by a polysomnography (PSG) which requires the patient's hospitalization with costs and discomfort for the patient. Ambulatory diagnostic devices may have an important role in order to mitigate these factors. The PSG consists on the acquisition of various electrical biosignals including electroencephalogram (EEG), electrooculogram (EOG) and electromyogram (EMG). The signals are segmented into epochs of 30 seconds and assigned to a sleep stage by an expert (Iber *et al*, 2007). This is a tedious and time consuming task. Automatic sleep stages classification (ASSC) is therefore an attractive solution. However, the general opinion is that most of the experts do not rely on ASSC software, because they usually present a low performance (i.e. present a high level of disagreement). One of the main reasons is due to the high variability between subjects which makes it difficult to obtain robust models for classification. The expert uses sometimes heuristics difficult to implement in the algorithms and combines a macro and micro perspective of the overall epochs. It should be highlighted that there is also some level of disagreement between experts.

This work describes part of an apnea detection system to be used in ambulatory situations by patients at home. It does not intend to substitute the PSG, but only to determine primarily if the patient is sleeping at the occurrence of the apnea episode, and secondly to determine in which sleeping stage it did occur. The stage classification relies only on EEG signals. This paper investigates several feature extraction methods to compare their performance aiming to achieve improved results in the following sleep detection stages: wake (W) vs. sleep (S), NREM (NR) sleep vs. REM (R) sleep, NREM N1 vs. NREM N2 + NREM N3, NREM N1 + NREM N2 vs. NREM N3, NREM N1 vs. NREM N2, NREM N2 vs. NREM N3 and NREM N1 vs. REM sleep (Iber *et al*, 2007). Moreover, a feature selection method based on the squared Pearson correlation coefficient (Guyon and Elisseeff, 2003), henceforth designated R-square criteria, is applied with the purpose of finding a reduced set of discriminative features. These features are used to provide additional information to the expert, and also to automatically classify each sleep stage with some degree of certainty. The classification is

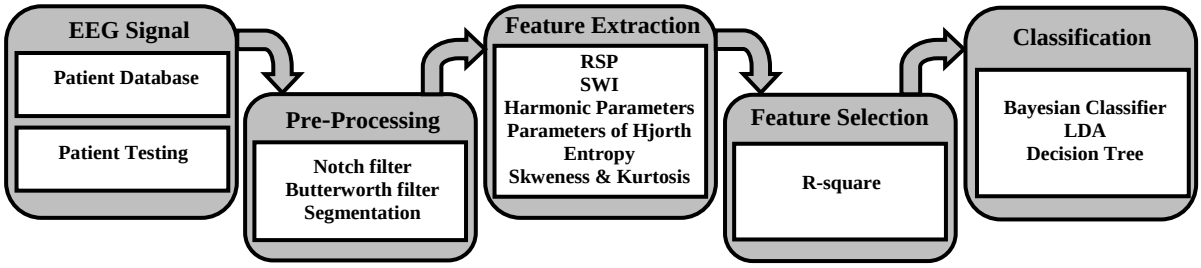


Figure 1: Classification methodology.

performed by a Bayesian classifier using 2-class detection. Scoring sleep is done according to rules of the American Academy of Sleep Medicine (AASM) Manual for Scoring Sleep (Iber *et al*, 2007), an actualization of the rules of Rechtschaffen and Kales (Rechtschaffen and Kales, 1968). According to AASM Manual, sleep is divided into five stages: wake, NREM (Non Rapid Eye Movement) sleep (N1, N2 and N3) and REM (Rapid Eye Movement) sleep. Considering only EEG signals, the wake stage is characterized by a low amplitude alpha activity (8-13 Hz); N1 by a low amplitude theta activity (3-7 Hz); in N2 the predominant frequencies are in the 0.7-4 Hz range and there is the arising of sleep spindles and K-complexes; N3 presents at least 20% of the epochs with delta activity (<2 Hz) with amplitude greater than 75 μV; REM is characterized by frequencies mostly between 2 and 6 Hz with low amplitude. Sleep staging based only on EEG presents some difficulties because different stages such as wake, REM and NREM N1 present similar patterns. The ASSC has been addressed by many research groups. In (Tang *et al*, 2007), Hilbert-Hang transform and wavelet transform were applied to extract harmonic parameters from EEG signals, (Hese *et al*, 2001) implemented a semi-automatic method based on k-means clustering algorithm. (Ebrahimi *et al*, 2008) used neuronal networks and wavelet packet coefficients to discriminate between different sleep stages. Doroshenkov *et al*. (2007) have developed a classification algorithm based on Hidden Markov Models using only EEG signals. (Zoubek *et al*, 2007) have used feature selection algorithms to find the relevant features extracted from PSG signals. Schwaibold *et al* (2003) have implemented a neuro-fuzzy algorithm to model the rules of Rechtschaffen and Kales. Although some studies show good performance, they are very limited to specific groups of patients and it has not been possible yet to create generalized models that provide results accepted by the experts. Moreover, it remains difficult to discriminate between certain sleep stages using only EEG signals.

2 DATABASE

Data from all-night PSG records were provided by the Laboratory of Sleep from *Centro Hospitalar de Coimbra*. The PSG was recorded by the model Somnostar Pro from Viasys at a sampling frequency of 200 Hz. The database comprises seven patients (five males and two females) with ages between 27 and 64 years old (mean = 50 years; standard deviation = 12.88 years). Only six EEG channels were used: F3-A2, C3-A2, O1-A2, F4-A1, C4-A1 and O2-A1. All recordings were segmented into epochs of 30 seconds and labelled by an expert.

The dataset was initially composed by 6558 epochs. In order to avoid the over-fitting in the learning and testing of algorithms, the number of sleep epochs in the database was reduced to 3000, balancing the distribution of epochs of different sleep stages according to a normal night sleep distribution as presented in Table 1. Since the sleep stages N2 and N1 are the ones with the highest and lowest occurrence during a normal night sleep, respectively, they were set as the stages with major and minor number of epochs in the dataset, respectively, and the other sleep stages have a number of epochs between these limits.

Table 1: Full and reduced datasets.

Sleep Stages	Wake	NREM			REM
		N1	N2	N3	
Full dataset	1293	784	2431	1154	896
Reduced dataset	560	410	760	520	750

3 AUTOMATIC SLEEP SCORING

The classification methodology is illustrated in the block diagram presented in figure 1. The EEG signals are filtered and segmented. Different types of features extraction are used. These features are

then selected using the correlation criteria R-square measure in order to provide the classification stage, a Bayesian-based classifier, with the most discriminative ones. The training process uses data from a pool of patients and some data from the patient being monitored, namely, the wake recorded epochs before the patient fall asleep. This way, the wake model can be improved. Moreover, the wake epochs can be used for calibration of sleep stages. The performance analysis of the of feature extraction algorithms was done through ten-fold cross validation. The patients' database is partitioned into ten groups with the same number of epochs from each sleep stage. Nine of them are used to perform the models of classification and one for testing. This process is repeated 10 times using a different group for testing.

4 FEATURE EXTRACTION AND SELECTION

In ASSC, the EEG is traditionally analyzed in frequency domain because, according with AASM Manual, each sleep stage is essentially distinguished by some spectral properties. However, temporal analysis provides also useful information. For each EEG channel, 34 features were extracted using several methods as described in the following.

Spectral analysis provides some of the most important features. For each sleep epoch, an autoregressive method solved by the Yule-Walker algorithm was applied to estimate the power spectral density (PSD) (Yilmaz *et al*, 2007). The spectrum is divided into ten frequency sub-bands as represented in Table 2.

Table 2: Spectral sub-bands used in RSP computation.

Bands	Sub-bands	Bandwidth $\{f_L, f_H\}$ (Hz)
Delta	Delta 1	{0.5,2.0}
	Delta 2	{2.0,4.0}
Theta	Theta 1	{4.0,6.0}
	Theta 2	{6.0,8.0}
Alpha	Alpha 1	{8.0,10.0}
	Alpha 2	{10.0,12.0}
Sigma	Sigma 1	{12.0,14.0}
	Sigma 2	{14.0,16.0}
Beta	Beta 1	{16.0,25.0}
	Beta 2	{25.0,35.0}

For each sub-band, the relative spectral power (RSP) was computed. This parameter is given by the ratio between the sub-band spectral power (BSP) and the total spectral power, i.e., the sum of all 10 BSP sub-bands. This normalization is important to increase classification robustness during the recording session.

Some spectral bands can be highlighted over slow wave bands by means of slow wave index (SWI) defined by the following ratios:

$$DSI = BSP_{Delta} / (BSP_{Theta} + BSP_{Alpha}) \quad (1)$$

$$TSI = BSP_{Theta} / (BSP_{Delta} + BSP_{Alpha}) \quad (2)$$

$$ASI = BSP_{Alpha} / (BSP_{Delta} + BSP_{Theta}), \quad (3)$$

where DSI, TSI and ASI stand for delta-slow-wave index, theta-slow-wave index and alpha-slow-wave index, respectively (Agarwal *et al*, 2001).

Harmonic parameters allow the analysis of a specific band in the EEG spectrum. They include three parameters: center frequency (f_c), bandwidth (f_σ) and spectral value at center frequency (S_{fc}), defined as follows (Tang *et al*, 2007):

$$f_c = \sum_{f_L}^{f_H} f P_{xx}(f) \Big/ \sum_{f_L}^{f_H} P_{xx}(f) \quad (4)$$

$$f_\sigma = \left(\sum_{f_L}^{f_H} (f - f_c)^2 P_{xx}(f) \Big/ \sum_{f_L}^{f_H} P_{xx}(f) \right)^{1/2} \quad (5)$$

$$S_{f_c} = P_{xx}(f_c), \quad (6)$$

where, $P_{xx}(f)$ denotes the PSD, which is calculated for the frequency bands $\{f_L, f_H\}$ (see Table 2).

The Hjorth parameters provide dynamic temporal information of the EEG signal. Considering the epoch x, the Hjorth parameters are computed from the variance of x, var(x), and the first and second derivatives x', x'' according to (Ansari-Asl *et al*, 2007)

$$Activity = \mathrm{var}(x) \quad (7)$$

$$Mobility = \sqrt{\mathrm{var}(x') / \mathrm{var}(x)} \quad (8)$$

$$Complexity = \sqrt{\mathrm{var}(x'') \times \mathrm{var}(x) / \mathrm{var}(x')^2}. \quad (9)$$

The entropy gives a measure of signal disorder and can provide relevant information in the detection of some sleep disturbs. It is computed from histogram of the EEG samples of each sleep epoch, according with (Zoubek *et al*, 2007)

$$Entropy = -\sum_{i=1}^{N} \frac{n_i}{n} \ln\left(\frac{n_i}{n}\right), \quad (10)$$

where n is the number of samples within the sleep epoch, N is the number of bins used in computation of histogram and n_i is the number of samples within the *ith* bin.

The skewness is a measure of symmetry. The kurtosis is a measure of wether the data are peaked or flat relative to a normal distribution. Defining the *kth* order moment m_k as (Zoubek *et al*, 2007)

$$m_k = \frac{1}{n}\sum_{i=1}^{n}\left(y(i) - \bar{y}\right)^k, \quad (11)$$

where n is the number of samples of an epoch and $\bar{y}$ is the mean of these samples, the skewness and kurtosis are given by

$$skewness = m_3 / m_2 \times \sqrt{m_2} \quad (12)$$

and

$$kurtosis = m_4 / m_2 \times m_2 . \quad (13)$$

Features are usually selected by wrapper or filter methods using sequential approaches. The results from wrappers methods are dependent of the choice of the classification algorithm. Our option fell on an R-square filter approach which is independent of the classifier, based on the Pearson correlation coefficient defined as (Guyon and Elisseeff, 2003):

$$\Re = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}, \quad (14)$$

where X and Y represent two random distributions of samples, and cov and var designates covariance and variance, respectively. Considering x_i and y_i as the sample values of feature i labelled with class 1 and class 2, respectively, the value $R(i)$ for the feature i is given by:

$$R(i) = \frac{\sum_{k=1}^{m}(x_{i,k} - \overline{x_i})(y_{i,k} - \overline{y_i})}{\sqrt{\sum_{k=1}^{m}(x_{i,k} - \overline{x_i})^2 \sum_{k=1}^{m}(y_{i,k} - \overline{y_i})^2}}, \quad (15)$$

where $\overline{x_i}$ and $\overline{y_i}$ represent the mean value of x_i and y_i of the m samples. The R-square, computed as $R(i)^2$, provide a level of discrimination between the two classes. High values of R-square indicate large inter-class separation and small within-class variance. The R-square provides a feature discrimination ranking.

5 BAYESIAN CLASSIFICATION

The conditional density function of the class i is modelled as a multivariate distribution under gaussian assumption

$$P(Y \mid \mu_i, \Sigma_i) = K \exp\left(-(Y-\mu_i)^T \Sigma_i^{-1} (Y-\mu_i)/2\right), \quad (16)$$

where,

$$K = 1/\left((2\pi)^{n/2}|\Sigma_i|^{1/2}\right), \quad (17)$$

Y is the feature vector resulting from concatenation of the extracted features, μ_i and Σ_i are respectively, the mean and covariance matrices computed for each class w_i from the training data. The Bayes decision function is written as:

$$\hat{w}(Y) = \arg\max\{\{\Delta_2 p(Y \mid w_1)P(w_1)\}, \{\Delta_1 p(Y \mid w_2)P(w_2)\}\}, \quad (18)$$

where $P(w_i)$ is the *ith* class prior probability and Δ_i an adjustment parameter to control the rate of false positives and false negatives (Heijden *et al*, 2004).

6 RESULTS AND DISCUSSION

The feature extraction process provides a vector of 204 features, 34 features per each EEG channel: 10 RSP, 3 SWI, 15 harmonic parameters, 3 Hjorth Parameters, 1 entropy feature, 1 skewness and 1 kurtosis. Next, the features are sorted in a decreasing order of level of discrimination by applying the R-squared based selection approach. Figure 2 shows the percentage of disagreement for wake/sleep detection between our ASSC system and expert classification (i.e. the percentage of epochs for which the automatic classification differs from manual classification made by the expert), as function of the number of features, i. e., the n-most discriminative features with n = 1,…,52. The disagreement values are obtained from a ten-fold cross validation. The lowest disagreement value was reached using the first 19 ranked features. Table 3 presents the results for each binary classifier, using 1, 2, 3, 19 most discriminative features and all 204 features. Selecting the relevant features reduces the number of features used in the ASSC leading to an increased robustness of the classifiers.

The feature selection also enables to identify the type of features and channels that lead to higher discrimination results for each 2-class discriminator

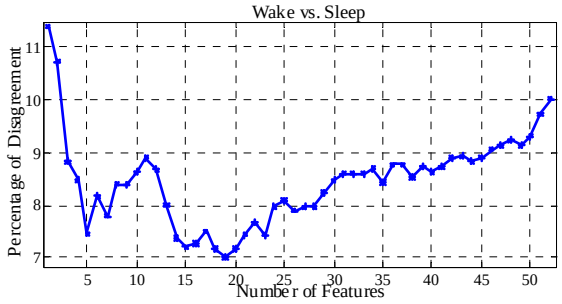


Figure 2: Percentage of disagreement vs. number of features used in wake vs. sleep classification.

Table 3: Percentage of disagreement obtained using 1, 2, 3 and the 19 most discriminative features and all 204 features.

	1	2	3	19	204
W vs. S	11,4	10,7	8,8	7,0	16,7
R vs NR	22,5	21,4	19,5	15,6	30,8
N1 vs. N2/N3	15,1	15,7	15,7	10,6	72,5
N1/N2 vs. N3	15,7	14,7	14,6	15,5	30,3
N1 vs. N2	21,9	22,6	18,5	15,6	63,9
N2 vs. N3	19,0	18,2	16,7	17,7	39,8
N1 vs. R	25,5	24,7	24,4	25,0	64,7
Mean	18,7	18,3	16,9	15,3	45,5

(Table 4). As it can be seen, the feature entropy (Ent), Skewness (Skw) and kurtosis (Krt) never appear in the 20 most discriminative features. On the other hand, the most frequents are the RSP and harmonic parameters. Analyzing the origin of the 20 most discriminative features for each case, the parameters of Hjorth (PHj) are most evident in N1/N2 vs. N3 and N2 vs. N3, but they have no weight in R vs. NR and N1 vs. R. The harmonic parameters are more frequent in W vs. S, N1 vs. N2/N3 and N1 vs. N2, but are not relevant in R vs. NR, N1 vs. N2/N3, N2 vs. N3 and N1 vs. R. For the RSP and SWI, they have a similar number of features in all discriminations, except for N1 vs. R, where the RSP has several features with good discrimination, and for N1 vs. N2, where SWI does not assume any importance. Analyzing the EEG channels, it can be seen that O1A2 (O1) and O2A1 (O2) are the most relevant in discrimination wake vs. sleep; F3A2 (F3) and F4A1 (F4) in REM vs. NREM; and C3A2 (C3) and C4A1 (C4) in N2 vs. N3. In the remaining discriminations, they all have a relatively uniform distribution, except in N1 vs. R, in which the channels O1A2 and O2A1 do not have any type of contribution. Figure 3 shows the type of features and channels that lead to higher discrimination results, taking all discriminators together. Summarizing, the best ranked discriminative features never include entropy features, skewness or kurtosis. These parameters are related to the signal shape. However, since the EEG signal patterns are very random, it is difficult to obtain useful information from these parameters.

Instead, the set of most discriminatory features between sleep stages was composed mainly by RSP and Harmonic Parameters. This result emphasizes the fact that the spectral analysis has more discriminative information than temporal signal analysis as already concluded in (Hese *et al*, 2001; Tang *et al*, 2007).

Table 4: Number of feature type and channels within the 20 most discriminative features.

		W vs. S	R vs. NR	N1 vs. N2/N3	N1/N2 vs. N3	N1 vs. N2	N2 vs. N3	N1 vs. R	Total
Features	**RSP**	5	4	6	6	5	6	13	45
	SWI	3	2	2	2	0	4	4	17
	HP	9	14	8	3	12	2	3	51
	PHj	3	0	4	9	3	8	0	27
	Ent	0	0	0	0	0	0	0	0
	Skw	0	0	0	0	0	0	0	0
	Krt	0	0	0	0	0	0	0	0
Channels	**F3**	1	6	6	3	4	2	5	27
	C3	1	3	4	5	4	5	5	27
	O1	6	1	3	4	2	4	0	20
	F4	2	5	3	2	4	1	5	22
	C4	5	3	3	4	4	5	5	29
	O2	5	2	1	2	2	3	0	15

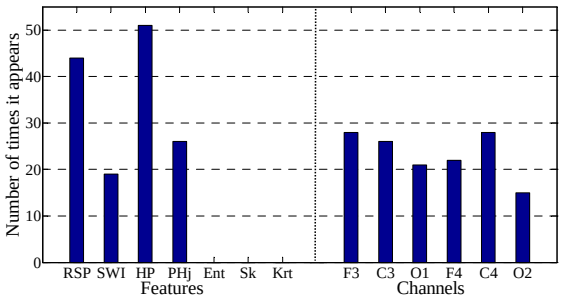


Figure 3: Number of times that each group of features and each channel appears in the 20 most discriminative features.

On the other hand, all the 6-six EEG channels provide useful features for sleep staging discrimination. Analyzing the results for each of the binary classifiers, there is greater disagreement in the case of N1 vs. R sleep. This situation relates to the fact that, in terms of EEG, the patterns presented in these two stages are very similar. Finally, a decision tree was implemented based on 2-class detection, as represented in Figure 4. At each step, a new level was introduced from a wake/sleep to all stages

classification. The results were compared with and without feature selection (Table 5). The improvements from feature selection are evident. The results obtained with our ASSC system are comparable to the ones obtained in other methods based on EEG only described in literature (zoubek *et al*, 2007; Doroshenkov *et al*, 2007).

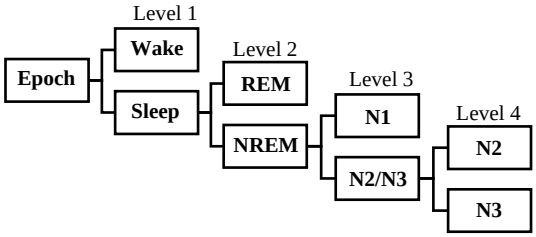


Figure 4: Decision tree based on 2-class detection.

Table 5: Disagreement obtained with using 19 most discriminative features and all 204 in 2, 3, 4 and 5 sleep stages classification.

Classification	Diasagreement (%)	
	All Features	19
2 Class	36	7
3 Class	62	18
4 Class	83	22
5 Class	83	29

7 CONCLUSIONS

In this paper, the use of several feature extraction methods was investigated in the context of EEG-based sleep staging. The first conclusion was that the most discriminative features were determined by RSP, SWI, Harmonic Parameters and Parameters of Hjorth. All the 6-EEG channels provide useful information. On the other hand, the application of the feature selection method improved, in general, the process of discrimination by selecting the set of features that provided a lower percentage of disagreement. One of the biggest problems in automatic sleep staging based on EEG is the similarity between patterns of different sleep stages such as REM and NREM N1. This can be improved recurring to other biosignals, such as EOG and EMG. Another problem in ASSC is the high level of variability between patients. Using an ambulatory system, the patient can perform periodic recordings at home. This way, the first session can be fully analysed by the expert. The labelled data can be used to obtain classification models specific to the patient. Further sessions can then use these robust user-dependent models. This approach is under research presently.

REFERENCES

Ansari-Asl. K., Chanel, G., Pun, T., A channel Selection Method for EEG Classification in Emotion Assessment Based on Synchronization Likelihood. In *EUSIPCO'07,* 1241-1245.

Doroshenkov, L., Konyshev, V., Selishchev, S., 2007, Classification of Human Sleep Stages Based on EEG Processing Using Hidden Markov Models. *Biomedical Engineering*, 41(1), 25-28.

Ebrahimi, F., Mikaeili, M., Estrada, E., Nazeran, H., 2008, Automatic Sleep Stage Classification Based on EEG Signals by Using Neural Networks and Wavelet Packet Coefficients. In *IEEE EMBS'08*, 1, 1151-1154.

Guyon, I., Elisseeff, A., 2003, An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.

Heijden, F., Duin, R., Ridder, D., Tax, D., 2004, *Classification, Parameter Estimation and State Estimation.* John Wiley & Sons.

Hese, P., Philips, W., Koninck, J., Walle, R., Lemahieu, I., 2001, Automatic Detection of Sleep Stages Using the EEG. In *IEEE EMBS'01,Proc. of*, 1994-1947.

Iber., C., Ancoli-Israel, S., Chesson, A., Quan, S., 2007, *The AASM Manual for the scoring of Sleep and Associated Events: Rules. Terminology and Technical Specifications* (1st ed.). Westchester, Illinois: American Academy of Sleep Medicine.

Rechtscheffen, A., Kales, A., 1968, *A Manual of Standardized Terminology, Techniques ans Scoring System for Sleep Stages of Human Subjects*, US Government Printing Office, National Institute of Health Publications, Washington DC.

Schwaibold, M., Harms, R., Scholer, B., Pinnow, I., Cassel, W., Penzel, T., Becker, H., Bolz, A., 2003, Knowledge-Based Automatic Sleep-Stage Recognition – Reduction in the Interpretation Variability. *Somnologie*, 7, 59-65.

Tang, W. C., Lu, S. W., Tsai. X. M., Kao. C. Y., Lee. H. H., 2007, Harmonic Parameters with HHT and Wavelet Transform for Automatic Sleep Stages Scoring. *Proc. of World Academy of Science. Engineering and Technology*, 22, 414-417.

Yilmaz, A., Alkan, A., Asyali, M., 2007, Applications of parametric spectral estimation methods on detection of power system harmonics. *Electric Power Systems Research* (2008), 78, 683-693.

Zoubek, L., Charbonnier, S., Lesecq, S., Buguet, A., Chapolot, F., 2007, Feature selection for sleep/wake stages classification using data driven methods. *Biomedical Signal Processing and Control*, 2, 171-179.

SOLUTION OF AN INVERSE PROBLEM BY CORRECTION OF TABULAR FUNCTION FOR MODELS OF NONLINEAR DYNAMIC SYSTEMS*

I. A. Bogulavsky
State Institute of Aviation Systems, Moskow Physical - Technical Institute, Moscow, Russia
bogus@gosniias.ru

Keywords: Polynomial approximation method, Uniformly small estimate errors, Mean-square optimal estimate.

Abstract: In this paper, we present a solution to the problem of correction of parameterized tabular nominal functions for the motion equations in a model of a nonlinear dynamical system using observations in discrete time. The correction vector is determined by the mean of the multi-polynomial approximation algorithm (MPA-algorithm) using observations of the noise functions of the components of the state vectors. The method of correction of tabular functions is demonstrated by correcting 204 parameters in an example involving a mathematical model of the motion of an F-16 aircraft.

1 INTRODUCTION

In this paper, an algorithm is presented for the a numerical process of correction of a nominal mathematical model of a nonlinear dynamical system based on experimental data. The problem of estimation of the vectors of mathematical model parameters (the traditional problem of identification of the constant unknown parameters) has been considered in many prior works (see, for example, Klein and Morelli, 2006, Cappe et al., 2005, Gordon et al., 1993, Doucet et al., 2000, Doucet et al., 2001, Ristic et al., 2004, Gosh et al., 2008, Namdeo et Manohar, 2007, Cotter et al., 2009, Boguslavskiy, 1996, Boguslavskiy, 2006, Boguslavskiy, 2008 and Boguslavskiy, 2009).

The statement of the problem we consider here differs from the traditional problem in that the nominal model –the model before the correction– contains several nominal (previously obtained) tabular functions of the components of the state vectors. The problem of identifying the correction vectors is as follows: it is necessary to construct an algorithm to correct the tabular functions by processing the observation data . To do this it is necessary to identify parameters that are not constant and depend on a flowing state vector of a dynamic system. These parameters are hidden (sleeping); they do not influence the evolution of system if the current state vector has not visited the corresponding areas of the phase space.

*This work has been supported by the Russian Foundation for Basic Research.

This singularity distinguishes our problem from traditional problems in which the evolution of the state vector does not influence the constant unknown parameters. The task differs from the task set in (Cotter et al., 2009), where the Bayesian approach is used to estimate a function of time that belongs to the mathematical model of a dynamical system.

An example of a model with a tabular function is the mathematical model of an aircraft with aerodynamic characteristics, i.e., the dimensionless coefficients of aerodynamic forces and moments (Klein and Morelli, 2006), given from tables of functions of components of state vectors (e.g., angles of attack and sliding) and components of the vector of control (e.g., angles of deviation of steering surfaces).

Modern computational methods and wind tunnel testing can provide, in many instances, comprehensive data about the nominal aerodynamic characteristics of the aircraft; these comprise the parameters of the mathematical model.

” However, there are still several motivations for identifying aircraft models from flight data, including:

1. Verifying and interpreting theoretical predictions and wind-tunnel test results (flight results can also be used to help improve ground-based predictive methods);

2. Obtaining more accurate and comprehensive mathematical models of aircraft dynamics for use in designing stability augmentation and flight control systems;

3. Developing flight simulators, which require an accurate representation of the aircraft in all flight regimes (many aircraft motions and flight conditions simply cannot be duplicated in the wind tunnel or computed analytically with sufficient accuracy or computational efficiency);

4. Expanding the flight envelope for new aircraft, which can include quantifying stability and predicting or controlling the impact of aircraft modifications, configuration changes, or special flight conditions;

5. Verifying aircraft specification compliance" (Klein and Morelli, 2006).

The nominal parameters for the problem of identifying actual aerodynamic characteristics are values that correspond to knots of one-dimensional or two-dimensional tables.

The correction vector of nominal (rated) parameters, defined by an algorithm handling the streams of digital information from the aircraft transmitters, has a very high dimension that is on the order of several tens or hundreds.

It should be noted that at NASA, projects based on the theory and practice of identification of aircraft by means of test flights are widely applied. An application of (Klein and Morelli, 2006) in the internet software package SIDPAC is published in the MATLAB M-files language (systems identification programs for aircraft), representing an implementation of the numerous algorithms recommended by NASA for identification problems.

The most common method of identification is the known nonlinear method of least squares, where the sum of the squares of the discrepancies, i.e., the differences between the actual measurements and their rated analogues, obtained by numerical integration of the system's equations of motion is computed for some realization of a vector of unknown parameters. The outcome of a successful identification accepts the vector of parameters, supplying a global minimum to the mentioned sum of squares of the discrepancies.

It is necessary to note that this criterion is statistically justified only for linear problems of identification, problems in which the measurements are linear in the unknown vector of parameters.

Significant computing difficulties arise when implementing a nonlinear method of least squares to correct the nominal parameters of an aircraft according to its test flights. The difficulties arise due to the large dimension of the correction vector and due to the existence of numerous relative minima for the sum of the squares of the discrepancies as functions of the correction vector, and also because of the use of variants of Newton's method, which requires a sequence of local linearizations to define the stationary points of the function.

The authors of the monograph (Klein and Morelli, 2006) presented a detailed exposition and analysis of known algorithms for the identification of parameters of the dynamic systems in chapters of[1]:[4 - 8]. However, only a regression method can be used for a practical investigation. The regression method presented in (Klein and Morelli, 2006) solves this problem subject to the following restrictions:

1. All components of the state vector are measured.

2. The algorithm builds a vector of estimates for the vector of derivatives $\dot{dx}$ at the moments of measurement,

3. The vector functions on the right-hand side of the equations of motion linearly depend on the estimated vectors.

4. Prohibition of mathematical modeling without the use of a Monte-Carlo method to analyze the theoretical observability of the components of the identified parameter vector if the laws of control are set beforehand by test flights of the aircraft and information about random errors of its transmitters.

In the monograph (Cappe et al., 2005), the problem of estimating the parameters is considered within the limits of the common problem of smoothing; this consists of the problem of constructing approximate conditional expectations for elements in a non-observable sequence if these elements influence observable elements by means of a given statistical mechanism. Various approaches to solving the smoothing problem by means of expectation-maximization (EM) methods are stated and investigated. However, the maximization operation requires the definition of a point global (but non-local!) extremum, that is generally not guaranteed by numerical methods.

In the last ten years, a significant number of studies (Gordon et al., 1993; Doucet et al., 2000; Doucet et al., 2001; Ristic et al., 2004; Gosh et al., 2008; Namdeo et Manohar, 2007), were published that represented the basic solution of the problem as the definition of the conditional expectation of a vector of parameters for a mathematical model of a nonlinear dynamical system. By means of multiple applications of Bayes' formula to the state vectors and with numerical quadratures, it is easy to determine the recurrence equations for the probability density function (pdf) of state vectors of the dynamical system. The actual solution of these recurrence equations, however, is not feasible because of the unwieldy dimensions of the integrals involved (Namdeo et Manohar, 2007). Therefore, several alternative strategies have been developed. One set of such alternatives consists

of developing suboptimal filtering, such as that based on linearization or transformations, and the other consists of methods that employ Monte Carlo simulation strategies (e.g., estimation with the sequential importance sampling particle filter) to approximately evaluate the multidimensional integrals in a recursive manner.

This direction is perceptive, but it is bulky and not suitable for the solution of practical applied (instead of model!) problems of nonlinear identification.

The MPA algorithm (Boguslavskiy, 1996; Boguslavskiy, 2006; Boguslavskiy, 2008; Boguslavskiy, 2009) makes use of this paper for the correction of tabular functions. The MPA algorithm is a new recursive algorithm that asymptotically and accurately solves the nonlinear problem of construction of the conditional expectation vectors of the vector of parameters for mathematical models of nonlinear dynamical systems, including random errors and perturbations with given distributions. Therefore, the MPA algorithm approximately solves the problem of quasi-optimal mean-squares estimation.

The MPA algorithm has none of the disadvantages of the NACA algorithm noted above, and in principle it differs from all of the abovementioned algorithms; a theoretical proof of its accuracy is presented in (Boguslavskiy, 1996; Boguslavskiy, 2006; Boguslavskiy, 2008; Boguslavskiy, 2009).

2 A STATEMENT OF THE PROBLEM OF CORRECTION OF FUNCTIONS DETERMINED IN THE FORM OF TABLES

Let $x \in R^m$ be a moving state vector of the mathematical model of the dynamical system and the components of the vector z be a subset of the components of x : $z \in x, z \in R^r, r < m$. The vector z is an argument of the tabulated functions. We suppose that the vectors z belong to the interior boundary of a parallelepiped $\Omega \in R^r$ under all realizable conditions of the dynamical system – a realizable vector of functions of $t : u(t)$. Choose a Cartesian coordinate system for R^r with axes parallel to the edges of the parallelepiped Ω. The parallelepiped Ω is covered by nodes numbered s^r, where s is given as an integer.

Here s^r is the number of nodes on which the table determines the tabulated functions. We denote each node by $z(i_1,...,i_r), 1 \le i_1,...i_r \le s$, where the integers $i_1,...i_r$ are the indices of the coordinates of this node. The nodes $z(i_1,...,i_r)$ are vertices .of the small parallelepipeds belonging to $\Omega \in R^r$.

We suppose that a presentation (a description) of the nominal mathematical model contains K tabular functions $\vartheta_j(z(i_1,...,i_r)), j = 1,...,K$, which are given on the specified nodes. The values $\vartheta_j(z(i_1,...,i_r))$ are the nominal values of the tabular function, which, according to the preceding experiments or in theory, all represent the vertices of the small parallelepipeds.

Each tabular function $\vartheta_j(z(i_1,...,i_r))$ Is the skeleton of a continuous function $f_j(z), j = 1,...,K$ as follows : 1) on the vertices of the small parallelepiped the values of the continuous function coincide with the values of the tabular function; 2) on other points of the parallelepiped, the values of the continuous function are linear combinations of the values of the tabular function at the vertices, such that these values are multilinear functions of the values of the tabular function.

Let $L_1,...,L_r$ be the lengths of the edges of any of the small parallelepipeds. Then in the coordinates $tt^1(\alpha_1),...,tt^r(\alpha_r)$ of the 2^r vertices, we can write the formula $tt^k(\alpha) = tt_0^k + L_k\alpha_k, k = 1...,r$, where $\alpha_1...,\alpha_r$ are independently determined with the values 0, 1.The values $\alpha_1 = 0,...,\alpha_r = 0$ correspond to the minimal values of all coordinates for a given small parallelepiped.

Let $z^1,...,z^r$ be the components of the vector xx, $\varphi_0(z^k) = L_k^{-1}(tt^k(1) - z^k), \varphi_1(z^k) = L_k^{-1}(z^k - tt^k(0)), k = 1,...,r$ be linear functions of these components $\varphi_0(z^k) = 0$ if $z^k = tt_0^k, \varphi_0(z^k) = 1$ if $z^k = tt_0^k + L_k$, $\varphi_1(z^k) = 0$ if $z^k = tt_0^k + L_k, \varphi_1(z^k) = 1$ if $z^k = tt_0^k$.

Then the components of the multilinear function take the form

$$f_j(z^1,...,z^r) = \sum_{\alpha_1,\cdots,\alpha_r=0,1} \varphi_{\alpha_1}(z^1)\cdots\varphi_{\alpha_r}(z^r)$$

$$\vartheta(tt_0^1 + L_1\alpha_1,...,tt_0^r + L_r\alpha_r), \qquad (2.1)$$

where the sum is over all binary values of an aspect $\alpha_1 \cdots \alpha_r$ and over all 2^r of the items.

The vector $f_j(z^1,...,z^r)$ is a nominal representation of the continuous functions defined by the tabular functions $\vartheta_j(z(i_1,...,i_r))$.

The corrections of the K tabular functions $\vartheta_j(z(i_1,...,i_r)), j = 1,...,K$ replace the values $\vartheta_j(...)$ with values $\vartheta_j(...) + \theta_j(...)$, where the correction terms $\theta_j(...)$ are the results of processing the new data.

We shall designate by $f_j(z^1,...,z^r,\theta_j)$ the functions obtained from $f_j(z^1,...,z^r)$ by substituting) functions $\vartheta_j(...) + \theta_j(...)$ for $\vartheta_j(...)$ in (2.1)

The new data are the results of observations values $y_1,...,y_N$ where $y_k = H_k(x_k) + \xi_k$, $y_k, x_k, k = 1,...,N$ are the results of observations of the vectors x at discrete instants, $H_k(...)$ are the given functions, and ξ_k are random errors with a given distribution.

Using the MPA algorithm, the correction vector is quasi-optimal in the mean square estimation of the variations of the components of the nominal vectors $\vartheta_j(z(i_1,...,i_r)), j = 1,...,K$. The experiment responds to these variations with new data about the dynamical system. The vector of the estimates has $K \times s^r$ components $\hat{\theta}_j(i_1,...,i_r), j = 1,...,K, 1 \le i_1,...,i_r \le s$.

The MPA algorithm is the Bayesian estimator of the parameters (Boguslavskiy, 2006). A priori information, which is necessary for the training and adjustment of the MPA algorithm by means of the Monte Carlo method, includes the segment lengths of expected scattering of values $\theta_j(...)$ of the corrections of the nominal tabular functions, expressed in terms of components of these functions. Therefore, the sum $\vartheta_j(...) + \theta_j(...)$ replaces random values $\vartheta_j(...)(1 + \rho_j \varepsilon(i_1,...,i_r))$, where $0 < \rho < 1$ and random the values $\varepsilon(...)$ are uniformly distributed on the segment $[-1,1]$.

If the nominal tabular function is smooth with respect to the components of the vector *xx*, i.e., it varies smoothly with changes in this component, then it is appropriate to require this smoothness to be preserved after the correction. The requirement is satisfied if the correction includes the increments. of the nominal values of the tabular function, This increments are the tabular functions obtained by varying a given component in the transfer from the given node to the subsequent node. These increments are approximately the derivatives of the tabular functions with respect to the given component.

If by a movement of the dynamical system the moving vector z is found inside or on the boundary of any small parallelepiped, then the equations of the model use domains of the nominal tabular functions $\vartheta_j(z(i_1,...,i_r))$ and the corresponding continuous functions for which the points $z(i_1,...,i_r)$ belong to this small parallelepiped Therefore the correction of values $\vartheta_j(z(i_1,...,i_r))$ is impossible if by a given $u(t)$ the current state vector x does not visit at any instant the small parallelepiped with the vectors $z(i_1,...,i_r)$. The significant influence of the area circumscribed by the control $u(t)$ on the correction result essentially distinguishes the problem of identification of the vector correction of tabular functions from the traditional problem of the identification of the parameters.

We illustrate the definitions on an example modeling the motion of an aircraft. The continuous functions corresponding to the tabular functions are piecewise linear approximations for the dimensionless coefficients of the aerodynamic forces and moments as the functions of the angles of attack.

We present a formalized statement of the problem of the correction of the tabular function, which is the problem of quasi-optimal identification of the variations of this function. The variations are estimated using data for the observed motion of the system. The equations of the motion model are written in the form

$$dx/dt = F(x, f_1(z^1,...,z^r,\theta_1),..., f_K(z^1,...,z^r,\theta_K), u, t). \tag{2.2}$$

The equation of the observation is written in the form

$$y_k = H_k(x_k) + \xi_k, \tag{2.3}$$

where $k = 1,...,N$.

Further, we designate by Y_N the vector whose components are $y_1,...,y_N$. In verifying the quality of the correction after realization of the estimations $\hat{\theta}_j(i_1,...,i_r), j = 1...,K, 1 \le i_1,...,i_r \le s$, we can use the sum of quadrates of differences, which is

$$\sum_{k=1,...,N} (y_k - \hat{y_k})^2,$$

where the values $\hat{y_k}$ are computed from (2.2), (2.3) after substituting the values $\hat{\theta}_j(i_1,...,i_r), i = 1,...,K, 1 \le i_1,...,i_r \le s$ for the values $\theta_j(i_1,...,i_r), i = 1,...,K, 1 \le i_1,...,i_r \le s$

3 IDENTIFICATION OF SOME PARAMETERS OF F-16 AIRCRAFT

1. The Driving Equations

It follows from monograph (Klein and Morelli, 2006) that the driving equations of an F-16 plane stabilize with respect to the magnitude of the airspeed velocity V and roll angle rotation ϕ ($\dot{V} = \dot{\phi} = 0$) and can be approximated in the following form:

$$\dot{\alpha} = q + (g cos\upsilon cos\phi + \bar{q}SC_Z/M)/V,$$

$$\dot{\beta} = -r + (g cos\upsilon sin\phi + \bar{q}SC_Y/M)/V,$$

$$p = 0,$$

$$\dot{q} = 160c_7 r + c_6 r^2 + \bar{q}S\bar{c}c_7 C_m,$$

$$\dot{r} = -(c_2 r + 160c_9)q + \bar{q}Sbc_9 C_n,$$

where the pitch angle rotation υ and the yaw angle rotation $\phi \simeq$ are constants (within a small maneuvering time), the angle of attack is α, the sideslip angle is β, p,q,r are the body-axis components of the aircraft's angular velocity, $\bar{q}$ is the dynamic pressure, S is the wing reference area, b is the wing span, $\bar{c}$ is the mean aerodynamic chord of the wing, M, c_2, c_6, c_7, c_9 are constants (see (Klein and Morelli, 2006), C_Y, C_Z, C_m, C_n are non-dimensional

coefficients that are directly proportional to the aerodynamic forces and moments

$$C_Y = -0.02\beta + 0.086(\delta_r/30) + (b/2V)C_{Y_r}(\alpha)r,$$

$$C_Z = C_{Z_0}(\alpha)(1-(\beta\pi/180)^2) - 0.19(\delta_s/25) + (\bar{c}/2V)C_{Z_q}(\alpha)q,$$

$$C_m = C_{m_0}(\alpha,\delta_s) + (\bar{c}/2V)C_{m_q}(\alpha)q + 0.1C_Z,$$

$$C_n = C_{n_0}(\alpha,\beta) + C_{n_{\delta_r}}(\alpha,\beta)(\delta_r/30) + (b/2V)C_{n_r}(\alpha)r - 0.1(\bar{c}/b)C_Y,$$

δ_s is the stabilizer deflection,δ_r is the rudder deflection, and $\alpha,\beta,\delta_s,\delta_r$ are in degrees.

Further, the functions $C_{Y_r}(\alpha), C_{Z_0}(\alpha), C_{Z_q}(\alpha)$, $C_{m_0}(\alpha,\delta_s), C_{m_q}(\alpha), C_{n_0}(\alpha,\beta), C_{n_{\delta_r}}(\alpha,\beta), C_{n_r}(\alpha)$ are nominal functions. They are defined from the vertices of the parallelepiped $\Omega \in R^4$ by $s = 8$. The functions accept ratings that are evaluated using experiments in a wind tunnel at a finite number of reference nodes, covering the domains Ω. The vectors of the nominal experimental data are $\vartheta_1(...),...,\vartheta_8(...))$ in the equations (2.1). For the functions $C_{Z_0}(\alpha), C_{Y_r}(\alpha), C_{Z_q}(\alpha), C_{n_r}(\alpha), C_{m_q}(\alpha)$ the vectors $\vartheta_1(...),...,\vartheta_5(...)$ have dimensions 12×1; accordingly, for the functions $C_{m_0}(\alpha,\delta_s), C_{n_0}(\alpha,\beta)$ the vectors $\vartheta_6(...),\vartheta_7(...)$ have dimensions 12×5 and 12×7. Furthermore, we do not correct the function $C_{n_{\delta_r}}(\alpha,\beta)$. Therefore, the number of nominal parameters defining these functions equals $12\times(5+5+7) = 204$.

The software package *SIDPAC* contains a file *F16 AERO SETUP Generates aerodynamic data tables* with ten one- and two-dimensional tables of nominal experimental data.

We emphasize that the nominal functions mentioned above are nonlinear functions of the arguments.

2. Parametric Model of Aerodynamic Parameters of the Subjects of Identification

We suppose that for all functions except $C_{\delta_r}(\alpha,\beta)$, the nominal experimental data differ from the true data by some random error vectors, which are designated $\theta_1,...,\theta_7$.

We consider the most complicated problem for the MPA algorithm, where at each of the points of the table, the actual parameter differs from the nominal parameter by a random magnitude subject to the a priori limits θ_i.

After collecting the measurements of the parameters of the perturbed driving of the aircraft, the MPA identification algorithm should estimate the 204 components of the vector of random errors, generating the vector of differences between the actual and nominal parameters.

Let A_i and B_i $(i = 1,...,204)$ be the i-th components of the nominal and actual (perturbed) vectors of the aerodynamic parameters corresponding to the 204 actual parameters subject to identification.

We suppose a fair parametrical model:

$$B_i = A_i + \Delta_i,$$

The vector Δ is the vector of perturbations of nominal parameters, i.e., the vector of errors of the aerodynamic parameters, and its component estimates are subject to our identification. For the structure of these components, we give the formula

$\Delta_i = A_i\rho_i\varepsilon_i, 0 < \rho_i < 1, -1 \le \varepsilon_i \le 1.$

The positive number ρ_i defines the maximum magnitude that the ratio of the random variable of perturbations Δ_i and nominal parameter A_i is allowed to attain under the conditions of our identification algorithm. Each ε_i is a random number that is uniformly and independently distributed.

3. Transients of Characteristics of the Nominal and Perturbed Movements

We suppose as above that the transients in the reduced driving equation of the aircraft F-16 are α,β,q,r over 20 sec., if at $t = 0$ $\alpha = \beta = 0.3rad., q = r = 10deg/sec$ and magnitudes δ_s,δ_r are constant and equal to $10deg$.

We shall discuss the precision of the estimate under the following assumptions: during the 20 sec. period, the current magnitudes α,β,q,r are measured at intervals of 0.05 sec ($N = 1600$). We suppose that the random errors of measurement are discrete white noise, which is limited by the product of the true measured magnitudes on the magnitude of the set ε. We shall suppose that the MPA algorithm supplies the magnitudes $\widehat{\Delta}_i, i = 1,...,204$, which are the estimates of the magnitudes $\Delta_i, i = 1,...,204$. To characterize the relative precision of the identification of the random parameters Δ_i we define ratios $\varepsilon_i = (\widehat{\Delta}_i - \Delta_i)/\Delta_i$

We must emphasize that the state vector of the aircraft corresponding to the modeled transient does not visit all the reference points in which the nominal experimental data are set. Therefore, for some values of i, the magnitude ε_i has an order of 1 or more. The corresponding values ϑ_i are not observable for the modeled transient, and also cannot be corrected by means of the MPA algorithm.

Table 1 presents a histogram of ε_i.

Table 1.

$2 \ge \lvert\varepsilon_i\rvert \ge 1$	$1 \ge \lvert\varepsilon_i\rvert \ge 0.5$	$0.5 \ge \lvert\varepsilon_i\rvert \ge 0.25$
37	44	37
$0.25 \ge \lvert\varepsilon_i\rvert \ge 0.1$	$0.1 \ge \lvert\varepsilon_i\rvert \ge 0.05$	$0.05 \ge \lvert\varepsilon_i\rvert$
29	17	17

The practical purpose of identification is to correct the nominal experiment data, and the outcome is to replace the nominal aerodynamic parameters with new

parameters. If the errors of identification are small, the driving characteristics of the aircraft, obtained by numerical integration after correction, should be close to the perturbed driving characteristics of the aircraft, as discovered early in real flight or by means of modeling. In a real flight situation, the errors of the nominal experimental data can only be estimated only over time and by observing the perturbed driving characteristics.

In Table 2 , we present the ratios of the difference between the characteristics of the corrected and perturbed movements and the difference between the characteristics of the nominal and perturbed movements as functions of discrete time with increments of 1 sec.

In Table 2, for example, the expression $\delta^{c,p}_{n,p}\alpha$ designates the difference ratio $(\alpha(corr) - \alpha(perturb))/(\alpha(nomin) - \alpha(perturb))$.

The labels $\delta^{c,p}_{n,p}\beta$, $\delta^{c,p}_{n,p}q$, $\delta^{c,p}_{n,p}r$ are defined similarly.

These ratios show how quickly the MPA algorithm reduces the difference between the corrected and perturbed movements compared to the difference between the nominal and perturbed movements.

Table 2.

sec	$\delta^{c,p}_{n,p}\alpha$	$\delta^{c,p}_{n,p}\beta$	$\delta^{c,p}_{n,p}q$	$\delta^{c,p}_{n,p}r$
0	-0.086	-0.050	0.215	-0.040
1	-0.094	-0.064	0.048	-0.112
2	-0.112	-0.067	0.012	-0.329
3	0.110	-0.054	-0.011	-0.024
4	-0.082	-0.050	-0.022	-0.130
5	-0.019	-0.046	-0.040	-0.083
6	-0.022	-0.042	-0.168	-0.114
7	-0.022	-0.039	-0.003	-0.126
8	-0.022	-0.036	-0.0105	-0.141
9	-0.022	-0.034	-0.029	-0.154
10	-0.022	-0.032	0.001	-0.079
11	-0.022	-0.031	-0.009	0.001
12	-0.022	-0.030	-0.007	0.108
13	-0.022	-0.028	-0.008	0.233
14	-0.022	-0.026	-0.008	0.297
15	-0.022	-0.025	-0.009	0.345
16	-0.022	-0.024	-0.009	0.0382
17	-0.022	-0.022	-0.009	0.434
18	-0.022	-0.021	-0.009	0.486
19	-0.022	-0.020	-0.009	0.599
20	-0.022	-0.019	-0.010	0.722

It follows from Table 2 that the corrected characteristics become close to, and often coincide with, the perturbed driving characteristics (to within 2 digits after the decimal point), in the absence of observational errors.

4 CONCLUSIONS

The data presented in this work show that a multipolynomial approximation algorithm can form a computational basis for creating an effective solution for inverse problems, thus identifying the parameters of a nonlinear dynamical system, including the system of aerodynamic parameters of an aircraft.

REFERENCES

Klein V., Morelli A.G. (2006) Aircraft System Identification: Theory and Methods In *AIAA.*

Cappe O., Moulines E., Ryden T. (2005) Interference in Hidden Markov Models In *Springer-Verlag, NewYork.*

Gordon N.J., Salmond, D.J., and Smith, A.F.M. (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation In *IEE Proceedings-F* 140, PP. 107-113.

Doucet, A., Godsill, S., and Andrieu C. (2000) On sequential Monte Carlo sampling methods for Bayesian filtering In *Statistics and Computing* 10, PP. 197-208.

Doucet A., de Fraitas, N., and Gordon, N. (2001) Sequential Monte Carlo Methods in Practice In *Springer-Verlag, New York.*

Ristic, B., Arulampalam, S., and Gordon, N. (2004) Beyond the Kalman Filter - Particle Filters for Tracking Applications, Artech House, Boston, London.

Ghosh,S, Manohar C.S., and Roy D. (2008) Sequential importance sampling filters with a new proposal distribution for parameter identification of structural systems In *Proceedings of Royal Society of London* A, 464, 25-47.

Namdeo V, and Manohar C.S. (2007) Nonlinear structural dynamical system identification using adaptive particle filters In *Journal of Sound and Vibration* 306, 524-563.

S L Cotter, M Dashti, J C Robinson and A M Stuart (2009) Bayesian inverse problems for functions and applications to fluid mechanics In *Inverse Problems* 25, 115008 (43pp).

Boguslavskiy J.A. (1996) A Bayes estimations of nonlinear regression and adjacent problems. In *Journal of Computer and Systems Sciences International* 4, , pp. 14 - 24.

Boguslavskiy J.A. (2006) Polynomial Approximations for Nonlinear Problems of Estimation and Control Fizmat, MAIK.

Boguslavskiy J.A. (2008) Method for the Non-linear identification of Aircraft Parameters by Testing Maneuvers In *International Conference on Numerical Analysis and Applied Mathematics* AIP Conf. Proc., 2008, V. 1048, pp 92-99.

Boguslavskiy J.A. (2009) A Bayes Estimator of Parameters of Nonlinear Dynamic Systememes In *Mathematical Problems in Engineering, 2009.*

Stone M. (1937) Applications of the Theory of Boolean Rings to General Topology In *Trans. Amer. Math. Soc.* -1937,-V.41.-P.375-481

MULTI-TERMINAL BDDS IN MICROPROCESSOR-BASED CONTROL

Václav Dvořák
Faculty of Information Technology, Brno University of Technology, Božetěchova 2, Brno, Czech Republic
dvorak@fit.vutbr.cz

Keywords: Microprocessor-based Control, Multi-Terminal Binary Decision Diagrams, MTBDD, Optimal Variable Ordering, Arbiters.

Abstract: The paper addresses software implementation of logic-intensive control algorithms whose implementation with the smallest memory footprint is often required in embedded systems. A presented heuristic method of Multi-Terminal Binary Decision Diagram (MTBDD) synthesis aims to minimize the cost of a resulting diagram and thus the required amount of memory to store it. Evaluation of Boolean functions then reduces to traversing a MTBDD, one or more variables in a single step, according to a required speed. In terms of program execution, the evaluation process essentially does a sequence of indirect memory accesses to dispatch tables. The presented method is flexible in making trade-offs between performance and memory consumption and may be thus useful for embedded microprocessor or microcontroller software.

1 INTRODUCTION

A microprocessor-based control system is today a fundamental component in many of the industrial control and automation applications. The new programmable logic controllers (PLCs) are based on embedded PC processors and are sometimes also referred to as programmable automation controllers (Gilvarry, 2009). Beside the operating system, an embedded PC uses a runtime environment for simulation of a PLC (soft PLC). New hardware platforms (such as the combination of the Intel Atom processor paired with the Intel System Controller Hub) offer low power consumption and footprint for fanless embedded applications. Performance and memory space depend on software that must offer typical control functions such as digital logic, PID, fuzzy logic and the capability to run model-based control. In this paper we are interested only in space- and time-efficient digital logic control based on evaluation of Boolean functions.

With a changeover from traditional PLC (Petruzella, 2004) to open platforms mentioned above, we think that the time is ripe to change also algorithms and programming of logic-intensive control: to trade off serial evaluation of Boolean functions for simultaneous group evaluation, redundant reading of Boolean variables for read-once techniques, ladder diagrams (Petruzella, 2004) for cube notation and Multi-Terminal Binary Decision Diagrams (MTBDDs). Beside PLCs, software evaluation of Boolean functions has been used in other areas like digital system simulation, formal verification and testing or specialized event processing (Sosic, 1996), where either a speed or a required memory were not that important. On the contrary, in embedded systems we do care for performance and memory space as well as for power consumption. We will demonstrate that presently used algorithms (ladder diagrams, PLA emulation, BDDs) are generally too slow and that faster evaluation is feasible.

Software implementation of Boolean functions will be assumed in a flexible form of a data structure describing the function and of a compiled program that reads the input vector and evaluates the function with the use of this data structure. The size of the code and of the data structure is one figure of merit, the other is the evaluation time from reading the input to generating the output.

The paper is structured as follows. In the following Section 2 we explain representation of Boolean functions by means of cubes and decision diagrams. In Section 3 we construct a MTBDD for the sample function specified by cubes using our heuristic approach for minimizing the MTBDD cost (and thus the size of relevant data structures – dispatch tables). In Section 4 we exemplify creation of branching programs and dispatch tables on the

Round Robin (RR) arbiter and show how to trade speed of evaluation for memory space. Results of MTBDD construction for RR arbiters of various size are also presented. The results are commented on in Conclusions.

2 CUBES AND DECISION DIAGRAMS

To begin our discussion, we define the following terminology. A system of m Boolean functions of n Boolean variables,

$$f_n^{(i)} : (Z_2)^n \to Z_2, \quad i = 1, 2, ..., m \qquad (1)$$

will be simply referred to as a multiple-output Boolean function F_n. Instead of a full function table, we prefer to use a shorthand description of a system (1) in a form of a PLA matrix, i.e., as a set of $(n+m)$-tuples, called *function cubes*, in which an element of $\{0, -, 1\}^n$ is called an *input cube* and element of $\{0, -, 1\}^m$ is called an *output cube*.

Symbols {0, 1, -} in the PLA matrix are interpreted the following way: each position in the input plane corresponds to an input variable where a (1) 0 implies that the corresponding input literal appears (un-)complemented in the product term. The uncertain value "-" can be either 0 or 1.

Definition 1. Compatibility relation ~ is defined on the set {0, 1, -}: all pairs except the pairs (0,1) and (1,0) are compatible (0 ~ 0, 1 ~ 1, - ~ -, 0 ~ -, 1 ~ -, - ~ 0, - ~ 1).

Compatibility relation is extended to cubes $\{0, -, 1\}^n$: two cubes are compatible if all their homothetic elements are compatible (Brzozowski, 1997).

Definition 2. A binary operation * (intersection or product) is defined on the set {0, 1, -}:
0*0 = 0, 1*1 = 1, -*- = -,
0*- = -*0 = 0, 1*- = -*1 = 1.
Operation * is not defined for pairs (0,1) and (1,0).
The intersection can be further extended to two or more compatible cubes if it is applied element-wise.

Function F_n is incomplete if it is defined only on set $D \subset (Z_2)^n$; $(Z_2)^n \setminus D = X$ is the don't care set (DC-set). The elements in X are input vectors that for some reason cannot occur. Our concern will be an incompletely specified integer (R-valued) function of n Boolean variables

$$F_n: D \to Z_R, \qquad (2)$$

$D \subseteq (Z_2)^n$, $Z_R = \{0,1,2, \ldots, R-1\}$, $R \le 2^m$, such that no two input cubes are compatible. A min-term applied to the input is thus contained in one and only one input cube. This restriction greatly simplifies algorithms described later on, and can be lifted in future. Output cubes are integer values that can be recoded back to output *binary* vectors $b \in \{0,1\}^m$ when desired. Function F_n is not defined on a don´t care set $X = (Z_2)^n \setminus D$.

We will use a function F_4: $D \to Z_5$, $D \subset (Z_2)^4$ with a map at Fig. 1 as a running example of a class of functions under our consideration. Here 6 cubes are mapped into 5 integer values. The function is not defined in $|X| = 6$ out of 16 points.

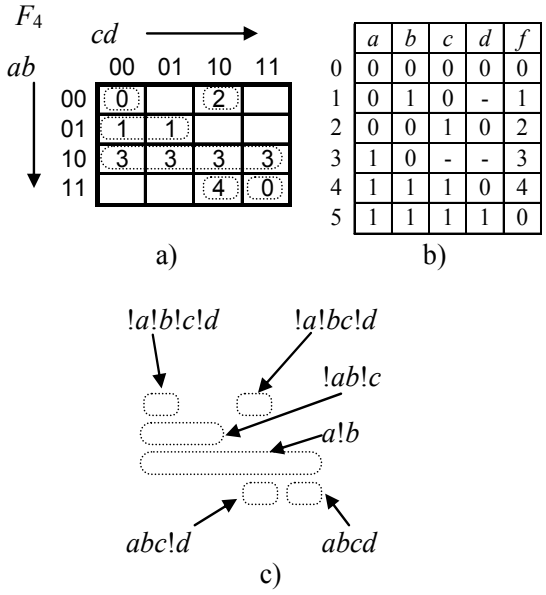


Figure 1: (a) The map of integer function F_4, (b) the equivalent cube specification and (c) product terms.

Machine representation of single-output Boolean functions frequently uses Binary Decision Diagrams (BDDs), which can have many forms, [Yanushkevich, 2006]. Integer-valued or multiple-output Boolean functions are frequently represented by Multi-Terminal Binary Decision Diagrams (MTBDDs) or by BDD for the characteristic function (BDD_for_CF), (Matsuura, 2007). The latter type has a drawback of a large size because input as well as output variables are used as decision variables; it is also more difficult to work with. From now on, we will therefore use only MTBDDs.

The DD size is the important parameter as it directly influences the amount of memory storing the DD data structure. However, the size of a DD is

very sensitive to variable ordering and finding a good order even for BDDs is an NP-complete problem (Yanushkevich, 2006). The size of DDs for random functions grows exponentially with the number of variables n for any ordering, but functions used in digital system design with few exceptions do have a reasonable DD size. One exception is the class of binary multipliers: for all possible variable orderings, the BDD size is exponential for n-bit inputs and $2n$-bit output (Bryant, 1991).

We will refer to BDDs or MTBDDs with the best variable ordering as to the optimal DDs. The term a "sub-optimal DD" will denote a DD with a size near to the optimal BDD.

3 MTBDD SYNTHESIS FROM CUBE SPECIFICATION

In this section we present a heuristic technique for a sub-optimal MTBDD synthesis. It is a generalization of the BDD construction by means of iterative disjunctive decomposition (Dvořák, 2007). Input variables are selected one after another in such a way that MTBDD cost is locally minimized.

Before formulation of the algorithm, we prefer to illustrate the synthesis technique on the F_4 example in Fig. 2. The integer function $z = F_4(a, b, c, d)$ of four binary variables is specified by cubes at the top of Fig. 2. In the meantime we will select a sequence of input variables for iterative decomposition randomly, e.g. *d, c, b, a*. A single variable (highlighted within tables in Fig. 2) will be removed from the function in one decomposition step. Starting with variable d, we inspect the set of input cubes with value 0 or 1 in column d and look for all possible compatible pairs of input cubes $e = (e_1, e_2, e_3, 0)$ and $e' = (e'_1, e'_2, e'_3, 1)$ *hiding* their values 0 and 1. One cube (...,0) may be compatible with several cubes (...,1) and vice versa. These pairs will be referred to as binary pairs (b-pairs).

Next we will identify input cubes with value "-" in column d. From each such cube $u = (u_1, u_2, u_3, -)$ we can create a compatible pair $u = (u_1, u_2, u_3, 0)$ and $u' = (u_1, u_2, u_3, 1)$ by substitution 0 and 1 for "-". These pairs will be referred to as unary pairs (u-pairs) because of their origin from one cube. Remaining cubes of two types, $q = (q_1, q_2, q_3, 0)$ or $r = (r_1, r_2, r_3, 1)$, are not compatible between themselves and neither with any cube in binary pairs; we will call them orphaned input cubes. This is because the compatible cubes $q = (q_1, q_2, q_3, 1)$ or $r = (r_1, r_2, r_3, 0)$ map to the don´t care values and therefore *are not listed* in the cube table. We can thus append each orphaned cube with the identical invisible input cube with DC output value. We will call these pairs appended pairs (a-pairs).

In our example in Fig. 2 we will find

- only one b-pair, cubes 4&5
- two u-pairs, cubes 2&2 and 3&3
- two a-pairs, cubes 0&x, 1&x.

When we do decomposition of function F_4 by removal of variable d,

$$F_4 = H(G(a, b, c), d), \tag{3}$$

we have to intersect all b-, u-, and a-pairs of compatible input cubes $u = (u_1, u_2, u_3)$ and $v = (v_1, v_2, v_3)$ in order to obtain cubes of a residual function G and map them into pairs of output values :

	a	b	c	d	z	comp. cubes	
0	0	0	0	0	0		
1	0	0	1	0	2	4&5	0:= (4,0)
2	0	1	0	-	1	2&2	1:= (1,1)
3	1	0	-	-	3	3&3	2:= (3,3)
4	1	1	1	0	4	0&x	3:= (0,**0**)
5	1	1	1	1	0	1&x	4:= (2,**2**)

						comp. cubes	
0	1	1	1		0		
1	0	1	0		1	3&4	0:= (3,4)
2	1	0	-		2	2&2	1:= (2,2)
3	0	0	0		3	0&x	2:= (0,**0**)
4	0	0	1		4	x&1	3:= (**1**,1)

						comp. cubes	
0	0	0			0		
1	1	0			1	0&3	0:= (0,3)
2	1	1			2	1&2	1:= (1,2)
3	0	1			3		

						comp. cubes	
0	0				0		
1	1				1	0&1	0:= (0,1)

Figure 2: Iterative decomposition of an integer function F_4 of 4 binary variables (replacement of DC values in bold).

$$\begin{array}{lll} F_4: & u = (u_1, u_2, u_3) & F_4\,(u_1, u_2, u_3, 0) = P \\ F_4: & \underline{v = (v_1, v_2, v_3)} & \underline{F_4\,(v_1, v_2, v_3, 1) = Q} \\ G: & u^* v = (z_1, z_2, z_3) & Z := [P, Q] \end{array} \tag{4}$$

For example, pair of values (4, 0) is produced by cubes 4 and 5 in the first table in Fig.2; without values of d are these cubes compatible and can be replaced in the new table of a residual function $G(a, b, c)$ by a single input cube 111 – their intersection. The removed variable d is left empty in all cubes of the following tables. A pair of output values (4, 0) from intersection of cubes 4&5 is replaced by a new

integer id (0), as indicated in Fig. 2 by the assignment 0 := (4, 0).

Unary pairs of cubes 2&2 and 3&3 produce output pairs of the same values (1, 1) and (3, 3) redefined to new identities 1 and 2. Finally input cubes 0 and 1 are appended with the same invisible cubes to produce output pairs (0, DC) and (2, DC). Now the DC values must be defined so as not to increase the number of already existing unique pairs. If merging with one already found unique pair is not possible, like in our case, we will use pairs of the same values (0, 0) and (2, 2) and give them new identities 3 and 4. Sometimes it may be useful to replace all DC values by a special default value that will be interpreted as "no _output" or "error".

Pairs of different output values correspond to a true decision node, whereas pairs of the same output values produce degenerate or false decision nodes, because variable *d* in fact does not decide anything. Nodes in the MTBDD are labeled by the new identities of output pairs. There is one true node (0) and four false nodes (1, 2, 3 and 4 shown as black dots) in the lowest level of the MTBDD in Fig. 3. Dashed edges are taken for 0-value and solid edges for 1- or both values of decision variables.

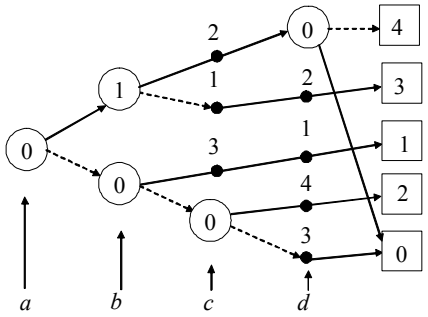


Figure 3: The MTBDD of function F_4 obtained by iterative decomposition.

By now, we have exhausted all possible pairs of compatible cubes of F_4 with $d = 1$ and $d = 0$ and have replaced them by new shorter cubes of the residual function *G*. The same procedure is repeated in the following decomposition steps until all variables have been removed. We move ahead in a backward direction, from the leaves of the MTBDD to its root, Fig. 3.

The remaining question not addressed as yet is, which variable should be used in any given step. We use a heuristic that strives to minimize the number of true nodes *t* in the current level of the MTBDD. In the case of a tie, a variable with a lower number of false nodes *f* is selected. In case of a tie again, a variable is chosen randomly.

The core of the above algorithm, the search for the best variable in step *i*, *i* in 1 to *n,* is given below (letters *S* stand for sets, *M* for tables):

// Determine the best variable v_k in step k //
M_{k-1}, a cube table of the $(k–1)^{th}$ residual function;
(M_0 is the cube table of the original function);
S_v, the set of input variables of the $(k–1)^{th}$ residual function;
v_k , the best variable in step k;

$v_{best} \leftarrow$ arbitrary variable from S_v,
$t_{best} \leftarrow$ size(M_{k-1}), $f_{best} \leftarrow 0$;
for all variables $v \in S_v$ do
  $M_p \leftarrow$ make b- and u-pairs(M_{k-1}, v);
  $S_p \leftarrow$ unique_output pairs(M_p);
  $S_m \leftarrow$ merge or add a-pairs(S_p);
  $t \leftarrow$ #true nodes(S_m);
  $f \leftarrow$#false nodes(S_m);
  if ($t < t_{best}$) or (($t == t_{best}$) and ($f < f_{best}$))
    then $v_{best} \leftarrow v$, $t_{best} \leftarrow t$, $f_{best} \leftarrow f$;
  endif
endfor
$v_k \leftarrow v_{best}$;

The whole algorithm for iterative decomposition has been implemented in the SW tool HIDET (Heuristic Iterative Decomposition Tool). It has been applied successfully to a class of arbiter and allocator circuits; parameters of some obtained MTBDDs are given in Section 4.

4 BRANCHING PROGRAMS WITH DISPATCH TABLES

Implementing multiple-output Boolean functions on a microprocessor can be done in several ways. Emulating PLC that evaluates one function after another in a sum-of-products form by redundant testing values of variables is slow and inefficient. A better way makes use of the whole processor word as 32 or 64 bits in parallel. The PLA matrix with *n* inputs and *m* outputs can be emulated in *n*+*m* steps. A product vector in the AND array is created by accumulating contributions from input variables: according to the value of an input variable, one of two masks is logically multiplied with the product vector created so far (and initially with all ones). Then *m* outputs are generated serially applying a single mask for each output to the product vector

and detecting presence of at least a single 1. However, if the number of cubes is larger than the word size, above steps must be repeated several times.

Finally, the method based on MTBDDs takes always n or a fraction of n steps. Provided that a (sub-)optimal MTBDD of a certain integer function is known, writing a branching program is a routine. We will illustrate it on the 4-input Round Robin Arbiter (RRA) with 4 input request lines r_0–r_3 and 4 grant outputs g_0–g_3. The n-bit priority register p_0–p_3 is maintained which points to the requester who is next. It contains a single 1 that rotates one position after a grant is issued. The MTBDD of this RRA obtained by HIDET tool is in Fig. 4. The speed of evaluation is given by the number of decision variables tested simultaneously.

The sample of a *symbolic* program with testing two binary inputs at a time is shown at Fig. 5. The best performance is obtained by hand coding the series of table lookups in assembly language and replacing switch statements by dispatch tables. The program uses 9 4-way and of 2 2-way dispatch tables. The size of dispatch tables varies depending on whether the input edge leads to a true decision node (L1-L5, L7-L10) or passes through one or more false nodes (L6, L11). In the assembly code, the base address of a dispatch table gets modified in two least significant bits by values of two variables under the test. Items in a dispatch table contain either the next base address or the terminal value. One bit is used to differentiate between these two formats. The total size of all dispatch tables is $9\times4+2\times2 = 40$ words and an arbitration decision is produced after four table lookups.

Had we used only single variable tests (a branching program with 2-way tables), we would need 17 dispatch tables of size 2, i.e. 34 words in total. However, the performance would be 2- times lower due to execution of a chain of 8 table lookups, one in each level of the MTBDD. Faster processing in three steps could test groups of 2, 3, 3 or 2, 2, 4 decision variables. The fastest execution would test 4 decision variables at a time and use 16-way branching. The features of various options are summarized in Table 1. The space × time product is a figure of merit of quality of the implementation. It gets the best (lowest) value for testing four variables at a time.

With the aid of HIDET tool, MTBDDs of many types of arbiters of different size have been obtained, among others priority encoders, RR, LGLP (Last Granted Lowest Priority) and LRS (Least Recently Serviced) arbiters. Cube tables were obtained automatically by means of small routines in C which enable scaling to the desired size. The results for RRAn arbiters with n inputs, m outputs and specified

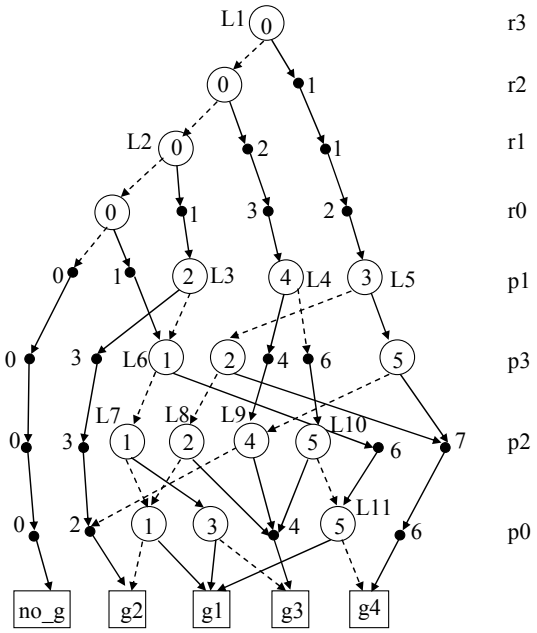


Figure 4: MTBDD of the 4-input RR arbiter

```
L1: input x ←r3r2 ;
    switch (x) {
      case 0:
        goto L2;
      case 1:
        goto L4;
      case 2:
        goto L5;
      case 3:
        goto L5;
    }
L2: input x ←r1r0;
    switch (x) {
      case 0:
        output no_g;
        goto End;
      case 1:
        goto L6;
      case 2:
        goto L3;
      case 3:
        goto L3;
    }
  …
```

```
L10: input x ←p2p0;
     switch (x) {
       case 0:
         output g4;
         goto End;
       case 1:
         output g1;
         goto End;
       case 2:
         output g3;
         goto End;
       case 3:
         output g3;
         goto End;
     }
L11: input x ←p0;
     switch (x) {
       case 0:
         output g4;
         goto End;
       case 1:
         output g1;
         goto End;
     {
   End:
```

Figure 5: A symbolic program for the 4-input RRA.

by #cubes are given in Tab. 2. The number of true nodes multiplied by 2 gives the lower bound on memory space (in words) for dispatch tables.

Table 1: Various RRA4 program options.

tested variables:	Σ dispatch table size	# table lookups	space x time
8 x 1	34	8	272
4 x 2	40	4	160
2, 3, 3	52	3	156
2, 2, 4	64	3	192
4, 4	72	2	144
8	256	1	256

Table 2: MTBDDs for Round Robin Arbiters.

	in	out	#cubes	# true nodes
	n	*m*		
RRA3	6	3	10	10
RRA4	8	4	17	17
RRA6	12	6	37	40
RRA8	16	8	65	75
RRA12	24	12	145	189

5 CONCLUSIONS

Programming a digital logic component of micro-processor-based control systems need not rely only on ladder diagrams anymore. Modern digital logic design offers multi-terminal BDDs that can specify groups of Boolean functions simultaneously, are non-redundant and allow direct conversion to branching programs with dispatch tables.

The advantages of the presented technique are twofold:

1. The transition from cube specification to the MTBDD and then to the assembly program is relatively easy and can be automated. The latter transition is of course depending on a target processor.
2. As soon as the MTBDD is known, the most suitable program implementation can be chosen trading-off performance for memory space (mainly to store dispatch tables).

The programming technique has been demonstrated on (but it is not limited to) the class of arbiter circuits. Currently it is applicable to integer functions of Boolean variables with don´t cares.

Future research will address multiple-output Boolean functions with compatible input cubes and incidentally with ternary output cubes $c \in \{0, -, 1\}^m$. This extension could provide appropriate design techniques for new classes of functions.

ACKNOWLEDGEMENTS

This research has been carried out under the financial support of the research grants “Natural Computing on Unconventional Platforms”, GP103/10/1517, “Safety and security of networked embedded system applications”, GA102/08/1429, "Mathematical and Engineering Approaches to Developing Reliable and Secure Concurrent and Distributed Computer Systems" GA 102/09/H042, all care of Grant Agency of Czech Republic, and by the BUT FIT grant FIT-10-S-1 and the research plan MSM0021630528.

REFERENCES

Bryant, R. E., 1991. On the complexity of VLSI implementations and graph representations of Boolean functions with applications to integer multiplication. In: *IEEE Transactions on Computers*, Vol. 40, pp. 205–213, 1991.

Brzozowski, J. A., Luba, T., 1997. Decomposition of Boolean Functions Specified by Cubes. *Research report CS-97-01*, University of Waterloo, Canada, p. 36.

Dvořák, V., 1997. Efficient Evaluation of Multiple-Output Boolean Functions in Embedded Software or Firmware, In: *Journal of Software*, Vol. 2, No. 5, 2007, pp. 52–63.

Gilvarry, I., 2009. IA-32 Features and Flexibility for Next-Generation Industrial Control. *Intel Technology, Journal*, Vol. 13, Issue 01, March 2009.

Matsuura, M., Sasao, T., 2007. BDD representation for incompletely specified multiple-output logic functions and its application to the design of LUT cascades, In: *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E90-A, No. 12, Dec. 2007, pp. 2770–2777.

Petruzella, F.D., 2004. *Programmable Logic Controllers,* McGraw Hill Science/Engineering/Math,

Sosic, R., Gu, J. and Johnson, R., 1996. The Unison algorithm: Fast evaluation of Boolean expressions. *ACM Transactions on Design Automation of Electronic Systems*, 1(4): pp. 456-477, Oct. 1996.

Yanushkevich, S. N., Miller, D. M., Shmerko, V.P., Stankovic, R. S., 2006. *Decision Diagram Techniques for Micro- and Nanoelectric Design Handbook*. CRC Press, Taylor & Francis Group, Boca Raton, FL.

MODELING SMART GRIDS AS COMPLEX SYSTEMS THROUGH THE IMPLEMENTATION OF INTELLIGENT HUBS

José González de Durana, Oscar Barambones
University College of Engineering, University of the Basque Country, Nieves Cano 12, 01006 Vitoria-Gasteiz, Spain
josemaria.gonzalezdedurana@ehu.es

Enrique Kremers, Pablo Viejo
European Institute for Energy Research, Karlsruhe Institute of Technology and EDF
Emmy Noether Str. 11, 76131 Karlsruhe, Germany
kremers@eifer.org

Keywords: Electrical grid, Hybrid renewable energy systems, Energy saving, Microgrid, Smart meter, Intelligent hub, Random graph, Complex system, Complex computer system, Scale free network, Agent based model.

Abstract: The electrical system is undergoing a profound change of state, which will lead to what is being called the smart grid. The necessity of a complex system approach to cope with ongoing changes is presented: combining a systemic approach based on complexity science with the classical views of electrical grids is important for an understanding the behavior of the future grid. Key issues like different layers and inter-layer devices, as well as subsystems are discussed and proposed as a base to create an agent-based system model to run simulations.

1 THE ELECTRICAL GRID AS A COMPLEX SYSTEM

The electrical grid as a whole can be considered as a complex system (more properly a Complex Computer System) whose aim is to assure a reliable power supply to all its consumers. Only regarding the grid from a multi-disciplinary point of view can help us understand the behavior of these systems. Despite conceptual advances in concrete fields like chaos theory or emergence in non-linear or self-organized systems, which were studied in the last decades, a unified theory of complexity does not yet exist.

Complex networks have been studied by several scientists. Erdös and Rényi (1959) suggested the modeling of networks as random graphs. In a random graph (Bollobás, 1998), the nodes are connected by a placing a random number of links among them. This leads to a Poisson distribution when considering the numbers of connections of the nodes, thus there are many nodes with a similar number of links.

Watts and Strogatz (1998) defined β as the probability of rewiring an edge of a ring graph and called these networks *small-world*. Analyzing networks with values $0 < \beta < 1$, they found that these systems can be highly clustered, with a relatively homogenous topology, and have small characteristic path lengths.

However, the study of networks in the real world has shown that there are many examples where this is not true but they exhibit a common property: the number of links k originating from a given node exhibits a power law distribution $P(k) \propto k^{-\gamma}$, i.e. few nodes having a large number of links. These networks are called scale-free and they are located in between the range of random and completely regular wired networks. Many systems in the real world such as neural networks, social networks and also the power grid, fulfill these properties.

Barabási and Albert (2002) mapped the topology of a portion of the World Wide Web and found that some nodes, which they called *hubs*, have many more connections than others and that the network as a whole exhibits a power-law distribution for the number of links connecting to a node. Using the Barabási-Albert network model, Chassin and Posse (2005) analyzed the topologies of the North American electric grid to estimate their reliability and calculated the exponent of scale-free power law as being $\lambda = 3.04$ for the U.S. eastern grid and $\lambda = 3.09$ for the western one.

Considering all of the advancements in complexity science, in this paper we will show how an electricity grid can be represented through a model as a complex system that can be used for simulations. First, the smart grid will be presented and some key issues dis-

cussed. Then, the approach for modeling the grid is explained and in the last section the simulation model is presented.

2 THE SMART GRID

The term *smart grid* as introduced by Amin and Wollenberg (2005), usually covers the entire spectrum of the electrical system, reaching from transportation over distribution up to the delivery. In common with earlier definitions, it contains two key elements: digital data processing and communication networks. Therefore, it can be said that what characterizes this *intelligent* grid is the existence of a flow of data and information, between the supplier company and the consumer, running in parallel with the energy flow (Singer, 2009).

With today's smart grid goals in mind, energy supply companies are in a transition process between our real electricity grid and the future smart grid, trying to improve the conventional network infrastructure, establishing the digital level (essence of the intelligent network) and also creating new business processes to carry out the capitalization and commercialization of the intelligent network.

The operation of the smart grid is far more complicated than the conventional power grid and in order to be operated, some special components like computers, sensors, remote controlled switching devices, as well as communication networks are necessary. For example, the current power grid is still not ready to admit microgrid connections. Connections made at present are experimental and almost always done manually, by taking care that a number of factors are fulfilled e.g. before realizing a connection.

Trying to model a microgrid, the network to which it will be connected should also be considered. Although a large amount of work in this area has been done, the main problem is that the current electricity grids are not yet adequately prepared for the transition to the smart grid. Therefore, in this article, the authors do not consider the inadequate existing grid, but instead focus on a hypothetical future network: the smart grid.

2.1 The two Layer Model

Concerning the upcoming challenges, especially facing the growing need of interaction of the different units of the smart grid, a two layer model is proposed (Kremers et al., 2010). These were identified as:

Physical Layer. The first layer is the physical structure of the electrical grid itself, including all the power transmission lines. It includes the power flows as well as all the electrical devices related to the correct operation of the grid.

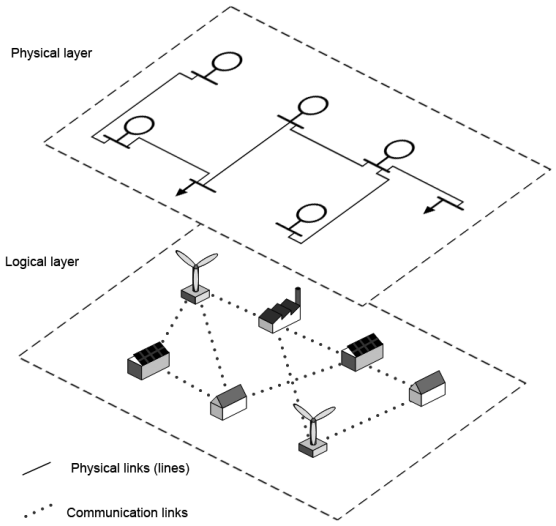


Figure 1: Different layers in an electrical microgrid.

Logical Layer. This second layer, which represents the main part of the upcoming generation of electrical grids, is not yet present, in contrast to the physical layer. This layer includes all the information exchange that has to be arranged to control distributed generation (DG), dispatchable loads and other *smart equipment* in future grids. It has to be underlined that the communication paths do not have to be the same as the links in the first layer, although they could be exploited for that aim. An example of this is Power Line Communication (PLC).

The current electricity grid could be seen as part of the first layer, whereas the second layer is still the focus of vast research and development. It represents all the information and communication technology linked in some way to the grid and its operation. It implements a system that allows real-time communication between the elements of the grid. In the E-Energie project (2010), an *Internet of Energy* is suggested as an analogy to computer networks. This medium could itself serve as a communication platform. More examples of the implementation of the logical layer are described in Kremers et al. (2010) and could be PLC, existing communication networks, wireless technologies, etc.

In the following sections, some key role playing concepts of the smart grid will be exemplified and discussed. First of all, smart metering as a technology under deployment is presented. Afterwards, the

concept of *intelligent hub* is introduced as a generic modeling approach for intelligent devices in the future grid. Finally, the microgrid as a sub-system of the smart grid is discussed.

2.2 Smart Metering

Traditionally, an energy meter measures only the consumption of the total energy during a specific time. This is used for billing the customer the total amount of energy he consumed. There is no way to obtain information on *when* the energy was consumed nor in what way. Smart meters are intended to provide more detailed information which will allow the supplier to adjust the pricing for consumption based on different parameters.

Electricity prices vary during a day or season, following the market offer-demand principle (especially with the introduction of renewable, non-dispatchable sources) or due to external factors such as temperature. Using a multiple tariff based system will allow for the reflection of these pricing changes to the final customer and thus entice him to make a more economical use of energy. These pricing signals shall help to reduce peak loads and sell more energy in off-peak periods, e.g. during the night.

The ESMA (European Smart Metering Alliance) defines a smart meter as an advanced meter with several functions, such as automatic data processing and transfer, automatic performing of measurements, which provides meaningful and up-to-date informations of consumption to the relevant actors and units of the energy system. Additionally, smart meters can provide support for measures to increase energy efficient consumption. Proof of the relevance of this device are the statements made by governments of different nations worldwide. For example, three of them have been chosen:

- Malta, where a pilot project is currently underway, with more than 5,000 smart counters being installed. The objective for 2012 is to have only smart meters in use.
- The United Kingdom, where in December 2009 the U.K. Department of Energy and Climate Change announced its intention to have smart meters in all homes by 2020.
- The U.S. where, according to Edison's Institute for Electric Efficiency, many of the country's largest electricity distribution companies have plans to install millions of meters in the coming years, with deployments to be complete between 2012 and 2015.

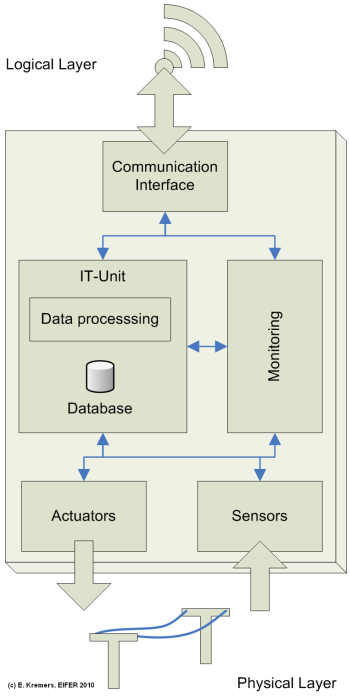


Figure 2: Architecture of an Intelligent Hub.

However, in the authors' opinion, energy savings will only be achieved when the meters are reinforced by new devices and directives such as information displays, time-varying pricing, energy audits and, in particular, some form of automatic load control. The fact that companies such as Intel, Cisco or Google are developing hardware and software for this growing market, seems to confirm that idea.

2.3 Intelligent Hubs as Interaction between the Layers

Having described the smart grid properties, the question of how the assumed measures can be implemented to make the grid *smarter* is apparent. The approach taken in this study is to model some specially designed generic units called Intelligent Hubs that:

- implement the communication functions of the logical layer,
- monitor the physical grid,
- perform data processing and evaluation,
- can take actions on the physical grid,
- can act as a local decision unit, and
- handle any interactions between the logical and physical layer.

The introduction of the intelligent hub arises from the idea that there are many different technologies and implementation possibilities for new infrastructure equipment, but no standard definition of these new intelligent units currently exists. For example, at household level a local load shedding module connected to a smart metering system could be implemented, or at a substation level new technologies that are be able to communicate with customers to send e.g. grid state signals, etc. are possible.

They all have in common that they share at least some of the characteristics of the intelligent hubs named above. This allows us to model a generic intelligent hub, which accomplishes with the specifications given and is able to simulate the behavior of these future equipment elements, even though a concrete implementation is not realized. It is important to underline that the intelligent hub is the link between the two layers, logical and physical, thus gathering information from both of them, being able to process it and actuating on the physical layer to perform changes. The acquisition of the data from the physical grid is performed by sensors, for example measuring units on the lines. The actuators are any kind of interaction with the power grid, such as demand control for example by direct (such as relays, operating on the line), or indirect means (dynamic demand reduction of the equipment).

The question of how far a smart meter can be seen as an intelligent hub or vice versa has to be analyzed further. There exist implementations of smart meters that seem to accomplish with some of the intelligent hub features (like load shedding functions), but in our opinion this already goes beyond the concept of metering. So, at least for modeling purposes, the smart meter will be seen as a part of the intelligent hub or an external unit linked to this, as the concept of the hub involves a much broader list of features, which can be summarized as the whole interaction between the two layers – even at different levels of the grid.

2.4 Microgrids as Smart Grid Subsystems

A microgrid is a set of small energy generators arranged in order to supply energy for a community of users in close proximity. It is a combination of generation sources, loads and energy storage, interfaced through fast-acting power electronics. Emerging from the general trend of the introduction of Renewable Energy Sources (RES), microgrids will mostly include this type of generation, so they form part of the Hybrid Renewable Energy Systems (HRES). Microgrids represent a form of decentralization of electrical networks. They comprise low- or medium-voltage distribution systems with distributed energy sources, storage devices and controllable loads.

During disturbances, the generation and corresponding loads can autonomously disconnect from the distribution system to isolate the load of the microgrid from the disturbance without damaging the integrity of the transmission grid. This mode is called *islanding* mode. From the point of view of the customer, it can be seen as a low voltage distribution service with additional features like an increase in local reliability, the improvement of voltage and power quality, the reduction of emissions, a decrease in the cost of energy supply, etc.

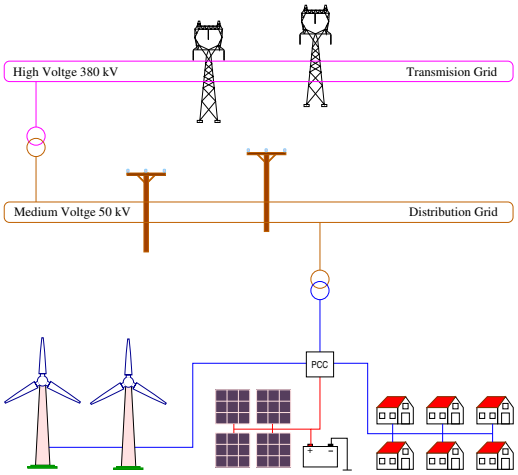


Figure 3: Integrated microgrid.

In Figure 3, a schematic drawing of an integrated microgrid can be seen, showing the Point of Common Coupling (PCC) and its electrical connections, but without representing the information channels. The authors have previously identified (Kremers et al., 2010) that electrical grids, as well as microgrids mostly satisfy the principal characteristics that distinguish them as true systems of systems, as defined by Maier (1998). The smart grid is constituted as a large complex system with operational and managerial independent elements (that are systems themselves), evolutionary development, emergent behavior and geographical distribution.

3 A COMBINED APPROACH FOR SMART GRID MODELING

For simulation purposes, one should not pay attention to accessory components but instead focus on the essential parts. It may occur that modeling some important elements is unnecessary for the operation, as

for example some electronic components that while being essential for the actual operation they are trivial or nonexistent for simulation.

So in this article, we focus on a very important element to consider both the actual operation of the microgrid and the simulation: the smart meter. Assuming that there is an intelligent hub at each network bus, a smart meter at each load bus and there exists communication among the nodes, the resulting complex computer system can be used as a basis for smart grid models.

Agent-based modeling tools are able to recreate complex system behavior such as those described here, unexpected emergent behavior in these systems, internal and external events, communications within the system, etc. In particular, local effects of the single units comprising the system can be modeled and their effects can be analyzed at the system level.

The combination of several approaches allows the creation of models that might abstract some details from the single unit models, but all in all create a much more realistic representation at the system level. The inclusion of some communication among the devices is fundamental here. This combined approach is the one followed in this work, as in the authors' opinion it is very advantageous when modeling future electrical systems.

Geographic network localization, distribution of processing and databases, interaction with humans, and unpredictability of system reactions to unexpected external events are also present in this kind of network.

4 THE SIMULATION MODEL

A combined approach model has been developed using Anylogic (XJTek, 2010), in which the grid nodes are represented by agents in the model, and each agent is provided of respective subsystem models (System Dynamics (SD), Discrete Events (DE), etc.). This kind of modeling has been chosen to allow for events such as the sudden elimination of one (or more) nodes, or connections between nodes, to be possible during the simulation, thus providing the possibility of a dynamic simulation of the electrical grid, including events such as failures, disasters and terrorist attacks.

The two-layer structure defined for the smart grid will be a key element due to its representation in the model. Apart from the agent-based approach, two other modeling paradigms are used: the SD paradigm for the physical layer and the DE paradigm for the logical layer.

The model is completely open, so it can be used to address a number of issues of design and computation that arise in such networks. Some of them coud be:

- Real grid, smart grid and microgrid simulation
- Grid and microgrid architecture design
- Centralized and decentralized control design
- Load connection and disconnection
- Microgrid connection and islanding modes
- Branch or node (e.g. substation) deleting
- Energy savings strategies

The approach presented here is intended to result in a suitable electrical microgrid model. It is a continuation of the authors' previous studies in this area (González de Durana et al., 2009; González de Durana and Barambones, 2009). In these works, the mesh method was used to obtain voltages at the mesh nodes and currents through the branches, assuming the voltages given by the generators are known. In a further study, however, a new *power flow method* has been created (Kremers et al., 2010). This new method was implemented using an combined approach in which agents represent the buses of the grid.

5 CONCLUSIONS AND OUTLOOK

A view of energy systems from the complex systems approach has been given, underlining the importance of viewing the energy system as such, especially for its future development. Modeling the energy system at system level is crucial to help us understand the interactions of the single units, and be able to observe system phenomena such as emergence and the behavior of the system as a whole. The requirement of the introduction of new perceptions of the energy system was shown with the proposition of the two layer model to represent the smart grid. Further, a device abstraction was done for modeling generic devices called intelligent hubs that are able to interact with this new environment.

An agent-based model for the simulation of microgrids is being implemented using AnyLogic. The model offers a clear two-layer structure, which allows for the representation of both physical and logical interactions between the elements. The logical layer offers a robust base to implement agent communication in real time. The physical layer provides the technical results of the power flow calculation integrated into the model. This allows real-time simulations of the

grid to be computed, which can provide valuable information prior to the implementation of the real grid. The first models of the intelligent hub have been developed and are currently being tested.

The model is mainly intended to design and test microgrids and can be used as a tool for the design, development and demonstration of control strategies, especially centralised supervisor control and decentralised load-dispatch control, the design and demonstration of microgrid operation strategies, the design and testing of microgrid communication buses and optimal microgrid design.

ACKNOWLEDGEMENTS

This work was made possible through cooperation between the University College of Engineering in Vitoria and EIFER in Karlsruhe. The authors are very grateful to the Basque Government for the support of this work through the project S-PE09UN12 and to the UPV/EHU for its support through the project GUI07/08.

REFERENCES

Amin, S. and Wollenberg, B. F. (2005). Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34–41.

Barabási, A. L. (2003). *Linked: How everything is connected to everything else and what it means*. Penguin Group New York.

Barabási, A. L. (2007). The architecture of complexity. *Control Systems Magazine, IEEE*, 27(4):33–42. 0272-1708.

Barabási, A. L. and Albert, R. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74.

Bollobás, B. (1998). *Modern graph theory*. Springer Verlag.

Chassin, D. P. and Posse, C. (2005). Evaluating north american electric grid reliability using the barabási-albert network model. *Physica A: Statistical Mechanics and its Applications*, 355(2-4):667–677.

Erdös, P. and Rényi, A. (1959). On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297.

European Commission. (2006). European smartgrids technology platform: Visions and strategy for europe's electricity networks of the future.

E-Energie Projekt. (2010) Bundesministerium fuer Wirtschaft und Technologie. http://www.e-energie.de

González de Durana, J., Barambones, O., Kremers, E., and Viejo, P. (2009). Complete agent based simulation of mini-grids. In *The Ninth IASTED European Conference on Power and Engineering Systems, EuroPES 2009*, volume 681, pages 046–188, Palma de Mallorca, Spain. Acta Press.

González de Durana, J. M. and Barambones, O. (2009). Object oriented simulation of hybrid renewable energy systems focused on supervisor control. In *IEEE Conference on Emerging Technologies and Factory Automation, 2009, ETFA 2009*, pages 1–8.

Hatziargyriou, N., Asano, H., Iravani, R., and Marnay, C. (2007). Microgrids - an overview of ongoing research, development, and demonstration projects. *IEEE power and energy magazine*, (july/august 2007):17.

Jiyuan, F. and Borlase, S. (2009). The evolution of distribution. *Power and Energy Magazine, IEEE*, 7(2):63–68. 1540-7977.

Karpov, Y. G., Ivanovski, R. I., Voropai, N. I., and Popov, D. B. (2005). Hierarchical modeling of electric power system expansion by anylogic simulation software. In *Power Tech, 2005 IEEE Russia*, pages 1–5.

Kremers, E., Lewald, N., Barambones, O., and González de Durana, J. (2009). An agent-based multi-scale wind generation model. In *The Ninth IASTED European Conference on Power and Engineering Systems, EuroPES 2009*, volume 681, pages 064–166, Palma de Mallorca, Spain. Acta Press.

Kremers, E., Viejo, P., González de Durana, J. M., and Barambones, O. (2010). A complex systems modelling approach for decentralized simulation of electrical microgrids. In *15th IEEE International Conference on Engineering of Complex Computer Systems*, page 8, Oxford.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284.

Singer, J. (2009). Enabling tomorrow's electricity system - report of the Ontario smart grid forum.

Valov, B. and Heier, S. (2006). Software for analysis of integration possibility of renewable energy units into electrical networks. In *The 5th International Conference Electric Power Quality and Supply Reliability*, Viimsi, Estonia. Proceedings Tallin University of Technology pp 173-177.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

XJ Technologies website. (2010). Anylogic. http://www.xjtek.com

POSTERS

MODELING RADIAL VELOCITY SIGNALS FOR EXOPLANET SEARCH APPLICATIONS

Prabhu Babu*, Petre Stoica
Division of Systems and Control, Department of Information Technology
Uppsala University, P.O. Box 337, SE-75 105, Uppsala, Sweden
{*prabhu.babu, ps*}*@it.uu.se*

Jian Li
Department of Electrical and Computer Engineering, University of Florida, Gainesville, 32611, FL, U.S.A.
li@dsp.ufl.edu

Keywords: Radial velocity method, Exoplanet search, Kepler model, IAA, Periodogram, RELAX, GLRT.

Abstract: In this paper, we introduce an estimation technique for analyzing radial velocity data commonly encountered in extrasolar planet detection. We discuss the Keplerian model for radial velocity data measurements and estimate the 3D spectrum (power vs. eccentricity, orbital period and periastron passage time) of the radial velocity data by using a relaxation maximum likelihood algorithm (RELAX). We then establish the significance of the spectral peaks by using a generalized likelihood ratio test (GLRT). Numerical experiments are carried out on a real life data set to evaluate the performance of our method.

1 INTRODUCTION

Extrasolar planet (or shortly exoplanet) detection is a fascinating and challenging area of research in the field of astrophysics. Till mid 2009, 353 exoplanets have been discovered. Some of the techniques available in the astrophysics literature to detect exoplanets are astrometry, the radial velocity method, pulsar timing, the transit method and gravitational microlensing. Among these methods, the radial velocity analysis is the most commonly used technique, in which the Doppler shift in the spectral lines and hence the radial velocity of the parent star is measured. The spectrum of the measured Doppler shifts is then analyzed to detect the exoplanet(s) revolving around the star.

Most often the radial velocity measurements are obtained at nonuniformly spaced time intervals due to hardware and practical constraints, which limits the application of commonly used spectral analysis methods. The most straightforward way to deal with this problem is to use the standard periodogram by ignoring the nonuniformity of data samples, which results in an inaccurate spectrum. In (Roberts et al., 1987), a method named CLEAN was proposed, which is based on iterative deconvolution in the frequency domain to obtain a clean spectrum from an initial dirty one. A periodogram related method is the least squares periodogram (also called the Lomb-Scargle periodogram) (Lomb, 1976; Scargle, 1982) which estimates the sinusoidal components by fitting them to the observed data. Most recently, (Yardibi et al., 2010; Stoica et al., 2009) introduced a new method called the Iterative Adaptive Approach (IAA), which relies on solving an iterative weighted least squares problem.

In this paper, we analyze the radial velocity data by using a relaxation maximum likelihood algorithm (RELAX) initialized with IAA estimates. The significance of the spectral peaks is then established via a generalized likelihood ratio test (GLRT). Numerical experiments are carried out on a real life radial velocity data set.

In Section 2, we describe the model used in this paper for the radial velocity data. Section 3 presents the RELAX and GLRT methods, and Section 4 contains the results for a real life example. Finally, the paper is concluded in Section 5.

*Corresponding author. This work was supported in part by the Swedish Research Council (VR).

2 DATA MODEL

Let $\{y(t_n)\}_{n=1}^N$ denote the radial velocity of a star measured at a set of possibly nonuniform time instants $\{t_n\}_{n=1}^N$. Based on the Keplerian model of planetary motion (Zechmeister and Kurster, 2009) (Cumming, 2004), the radial velocity data are modeled as follows:

$$y(t_n) = C_0 + \sum_{m=1}^{M} \beta_m[\cos(\omega_m + \nu_m(t_n)) + e_m\cos(\omega_m)], \quad n = 1, \cdots, N \tag{1}$$

where C_0 is the constant radial velocity, and

$$\begin{aligned} \tan(\nu_m(t_n)) &= \sqrt{\tfrac{1+e_m}{1-e_m}}\tan(E_m(t_n)) \\ E_m(t_n) - e_m\sin(E_m(t_n)) &= \tfrac{2\pi(t_n - T_m)}{P_m}, \end{aligned} \tag{2}$$

M: Number of exoplanets revolving around the star. The number of planets M is usually unknown.

e_m: Eccentricity of the orbit of the m^{th} planet.

ω_m: Longitude of the periastron for the m^{th} planet.

P_m: Orbital period of the m^{th} planet; P_m is related to orbital frequency f_m by $f_m = \frac{1}{P_m}$.

T_m: Periastron passage time of the m^{th} planet.

β_m: Radial velocity amplitude of the m^{th} planet.

$\nu_m(t_n)$, $E_m(t_n)$: True and eccentric anomaly of the m^{th} planet, with t_n denoting their time dependence.

We divide the entire 2D space $\mathcal{G}$, defined as $\mathcal{G} = \{(e,f),\, 0 \le e < e_{\max},\, -\frac{f_{\max}}{2} < f < \frac{f_{\max}}{2}\}$, into a grid of prespecified size K. We point out here that the grid $\mathcal{G}$ does not include the parameter T, and that T is taken to be zero for the time being but will be estimated as described later on. The choices of $e_{\max}$ and $f_{\max}$ in $\mathcal{G}$ depend on the sampling pattern: to determine them, we calculate the spectral window defined as:

$$W(e,f) = \left|\frac{1}{N}\sum_{n=1}^{N}\exp(j\nu(t_n))\right|^2, \quad 0 \le e < 1, -\infty \le f \le \infty. \tag{3}$$

For any choice of (e,f) and t_n, there exists a $\nu(t_n)$ obtained via (2). Following (Eyer and Bartholdi, 1999), the parameters $e_{\max}$ and $f_{\max}$ are chosen such that the region $\mathcal{G}$ leads to an unambiguous $W(e,f)$ (see (Babu et al., 2010) for more details).

3 PARAMETER ESTIMATION AND STATISTICAL SIGNIFICANCE TESTING: RELAX AND GLRT

The estimates obtained from IAA are usually fairly accurate. However, if the grid ($\mathcal{G}$) is not chosen fine enough (to reduce the computation time), then IAA might miss some true peaks (see (Babu et al., 2010) for an elaborate discussion on IAA for radial velocity data). In that case, applying RELAX (Li and Stoica, 1996), a parametric iterative estimation algorithm, can refine the IAA estimates. Algorithm 1 briefly describes the steps involved in RELAX. The P largest peaks picked from IAA are used as initial estimates for RELAX, which has a beneficial effect on the convergence of RELAX compared with using other more arbitrary initial estimates. In the case of radial velocity data, the choice of $P = 5$ peaks appears to be reasonable for most applications. RELAX generally converges within a few iterations (typically in less than 10 iterations).

Next we note that, under the assumption the noise in the data is Gaussian distributed, the RELAX estimates are optimal in the maximum likelihood sense (Li and Stoica, 1996). We can then use the generalized likelihood ratio test (GLRT) to establish the statistical significance of the estimated planet parameters. We first apply RELAX to the largest IAA peak and use GLRT to test the null hypothesis that there are no planets (or, in other words, that the data is made only of white noise) against the hypothesis that there is at least one exoplanet. If the test rejects the null hypothesis then we will proceed and apply RELAX to the two largest peaks and subsequently test the hypothesis that there is one exoplanet in the data against the hypothesis that there are at least two exoplanets; and so on. As an example, for the following hypotheses

H_0: There are no planets.

H_1: There is at least one exoplanet with eccentricity $\hat{e}_1$, orbital frequency $\hat{f}_1$ and periastron passage time $\hat{T}_1$.

the log-likelihood (LL) functions are given by:

$$\begin{aligned} &\mathrm{LL}(H_0) = \\ &\quad -\tfrac{N}{2}\ln\left(\sum_{n=1}^{N}|y(t_n)|^2\right) + C, \\ &\mathrm{LL}(H_1) = \\ &-\tfrac{N}{2}\ln\left(\sum_{n=1}^{N}|y(t_n) - \hat{r}_1\cos(\nu(t_n)) - \hat{q}_1\sin(\nu(t_n))|^2\right) + C \end{aligned} \tag{4}$$

where C is an additive constant, $\nu(t_n)$ is calculated from the RELAX estimates ($\hat{e}_1$, $\hat{f}_1$, $\hat{T}_1$), and $\hat{r}_1, \hat{q}_1$ are the least square estimates of r, q corresponding to ($\hat{e}_1$, $\hat{f}_1$, $\hat{T}_1$), see Algorithm 1. Under the assumption that hypothesis H_0 is true, the log-likelihood-ratio, defined as $2\,(\mathrm{LL}(H_1) - \mathrm{LL}(H_0))$, is asymptotically a random variable with a chi-square distribution. Then the GLRT is given by

$$2(\mathrm{LL}(H_1) - \mathrm{LL}(H_0)) \underset{H_0}{\overset{H_1}{\gtrless}} \Lambda \tag{5}$$

where Λ denotes a fixed threshold. The threshold is usually chosen such that $\text{prob}(X \leq \Lambda) = \xi$, where $X \sim \chi^2_5$ denotes a chi-square distributed random variable with 5 degrees of freedom (because of the 5 unknowns per planet in the data model, namely e, f, T, r and q), and ξ determines the significance level of the test. Choosing $\xi = 0.99$ gives a false alarm probability of 0.01 and the corresponding threshold is $\Lambda = 15$.

4 A REAL LIFE EXAMPLE: HD 208487

In this section, we consider the application of the algorithm introduced in the previous section to a real life radial velocity data set. Our goal is to detect the exoplanets present in a star system and estimate their eccentricities, frequencies and periastron passage times. We will show the following plots:

- Amplitude vs. orbital frequency for IAA and RELAX (eccentricity and periastron passage time values for the peaks in the amplitude spectrum are indicated in the plots).
- Likelihood ratio vs. the planet number.
- Observed and fitted data sequences.

The data set used here consists of 31 samples of radial velocity measurements of the star HD 208487. The parameters $e_{\max}$ and $f_{\max}$ are determined from the spectral window to be 0.5 and 1 cycles/day. The spectrum obtained using IAA is shown in Fig.1(a), which indicates the presence of more than one planet. The 5 largest peaks in the IAA spectrum are picked up and are used to initialize RELAX. The GLRT plot shown in Fig.1(c) suggests the existence of three planets in the HD 208487 star system with the following parameters (see Fig.1(d) and also Table 1): ($e_1 = 0.326$, $f_1 = 0.0078$ cycles/day, $T_1 = 130.9$ days), ($e_2 = 0.315$, $f_2 = 0.069$ cycles/day, $T_2 = 14.2$ days) and ($e_3 = 0$, $f_3 = 0.0408$ cycles/day, $T_3 = 2.9$ days). However (Tinney et al., 2005) reported that the star has only one planet with an orbital frequency of 0.0077 cycles/day. Fig. 1(e) and 1(f) show the plots of measured data and the fitted data obtained assuming the existence of one and, respectively, three planets. It is seen clearly from these figures that the three planet model fits the measured data much better than a single planet model.

5 CONCLUSIONS

The real life example discussed in the paper suggests that our algorithm successfully detects the presence of spectral peaks (planets) in radial velocity data and accurately identifies both their frequencies and eccentricities as well as their periastron passage times. The example used here is typical of cases usually encountered in exoplanet search and hence the proposed algorithm is believed to be an effective and useful tool.

REFERENCES

Babu, P., Stoica, P., Li, J., Chen, Z., and Ge, J. (2010). Analysis of radial velocity data by a novel adaptive approach. *The Astronomical Journal*, 139:783–793.

Cumming, A. (2004). Detectability of extrasolar planets in radial velocity surveys. *Monthly Notices of the Royal Astronomical Society*, 354(4):1165–1176.

Eyer, L. and Bartholdi, P. (1999). Variable stars: Which Nyquist frequency? *Astronomy and Astrophysics, Supplement Series*, 135:1–3.

Li, J. and Stoica, P. (1996). Efficient mixed-spectrum estimation with applications to target feature extraction. *IEEE Transactions on Signal Processing*, 44(2):281–295.

Lomb, N. R. (1976). Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(1):10–33.

Roberts, D. H., Lehar, J., and Dreher, J. W. (1987). Time series analysis with CLEAN. I. Derivation of a spectrum. *The Astronomical Journal*, 93(4):968–989.

Scargle, J. D. (1982). Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *Astrophysical Journal*, 263:835–853.

Stoica, P., Li, J., and He, H. (2009). Spectral analysis of nonuniformly sampled data: A new approach versus the periodogram. *IEEE Transactions on Signal Processing*, 57(3):843–858.

Stoica, P. and Moses, R. (2005). *Spectral Analysis of Signals*. Prentice Hall, Upper Saddle River, N.J.

Tinney, C., Butler, R., Marcy, G., Jones, H., Penny, A., McCarthy, C., Carter, B., and Fischer, D. (2005). Three Low-Mass Planets from the Anglo-Australian Planet Search 1. *The Astrophysical Journal*, 623(2):1171–1179.

Yardibi, T., Li, J., Stoica, P., Xue, M., and Baggeroer, A. B. (2010). Iterative adaptive approach for sparse signal representation with sensing applications. *IEEE Transactions on Aerospace and Electronic Systems*, 46:425–443.

Zechmeister, M. and Kurster, M. (2009). The generalised Lomb-Scargle periodogram. *Astronomy and Astrophysics*, 496(2):577–584.

APENDIX

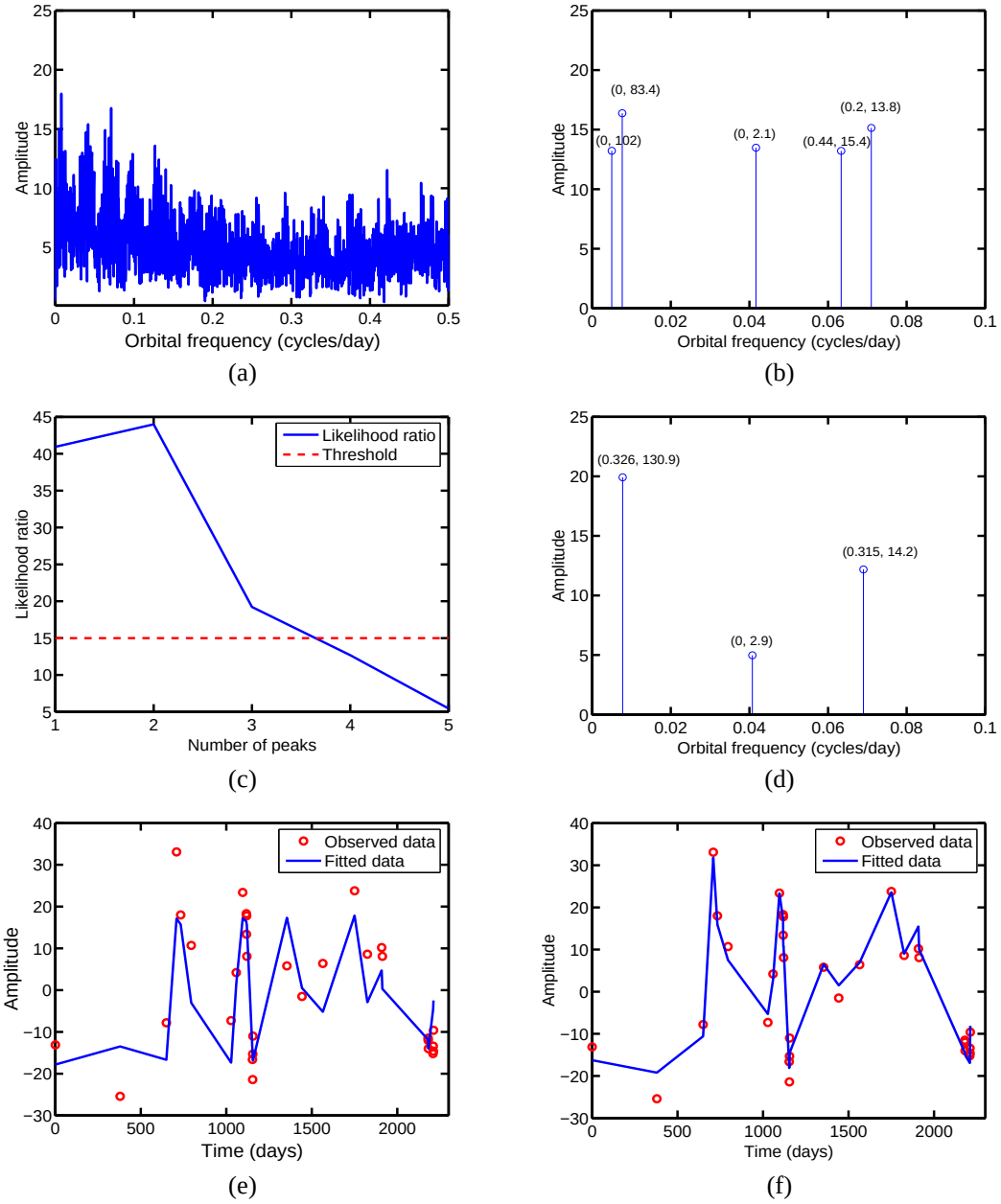


Figure 1: HD 208487: (a) the IAA spectrum, (b) the 5 largest peaks in the IAA spectrum, (c) the likelihood ratio, (d) the RELAX spectrum, (e) comparison of the observed data and fitted data obtained from the parameters of the planet reported in (Tinney et al., 2005) and (f) comparison of the observed data and fitted data obtained from the three detected planets.

Table 1: Parameters of the planets of HD 208487 star system. *) The third planet becomes statistically insignificant if the false alarm probability is decreased from 10^{-2} to 10^{-4}, in which case the threshold becomes $\Lambda = 25$.

Planet No.	**Previous work**				**This work**			
	e	f(cycles/day)	T(days)	β	e	f(cycles/day)	T(days)	β
1	0.24	0.0077	92	20	0.3260	0.0078	130.9	19.9
2	-	-	-	-	0.3150	0.0690	14.2	12.18
3*)	-	-	-	-	0	0.0408	2.9	4.96

Algorithm 1: RELAX.

Let $\{(e_p^0, f_p^0, T_p^0)\}_{p=1}^P$ denote the parameters of the P most dominant planets in the IAA spectrum (e.g. $P = 5$). The initial values of their radial velocity amplitudes $\{(r_p^0, q_p^0)\}_{p=1}^P$ are taken to be zero.

for $p = 1$ to P **do**
  for $i = 1$ to I (e.g. $I = 10$) **do**
    for $u = 1$ to p **do**
      $y_u^i(t_n) = y(t_n) -$
      $\sum_{k=1,k\neq u}^{p} \left(r_k^{i-1} \cos(\nu_k^{i-1}(t_n)) + q_k^{i-1} \sin(\nu_k^{i-1}(t_n)) \right)$
      where ν_k^{i-1} is obtained from
      $(e_k^{i-1}, f_k^{i-1}, T_k^{i-1})$. Then
      (e_u^i, f_u^i, T_u^i)
      $= \arg\min_{\{e,f,T\}} \min_{\{r,q\}} \sum_{n=1}^{N} \left| y_u^i(t_n) - r\cos(\nu(t_n)) - q\sin(\nu(t_n)) \right|^2$
      The inner minimization with respect to $\{r, q\}$ in the above equation is a least squares problem and the estimates $\{r_u^i, q_u^i\}$ can be obtained analytically (see (Stoica and Moses, 2005)). The minimization with respect to $\{e, f, T\}$ is carried out via a 3D grid search performed around $(e_u^{i-1}, f_u^{i-1}, T_u^{i-1})$.
    end for
  end for
  for $u = 1$ to p **do**
    $(e_u^0, f_u^0, T_u^0) \leftarrow (e_u^I, f_u^I, T_u^I)$ and $(r_u^0, q_u^0) \leftarrow (r_u^I, q_u^I)$.
  end for
end for
$\{(\hat{e}_p, \hat{f}_p, \hat{T}_p) \leftarrow (e_p^I, f_p^I, T_p^I)\}_{p=1}^P$ and $\{(\hat{r}_p, \hat{q}_p) \leftarrow (r_p^I, q_p^I)\}_{p=1}^P$

SHARED MEMORY IN RTAI SIMULINK FOR KERNEL AND USER-SPACE COMMUNICATION AT THE EXAMPLE OF THE SDH-2

QRtaiLab For SDH-2 Matrix Visualization

Thomas Haase, Heinz Wörn
Institute for Process Control and Robotics (KIT), Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany
haase@kit.edu

Holger Nahrstaedt
Control Systems Group, TU Berlin, D-10587, Berlin, Germany
nahrstaedt@control.tu-berlin.de

Keywords: QRtaiLab, RTAI linux, Matlab simulink Real-time workshop, SDH2, Shared memory.

Abstract: At the Institute for Process Control and Robotics reactive grasping skills are developed to enhance the Multi-fingered SCHUNK Dextrous Hand 2 (SDH2) in order to fulfill industrial needs. Therefore, RTAI Linux and Matlab - Simulink RTW are used as application development system (RTAI, 2010),(Mathworks, 2010). The exchange of data between the Multi-fingered hand and the computer system is possible by means of a C++ library. By reason that this SDH2 C++ library could not be used in Real-Time kernel programs this paper presents an approach of how to combine Real-Time Simulink models (RTAI) with user-space tasks. Therefore a shared memory based interface within Simulink S-Functions is established. The RTAI Target Language Compiler remains unaffected. The designed interface is described in detail. It represents a contribution to the further development of RTAI. In addition a possibility of how to debug and visualize tactile sensor matrices with QRtaiLab is presented.

1 INTRODUCTION

The Real-Time Application Interface for Linux (RTAI) combined with the Matlab/Simulink Real-Time Workshop offers the possibility to generate C code from Simulink models. Therefore, RTAI uses the configurable code generator called Target Language Compiler (Quaranta and Mantegazza, 2001). The resulting C code can be used to run Real-Time applications within an RTAI patched Linux kernel. It is not possible to start and to debug the executables with Simulink. The required interaction with these Real-Time modules is realized with the help of user interfaces like QRtaiLab and Xrtailab. At the Institute for Process Control and Robotics such an RTAI Real-Time Simulink system was chosen to develop reactive grasping skills for the SDH2. RTAI was chosen because the community project RTAI (RTAI, 2010) is up-to-date and offers a Simulink Target Language Compiler (TLC). In (Quaranta and Mantegazza, 2001) a different solution is proposed that uses the Tornado/VxWorks Target Language Compiler distributed together with MATLAB for creating the Real-Time code for Linux. Therefore some compatibility wrappers to the Simulink VxWorks TLC interface have to be applied. Unfortunately no user space communication is mentioned and a new adapted wrapper has to be built to communicate with the

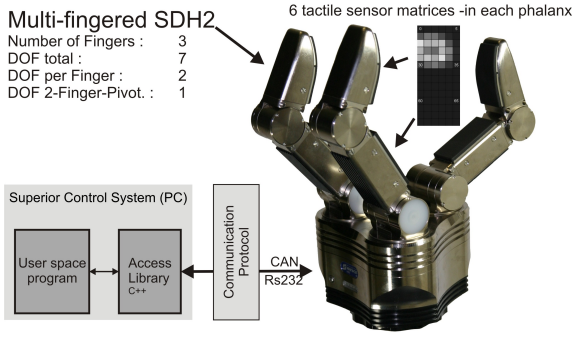


Figure 1: Multi-fingered SDH2.

SDH2. (W.E. Dixon, 2001) and (Ramadurai, 2001) introduce a Real Time Linux Target (RTLT) as reflection to the Real Time Windows Target (RTWT). RTLT seems to be a software application that gives Simulink the ability to run on a standard PC with hard Real-Time constraints. Unfortunately it seems that further development of RTLT has been stopped several years ago. The described computer aided control system design (CACSD) is only available for RTWT (D. M. Dawson, 2002).

In 2008 the final version of the Multi-fingered Dextrous Hand SDH-2 (SCHUNK GmbH & Co. KG, 2010) was introduced, see Figure 1. Each finger possesses two independent degrees of freedom. Combined with an extra pivoting joint the SDH2 provides seven degrees of freedom. Six tactile sensor arrays are included, one in each phalanx. Among others, they offer the feasibility to detect object contact or surface characteristics. The data exchange with the SDH2 is implemented by means of a C++ library. This library is provided by the manufacturer and is prepared for working in user space. Unfortunately, the RTAI Simulink toolbox supports neither a user-space communication nor a possibility to integrate the user-space SDH2 C++ library. Therewith, a direct data exchange among the SDH2 and the Real-Time executables in kernel space is not possible. In order to retain the possibility to use RTAI Simulink a way for exchanging the desired control and sensor data with these Real-Time programs has to be found. A universally valid solution is shown in Figure 2. The data exchange with the SDH2 is not changeable. A user-space transceiver exchanges all required information with the robot hand. A special interface offers these data to the Real-Time executables in kernel space. The challenge consists of designing an interface that allows the RTAI Target Language Compiler to remain unaffected. If this will be feasible a contribution to the further development of RTAI can be achieved. In the following sections an approach of how to combine Real-Time Simulink models (RTAI) with user space tasks is presented. Therefore a shared memory based interface within Simulink S-Functions is established. Generating named memory in kernel space is the determining advantage of this inter-task communication mechanism. Another advantage is, that there are no data queues and therewith only one actual dataset is given. This is crucial for transferring a large amount of tactile sensor data. Both, user-space program and kernel module, should be able to address this allocated memory. The allocation of shared memory in user-space programs is a well-known operation. In the next sections it is shown how to design a Simulink S-Function that realizes the access

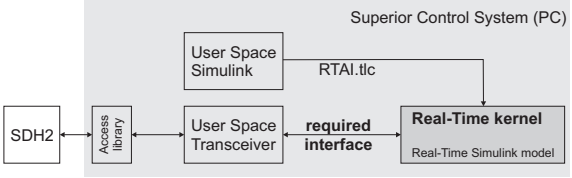


Figure 2: Global Concept of the Required Interface.

and therewith the communication for the Real-Time executable. This interface enhances the amount of possibilities in developing (semi-) Real-Time control systems with Matlab / Simulink using RTAI Linux. In addition, all RTAI files stay untouched and the RTAI Target Language Compiler (RTAI.tlc) is used anymore.

2 THE SHARED MEMORY INTERFACE

2.1 Basic Concept

Figure 3 shows the basic concept of the RTAI SDH2 simulation environment. Simulink is used to design software algorithms. The RTAI Target Language Compiler generates the Real-Time code that could be controlled and debugged with the help of QRtaiLab / Xrtailab. The Real-Time Simulink code

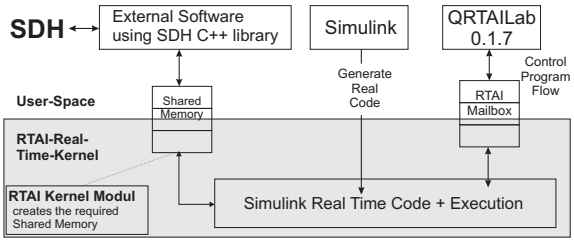


Figure 3: Concept of SDH-2 RTAI application development system.

is able to communicate with a shared memory module to exchange information and sensor data with the user-space transceiver module. The development system from Figure 3 requires a user-space program that communicates with the SDH2, an additional kernel tool for setting up the shared memory and some RTAI Simulink tools. Furthermore, for constructing and proceeding Simulink based Real-Time applications, a patched RTAI Linux kernel and the RTAI toolbox for Simulink are required. The Target Language Compiler (RTAI.tlc) that generates the Real-Time (RTAI) code from the Simulink models is part of the RTAI Simulink toolbox(Quaranta and Mantegazza, 2002).

As mentioned in section 1, it is not possible to execute and to debug the created Real-Time code with Simulink itself. All RTAI Simulink modules generate mailboxes for inter-task communication of all desired measurable signals. The scopes and displays in Simulink are not able to import the mailbox data. Therefore, additional software tools like Xrtailab or QRtaiLab are required to receive and to visualize measured values. Based on the demand for visualization of tactile sensor matrices, QRtaiLab 0.1.7 or newer is used instead of Xrtailab and is presented in section 4.
All required software from Figure 3 is described in detail within the following sections. It is shown how to create the SDH2 Simulink S-Function on the basis of the RTAI Simulink library.

2.2 The RTAI Kernel Module

For being able to access named shared memory in user-space programs, it has to be created by a Real-Time task first. This kernel module presented here creates the shared memory required for the Real-Time simulations. The common rtai-shm kernel module has to be loaded into the kernel before this tool is able to create the memory. As an example, the following enumeration shows all the different shared structures that are desired for working with the SDH2:

1. tactile sensors: to import sensor data from user-space into kernel-space
2. control: exchange the SDH2 control protocol
3. exporting joint angles into user-space (i.e. simulation)

The generation of the tactile sensor data array is shown more precisely in Listing 1. Simulink S-Function, RTAI kernel module and user-space communication software are all using the same structure 'TAK'. The tactile shared memory is generated within the RTAI kernel module, Figure 3:

```
#define SHM_Name "name/ID"                          1
static RT_TASK  t1;                                 2
typedef struct TAK{                                 3
   int Matrix1[84];                                 4
   int Matrix2[78];                                 5
   ...                                              6
} MSG_TAK;                                          7
...                                                 8
rt_set_periodic_mode();                             9
taktil =                                            10
     rtai_kmalloc(nam2num(SHM_Name),
     sizeof(struct TAK));
period = start_rt_timer();                          11
```

Listing 1: Kernel Module Listing.

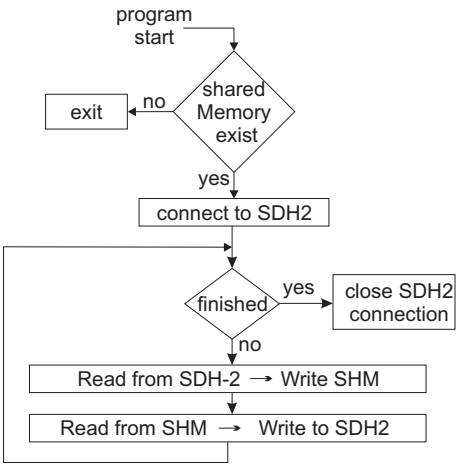


Figure 4: Program flow chart of user space software.

The variable "name/ID" is the unique identifier which allows to access the SHM. The defined struct (Line 3 to 7) contains all required arrays. This RTAI kernel module creates that named shared memory "name/ID" in Line 11. Now, all software modules in kernel- and in user-space are able to access the SHM. It is recommended that all structures and defines are swapped out into a shared header file.

2.3 User Space Control Transceiver

The user-space transceiver software from Figure 3 exchanges the predefined control and sensor data with the SDH2. It includes all data to and reads from the shared memory. Figure 4 illustrates the flow chart. It is recommended, that the kernel module from section 2.2 has already created the SHM before this transceiver is started. The program uses the C++ SDH2 library in user-space. Unfortunately this code is not running under Real-Time constraints. Due to the shared memory the operating speed of the transceiver is independent from any time scale of all Simulink Real-Time executables in kernel space. The program is kept as small as and therewith as fast as possible to ensure minimal reaction times. Since the SDH2 uses a CAN or an RS232 interface, the transceiver reaches the currently maximum possible operating frequency f_o with up to $f_o = 60Hz$.

3 THE SHARED MEMORY S-FUNCTION

The goal of this paper is the description of how to set up RTAI Simulink S-Functions using special shared memory arrays. The usage of the common

RTAI Target Language Compiler has to be ensured without limitations. The basic requirement for each S-Function is the availability of all shared memory structures used in kernel space. Required header files should be integrated into the S-Functions as shown in listing 2

```
#ifndef MATLAB_MEX_FILE
   ...
   #include <rtai_shm.h>
   ...
#endif
```

Listing 2: Integration of header files to allocate shared memory.

To make sure that the included RTAI modules are only addressed if working in Real-Time, Simulink defines MATLAB_MEX_FILE if working in normal Non-Real-Time mode.

3.1 Shared Memory Input

The following listing presents an essential abstract of how to input data from named shared memory into an RTAI Real-Time executable.

```
static void
     mdlInitializeSizes(SimStruct *S){
 if(!ssSetOutputPortDimensionInfo
      (S,0,&d))return;
 if(!ssSetOutputPortDimensionInfo
      (S,0,1 ))return;
}
static void mdlStart(SimStruct *S){
#ifndef MATLAB_MEX_FILE
 static struct MSG_TAK *msg;
 if (!(msg=rtai_malloc(nam2num(
      "name/ID"),sizeof(MSG_TAK)))){
    printf("no shared memory");
    exit(1);}
 ssGetPWork(S)[0]= (void *)msg;
#endif
}
static void mdlOutputs(SimStruct *S,
     int_T tid){
 double *M1   =
      ssGetOutputPortRealSignal(S,0);
 double *i    =
      ssGetOutputPortRealSignal(S,1);
#ifndef MATLAB_MEX_FILE
 MSG_ID *msg = (MSG_ID
      *)ssGetPWorkValue(S,0);
 M1[i] = msg->Matrix1[i]; //array
 *i = msg->variable; // scalar
#endif
}
```

Listing 3: RTAI Simulink Input Listing.

The given code snippet in Listing 3 demonstrates an S-Function with two output ports (lines 2 and 3). The first output contains a tactile matrix; variable d specifies the information about the dimensionality of the output port. The 'mdlStart' function initializes the named shared memory "name/ID". If no matching shared memory is found the execution of the Real-Time model is aborted. Otherwise a pointer to the initialized data structure is stored in the S-Function PWork vector for being addressable within further functions (line 11). Function 'mdloutputs' accesses this PWork vector and assigns the data to the output ports. The lines 19 and 20 demonstrate the difference in assigning arrays and scalars. To make sure that the shared memory is only assigned within the Real-Time kernel and not within the Simulink software, the code fragment in lines 6 and 17 are necessary. If the S-Function is built as MEX-file with the mex command, MATLAB_MEX_FILE is automatically defined. The RTAI Target Language Compiler is able to handle this code. It is important, that the SHM Input S-Function accesses to the SHM struct defined in listing 1.

3.2 RTAI Simulink Output

Writing data into the assigned shared memory is nearly equal as shown in section 3.1 and is given in Listing 4.

```
static void
     mdlInitializeSizes(SimStruct *S)
{  if (!ssSetNumInputPorts(S, 2))
     return;
    ssSetInputPortWidth(
         S, 0, d );
    ssSetInputPortWidth(
         S, 1, 1 );
}
static void mdlOutputs(SimStruct *S,
     int_T tid)
{
InputRealPtrsType uPtrs1 =
     ssGetInputPortRealSignalPtrs(S,0);
InputRealPtrsType u1 =
     ssGetInputPortRealSignalPtrs(S,1);
#ifndef MATLAB_MEX_FILE
        MSG_TAK *msg   = (MSG_TAK
             *)ssGetPWorkValue(S,0);
      msg->P1 = (double)*uPtrs1[0];
      msg->P7 = (double)*uPtrs1[6];
      msg->Move  = (double)*u1[0];
      msg->Variable  = 4;
#endif
```

Listing 4: RTAI Simulink Output Listing.

MdlStart allocates the named shared memory and stores a pointer to it within the PWork vector. If running in Real-Time kernel, 'Matlab_Mex_File' is not defined and lines 11 to 15 are executed. Lines 11 to

14 demonstrate how to assign values to certain shared variables.

4 QRTAILAB

By means of the Real-Time Workshop (RTW) and the RTAI Target Language Compiler it is possible to create Real-Time C Code from Simulink models. As mentioned in section 2.1, Simulink is not able to execute and to debug the models. Instead of Simulink, QRtaiLab is able to start, to control and to debug the generated code. The open source software QRtaiLab offers nearly the same functionality as Xrtailab. Xrtailab is provided through RTAI-Lab, which is a component of RTAI. Mailboxes are used as inter-task communication. This realizes an exchange of data with the Real-Time Code. In addition it is possible to display and to log the control data. As Xrtailab uses the cross-platform C++ GUI toolkit (EFLTK), there was a need for reprogramming the software using QT4 (cross-platform application and UI framework). The problem was that EFLTK is not under active development and offers only limited functionality. EFLTK must be compiled and installed manually and needs a manually compiled Mesa-library. Compared to Xrtailab QRtaiLab is much easier to install and does not use OpenGL, which results in reduced hardware requirements. QRtaiLab also offers some additional features:

- saving / loading of block parameters
- auto scaling for scope signals
- visualization of small matrices

The last feature may be used to verify tactile sensor matrices from the SDH2. During the development of tactile reactive grasping skills, it is essential to visualize these matrices. QRtaiLab offers this visualization by using existing mailbox algorithms within the RTAI library. A Matrix is transferred to QRtaiLab using the "*RTAI_Log*" - block from the original RTAI library, Figure 5. Based on the large amount of data within each tactile sensor matrix and the limited data transmission the maximum size of a matrix is restricted to $[15 \times 10]$.

5 HOW TO CONFIGURE AND START A SDH-2 REAL-TIME SIMULINK MODEL

To create and run a Simulink model as Real-Time model in RTAI kernel it's necessary to take the following steps.

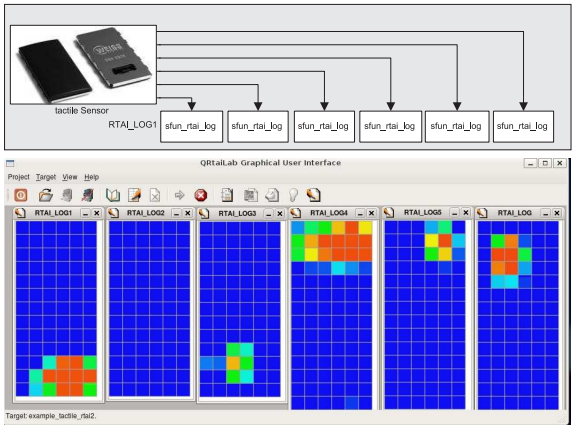


Figure 5: Visualization of tactile sensor matrices with QRtaiLab.

1. activate RTAI kernel: insert kernel modules
2. generate the desired named shared memory (section 2.2)
3. create a Simulink model, configure the model for RTAI Real-Time simulation:
   - edit solver: fixed step size (e.g. 5ms), discrete (no continuous states)
   - Real-Time Workshop: Target selection: rtai.tlc
   - use normal and **not** external mode
4. integrate the designed S-Functions from section 3.1 and 3.2 to connect and communicate with the shared memory
5. generate required C-Code for Real-Time kernel →(tools/real time workshop/build model)
6. open the current Matlab directory and start the executable (e.g. → ./modelXY -v -w)
7. start user space transceiver software to communicate with the SDH2
8. start QRtaiLab; connect to target and start the simulation

6 CONCLUSIONS AND FUTURE WORKS

Working with RTAI Simulink is quite different to working with Real-Time Simulink on Windows using xPC or Real-Time Windows Target. Installing and establishing an RTAI Linux is time-consuming. Unfortunately Simulink may not be used to execute and debug the created Real-Time models. Therefore, additional software is required. The possibilities with regard to the RTAI kernel configurations are

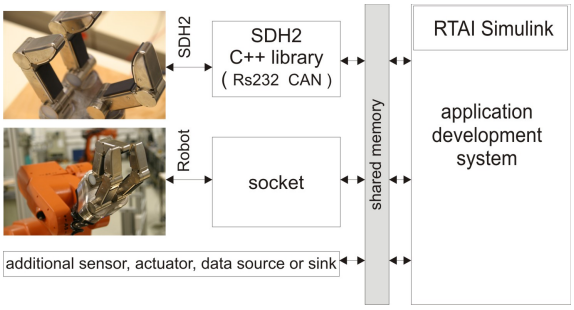


Figure 6: SHM Simulink Inter Task Communication.

very powerful. The realized data exchange enables software engineers to expand their Simulink models and to build up control systems for all available hardware, Figure 6. User-defined protocols may be designed to minimize the amount of information. Even the integration of kernel and user-space sockets into Real-Time models is feasible (Kiszka, 2004). The implementation of shared memory is very efficient. Within a short time it is possible to expand the transferred data set, to establish new named SHM and to adapt the required Simulink S-Functions. Additionally it is possible to create S-Functions which are able to read and write to some shared memory. Even the access to different SHM in one S-Function is feasible. Up to now it was possible to run different Real-Time executables at the same time on one system. With this presented SHM interface it is now possible to communicate and exchange data between different Real-Time modules. Each module and each communication can be constructed and designed within the Simulink environment. The design of parallel and distributed systems with Simulink becomes possible. The SHM interface is very stable and guarantees a high degree of operational reliability. The presented Simulink model of figure 5 is an example application of the designed SHM interface. High accessing frequencies f_a of $f_a > 1kHz$ can be achieved without difficulty. Application crashes (QRtaiLab, User-space transceiver, Real-Time executable) do not influence the interface.
All in all the realized Real-Time-Simulink seems to be more comprehensive than RTWT. This, however, is countered by a longer training period.

6.1 Future Works

In the near future it should be possible to communicate with the SDH2 even in kernel space. Therefore, an extension of the SDH2 C++ library is necessary. Aside from the facts that the sampling rate could be increased and that no additional software will be required for working with the robot hand, it could even facilitate the working environment. RTAI Simulink uses mailboxes for inter-task communication. This paper shows how to use shared memory. Perhaps shared memory is more suitable to transfer time-critical information particularly with regard to matrices. Raising time delays because of growing message queues are unfeasible. This will be essential if a large amount of data has to be transferred. While working on reactive grasping skills it will be necessary to debug different evaluation algorithms. Especially for large tactile matrices the mailbox communication becomes unusable. It was shown in section 4 that the maximum size of a matrix is restricted to $[15 \times 10]$. Maybe the usage of shared memory is able to solve the problem. Furthermore, the assembling of own SHM-based Simulink scopes is considerable. That is based on the fact that the usage of external software to debug the Real-Time Simulink models slows down the developing work.

ACKNOWLEDGEMENTS

This project is supported by SCHUNK GmbH & Co. KG, Lauffen.

REFERENCES

D. M. Dawson, W. D. (2002). Matlab-based control systems laboratory experience for undergraduate students: toward standardization and shared resources.

Kiszka, J. (2004). Real-time ethernet on top of rtai. Technical report, *University of Hannover, ISE - Real Time Systems Group*, Germany. www.rts.uni-hannover.de.

Mathworks (2010). Generate c code from simulink models and matlab code. *http://www.mathworks.com/products/rtw/*.

Quaranta, P. and Mantegazza, P. (2001). Using matlab simulink rtw to build real-time control applications in user space with rtai-lxrt. *Technical report*.

Quaranta, P. and Mantegazza, P. (May 27 2002). Interfacing linux rtai with matlab and simulink through real time workshop. *Technical report, University of Applied Sciences of Southern Switzerland (SUPSI), Dept. of cs and ee (DIE)*.

Ramadurai, S. (2001). Using real time linux target to compile and execute a simulink program. *Technical report, Clemson University, Intelligent Systems*.

RTAI (2010). The realtime application interface for linux from diapm. *https://www.rtai.org*.

SCHUNK GmbH & Co. KG (2010). Servo-electric 3-finger gripping hand sdh. *http://www.schunk.com*.

W. E. Dixon, D. M. D. (2001). Towards the standardization of a matlab-based control systems laboratory experience for undergraduate students. *Technical report*.

REUSABLE STATE MACHINE COMPONENTS FOR EMBEDDED CONTROL SYSTEMS

Krzysztof Sierszecki, Feng Zhou and Christo Angelov
Mads Clausen Institute for Product Innovation, University of Southern Denmark, Alsion 2, Soenderborg, Denmark
{ksi, zhou, angelov}@mci.sdu.dk

Keywords: Embedded control systems, Component-based design, Reusable and reconfigurable components, State machines.

Abstract: The paper presents a software design method for embedded applications, featuring reconfigurable components such as a State Machine (SM) function block operating in conjunction with a composite Signal Generator (SG) function block. The method emphasizes separation of concerns, whereby the State Machine realizes the reactive aspect of system behaviour in separation from the transformational aspect, which is delegated to the Signal Generator. Instances of these function blocks can be used to configure event-driven state machines executed periodically in the context of control system tasks (actors). When activated, the SM determines the control step that has to be executed in response to a particular event. The control step is then indicated to the SG, which generates the corresponding control signals. The SM has been implemented using a new Binary Decision Diagram (BDD)-based design pattern, resulting in a simple, yet powerful component capable of processing both discrete and continuous signals, which can be used to efficiently implement control actors for sequential and hybrid control applications.

1 INTRODUCTION

The conventional implementation of state machines is based on manual encoding of an abstract model representing either the behaviour or the structure of the state machine. In the former case, the behavioural model, i.e. the state transition graph, is converted into code using various kinds of design patterns, such as the switch-case design pattern (Samek, 2002). In the latter case, the software implementation models the hardware structure of the state machine. The resulting program computes the state transition logic functions and executes the actions that are associated with various states. In particular, that is how sequential control programs are developed for industrial automation systems, where control logic is encoded using domain-specific languages, such as those defined in standards IEC 61131-3 (John and Tiegelkamp, 2001) and IEC 61499 (Lewis, 2001).

In both cases, conventional design methods have a major shortcoming: the resulting implementation is not reusable, because the logic of the state machine is built into the code. Consequently, a new program has to be developed whenever an application is created or modified. This is a time-consuming and error-prone process whose complexity grows rapidly with the number of states and state transitions. To some extent, the situation can be alleviated via automated program generation using validated models, but code reusability is still a problem.

This problem can be solved by developing reusable state machine components, featuring standard state machine drivers operating on re-configurable data structures (Wang and Shin, 2002), (Wagner and Wolstenholme, 2003). The resulting software artifact can be viewed as an object of type ‘state machine’, which may have multiple instances defined by the contents of the encapsulated data structures (configuration tables). These can be configured and re-configured using a dedicated configuration tool. In this way, conventional software development is replaced by the configuration of reusable components and consequently, manual coding of state machines can be largely reduced and even eliminated.

This design philosophy has been adopted and further refined in a reconfigurable state-machine component for embedded control systems (Angelov et al., 2005). With that component, it is possible to invoke signal-processing function blocks (FBs) within the states of the execution control state

machine. However, the complexity of the component model is relatively high because it combines both reactive (state-based) and transformational behaviour in the context of hybrid control systems. This has motivated the development of the master-slave model presented in (Angelov et al., 2008) where system tasks (*actors*) are configured using stateful components of lower complexity, i.e. a state-machine function block coupled to a modal function block. However, in this model the state machine can process only binary event signals that are generated by pre-processing function blocks such as comparators, counters, timers, etc., which may result in increased complexity of the corresponding function block networks.

The above problem has been addressed with a design method featuring a new State Machine function block operating in conjunction with a composite Signal Generator, which is presented in this paper. The discussion is illustrated with a running example – a state machine used to implement one of the control actors of a Medical Ventilator Control System (Zhou et al., 2009).

The rest of the paper is structured as follows: Section 2 presents the design model of a state machine composed of State Machine and Signal Generator components and focuses on the implementation of a reconfigurable function block of class State Machine, using a design pattern that integrates pre-processing and control functions. Section 3 presents briefly the design pattern of the Signal Generator function block. A summary of the proposed software design method and its implications is given in the concluding section of the paper.

2 RECONFIGURABLE STATE MACHINE FUNCTION BLOCK

Embedded control system actors may exhibit complex stateful behaviour. Such actors can be built from reconfigurable software components, i.e. *State Machine* (*SM*) and *Signal Generator* (*SG*) function blocks. This approach emphasizes separation of concerns: the SM implements reactive behaviour by selecting the control step to be executed in response to a transition event defined in terms of one or more event signals that are sampled when the host actor is triggered. The control step is specified in terms of one or more output signals. These are generated by invoking a sequence of function blocks inside the SG - a composite component, which implements the transformational aspect of actor behaviour - from input signals to output signals (see Fig. 1).

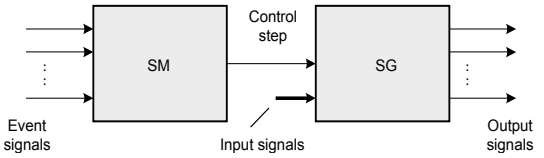


Figure 1: State Machine and Signal Generator function blocks.

The SM function block can be implemented using a new version of the *State Logic Controller (SLC)* design pattern originally introduced in (Angelov et al., 2005). The SLC employs a data structure that represents the state transition graph of a Moore machine realizing the desired control behaviour. It can be efficiently encoded as a table containing binary decision diagrams that represent the next-state mappings of various states *s* in the set of states *S*, and the corresponding control steps, in accordance with the state transition graph.

The next-state mapping of a state *s* is defined as the subset $Fs = \{s'\}$ involving those states that are immediate successors of *s*. Hence, a state transition graph can be symbolically represented by specifying the next-state mappings of all states $s \in S$, whereby the transition arcs are defined as tuples $(s, s' \mid s' \in F_s)$ that are associated with the corresponding transition events and event-priority symbols.

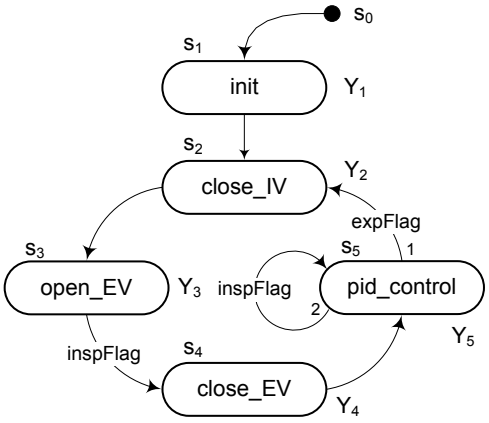


Figure 2: Medical Ventilator Control System: control actor state machine.

This technique will be illustrated with a running example, i.e. a control state machine encapsulated in the Volume Control Ventilation (VCV) actor of a Medical Ventilator Machine (Zhou et al., 2009), see Fig. 2. Its state transition graph can be represented as follows:

$Fs_0 = s_1 / Y_1 [c]$
$Fs_1 = s_2 / Y_2 [c]$
$Fs_2 = s_3 / Y_3 [c]$
$Fs_3 = s_4 / Y_4 [e_1], s_3 / NOP [!e_1]$
$Fs_4 = s_5 / Y_5 [c]$
$Fs_5 = s_2 / Y_2 [e_2, 1], s_5 / Y_5 [e_1, 2], s_5 / NOP [!e_1, !e_2],$

where s_0 denotes the initial pseudo-state, $s_1 - s_5$ denote operational states; $Y_1 - Y_5$ denote the corresponding control steps – initialization *(init)*, close inspiration valve (*close_IV*), open expiration valve (*open_EV*) close expiration valve *(close_EV)*, PID control of inspiration valve (*PID_control*); e_1 and e_2 denote events represented by signals *inspFlag* and *expFlag* respectively, and *c* denotes the default clock event; bracketed expressions denote the corresponding triggering events or <event – event-priority> pairs (when necessary).

Next-state mappings can be conveniently represented by means of binary decision diagrams, as shown in Fig. 3 for the example state machine. In these diagrams, circular nodes denote event signals that have to be tested in order to determine the current state/step to be executed from among the subset of successors of the previous state/step. These are tested in a predefined sequence that reflects the priority of the corresponding transitions.

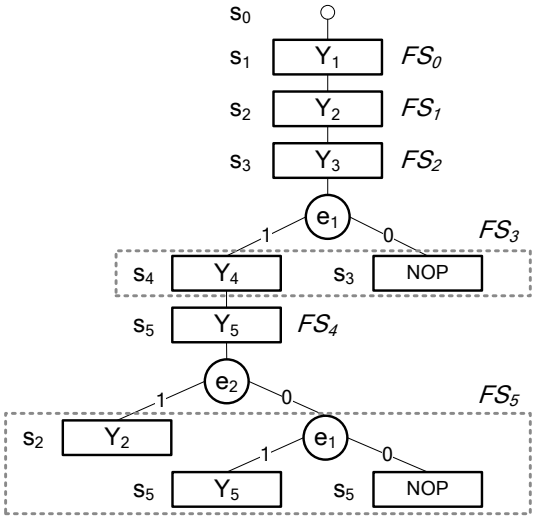


Figure 3: Binary decision diagrams for next state (step) mappings.

For example, it is possible to make a transition from s_5 to either s_2 or s_5, whereby the former transition has higher importance, i.e. lower event priority than the other one. That is encoded in the BDD whereby the event signal e_2 is checked first and the transition to s_2 – taken if e_2 is *true*; the transition to s_5 will be taken only if e_2 is *false* and e_1 – *true*. In case neither of the event signals is present, the parsing of the BDD ends up in a no-operation (NOP) node, meaning that no transition is taken and the previous state has to be maintained in the current period without executing a control action.

Table 1: Step Sequence Table (the BDD Table).

	Node	*SuccTrue / NextStateM*	*SuccFalse*	*Mapping*
0	Y_1	*1*	x	FS_0
1	Y_2	*2*	x	FS_1
2	Y_3	*3*	x	FS_2
3	e_1	*4*	*5*	FS_3
4	Y_4	*6*	x	
5	*NOP*	*3*	x	
6	Y_5	*7*	x	FS_4
7	e_2	*1*	*8*	FS_5
8	e_1	*6*	*9*	
9	*NOP*	*7*	x	

The binary decision diagrams of the next-state mappings can be encoded in a *Step Sequence Table*, (also called the BDD Table) as shown in Table 1. It consists of the columns *Node, successorTrue / NextStateMapping* and *successorFalse*, whereby the first column contains symbols denoting BDD nodes, and the other two columns – pointers to rows containing the corresponding BDD elements. The rows are grouped into segments containing the next-state mappings of states $s_0 - s_5$.

The Step Sequence Table can be interpreted much in the same way as its graphical counterpart. This can be done by a standard routine – a *State Machine Driver* (*SMD*), which is activated periodically by the host actor. Within each cycle, the SMD processes the BDD segment containing the successor states/steps of the state visited in the previous cycle, in order to determine the current state/step. If a state transition has taken place, the control step index variable is updated accordingly, and the associated Signal Generator function block is subsequently invoked to execute the corresponding control step. However, it is executed only when the state is visited for the first time; it will not be executed in subsequent cycles if the state is maintained, unless a self-transition is explicitly specified (execute-once semantics).

The above discussion is based on the assumption that event signals are Boolean variables supplied by external components, e.g. pre-processing function

blocks such as digital and analogue signal comparators, timers, event counters, etc. However, this may result in relatively complex function block networks modelling application actors. That problem can be eliminated by executing pre-processing operations as internal functions of the state machine function block. In that case, the condition nodes of the BDD may be interpreted as various types of compare, event counting and timer operations whose result is tested in order to make a branching decision within the BDD.

Each node of the BDD is thus associated with an operation that has to be executed by the State Machine Driver when processing the node. To that end, an operation code is used to specify the node operation, e.g. a control step executed in a state node. Likewise, it is necessary to specify operation codes (or function pointers) for various Boolean test, compare, and counter/timer operations executed in condition-testing nodes of the BDD.

The BDD table of the state machine function block can be encoded using records of the following format:

```
BDD_Record  = Operation, CondTest |
              Operation, ControlStep;

Operation   = CondOp1 | ... | CondOpk|
               CtrlStepOp;

CondTest    =  Operand1, Operand2,
               SuccessorTrue,
               SuccessorFalse;

ControlStep = ControlStepIndex,
               NextStateMapping;
```

where the *Operation* code specifies one of the following node-processing operations:

- Boolean operations
- Compare Integer operations
- Compare Real operations
- Count Events operation
- Control Step operation

In case of condition evaluation, the *CondTest* part of the BDD record contains operand fields, which are interpreted in the context of the executed node-processing operation as follows:

- *Boolean* operations use *Operand1* and *Operand2* as pointers to the tested variable locations.
- *Compare Integer* and *Compare Real* operations use *Operand1* as a pointer to the first compared variable and *Operand2* – as a pointer to the second compared variable or constant.
- The *Count Events* operation uses *Operand1* as a pointer to the counted event variable. The initial value of the event counter and the event counter itself are passed as parameters via the *Operand2* field.

The remaining two items of the condition-test record are used to implement branching decisions, as follows:

- *Successor1* is used as a pointer to the next line to be processed if the test/compare/counter operation returns *True.*
- *Successor0* is a pointer to the next line to be processed if the test/compare/counter operation returns *False*.

In case of control step execution, the *Operation* field is accompanied by a *ControlStep* field comprising:

- *ControlStepIndex* – an index of the control step that has to be executed in the current state. A NOP encoding of the control step index denotes *no operation.*
- *NextStateMapping* – a pointer to the first line of the corresponding next-state mapping.

The above operations are executed by the State Machine Driver while processing binary decision diagrams, as follows:

Boolean operations and compare operations are implemented by means of C-library compare routines, which return *True* or *False* depending on the result.

The *Count Events* operation interprets *Operand1* as a pointer to the input variable of the event counter. If that is a *NULL* pointer, the event counter is driven by the periodic timing events triggering the host actor, and operates as a timer measuring time intervals that are multiples of the actor period. The initial value of the event counter and the counter itself are passed as a pointer to a dedicated data structure in the second operand field. The operation returns *False* if [*counter*] != *0* after decrementing the counter; if [*counter*] = *0*, the operation re-initializes the counter and returns *True*.

The control step index is supplied to the SG as an input parameter used to invoke the corresponding sequence of function blocks in order to generate the required control signals. If the SM state in the current cycle is the same as in the previous one (*NOP* BDD node), a *NOP* control step index is generated, in accordance with the adopted execute-

once semantics. However, a self loop may be used if a control step has to be executed repeatedly in successive periods (e.g. *PID in state s_5 of Fig. 2).*

The algorithm given below can be used to implement a state machine driver for a reusable and reconfigurable function block of class State Machine:

```
void StateMachineDriver(void *FB)
{
  // Restore execution history
  BDD_Record *r = FB->tableRecord;

  // Determine current step and update
  // output
  do {
    // Condition-testing node?
    if (r->operation != CTRL_STEP_OP)
    {
      if ((r->operation)(r->operand1,
                         r->operand2))
      { // True
        r = r->successorTrue;
      }
      else { // False
        r = r->successorFalse;
      }
    }
    else {
      // Control step node?
      // Update control step index
      FB->ctrlStepIndex =
                      r->ctrlStepIndex;

      // Save execution history
      if ( r->ctrlStepIndex != NOP )
        FB->tableRecord =
                    r->nextStateMapping;

      return; // Leave the driver
    }
  } while( TRUE );
}
```

The SM function block instance is invoked with a pointer to an execution record of type *StateMachine* denoted as *FB*, which contains relevant data, such as output buffer for the control step index variable as well as a *tableRecord* history variable, i.e. a pointer to the first line of the next-state mapping segment, to be processed during the next activation of the SM function block.

3 SIGNAL GENERATOR FUNCTION BLOCK

The Signal Generator is a composite function block containing instances of function blocks that are to be invoked within statically defined execution schedules - control step (CS) sequences, in order to generate the control signals associated with the corresponding control steps.

To that end, the control step index generated by the SM is used to access a table containing records such as < *CSsequenceStart, CSsequenceLength* >, where each line corresponds to one particular control step. These two parameters are used to access a *Function Block Table* (*FBT*) where each line corresponds to a function block instance specified by the record < *FBfunction, FBinstance* >.

In particular, the *CSsequenceStart* is used to access a FBT record specifying the first function block instance of a control step sequence, and *CSsequnceLength* – the number of function block instances that have to be invoked in order to execute the control step. Hence, the FBT can be viewed as a concatenation of control step sequences that are specified by the corresponding sub-networks of the function block network encapsulated in the Signal Generator.

It is possible that several control steps generate one and the same continuous control signal, e.g. a control signal that is generated in both manual and automatic mode of operation. In that case, a Multiplexor FB shall be used, whereby different Multiplexor functions are invoked to switch the corresponding input signals to the multiplexor output.

Discrete (on/off) control signals can be generated by means of another kind of function block that may be invoked within the Signal Generator – the *Discrete Control Function Block (DCFB).* The DCFB employs the concept of *control memory* storing binary control words: a particular word is accessed using the corresponding control step index, and is subsequently stored in the DCFB output buffer.

Discrete control signals can also be generated by means of a digital multiplexor function block. In this case, the control step index is used to select an input binary word to be switched to the multiplexor output in order to generate the corresponding on/off control signals. This solution is preferable for applications featuring a small number of discrete control steps, as is the case with the example state machine of Fig. 1.

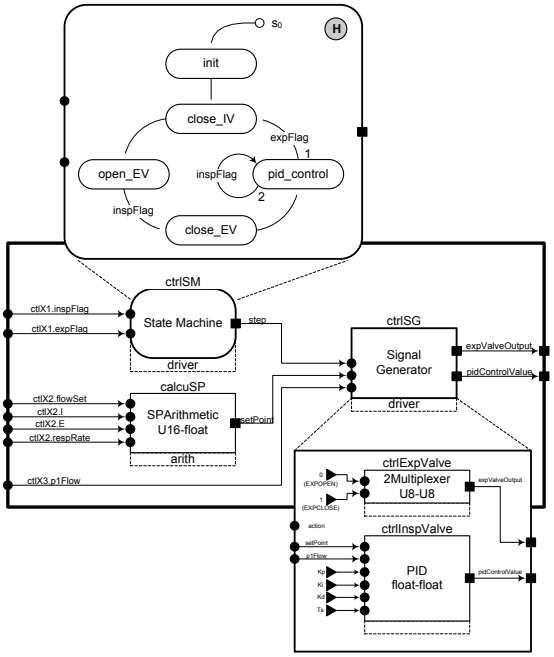


Figure 4: Control actor state machine implementation.

The Signal Generator of the example state machine is shown in Fig. 4. It incorporates two function block instances, i.e. a Multiplexor FB instance generating on/off control signals for the control steps *open_EV* and *close_EV*, and a PID FB instance generating the signal *pid_control* and *close_IV*. The PID function block encapsulates three functions: *initialize()*, *PID()* and *stop()*. The first one is invoked when executing the *init* control step and the other two – when executing the control steps *pid_control* and *close_IV* (by applying a zero voltage to the inspiration valve of the ventilator).

The combination of state machine and signal generator can be used to engineer sequential and modal continuous control systems, as well as systems generating continuous control signals in some states and discrete on/off control signals in other states, as shown in the example.

4 CONCLUSIONS

The paper presents a software design method for embedded control applications, which employs two types of reconfigurable component that can be used to configure control system tasks (actors) – State Machine and Signal Generator function blocks. The State Machine function block realizes the reactive (control flow) aspect of actor behaviour, in separation from the transformational (data flow) aspect, which is assigned to the Signal Generator.

The presented version of the State Machine function block is capable of processing any kind of input signal – Boolean, binary-coded or analogue in order to compute Boolean event variables needed to implicitly select the state to be activated, and to explicitly select the control step to be executed in that state. The index of the control step is then indicated to the Signal Generator, in order to activate the corresponding FB sequence used to generate the corresponding control signals.

The State Machine has been implemented as a reusable and reconfigurable function block using a new BDD-based State Logic Controller design pattern, resulting in a simple, yet powerful component that can be combined with a reconfigurable Signal Generator to efficiently implement state machines of arbitrary complexity for a broad range of sequential and hybrid control applications.

REFERENCES

Samek, M., 2002. *Practical Statecharts in C/C++: Quantum Programming for Embedded Systems*, CMP Books.

John, K-H., Tiegelkamp, M., 2001. *IEC61131-3: Programming Industrial Automation Systems*, Springer.

Lewis, R., 2001. *Modeling Control Systems Using IEC 61499*, Institution of Electrical Engineers.

Wagner, F., Wolstenholme, P., 2003. Modeling and Building Reliable, Re-usable Software. In *10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*. Huntsville, USA.

Wang, S., Shin, K.G., 2002. Constructing Reconfigurable Software for Machine Control Systems. In *IEEE Trans. on Robotics and Automation*, vol. 18, No 4

Angelov, C., Sierszecki, K., Marian, N., 2005. Design Models for Reusable and Reconfigurable State Machines. In *Lecture Notes in Computer Science*, v. 3824, Springer

Angelov C., Ke, X., Guo Y., Sierszecki K., 2008. Reconfigurable State Machine Components for Embedded Applications. In *SEAA 2008*, *34th EUROMICRO Conference on Software Engineering and Advanced Applications*, IEEE Computer Society

Zhou, F., Guan, W., Sierszecki, K., Angelov, C., 2009. Component-Based Design of Software for Embedded Control Systems: the Medical Ventilator Case Study. In *ICESS 2009*, *International Conference on Embedded Software and Systems*, IEEE Computer Society

APPLICATION OF HIERARCHICAL MODEL METHOD ON OPEN CNC SYSTEM'S BEHAVIOR RECONSTRUCTION

Yongxian Liu, Chenguang Guo, Jinfu Zhao, Hualong Xie
Institute of Advanced Manufacturing and Automation, Northeastern University, Shenyang, Liaoning, China
yxliu@mail.neu.edu.cn, gchg_neu@163.com, susanzhaojf@hotmail.com, hlxie@mail.neu.edu.cn

Weitang Sun
Shenyang Institute of Computer Science, Chinese Academy of Sciences, Shenyang, Liaoning, China
82603180@qq.com

Keywords: Finite state machine, Hierarchical model, State table, Behavior reconstruction.

Abstract: There are many shortcomings in the open CNC system control program developed by the traditional programming mode, such as maintenance difficulties and poor portability. The application and development of FSM in CNC system are researched in this paper, and the basic principles of FSM and reconstruction mechanism of FSM are introduced. The reconstruction process based on FSM by the application of hierarchical modeling method and status table are also constructed. At last, the adaptive control function of automatic adjusting feeding speed in three axis CNC milling machine is extended to realize the function definition of the software unit in control system and control logic separation.

1 INTRODUCTION

NC system is an abbreviated form of numerical control system. It is a complex multi-tasking controller with different levels of real-time requirements. In the system, each object's function, behavior, starting process and their mutual relationships between the operating system modeling must be a clear description. This is directly related to the system's performance and operating reliability. Kruth et al. (2001) use FSM to model the planning and monitoring knowledge in Machine tool control. Li et al. (2005) apply a hierarchical finite state machine to the system behaviors' model o improve the reconfigurability in an open architecture control system. Ma et al. (2007) propose a dynamical behavioral modeling and describe it in hierarchy finite state machine model. Aiming at the characteristics of modularity and reconfigurable in open architecture computer numerical control (CNC) system, Wang et al. (2007) adopted finite state machine to create the dynamically modeling. Lid et al. (2008) use hierarchic finite state machine to describe dynamical behaviors of the controller. Using FSM method, the machine control flow can be separated with the mechanical parts, and it can be changed separately. This approach greatly increases the openness of the CNC system by limiting the control program which depends on the particular machine behavior and operating in a local part. Based on the working principle and the hierarchical modeling method of finite state machine, we create the model of complex system to explore the application of finite state machine in numerical control system software development process.

2 THEORETICAL BACKGROUND

2.1 Finite State Machine

Finite state machine consists of the following elements:

1) State: The basic component of behavioral model. It reflects the stage and activity of an object in the system.

2) Transfer: The process from one state to another state of the objects.

3) Event: The events and conditions that caused the state transformation of object.

4) Action: The action of object when the state changes.

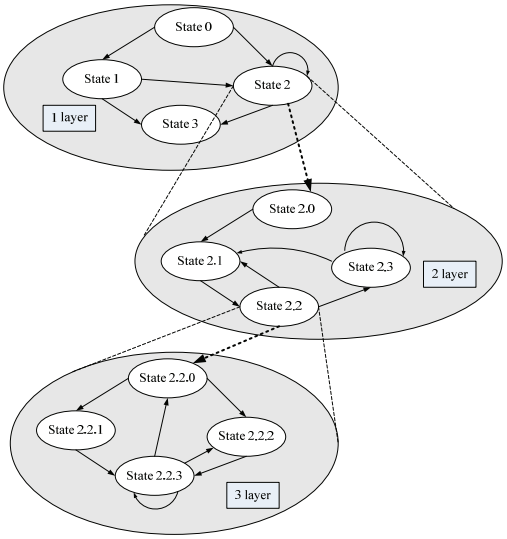


Figure 1: Hierarchical FSM model denotation for system behaviours.

For complex systems, such as using the above basic FSM model, there may be hundreds of the state, resulting in decreased efficiency of the system, and it is difficult to validate and maintain the system. Therefore, we need to expand the basic FSM model to be a hierarchical FSM model. In figure 1 using hierarchical modeling can make the system in a structured, hierarchical expression(Pritschow, Sperling, 1993).

2.2 State Table

Open CNC system's FSM consist of state set, event set, transfer set and action set, we call this state table (Wang, 2003).

State table can be used linked list structure. In the linked list, each unit contains a state sign and a pointer points to the transfer set. The structure of the state table as shown in Figure 2, every state has a number of transfers. In figure 2, state 7 is the composite state, which corresponds to the state of 7.1 and 7.2 in the lower layer of FSM.

When the needs of the system's changes, the system behavioral changes, such as add, delete, and replace in the state table. In figure 2, we replace "`action 3()`" with "`modifiedAction ()`", add a newState, new transfer "`&tran7`" and new event "`newEvent1`". Through the modification of the state table, we can reconstruct the system which is based on FSM.

In order to create and revision status table, the FSM-based class library can be used. The class library not only provides the API of state table's definition, query and modify, but also can be the drive center of FSM. Based on the current state, the class library finds the corresponding transfer, trigger required routine, so as to realize the system's specific functions.

3 MODULE DESCRIPTION AND RECONSTRUCTION PROCESS BASED ON FSM

3.1 Systems Module Description

CNC software is a real-time and multitasking software, it has two types of tasks: management and control. System management section contains input, I/O process, display, diagnose, etc. The control section contains decoding, cut adjust, speed processing, interpolation, position control, etc. Interpolation and position control is a real-time task. Decoding, cut adjust and speed processing is a condition task. The software should ensure synchronization between tasks.

The of function modules of a numerical control system include: Coordination Module, Pretreatment module, PLC module, Interpolation module, Motion control module.

Take the three coordinates CNC milling machine as an example, the simplified FSM models of coordination module, interpolation module and motion control module are shown in figure3.

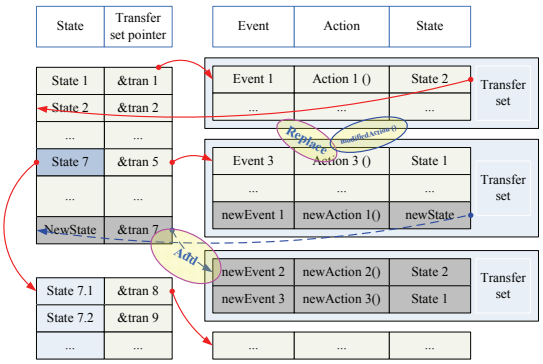
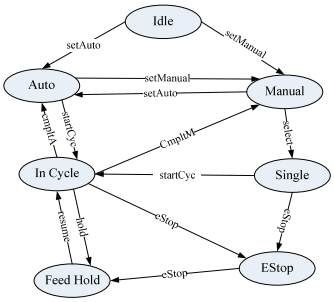
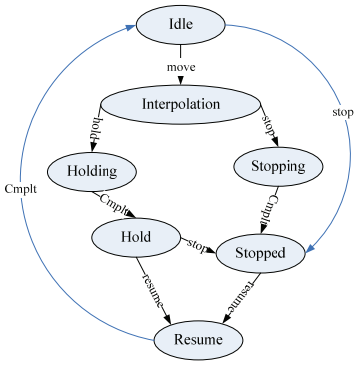


Figure 2: Structure of state table.

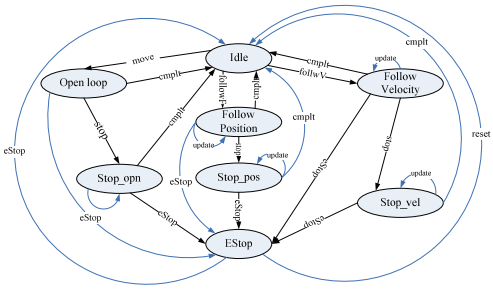
The complete FSM is formed by all the modules together according to the hierarchical structure. Task coordinator is dispatch center of the control software, and FSM is located in the top-level of the system. Other modules run in parallel under the management of FSM.



a: Coordination module.



b: Interpolation module.



c: Motion control module.

Figure 3: Simplified FSM models.

3.2 Flow of Reconfiguration

From the perspective of FSM, a new control flow and parts reconstruction means state, transfer, event and action's change. And this information is described with a state table which is not dependent on the system. Consequently, the reconstruction of the system means that the reconstruction of FSM's state table. Figure 4 shows the flow of reconfiguration system with FSM.

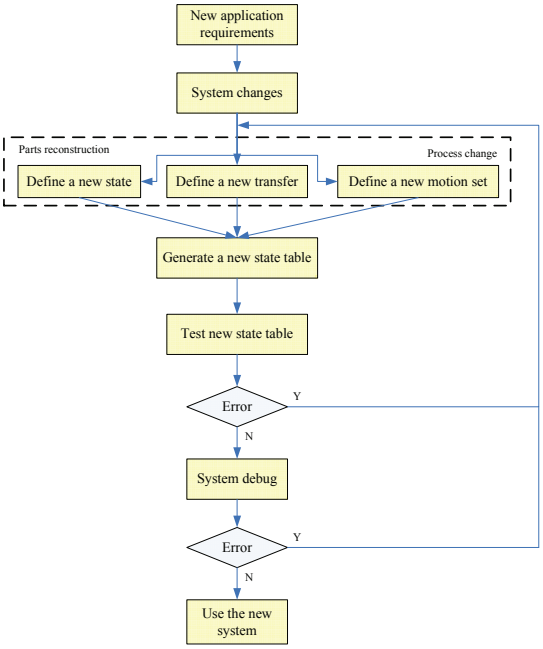


Figure 4: Flow of reconfiguration system with FSM.

4 THE CASE STUDY OF OPEN CNC SYSTEM'S FUNCTION EXPANSION

The following example is to expand the adaptive control function of automatic adjusting feeding speed in the three axis CNC milling machine. Due to adopted FSM model, only motion control module related with the motion control needs to modify in the system, and the other modules are no need to change. This method greatly reduced the system function expansion of programming workload.

As shown in figure 5, the modification of state table is consisted of adding a state of `"Velocity_Adjust"`, a transfer related with adjustable speed and a transfer related with the state of `"Follow_ Velocity"`. These new entries can be created with the method of `addTransition` provided by FSM-based class library. This function prototype is: `CFiniteStateMachine :: addTransition (string state, string event, string nextState, CFSMAction action)`.In this function, `CfiniteStateMachine` indicates the class of FSM. For example, `string` indicates a string class, `state` indicates the name of object's state, `event` indicates the name of the events that object received, `nextState` indicates the name of the next state that object will transfer, `CFSMAction` indicates action class of FSM, `action`

indicates the action triggered in the transfer process of object.

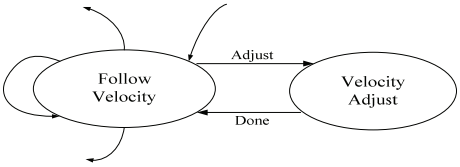


Figure5: Change of FSM state.

As shown above, it is very convenient and efficient to change status table with the method provided by FSM base class library. In addition, module developers still need to complete the programming of `AdjustAction` and `DoneAction`.The following is part codes of `AdjustAction ()`, it can realize the function of speed adjustment.

Currently popular adaptive speed control algorithm can be added in the above codes to satisfy all kinds of needs of users. According to adding speed regulating adaptive control functions in three axis CNC milling machine, the open CNC system can integrate other external sensor signals to implement users' unique control strategy.

5 CONCLUSIONS

As the dynamic behavior model of the system, FSM has the ability of behavior reconstruction. This method greatly increases the openness of the system. The model of behavior which is based on finite state machine stipulates the system behavior and control flow, cuts down the development cycle of the CNC system and enhances the reliability of the system. At the same time, FSM model can realize the function definition and control logic separation of software unit. It can improve flexibility of system reconstruction. Finally, the system's reconstruction based on finite state machine represents the reconstruction of state table. This can simplify the reconstruction process of the numerical control system absolutely.

ACKNOWLEDGEMENTS

This work was supported by Major national science and technology special projects during the 10th five-year plan (No. 2006BAF01A19), Key Scientific and Technological Project of Liaoning Province (No. 2006219008), Key Scientific and Technological Project of Shenyang City (No. 1071114-2-00), Postdoctoral Science Foundation under Grant 20080441093 and Key Laboratory Foundation of Liaoning Province under Grant 2008S088.

REFERENCES

Kruth, J. P., Van Ginderachter, T., Tanaya, P.I., Valckenaers, P. (2001). The use of finite state machines for task-based machine tool control. *Computers in Industry*, 46(3), pp. 247-258.

Li D. Y., Wang Y. Z., Fu H. Y. (2008). An open architecture motion controller for CNC machine tools. *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, Publisher: Inst. of Elec. and Elec. Eng.

Li X., Wang Y. Z., Liang H. B., Zhong L. (2005). Finite state machine application in open CNC. *Computer Integrated Manufacturing Systems*, 11(3), pp. 428-432.

Ma X. B., Han Z. Y., Wang Y. Z., Fu H. Y. (2007). Development of a PC-based open architecture software-CNC. *Chinese Journal of Aeronautics*, 20(3), pp. 272-281.

Pritschow G., Sperling W. (1993). *Open controllers -a challenge for the future of the machine tool industry*, Ann. CIRP 42 (1).

Wang Y. H. (2003). Study on a reconfigurable model of an open CNC kernel. *Journal of Materials Processing Technology*, 138(1-3), pp. 472-474.

Wang Y. Z., Liu T., Fu H. Y., Han Z. Y. (2007). Open architecture CNC system HITCNC and key technology. *Chinese Journal of Mechanical Engineering*, 20(2), pp. 13-16.

RECOGNIZING USER INTERFACE CONTROL GESTURES FROM ACCELERATION DATA USING TIME SERIES TEMPLATES

Pekka Siirtola, Perttu Laurinen, Heli Koskimäki and Juha Röning
Intelligent Systems Group, P.O. BOX 4500, FI-90014, University of Oulu, Finland
{*pesiirto, perttu, hejunno, jjr*}*@ee.oulu.fi*

Keywords: Gesture recognition, Accelerometer, Template-based matching.

Abstract: This study presents a method for recognizing six predefined gestures using data collected with a wrist-worn triaxial accelerometer. The aim of the study is to design a gesture recognition-based control system for a simple user interface. The recognition is done by matching the shapes that user's movements cause to acceleration signals to predefined time series templates describing gestures. In this study matching is done by using three different trajectory distance measures, the results show that the weighted double fold gives the best results. The superiority of this distance measure was shown using a statistical significance test. A user-dependent version of the method recognizes gestures with accuracy of 94.3% and a recognition rate of the user-independent version is 85.5%. This work was supported by the EU 6th Framework Program Project XPRESS.

1 INTRODUCTION AND RELATED WORK

In some situations gesture recognition is a good option for handling human-computer interaction because it enables natural interaction and no input devices, such as a keyboard and a mouse, are needed. In fact, in recent years gesture recognition systems have become more widely known among the public as new products controlled by gestures have become available. For instance gesture-controlled game consoles have recently appeared in stores.

This work studies the recognition of six gestures: *punch - pull*, *pull - punch*, *left - right*, *right - left*, *up - down* and *down - up*. These gestures were selected for this study because the future purpose of the gesture recognition system is to control a simple user interface. The interface view is a table and each cell of the table is a button. Using gestures, the user can decide which button to push. All the gestures selected for this study include two phases, action and counter-action, because it is natural for a human to return the hand to the original position after each performed gesture. Moreover, the gestures of this study were selected so that they can be performed by moving hand along one out of three coordinate axis so gestures contain movement mainly in one dimension, though the data is tridimensional. Therefore, for each gesture, two out of three acceleration channels are considered useless and are removed in order to improve the recognition rates.

Mainly two different types of methods have been used to recognize gestures: template-based methods and HMM methods. However, in (Ko et al., 2008) it is shown that gestures can be recognized more accurately using templates than by using HMM. Several template-based gesture recognition systems are proposed in the literature. In (Corradini, 2001) dynamic time warping (DTW) was used to recognize a small gesture vocabulary from offline data. The study did not use body-worn sensors, instead the system was trained with video sequences of gestures. A recognition accuracy of 92% was attained when five gestures such as stopping and waving were recognized.

In (Stiefmeier and Roggen, 2007) gesture signals were transformed into strings to make similarity calculations faster and real-time. The study used several inertial sensors: the sensors were attached to the lower arms, upper arms and the torso of the body. Human motion was presented by strings of symbols, and by combining the data provided by different sensors, the relative position of the arms with respect to the torso was computed. The method was demonstrated by spotting five predefined gestures from a bicycle maintenance task. An average classification rate of 82.7% was achieved when the method was tested with three persons.

Methods similar to those in our study were used in (Ko et al., 2008). The study used two wrist-worn

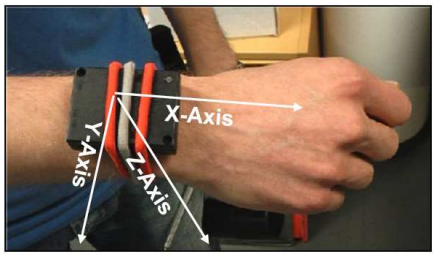


Figure 1: Accelerometer attached to the user's active wrist.

accelerometers, one on each wrist, and DTW as a distance measure. The frequency of the accelerometers was 150 Hz. Because DTW was used, the parameters for endpoint detection had to be defined by hand or by using a complex automated way. This made the DTW approach less generic. To make recognition faster, Ko *et al.* transformed the signal into a more compact representation by sliding a window of 50 samples with a 30-sample overlap through the signal. This way the number of points was reduced, making the system faster but at the same time making the system less sensitive to fast changes in the signal. Therefore, if fast movement is an important part of the gesture, this can cause problems. In the study 12 gestures of a cricket umpire, performed by four actors, were recognized. The system was tested using many different settings, for example offline and online. Each actor performed each gesture only once, and these data were used for testing, so the total number of gestures in the test data was only 48. The accuracy of the system was 93.75% when recognition was done using one template per gesture, as was done in our study, also.

The paper is organized as follows: Section 2 describes sensors and data sets. Section 3 introduces the techniques and gestures used in this study. Section 4 evaluates the performance and accuracy of the proposed method with the data sets presented in Section 2. Finally, conclusions are discussed in Section 5.

2 DATA SET

The data were collected using a mobile device equipped with a 3D accelerometer, 3D gyroscope, 3D magnetometer and two proximity sensors. In this study only accelerometers were used and the measuring device was attached to the active wrist of the user, see Figure 1. The sampling frequency of the accelerometer was 100Hz.

The data were collected from seven persons. Two separate *gesture data* sets were collected from each person: a training data set that included five repetitions each of six gestures and a test set that included ten repetitions of each gesture. These data sets were used to test how well the presented method detects the performed gestures from continuous data streams.

In addition, a *performance data* set around 30 minutes long that does not include any gestures was also collected from each person. This data set included other activities such as walking and working. This data set was used to test the speed and accuracy of the gesture recognition method. Accuracy was tested with this data set by testing how many false positive results the system found from a signal that did not include any predefined gestures.

3 METHODS

The purpose of the proposed method is to find predefined gestures from continuous accelerometer data streams. Basically, the system compares the shapes of studied signals with the shapes of template patterns describing gestures the system is trained to recognize. If the shape of the studied gesture is similar to the shape of some template, we know which gesture is performed. The quality of the proposed method depends mostly on four things: the quality of the templates, the accuracy of the similarity measure, selection of a proper similarity limit and the goodness of the sliding method. Of course, pre-processing also has its own important role.

3.1 Data Pre-processing

The raw acceleration data were pre-processed by first smoothing and then compressing them.

Smoothing was done using moving average (MA) filter and same weight were given to each point.This way the number of disturbances could be reduced and the signal became smoother and easier to handle.

After the smoothing, the signals were compressed in order to speed up calculations. The data were compressed so that they contained points of the original data where the derivative is equal to zero. Nevertheless no more than m sequential points were allowed to be removed from the original data. Therefore, if the number of points between two sequential derivative points was r and $r > m$, $r, m \in \mathbb{Z}_+$ then $\lfloor r/m \rfloor$ points, located at equidistant intervals, were also included in the compressed signal, see Figure 2.

3.2 Choosing Time Series Templates

The gestures of the study includes two phases, action and counter action. The use of gestures consisting of only one phase seemed to confuse users and the recognition system, because users tend to move their

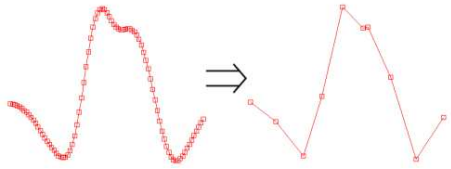


Figure 2: Template and a compressed version of it.

hand back to its original position. So, if both the action and counter-action are predefined as gestures that the system is trying to recognize, users easily accidentally perform two gestures instead of one. Selecting the gestures so that they contain an action and a counter-action solves this problem.

All six gestures of the study were selected so that movement is performed along only one out of three coordinate axis and thus only the data of this acceleration channels is needed in recognition, see Figure 3. So, the data given by two other channels is not important and it can be considered that it mostly consist disturbances, white noise and other non-valid information and therefore these channels are not used in recognition. Thereby the gestures and gesture templates are one-dimensional but the data are tri-dimensional and therefore templates are only needed to slid through one acceleration channel. The sensor was attached to the wrist so that the templates of the gestures *punch - pull* and *pull - punch* are slid through the x-axis accelerometer data, because these gestures cause mainly x-axis movement, see Figure 1. Correspondingly, *left - right* and *right - left* are slid through the y-axis data and *up - down* and *down - up* through the z-axis data. Elimination of two acceleration channels makes gesture recognition not only more accurate but also faster, because similarity calculation is faster from the one-dimensional acceleration signal than from the tri-dimensional signal.

3.2.1 User-dependent Case

In the user-dependent case, a *class template*, which is a template that is used to recognize a certain gesture, was selected for each gesture using a training data set. The class templates for each gesture were labeled from the training data set and they were used as training templates. Among these training templates, one at a time was selected as a candidate class template and used to recognize other training templates. As a class template describing gesture A was selected candidate class template $P_{A,i}$ which minimizes the sum

$$\sum_{j=1}^{n} d(P_{A,i}, P_{A,j}), \qquad (1)$$

when $1 \leq i \leq n$ and n is the total number of training templates of class A, $P_{A,j}$ is a training template of gesture A and $d(\cdot,\circ)$ is some similarity measure.

3.2.2 User-independent Case

A user-independent version of the presented gesture recognition system was tested using gesture templates selected in three different ways. The first two were suggested by (Ko et al., 2008).

Minimum Selection. In the case of minimum selection a class template describing gesture A was selected using Equation 1. In the user-dependent case the training and test data sets were performed by the same person, but in the user-independent case Equation 1 was applied to the training template set extracted from six persons. One person was left out as a test person.

Average Selection. Average selection was also done using Equation 1. Now the data of six persons were also used for training and the data of one person were left out for testing. Equation 1 was performed separately for each of the six training data sets to find six templates that have minimum inter-class distances, and the resultant six class templates were combined as one average template using the method presented in (Gupta et al., 1996). The method was used, though in (Niennattrakul and Ratanamahatana, 2007) it is claimed that the method does not produce the real average of two templates. Still, this DTW-based method works really well, giving a good estimation of the average template of two templates, and no better averaging methods seem to be available.

Evolutionary Selection. Evolutionary selection of a class template was done using a slightly modified version of the algorithm presented in (Siirtola et al., 2009). This evolutionary algorithm produces an optimal template describing some periodic time series. In this case the training data sets of six persons were fused so that the training gestures of each gesture A were combined as a periodic time series. This time series was given as an input to the algorithm presented in (Siirtola et al., 2009), and using it an optimal template describing the periods was found. The purpose of the algorithm is to find a template P that maximizes the fitness function

$$f(P) = \frac{\text{Number of found gestures using } P}{\text{Correct number of gestures}}. \qquad (2)$$

Template P which maximizes this function was selected as the class template.

3.3 Sliding and Decision Making

The purpose of sliding is to find every shape of time series T that is similar to class template P. In the case

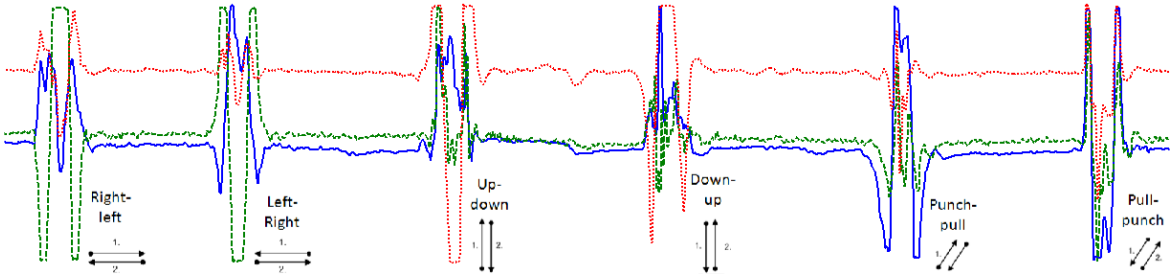


Figure 3: Gestures and corresponding tri-axial acceleration.

of online recognition, the functioning of the sliding method is in an important role because the starting point and ending point of the performed gesture is not known in advance. This means that if sliding method cannot find these points, it is not possible to recognize the gestures, either. In this study sliding method presented in (Siirtola et al., 2009) was used.

If more than one template P_i is found similar to some subsignal S, then the class of template P_i for which the ratio $\frac{d(S_k,P_i)}{\delta_i}$ is the smallest, where δ_i is predefined similarity limit for template P_i, is considered as a class of subsignal S_k. Note that different templates can have different similarity limits because some gestures are more difficult to perform and recognize than others.

4 EXPERIMENTS

4.1 Gesture Data

The gesture data presented in Section 2 were tested in two cases: a user-dependent case where the gesture recognition method was trained and tested with the same person's data, and a user-independent case where the data of the test person were not used in training.

4.1.1 User-dependent Case

User-dependent version of the method was tested using three different distance measures: *weighted double fold* (WDF) distance measure (Siirtola et al., 2008), *double fold* (DF) (Laurinen et al., 2006) and DTW. Also two different point-to-point distance measures were tested, Euclidean distance (ED) and Chebychev distance (CD).

The results (see Table 1) show that the combination of WDF and ED produces the highest total recognition accuracy; on average 94.3% of the gestures were recognized correctly. In fact, this combination gave the best recognition rates for six out of seven test persons. It seems that user-dependent version is very reliable because the gestures of every person can be recognized with an accuracy of at least 90%. When DTW and ED are used, the total recognition rate is 4.3 percentage units smaller. According to paired t-test with 6-degrees of freedom and $p = 0.95$ this improvement is statistically significant.

Note that the recognition rates drop when CD is used instead of ED as a point-to-point distance measure. The results show that using Chebychev distance and DTW or WDF, the gestures of some persons can be recognized with very high accuracy but the gestures of other persons seem to be difficult to recognize. For instance, using WDF the difference between the highest and lowest rates is almost 40 percentage units. CD considers only one dimension relevant, but the results show that by considering both dimensions relevant, as is done in the case of ED, better recognition rates are gained.

The good results using WDF came as no surprise since WDF is specially designed to measure the similarity of sparse signals, where the data points of the signals are not distributed at equal-length intervals (Siirtola et al., 2008). Compression presented in Section 3.1 produces such sparse signals.

4.1.2 User-independent Case

In the user-independent case a combination of WDF and ED was used as a distance measure because the results of Table 1 show that this combination gives the highest recognition rates.

Three different ways of choosing class templates for user-independent gesture recognition were introduced in Section 3.2.2. These methods were compared and the results are given in Table 2.

The highest recognition accuracy of 85.5% was achieved by using evolutionary selection. This template choosing method produced the best recognition results for five out of seven test persons. Based on these results it can be seen that the proposed method can be used for reliable user-independent gesture recognition. The other two methods seem to be almost equally accurate between themselves by recognizing gestures with an accuracy around 82%.

Table 1: Recognition accuracy in a user-dependent case. Comparison of local distance measures and similarity measures.

Measure / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
DTW + ED	95.0%	95.0%	93.3%	83.3%	88.3%	93.3%	81.7%	90.0%
DTW + CD	90.0%	91.7%	95.0%	86.7%	60.0%	91.3%	90.0%	86.4%
WDF + ED	95.0%	96.7%	98.3%	90.0%	90.0%	98.3%	91.7%	**94.3%**
WDF + CD	96.7%	65.0%	91.7%	71.7%	58.3%	96.6%	81.7%	88.6%
DF + ED	95.0%	91.7%	96.7%	78.3%	88.3%	85.0%	81.7%	88.1%
DF + CD	75.0%	88.3%	86.7%	60.0%	66.7%	70.0%	78.3%	74.6%

Table 2: Recognition accuracy in a user-independent case using different template choosing methods. MS = Minimum selection, AS = Average selection, ES = Evolutionary selection.

Method / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
MS	91.7%	81.7%	91.7%	85.0%	85.0%	58.3%	81.7%	82.1%
AS	100.0%	81.7%	91.7%	85.0%	85.0%	60.0%	76.7%	82.9%
ES	100.0%	83.3%	90.0%	80.0%	90.0%	68.3%	86.7%	**85.5%**

Table 3: User-independent recognition results using the evolutionary template selection method.

Gesture / Test person	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6	Person 7	Total
Punch-Pull	100.0%	90.0%	90.0%	70.0%	70.0%	80.0%	80.0%	82.6%
Pull-Punch	100.0%	90.0%	90.0%	60.0%	90.0%	80.0%	90.0%	85.7%
Right-Left	100.0%	70.0%	100.0%	100.0%	80.0%	50.0%	90.0%	84.3%
Left-Right	100.0%	100.0%	60.0%	100.0%	100.0%	50.0%	70.0%	82.6%
Up-Down	100.0%	80.0%	100.0%	70.0%	100.0%	80.0%	100.0%	90.0%
Down-Up	100.0%	70.0%	100.0%	80.0%	100.0%	70.0%	90.0%	87.1%
Total	100.0%	83.3%	90.0%	80.0%	90.0%	68.3%	86.7%	85.5%

When the results of the best methods of the user-dependent and -independent versions are compared, it can be seen that in most cases the user-independent version using evolutionary template selection gave around 10 percentage units worse results than the user-dependent version using WDF and ED. Still, the gestures of every person were recognized with high accuracy using evolutionary selection: the recognition rates for the gestures of person 1 were in fact better using the user-independent version. The only difference was person 6, whose gestures were recognized user-independently with an accuracy of only 68.3%. Using user-dependent templates, the gestures of person 6 were recognized almost perfectly, at a rate of 98.3%. Therefore, the problem is not that the gestures of the test data of person 6 were of low quality and impossible to recognize. One explanation for the weak user-independent recognition results is that person 6 had his/her own personal way of performing the gestures; person 6 especially seemed to perform the left-right and right-left gestures differently than the others. These gestures were recognized with an the accuracy of only 50%, see Table 3. Because persons seem to have at least two different ways of performing gestures, it could be wise to choose at least two templates per gesture, and not just one as was done in this study, to make user-independent gesture recognition more reliable.

4.2 Performance Test Data

Performance test data were collected to test the performance and accuracy of the gesture recognition system. These data did not include any of the six gestures and therefore all the detected gestures could be considered as false positive.

The gesture recognition system was tested using a Pentium D (3GHz, 2GByte RAM)) powered computer, and the results presented in Table 4 show that the running time of the presented method was about 15.0% of the duration of the performance test data sequences. This means the system is over six times

Table 4: Performance and accuracy of the method.

Person	Duration of performance data	CPU time for template matching	False positive results
1	1732s	306s	0
2	1672s	296s	2
3	1604s	340s	0
4	1557s	358s	3
5	1609s	218s	0
6	1791s	297s	5
7	1609s	206s	0
Total	11574s	1745s	10

faster than real-time, without any optimization, therefore the method can be used online.

A gesture recognition system is not allowed to produce false positive results often, because it would make the user-interface very frustrating to use. Table 4 also shows that the method is very accurate, meaning that it very seldom produced false positive results. The test sequences were all together over three hours long and the number of false positive results was only 10. So, on average, the proposed method produced one false positive result per 20 minutes.

5 CONCLUSIONS

This article presented a gesture recognition method for recognizing six predefined gestures. The method is based on template matching and the results show that it can recognize gestures very accurately and in real time. Three different distance measures were tested and the best results were achieved using weighted double fold distance measure. A user-dependent version of the system can recognize gestures with an accuracy of 94.3% when WDF distance measure is used. It was also shown that the improvement gained using WDF is statistically significant. User-independent version of the method can rocognize gestures with an accuracy of 85.5%. Compared with other studies, the recognition rates are really competitive. Most other studies use more than one sensor, unlike this study, and therefore the achieved results can be considered state-of-the-art.

The presented method works really well. It seldom produces false positive results and can recognize gestures with high accuracy. Still, the accuracy of the user-independent version could be improved by choosing more class templates, because people seem to have at least two different ways of performing gestures. Now only one template per gesture was used. The problem is that this would of course make the system slower.

The presented gesture recognition system is designed to control a simple user interface, and the next task is to fuse the gesture recognition system and the interface together.

ACKNOWLEDGEMENTS

This study was carried out with financial support from the Sixth Framework Programme of the European Community for research, technological development and demonstration activities in an XPRESS (FleXible Production Experts for reconfigurable aSSembly technology) project. It does not necessarily reflect the Commission's views and in no way anticipates the Commission's future policy in this area.

Pekka Siirtola would like to thank GETA (The Graduate School in Electronics, Telecommunications and Automation) for financial support.

REFERENCES

Corradini, A. (2001). Dynamic time warping for off-line recognition of a small gesture vocabulary. In *RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 82, Washington, DC, USA. IEEE Computer Society.

Gupta, L., Molfese, D., Tammana, R., and Simos, P. (1996). Nonlinear alignment and averaging for estimating the evoked potential. *Biomedical Engineering, IEEE Transactions on*, 43(4):348–356.

Ko, M., West, G., Venkatesh, S., and Kumar, M. (2008). Using dynamic time warping for online temporal fusion in multisensor systems. *Inf. Fusion*, 9(3):370–388.

Laurinen, P., Siirtola, P., and Röning, J. (2006). Efficient algorithm for calculating similarity between trajectories containing an increasing dimension. pages 392–399. Proc. 24th IASTED international conference on Artificial intelligence and applications, February 13 - 16, Innsbruck, Austria.

Niennattrakul, V. and Ratanamahatana, C. (2007). Inaccuracies of shape averaging method using dynamic time warping for time series data. In *ICCS '07: Proceedings of the 7th international conference on Computational Science, Part I*, pages 513–520, Berlin, Heidelberg. Springer-Verlag.

Siirtola, P., Laurinen, P., and Röning, J. (2008). A weighted distance measure for calculating the similarity of sparsely distributed trajectories. In *ICMLA'08: Proceedings of the Seventh International Conference on Machine Learning and Applications*.

Siirtola, P., Laurinen, P., and Röning, J. (2009). Mining an optimal prototype from a periodic time series: an evolutionary computation-based approach. In

Congress on Evolutionary Computation (CEC 2009), pages 2818–2824.

Stiefmeier, T. and Roggen, D. (2007). Gestures are strings: Efficient online gesture spotting and classification using string matching. In *In: Proceedings of 2nd International Conference on Body Area Networks (BodyNets)*.

CLASSIFICATION OF POWER QUALITY DISTURBANCES VIA HIGHER-ORDER STATISTICS AND SELF-ORGANIZING NEURAL NETWORKS

Juan José González de la Rosa, José Carlos Palomares, Agustín Agüera
Univ. Cádiz, Electronics Area, Research Group PAIDI-TIC-168
EPSA, Av. Ramón Puyol S/N, E-11202-Algeciras-Cádiz, Spain
juanjose.delarosa@uca.es

Antonio Moreno Muñoz
Univ. Córdoba, Electronics Area, Research Group PAIDI-TIC-168
Campus Rabanales, Ed. L. Da Vinci, E-14071-Córdoba, Spain
amoreno@uco.es

Keywords: Higher-Order Statistics (HOS), Neural classifiers, Power-quality.

Abstract: This work renders the classification of Power Quality (PQ) disturbances using fourth-order sliding cumulants' maxima as the key feature. These estimators are calculated over high-pass filtered real-life signals, to avoid the low-frequency 50-Hz sinusoid. Four types of electrical AC supply anomalies constitute the starting grid of a competitive layer performance, which manages to classify 90 signals within a 2D-space (whose coordinates are the minima and the maxima of the sliding cumulants calculated over each register). Four clusters have been clearly identified via the competitive network, each of which corresponds to a type of anomaly. Then, a Self-Organizing Network is conceived in order to guess additional classes in the feature space. Results suggest the idea of two additional sets of signals, which are more related to the degree of signals' degeneration than to real new groups of anomalies. We collaterally conclude the need of additional features to face the problem of subclass division. The experience sets the foundations of an automatic procedure for PQ event classification.

1 INTRODUCTION

Power Quality (PQ) analysis is becoming a key factor for the economy because equipment is highly sensitive to the power line signal's imperfections (Moreno and *et al*, 2007; IE3, 1995b). As a consequence, malfunctioning not only has to be detected, but also predicted and diagnosed, to identify the cause and prevent the system from a similar shock. This is reflected a posteriori in an increase in the amount and quality of the industrial production. The solution for a PQ problem implies the acquisition and monitoring of long data records from the energy distribution system, along with a detection and classification strategy, which allows the identification of the cause of these voltage anomalies. These perturbations can be considered as non-stationary transients, so it is necessary a battery of observations to obtain a reliable characterization. The goal of the signal processing is to get a feature vector from the target data, which constitute the input to the computational intelligence modulus, with the task of classification. Traditional measurement algorithms are mainly based in spectral analysis and wavelet transforms. Complementary second-order methods are based on the independence of the spectral components and the evolution of the spectrum in the time domain. Others are threshold-based functions, linear classifiers and Bayesian networks (De la Rosa et al., 2009,).

Recent works are bringing a higher-order statistics (HOS) based strategy, dealing with PQ analysis (De la Rosa et al., 2007; Ömer Nezih Gerek and Ece, 2006,), and other fields of Technology (De la Rosa et al., 2004; De la Rosa et al., 2008,). They are based in the following argument. Without perturbation, the 50-Hz of the voltage waveform exhibits a Gaussian behavior. Deviations can be detected and characterized via HOS; non-Gaussian processes need at least 3^{rd} and 4^{th}-order statistical characterization in order to be characterized, because 2^{nd}-order moments and cumulants could be not capable of differentiate non-Gaussian events.

Concretely, the problem of differentiating between a transient of long duration named oscillatory

(within a signal period) and a short duration transient, or impulsive transient (25 per cent of a cycle), has been outcome under controlled conditions in (De la Rosa et al., 2009,), and the idea of differentiating between healthy signals and signals with transients was pointed out and accomplished in (De la Rosa and Muñoz, 2009,). This problem was previously described in (Bollen et al., 2005) and matches HOS category, in the following sense. The short transient could also bring the 50-Hz voltage to zero instantly and, generally affects the sinusoid dramatically. By the contrary, the long-duration transient could be considered as a modulating signal (the 50-Hz signal is the carrier), and is associated to load charges (Bollen et al., 2005). Similarly, considering the statistical deviation from the Gaussian behavior that power disturbances add to the power line, it seems appropriate to launch the task of higher-order classification of more types of electrical anomalies, also considering the confluence of various perturbations in the same measurement register.

The contribution of this paper consists of the application of fourth-order central cumulants at zero lags to characterize PQ events in the time-domain (measuring maxima and minima values of higher-order cumulant sequences), along with the use of competitive layer and SOM as the classification tools. Four different sets of signals have been a priori established and confirmed using a competitive layer. The first set comprises *healthy* sine-waves from the power 50 Hz-line. Then, we consider signals with oscillatory mono-frequency (long duration) transients of relatively high amplitude; we also consider for the second set the signals with harmonics, which distort the shape of the sine-wave producing a not very high valued fourth-order cumulant. The third group gathers features' anomalies which appeared simultaneously in a signal, corresponding to impulsive transients (of short duration), and/or a weak sag (RMS descent), and/or oscillatory high-amplitude events. Finally, signals clearly affected by high-amplitude impulsive transients and/or deep sags are contained in the fourth set. Sets ♯3 and ♯4 may be joined in one, but signals in set ♯4 are clearly more affected and probably by only one type of perturbation. On the other hand, signals belonging to set ♯3 are generally affected by several anomalies. Consequently, four classes have been established with the possibility od upgrading the detection towards 6 clusters.

The paper is structured as follows. The following Section 2 explains the fundamentals of power quality monitoring. Higher-Order Statistics are outlined then in Section 3, to be followed by a summary on competitive layers and self-organizing networks in Section . Finally, results are presented in Section 5 and conclusions are drawn in Section 6.

2 POWER-QUALITY CHARACTERIZATION

As more and more electronic equipments enter the residential areas and business environment, the subjects related to PQ and its relationship to vulnerability of installations is becoming an increasing concern to the users. Particularly has arisen and increased the need to protect sensitive electronic equipment from damaging over-voltages. Things like lightning, large switching loads, non-linear load stresses, inadequate or incorrect wiring and grounding or accidents involving electric lines, can create problems to sensitive equipment, if it is designed to operate within narrow voltage limits, or if it does not incorporate the capability of filtering fluctuations in the electrical supply (Bollen et al., 2005; Moreno and *et al*, 2007; Paul, 2001).

The two main regulated aspects of PQ are the following:

- Technical PQ, which includes: Continuity of supply or reliability (long interruptions) and Voltage quality (voltage level variations and voltage disturbances).
- Commercial services associated to the wires (such as the delay to get connected to the grid, etc.) as well as commercial services for energy retail to regulated customers.

Assessment of voltage quality and power disturbances involves looking at electromagnetic deviations of the voltage or current from the ideal single-frequency sine wave of constant amplitude and frequency. A consistent set of definitions can be found in (IE3, 1995b). Regulation in European countries proposes to use the standard EN-50160 to define the voltage quality ranges. This norm actually describes the electricity through the technical characteristics that it has to fulfill to be considered as a compliant product. But there are a lot of undefined aspects; besides the fact that most of the regulator has yet to publish the technical criteria to measure and control all the voltage quality characteristics and decide what would be the penalization. The fact is that the only voltage quality aspect that is now enforced is the maximum voltage level variation settled to ± 7 % (which is actually different to the ± 10 % fixed on the EN-50160). But even this aspect is not yet controlled and there is not any defined procedure to determine if the limit has been reached.

On the other hand, the presence of disturbances on power distribution also affect the energy efficiency of the system. As far as energy efficiency is concerned in a power distribution system, the two dominant factors in PQ are its unbalanced and harmonic distortion. In an electrical installation when single-phase loads (especially those with non-linear characteristics), are not evenly and reasonably distributed among the three-phases of the supply, we are in the presence of unbalance. Voltage unbalance in a three-phase system causes three-phase motors to draw unbalanced current. This phenomenon causes additional power losses in conductors and motors and can cause the rotor of a motor to overheat.

Among all categories of electrical disturbances, the voltage sag (dip) and momentary interruption are the nemeses of the automated industrial processes. Voltage sag is commonly defined as any low voltage event between 10 and 90% of the nominal RMS voltage lasting between 0.5 and 60 cycles. Momentary voltage interruption is any low-voltage event of less than 10% of the nominal RMS voltage lasting between 0.5 cycles and 3 seconds. In medium voltage distribution networks, voltage sags are mainly caused by power system faults. Fault occurrences elsewhere can generate voltage sags affecting consumers differently according to their location in the electrical network. Even though the load current is small compared to the fault current, the changes in load current during and after the fault strongly influence the voltage at the equipment terminals. It has been discovered that the 85% of power supply malfunctions attributed to poor PQ are caused by voltage sag or interruptions of fewer than one second duration. Starting large motors can also generate voltage sags, although usually not so severe. In comparison with interruptions, voltage sags affect a larger number of customers and for some customers voltage sags may cause extremely serious problems. These can create problems to sensitive equipment if it is designed to operate within narrow voltage limits, or it does not have adequate ride-through capabilities to filter out fluctuations in the electrical supply.

Over-voltage is an RMS increase in the AC voltage, at the power frequency, for durations greater than a few seconds, and can be the result of a programmed utility operation, or the effect of an external eventuality (IE3, 1995a). Under normal operating conditions, the steady-state voltage is regulated by the utility within a limits band accepted by the EN-50160. Deviations from these limits are rare, and the utility can actuate readily to correct them, if known their occurrence, by acting on conventional distribution technologies, such as tap-changing transformers (Moreno et al., 2007).

However, under the typical operating conditions of a power system there is risk of damaging due to a momentary excess of voltage. Although by themselves they would be described as ”abnormal”, it is possible to distinguish between surges and swells. A surge is an over-voltage that can reach thousands of volts, lasting less than one cycle of the power frequency, that is, less than 16 milliseconds. A swell is longer, up to a few seconds, but does not exceed about twice the normal line voltage.

Power system surges, based on waveform shapes, can be classified into ”oscillatory transients” and ”impulsive transients” (IE3, 1995b; Paul, 2001) and they are the goal of the present research work. Oscillatory transient surges show a damped oscillation with a frequency range from 400 Hz to 5 kHz or more. Impulsive transient surges present a fast rise time in the order of 1 ns-10 μs over the steady state condition of voltage, current or both, that is unidirectional in polarity (primarily either positive or negative), reaching hardly twice the peak amplitude of the signal. They are damped quickly, presenting a frequency range from 4 kHz to 5 MHz, occasionally reaching 30 MHz.

Categorization of electrical transients based on waveform shapes and their underlying causes (or events) has been studied in (Bollen et al., 2005), and a few previous studies (De la Rosa et al., 2007; Ömer Nezih Gerek and Ece, 2006,) using HOS for feature extraction of electrical signals have shown the possibility of distinguish transients based on details beyond the second-order. In a real-life 50-Hz power line signal, it is very common to find these transients. In Fig. 1 we show an example of anomalous signal, including transients which are not classified between short-duration and long-duration. We show the computation of three higher-order time-domain statistics in order to introduce them qualitatively. The second-order estimator operates as an increase-of-power detector, showing the bumps associated to the increase of power, which in turn are associated to the anomalies of the power-line sine wave, but the third and fourth-order sliding cumulants have to be interpreted further. The most intuitive procedure is to calculate their maxima and minima.

Once the foundations of PQ have been settled down, in the following Section we present higher-order statistics in the time-domain in order to present the signal processing tool, along with a basic example which shows the performance of the statistical estimators which have been used in the computation of the cumulants. This example also motivates the use of HOS in time-series characterization.

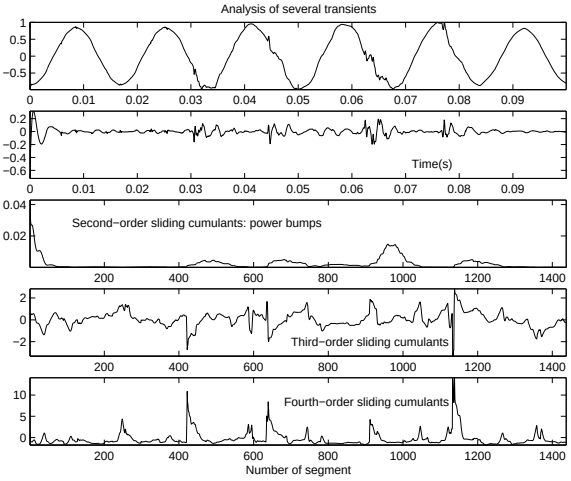


Figure 1: Several transients in the power line 50-Hz sine wave, and the computation of time-domain statistics. The signal has been previously normalized and high-pass filtered in order to remain with the transients.

3 HIGHER-ORDER STATISTICS

Higher-order cumulants are used to infer new properties about the data of non-Gaussian processes (De la Rosa et al., 2004,). In multiple-signal processing it is very common to define the combinational relationship among the cumulants of r stochastic signals, $\{x_i\}_{i\in[1,r]}$, and their moments of order $p, p \leq r$, given by using the *Leonov-Shiryaev* formula (Nikias and Mendel, 1993; Mendel, 1991)

$$Cum(x_1,...,x_r) = \sum(-1)^{p-1}\cdot(p-1)!\cdot E\{\prod_{i\in s_1} x_i\} \cdot E\{\prod_{i\in s_2} x_j\}\cdots E\{\prod_{i\in s_p} x_k\}, \quad (1)$$

where the addition operator is extended over all the partitions, like one of the form $(s_1, s_2, \ldots, s_p)$, $p = 1, 2, \cdots, r$; and $(1 \leq i \leq p \leq r)$; being s_i a set belonging to a partition of order p, of the set of integers *1,...,r*.

Let $\{x(t)\}$ be an rth-order stationary random real-valued process. The rth-order cumulant is defined as the joint rth-order cumulant of the random variables $x(t)$, $x(t+\tau_1)$,..., $x(t+\tau_{r-1})$,

$$C_{r,x}(\tau_1,\tau_2,\ldots,\tau_{r-1}) = Cum[x(t), x(t+\tau_1),\ldots,x(t+\tau_{r-1})] \quad (2)$$

The second-, third- and fourth-order cumulants of zero-mean $x(t)$ can be expressed via:

$$C_{2,x}(\tau) = E\{x(t)\cdot x(t+\tau)\} \quad (3a)$$

$$C_{3,x}(\tau_1,\tau_2) = E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\} \quad (3b)$$

$$\begin{aligned} &C_{4,x}(\tau_1,\tau_2,\tau_3) \\ &= E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\cdot x(t+\tau_3)\} \\ &- C_{2,x}(\tau_1)C_{2,x}(\tau_2-\tau_3) \\ &- C_{2,x}(\tau_2)C_{2,x}(\tau_3-\tau_1) \\ &- C_{2,x}(\tau_3)C_{2,x}(\tau_1-\tau_2) \end{aligned} \quad (3c)$$

By putting $\tau_1 = \tau_2 = \tau_3 = 0$ in Eq. (3), we obtain

$$\gamma_{2,x} = E\{x^2(t)\} = C_{2,x}(0) \quad (4a)$$

$$\gamma_{3,x} = E\{x^3(t)\} = C_{3,x}(0,0) \quad (4b)$$

$$\gamma_{4,x} = E\{x^4(t)\} - 3(\gamma_{2,x})^2 = C_{4,x}(0,0,0) \quad (4c)$$

The expressions in Eq. (4) are measurements of the variance, skewness and kurtosis of the distribution in terms of cumulants at zero lags (the central cumulants).

Normalized kurtosis and skewness are defined as $\gamma_{4,x}/(\gamma_{2,x})^2$ and $\gamma_{3,x}/(\gamma_{2,x})^{3/2}$, respectively. We will use and refer to normalized quantities because they are shift and scale invariant. If *x(t)* is symmetrically distributed, its skewness is necessarily zero (but not *vice versa*); if *x(t)* is Gaussian distributed, its kurtosis is necessarily zero (but not *vice versa*). In the experimental section, results are obtained by using sliding cumulants, i.d. a moving window in the time domain over which to compute the each cumulant.

To show the relevance of HOS an illustrative example is prepared. Four noise processes: Gaussian; uniform; exponential and Laplacian, previously catalogued in, and indistinguishable from the second-order perspective, are presented in this subsection in order to illustrate the importance of introducing higher-order cumulants. The *4*th-order cumulants are computed according to the estimate given in (De la Rosa et al., 2009,). We consider a 2048-point sample register for each random set of data. The four identical autocorrelation sequences contrast to the fourth-order ones, where substantial differences are observed, specially those corresponding to zero time lags. This can be seen in Fig. 2, where the *4*th-order cumulant sequences are depicted. The theoretical values of the cumulants at zero time-lag are: 0 (Gaussian), -1 (uniform), 6 (Exponential), 12 (Laplacian). The difference between the theoretical and the experimental value is due to the lack of averaging (only one sample register is consider). The convergency of the estimate is assured.

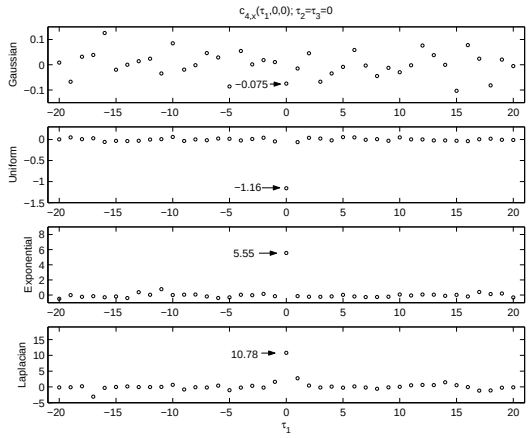


Figure 2: *4*th-order cumulant sequences for the four noise processes. Sample values at zero time lag are included in each sub-figure.

4 COMPETITIVE LAYERS AND SELF-ORGANIZING MAPS

In a competitive layer neurons distribute themselves to recognize frequently presented input vectors. The competitive transfer function accepts a net input vector **p** for a layer (each neuron competes to respond to **p**) and returns neuron outputs of 0 for all neurons except for the winner, the one associated with the most positive element of net input. If all biases are 0, then the neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1.

The winning neuron will move closer to the input, after this has been presented. The weights of the winning neuron are adjusted with the *Kohonen* learning rule (0.9 in the present case). Supposing that the *i*th-neuron wins, the elements of the *i*th-row of the input weight matrix (**IW**) are adjusted as shown in Eq. (5):

$$\mathbf{IW}_i^{1,1}(q) = \mathbf{IW}_i^{1,1}(q-1) + \alpha\left[\mathbf{p}(q) - \mathbf{IW}_i^{1,1}(q-1)\right], \tag{5}$$

where **p** is the input vector, q is the time instant, and α is the learning rate. The neuron whose weight vector was closest to the input vector is updated to be even closer. The result is that the winning neuron is more likely to win the competition the next time a similar input is presented. As more inputs are presented, each neuron in the layer closest to a group of input vectors soon adjusts its weights toward those inputs. Eventually, if there are enough neurons, every cluster of similar input vectors will have a neuron that outputs 1 when a vector in the cluster is presented, while outputting a 0 at all other times. Thus, the competitive network learns to categorize the input vectors.

Self-Organizing Maps (SOM) learn to classify feature input vectors according to how they are grouped in the input space. SOM differ from competitive layers in that neighbor-neurons learn to recognize neighboring sections of the input space. Thus, SOM learn both the distribution (as do competitive layers) and topology of the input vectors they are trained on. Consequently, instead of updating only the winning neuron, all neurons in its neighborhood are updated using the *Kohonen* rule. The neurons in the layer of a SOM are arranged originally in physical positions according to a topology function. A distance function allows the calculation of the distances between neurons. Thus, for the i*th* neighboring neuron, in the q*th* instant, we have the weight vector **w**, in Eq. (6):

$$\mathbf{w}_i(q) = \mathbf{w}_i(q-1) + \alpha[\mathbf{p}(q) - \mathbf{w}_i(q-1)]. \tag{6}$$

Thus, when a vector is presented, the weights of the winning neuron and its closest neighbors move toward. Consequently, after many presentations, neighboring neurons will have learned vectors similar to each other.

5 EXPERIMENTAL RESULTS

As conveyed in previous sections, the experiment comprises two phases. The feature extraction, and first stage, is based on the calculation of the maxima and minima of the 4*th*-order central cumulants at zero lags for each data recording; i.d., each signal is characterized in a 2-D space by a vector, whose coordinates correspond to the local maxima and minima of the 4^{th}-order central cumulants. A number of 90 different measured power-line signals were selected, containing different PQ anomalies. Secondly, the classification stage (on the 90 feature vectors) is based on the application of ANN as classification tools in a twofold frame. The mission of the competitive layer consists of confirming the existence of four different sets of signals' classes (a priori established in the research). Additionally, the SOM network is conceived to guess additional possible classes and, in case of finding out more, determine their nature and relationship with the firstly proposed four groups of features.

Each cumulant is computed over 50 points; this window's length (50 points) has been selected neither to be so long to cover the whole signal nor to be very short to loose information. The algorithm calculates the cumulant over 50 points, and then it jumps to the following starting point (next 50-point overlapped

group); as a consequence we have 98 per cent overlapping sliding windows (49/50=0.98). Then each computation over a window (called a segment) outputs a 4^{th}-order cumulant.

Besides, each *4*th-order cumulant, $Cum_{n,x}[i]$, associated to the *i*th computation segment has been normalized by $(Cum_{2,x}[i])^2$, in order to obtain categorization results associated to the shape of the sliding cumulants. This gives a real statistical characterization. If the cumulants are not normalized, the maxima and minima also gather information regarding the absolute value of the cumulants. The higher-order (n>2) normalized cumulants are the skewness and the kurtosis.

Before the computation of the biased cumulants, two pre-processing actions have been performed over the sample signals. First, they have been normalized because they exhibit very different-in-magnitude voltage levels. This disparity of voltage levels cannot influence the results of the categorization. Secondly, a high-pass digital filter (5th-order Butterworth model with a characteristic frequency of 150 Hz) eliminates the low frequency components which are not the targets of the experiment.

Once filtered, each signal contains one or more types of PQ events. Four different sets of signals have been a priori settled down empirically, based on the qualitative human knowledge, and then confirmed using a competitive layer. The first set comprises *healthy* sine-waves from the power 50 Hz-line. Then, we consider signals with oscillatory mono-frequency (long duration) transients of relatively high amplitude; we also consider for the second set signals with harmonics, which distort the shape of the sine-wave producing a not very high valued fourth-order cumulant. The third group gathers features' anomalies which appeared simultaneously in a signal, corresponding to impulsive transients (of short duration), and/or a weak sag (RMS descent), and/or oscillatory high-amplitude events. Finally, signals clearly affected by high-amplitude impulsive transients and/or deep sags are contained in the fourth set. Sets ♯3 and ♯4 may be joined in one, but signals in set ♯4 are clearly more affected and probably by only one type of perturbation. On the other hand, signals belonging to set ♯3 are generally affected by several anomalies. Consequently, four classes have been established with the possibility of upgrading the detection towards 6 clusters. The limits for the four classes' intervals, in units of cumulants maxima are: [0,7], [7,12], [12,20], [20,40]. These classes can be appreciated in Fig. 3, in the upper sub-graph, for the competitive layer training results. The lower subgraph in Fig. 3, shows the results of applying the SOM network and can be seen the shifting phenomenon that occurs for the final weights vector after the training stage of the NNT over 50 epochs. This result conveys the idea of the SOM network used to refine the classification, more than performing the coarse sub-division in anomalies' subclasses.

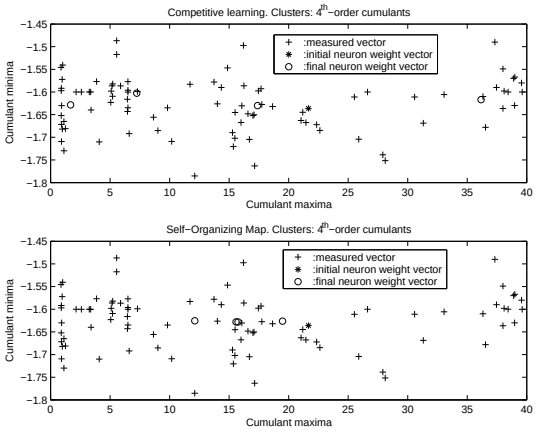


Figure 3: Four clusters for 4 types of signals. A *randtop* 2×2 topology has been selected over 50 epochs training. Upper graph: competitive layer performance. Down graph: SOM performance.

The separation between classes (inter-class distance) is well defined in the 2-D feature graph for the competitive layer. Consequently, the four types of PQ events are clustered. The correct configuration of the clusters is corroborated during the simulation of the neural network, in which we have obtained an approximate classification accuracy of 95 percent. During the simulation, new signals (randomly selected from our data base) were processed using this methodology. The accuracy of the classification results increases with the number of data. To evaluate the confidence of the statistics a significance test has been conducted. As a result, the number of measurements is significantly correct.

An attempt to classify signals according a 6-cluster pattern has been developed. The limits for the four classes' intervals, in units of cumulants maxima are: [0,2], [2,7], [7,12], [12,20], [20,30], [30,40]. The new proposed intervals (added to the four classes proposal) are related to graded anomalies. The training results are displayed in Fig. 4, conveying the idea that, new classes are not really new anomalies.

In fact, despite the fact that the competitive layer manages to classify the signals into 6 classes (we force it), when we apply SOM networks, due to the influence of the close neighbor neurons, the final weight vectors are shifted or moved to new positions, depending on the geometry of the network (rand-to, hex-top). This clearly confirms the idea of a graded

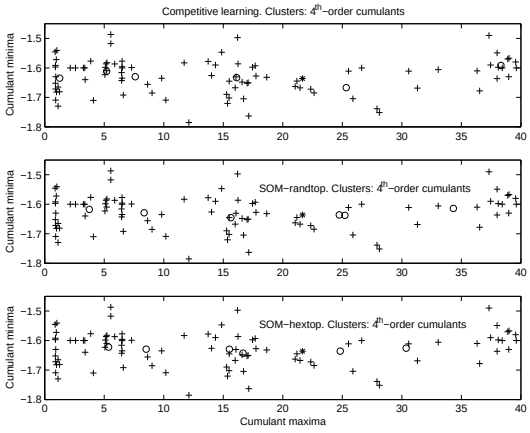


Figure 4: Six clusters for 6 possible types of signals after 50 epochs training. A 2×3 topology has been selected. Upper graph: competitive layer performance. Middle graph: SOM performance for a rand-top topology. Down graph: SOM performance for an hex-top topology.

anomaly, because the weight vectors are not located in the same position for both types of network's geometry.

6 CONCLUSIONS

In this paper an automatic procedure to classify electrical PQ anomalies has been proposed. The method comprises two stages. The first includes preprocessing (normalizing and filtering) and outputs the 2-D feature vectors, each of which coordinate corresponds to the maximum and minimum of the central *4*th-order cumulants. The second stage is based in computational intelligence and uses a competitive layer to confirm the existence of 4 classes, related to the different groups of anomalies. Then a SOM network confirms that newly added classes (proposed empirically) are not really new. New sub-divisions are related to degree of the degree of the anomaly. The geometry of the SOM network confirm this fact, moving the final weight vectors to different positions. Future work is designed to deal with a great number of signals (more than 90), trying to guess more classes with the aim of generalizing the method.

ACKNOWLEDGEMENTS

The authors would like to thank the *Spanish Ministry of Science and Innovation* for funding the research project TEC2009-08988. Our unforgettable thanks to the trust we have from the *Andalusian Government* for funding the Research Unit PAIDI-TIC-168 in *Computational Instrumentation and Industrial Electronics*.

REFERENCES

(1995a). IEEE Guide for service to equipment sensitive to momentary voltage disturbances. Technical Report IEEE Std. 1250-1995, The Institute of Electrical and Electronics Engineers, Inc.

(1995b). IEEE Recommended practice for monitoring electric power quality. Technical Report IEEE Std. 1159-1995, The Institute of Electrical and Electronics Engineers, Inc.

Bollen, M. H. J., Styvaktakis, E., and Gu, I. Y.-H. (2005). Categorization and analysis of power system transients. *IEEE Transactions on Power Delivery*, 20(3):105–118.

Mendel, J. M. (1991). Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305.

Moreno, A., Flores, J., Oterino, D., and De la Rosa, J. J. G. (2007). Power line conditioner based on ca pwm chopper. In *ISIE 2007, Proceedings of the 2007 IEEE International Symposium on Industrial Electronics*, pages 2454–2456, June 2007.

Moreno, A. and *et al* (2007). *Mitigation Technologies in a Distributed Environment*. Power Systems. Springer-Verlag, 1 edition.

Nikias, C. L. and Mendel, J. M. (1993). Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, pages 10–37.

Ömer Nezih Gerek and Ece, D. G. (2006). Power-quality event analysis using higher order cumulants and quadratic classifiers. *IEEE Transactions on Power Delivery*, 21(2):883–889.

Paul, D. (2001). Low-voltage power system surge overvoltage protection. *IEEE Transactions on Industry Applications*, 37(1):223–229.

De la Rosa, J. J. G., Lloret, I., Puntonet, C. G., and Górriz, J. M. (2004). Higher-order statistics to detect and characterise termite emissions. *Electronics Letters*, 40(20):1316–1317. Ultrasonics.

De la Rosa, J. J. G., Lloret, I., Puntonet, C. G., Piotrkowski, R., and Moreno, A. (2008). Higher-order spectra measurement techniques of termite emissions. a characterization framework. *Measurement (Ed. Elsevier)*, 41(1):105–118. Available online 13 October 2006.

De la Rosa, J. J. G., Moreno, A., and Puntonet, C. G. (2007). A practical review on higher-order statistics interpretation. application to electrical transients characterization. *Dynamics of continous discrete and Impulsive Systems-Series B: Applications and Algorithms*, 14(4):1577–1582.

De la Rosa, J. J. G. and Muñoz, A. M. (2009). Higher-order characterization of power quality transients and their classification using competitive layers. *Przegąd Elektrotechniczny-Electrical Review*, 10(Issue 85):284–289.

De la Rosa, J. J. G., Muñoz, A. M., Gallego, A., Piotrkowski, R., and Castro, E. (2009). Higher-order characterization of power quality transients and their classification using competitive layers. *Measurement (Ed. Elsevier)*, 42(Issue 3):478–484.

SURVEY OF ESTIMATE FUSION APPROACHES

Jiří Ajgl and Miroslav Šimandl
Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen
Univerzitní 8, Plzeň, 306 14, Czech Republic
{*jiriajgl, simandl*}*@kky.zcu.cz*

Keywords: Dynamic systems, State estimation, Optimal estimation, Sensor fusion, Filtering problems.

Abstract: The paper deals with fusion of state estimates of stochastic dynamic systems. The goal of the contribution is to present main approaches to the estimate fusion which were developed during the last four decades. The hierarchical and decentralised estimation are presented and main special cases are discussed. Namely the following approaches, the distributed Kalman filter, maximum likelihood, channel filters, and the information measure, are introduced. The approaches are illustrated in numerical examples.

1 INTRODUCTION

The classical estimation theory deals with estimating the value of some attribute by using measured data. (Simon, 2006) reviews the optimal state estimation techniques for linear systems and their extension to non-linear systems. However, there are other dimensions of the estimation problem. The direction discussed here is the multisensor problem that assumes the system state to be estimated by multiple estimators. Each estimator uses different data sets and it can communicate its estimate to the other estimators. The question is how to combine multiple estimates to obtain optimal results.

The key issues in the multisensor fusion are communication and dependences. In practise, it is possible to communicate raw measurements among estimators. In such a case, each estimator can process the measurements only and no estimate fusion is required. But in the case of the on-line state estimation of dynamic systems the out-of-sequence problems occurs. Updating the estimate by an old measurement is complicated, see (Bar-Shalom, 2002) or (Challa et al., 2003). Moreover, in general network of estimators, the estimators must log a list of all measurements they have processed or the measurement must be passed with a list of estimators that have processed it. Otherwise the multiple processing of the same data is inevitable.

If two estimators use measurements with dependent errors, their estimates will be dependent. A nonzero state noise causes dependence of the estimates as well as the communication of the estimates with the consequent fusion. The fused estimate and the estimates before fusion are obviously dependent. In a rooted tree estimator network, some restarts of the estimators can be applied to solve the communication dependence problem, see (Chong et al., 1999).

In the fusion point of view, the classical estimation is named as centralised. A central estimator processes raw measurements only. If the estimators are organised in a rooted tree, the root is called a fusion centre and the fusion is denoted as hierarchical or distributed. If there is not a fusion centre, the fusion is decentralised. Only a local knowledge of the network is usually assumed in these cases. The above mentioned approaches have been introduced in the literature by different ways during last decades. However, a unique survey of the approaches is missing.

Therefore, the aim of the paper is to give a survey of main results in estimate fusion and to show numerical illustrations. Both hierarchical and decentralised estimation are presented and discussed. In the hierarchical framework, namely the distributed Kalman filter and the fusion based on the maximum likelihood estimation are considered. In the decentralised framework, the stress is laid on the channel filters and the information measure approach.

The paper is organised as follows. Section 2 defines the fusion problem, section 3 and 4 discuss the hierarchical and decentralised approaches, respectively. A numerical example is given in section 5 and finally section 6 summarises the fusion problems.

2 PROBLEM STATEMENT

Let the discrete-time stochastic system be described by state transition and measurement conditional probability density functions

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k), \quad (1)$$

$$p(\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \ldots, \mathbf{z}_k^{(N)}|\mathbf{x}_k). \quad (2)$$

where $\mathbf{z}_k^{(j)}$, $j = 1, \ldots, N$, are local measurements at time k, $k = 0, 1, \ldots$ and the initial condition $p(\mathbf{x}_0)$ is known. Let the system be linear gaussian. In such case, analytical solutions to estimation problems exist. The linear gaussian system can be described by state and measurement equations

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k, \quad (3)$$

$$\mathbf{z}_k^{(j)} = \mathbf{H}^{(j)}\mathbf{x}_k + \mathbf{v}_k^{(j)}, j = 1, \ldots, N, \quad (4)$$

where $\mathbf{F} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{H}^{(j)} \in \mathbb{R}^{n_z^{(j)} \times n_x}$, and $\mathbf{G} \in \mathbb{R}^{n_x \times n_w}$ are known matrices, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the immeasurable system state and $\mathbf{z}_k^{(j)} \in \mathbb{R}^{n_z^{(j)}}$ is the local measurement coming from j-th sensor. The variables $\mathbf{w}_k \in \mathbb{R}^{n_w}$ and $\mathbf{v}_k^{(j)} \in \mathbb{R}^{n_z^{(j)}}$ represent the state and measurement white Gaussian noises with zero mean and with known covariance matrices $\mathbf{Q}$, $\mathbf{R}^{(jj)}$, respectively. The processes $\{\mathbf{v}_k^{(j)}\}$ are independent of the process $\{\mathbf{w}_k\}$ and all of them are independent on the system initial state described by the Gaussian pdf $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0 : \overline{\mathbf{x}}_0, \mathbf{P}_0)$. The measurement error processes $\{\mathbf{v}_k^{(j)}\}$ can be generally mutually dependent, with cross-correlations $,\mathbf{R}_k^{(ij)} = \mathrm{E}(\mathbf{v}_k^{(i)}\mathbf{v}_k^{(j)\mathrm{T}})$, but there are often assumed to be independent, $\mathbf{R}_k^{(ij)} = 0$ for $i \neq j$.

Let each sensor have its estimator, i.e. there exist N state estimates $\hat{\mathbf{x}}^{(j)}$, $j = 1, \ldots, N$, with corresponding error covariance matrices $P^{(j)}$. The estimators are connected with some others by data link. The communication network can be described by a directed graph with nodes in each sensor and with edges representing the oriented data links. It is assumed that measurements coming from other sensor nodes can not be processed directly, e.g. due to the unknown measurement equation of the respective sensors, or the communication of the measurements would be ineffective. So it is assumed that only the estimates are communicated. The goal of the fusion is to combine local estimates.

3 HIERARCHICAL FUSION

In the hierarchical fusion, the local estimates are communicated to a fusion centre. The method are based on the classical one-sensor estimation, which is described in subsection 3.1. The distributed Kalman filter extracts independent information from the estimates and is discussed in subsection 3.2. In the maximum likelihood approach, the estimates are regarded as dependent measurements. The respective fusion is shown in subsection 3.3.

3.1 Optimal Centralised Estimate

In the case of one sensor system, there is no fusion of estimates. The classical Kalman filter solution is the exact Bayessian solution to the filtering problem for a linear Gaussian system. You can see (Simon, 2006) for many numerical approximations to the exact solution for non-linear systems. The Kalman filter estimate is a standard against which other methods can be compared. The filtering (measurement update) equations

$$\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} = \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{z}_k, \quad (5)$$

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{H}_k, \quad (6)$$

can be interpreted as a fusion of the predictive estimate with the information based on the last measurement only. The prediction (time update) equations

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k\hat{\mathbf{x}}_{k|k}, \quad (7)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k\mathbf{P}_{k|k}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k. \quad (8)$$

correspond to the dynamics of the system. If more explicit notation is required further in this article, the general conditional pdf notation will be used. The exact Bayessian solution is given by

$$p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{k-1}) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1}) \quad (9)$$

$$p(\mathbf{x}_{k+1}|\mathbf{Z}_k) = \int_R p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_k)\mathrm{d}\mathbf{x}_k \quad (10)$$

where $\propto$ means proportional to and $\mathbf{Z}_k \triangleq \{\mathbf{z}_k, \mathbf{Z}_{k-1}\}$ denotes the set of all measurements up to the time k.

The centralised estimator is a hypothetical estimator which assumes that all measurements are immediately available to the estimator and that the correspondent measurement equations are known at the centre. The local measurement equations (4) can be merged to one equation with

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}_k^{(1)} \\ \vdots \\ \mathbf{z}_k^{(N)} \end{bmatrix}, \ \mathbf{H}_k = \begin{bmatrix} \mathbf{H}_k^{(1)} \\ \vdots \\ \mathbf{H}_k^{(N)} \end{bmatrix}, \ \mathbf{v}_k = \begin{bmatrix} \mathbf{v}_k^{(1)} \\ \vdots \\ \mathbf{v}_k^{(N)} \end{bmatrix}, \quad (11)$$

$\mathbf{R}_k = [\mathbf{R}_k^{(ij)}]_{i,j=1}^N$. The centralised Kalman filter is given by (5)-(8) and (11).

3.2 Distributed Kalman Filter

The distributed Kalman filter consists of N local Kalman filters which send their estimates to one fusion centre. It is also possible to distribute the local filters recursively. The name hierarchical Kalman filter is also used. Note that the term decentralised is misused in the literature to express that this filter is not the centralised one.

The main assumption is the independence of the local measurement errors,

$$\mathbf{R}_k^{(ij)} = 0, \quad i \neq j. \tag{12}$$

Then the pieces of information gained from the same time measurements are independent and can be simply summed up. The fusion centre filtering equation can be derived from (5), (6) with the use of (11) as

$$\begin{aligned}\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} &= \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1} + \\ &+ \sum_{j=1}^{N}\left(\mathbf{P}_{k|k}^{(j)^{-1}}\hat{\mathbf{x}}_{k|k}^{(j)} - \mathbf{P}_{k|k-1}^{(j)^{-1}}\hat{\mathbf{x}}_{k|k-1}^{(j)}\right), \quad (13)\end{aligned}$$

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{j=1}^{N}\left(\mathbf{P}_{k|k}^{(j)^{-1}} - \mathbf{P}_{k|k-1}^{(j)^{-1}}\right), \quad (14)$$

where indexes $^{(j)}$ denotes the local estimates. The fusion centre predictive equations are identical to (7), (8). It is possible to compute the predictive estimates at each local estimator, but it requires to send predictive estimate to the fusion centre. Instead of that, the fusion centre predictive can be send to each local estimator where it replaces the local estimate

$$\hat{\mathbf{x}}_{k+1|k}^{(j)} \leftarrow \hat{\mathbf{x}}_{k+1|k}, \quad \mathbf{P}_{k+1|k}^{(j)} \leftarrow \mathbf{P}_{k+1|k}, \tag{15}$$

$j = 1,\ldots,N$. This feedback brings the globally optimal estimate to each local estimator and the estimation is expected to be better if the extension to nonlinear systems approximated by linearisation is considered.

The distributed Kalman filter for the system with dependent noises is discussed in (Hashemipour et al., 1988). (Berg and Durrant-Whyte, 1992) minimise the communication by reducing the dimension of the estimated state at each local estimator and using internodal transformations; there is no communication of the state components that are not influenced by the measurement.

The fusion centre filtering equations (13), (14) can be written by the conditional densities as

$$p(\mathbf{x}_k|\mathbf{z}_k,\mathbf{Z}_{k-1}) \propto p(\mathbf{x}_k|\mathbf{Z}_{k-1})\prod_{j=1}^{N}\frac{p(\mathbf{x}_k|\mathbf{z}_k^{(j)},\mathbf{Z}_{k-1})}{p(\mathbf{x}_k|\mathbf{Z}_{k-1})}, \tag{16}$$

where the feedback is given by

$$p(\mathbf{x}_k|\mathbf{z}_k^{(j)},\mathbf{Z}_{k-1}) \leftarrow p(\mathbf{x}_k|\mathbf{Z}_k), \tag{17}$$

$j = 1,\ldots,N$, and is analogous to (15). Note that the division by the predictive density $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ can not be easily extended to general non-Gaussian densities.

3.3 Fusion by the Maximum Likelihood

This subsection discusses the fusion of dependent estimates at a fusion centre. The cornerstone idea is to treat the local estimates as if they were measurements. It arises from the identity, see (Li et al., 2003),

$$\hat{\mathbf{x}}_{k|k}^{(j)} = \mathbf{x}_k + (\hat{\mathbf{x}}_{k|k}^{(j)} - \mathbf{x}_k) = \mathbf{x}_k + (-\tilde{\mathbf{x}}_{k|k}^{(j)}) \tag{18}$$

where $\tilde{\mathbf{x}}_{k|k}^{(j)}$ is the error of the estimate at the j-th estimator. The covariance matrices of these measurements are the error covariance matrices $\mathbf{P}_{k|k}^{(jj)} = \mathbf{P}_{k|k}^{(j)}$. Assuming the local estimates are obtained by Kalman filters with Kalman gains $\mathbf{K}_k^{(j)} = \mathbf{P}_{k|k}^{(jj)}\mathbf{H}_k^{(j)\mathrm{T}}\mathbf{R}_k^{(j)^{-1}}$, the cross-covariances $\mathbf{P}_{k|k}^{(ij)} = \mathrm{E}(\tilde{\mathbf{x}}_{k|k}^{(i)}\tilde{\mathbf{x}}_{k|k}^{(j)\mathrm{T}})$ are given by

$$\begin{aligned}\mathbf{P}_{k|k}^{(ij)} &= (\mathbf{I}_{n_x} - \mathbf{K}_k^{(i)}\mathbf{H}_k^{(i)})\mathbf{P}_{k|k-1}^{(ij)}(\mathbf{I}_{n_x} - \mathbf{K}_k^{(j)}\mathbf{H}_k^{(j)})^{\mathrm{T}} + \\ &+ \mathbf{K}_k^{(i)}\mathbf{R}_k^{(ij)}\mathbf{K}_k^{(j)\mathrm{T}}, \quad (19)\end{aligned}$$

where $\mathbf{I}_{n_x}$ is the identity matrix of the size n_x, with the initial condition $\mathbf{P}_{0|-1}^{(ij)} = \mathbf{P}_0$. The predictive covariance $\mathbf{P}_{k|k-1}^{(ij)}$ is computed by (8).

Then the fusion centre measurement equation is given by

$$\mathbf{z}_k^{FC} = \mathbb{I}_N\mathbf{x}_k + \xi_k \tag{20}$$

where $\mathrm{cov}(\xi_k) = \mathbf{P}_k = [\mathbf{P}_{k|k}^{(ij)}]_{i,j=1}^N$ and

$$\mathbf{z}_k^{FC} = \begin{bmatrix}\hat{\mathbf{x}}_{k|k}^{(1)}\\ \vdots \\ \hat{\mathbf{x}}_{k|k}^{(N)}\end{bmatrix}, \ \mathbb{I}_N = \begin{bmatrix}\mathbf{I}_{n_x}\\ \vdots \\ \mathbf{I}_{n_x}\end{bmatrix}, \ \xi_k = \begin{bmatrix}-\tilde{\mathbf{x}}_{k|k}^{(1)}\\ \vdots \\ -\tilde{\mathbf{x}}_{k|k}^{(N)}\end{bmatrix}. \tag{21}$$

Unfortunately, the process $\{\xi_k\}$ is correlated with $\mathbf{x}_k$ and it is coloured, so it is not possible to use a Kalman filter in the fusion centre. But the central estimate can be obtained, see (Chang et al., 1997), by the maximum likelihood method

$$\hat{\mathbf{x}}_{k|k} = (\mathbb{I}_N^{\mathrm{T}}\mathbf{P}_k^{-1}\mathbb{I}_N)^{-1}\mathbb{I}_N^{\mathrm{T}}\mathbf{P}_k^{-1}\mathbf{z}_k^{FC}, \tag{22}$$

$$\mathbf{P}_{k|k} = (\mathbb{I}_N^{\mathrm{T}}\mathbf{P}_k^{-1}\mathbb{I}_N)^{-1}. \tag{23}$$

Note that the above fusion requires to send the Kalman filter gains $\mathbf{K}_k^{(j)}$, $j = 1,\ldots,N$ to the fusion centre to compute the cross-correlations of the estimates (19). The measurement matrices $\mathbf{H}_k^{(j)}$ must be known at or sent to the fusion centre also.

4 DECENTRALISED FUSION

In the decentralised fusion, information is processed localy. The channel filters enable to obtain a globally optimal solution in a tree network and they are described in subsection 4.1. The information measure approach discussed in subsection 4.2 sacrifices the Bayessian optimality for the the possibility to be easily used in an arbitrary network.

4.1 Channel Filters

The principle of the channel filter approach, that was introduced in (Grime and Durrant-Whyte, 1994), is the same as that of the distributed Kalman filter in fact. The new information is extracted and summed up. The necessary condition is that there is one and only one way of the information propagation, i.e. the network structure is a tree. The density notation will be used to explicitly denote the set of the measurements that were exploited by each estimator.

The essential rule of the estimate fusion is

$$p(\mathbf{x}_k|\mathcal{Z}_A \cup \mathcal{Z}_B) = \frac{p(\mathbf{x}_k|\mathcal{Z}_A)p(\mathbf{x}_k|\mathcal{Z}_B)}{p(\mathbf{x}_k|\mathcal{Z}_A \cap \mathcal{Z}_B)}. \qquad (24)$$

The posterior probability density function of the state conditioned on the union of two measurement sets is equal to the product of the densities conditioned on each measurement set divided by the density conditioned on the intersection of the measurement sets.

The equation (24) is the core of the channel filters. It is assumed that all local measurement errors are independent, (12). Thus, the measurement density can be factorised, $p(\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \ldots, \mathbf{z}_k^{(N)}|\mathbf{x}_k) = \prod_{j=1}^{N} p(\mathbf{z}_k^{(j)}|\mathbf{x}_k)$.

First, all local estimators filter their predictive estimates according to (9). Then the filtering estimates are communicated to the neighbouring estimators. The fusion is given by a repeated use of the fusion rule (24) as

$$p(\mathbf{x}_k|\mathcal{Z}_k^j) = p(\mathbf{x}_k|\mathbf{z}_k^{(j)}, \mathcal{Z}_{k-1}^j) \prod_{i \in \mathcal{N}_j} \frac{p(\mathbf{x}_k|\mathbf{z}_k^{(i)}, \mathcal{Z}_{k-1}^i)}{p(\mathbf{x}_k|\mathcal{Z}_{k-1}^j \cap \mathcal{Z}_{k-1}^i)}, \qquad (25)$$

where $\mathcal{Z}_k^j = (\mathcal{Z}_{k-1}^j \cup \mathbf{z}_k^{(j)}) \bigcup_{i \in \mathcal{N}_j} (\mathcal{Z}_{k-1}^i \cup \mathbf{z}_k^{(i)})$ is the set of the measurements that were exploited by the j-th estimator at the time k after the fusion with the incoming estimates $p(\mathbf{x}_k|\mathbf{z}_k^{(i)}, \mathcal{Z}_{k-1}^i)$, $\mathcal{N}_j$ is the set of the neighbours of the j-th estimator that have sent their estimates to it, and $p(\mathbf{x}_k|\mathcal{Z}_{k-1}^j \cap \mathcal{Z}_{k-1}^i)$ is the estimate of the channel filter ij. The fusion (25) uses the fact that the measurement errors are independent and thus

$$(\mathcal{Z}_{k-1}^i \cup \mathbf{z}_k^{(i)}) \cap (\mathcal{Z}_{k-1}^j \cup \mathbf{z}_k^{(j)}) = \mathcal{Z}_{k-1}^i \cap \mathcal{Z}_{k-1}^j. \qquad (26)$$

The predictive estimates are computed according to (10) and the channel filter estimate is given by

$$p(\mathbf{x}_k|\mathcal{Z}_k^j \cap \mathcal{Z}_k^i) = \frac{p(\mathbf{x}_k|\mathbf{z}_k^{(j)}, \mathcal{Z}_{k-1}^j)p(\mathbf{x}_k|\mathbf{z}_k^{(i)}, \mathcal{Z}_{k-1}^i)}{p(\mathbf{x}_k|\mathcal{Z}_{k-1}^j \cap \mathcal{Z}_{k-1}^i)} \qquad (27)$$

where the equations (24), (26) and the relation

$$(\mathcal{Z}_{k-1}^i \cup \mathbf{z}_k^{(i)}) \cup (\mathcal{Z}_{k-1}^j \cup \mathbf{z}_k^{(j)}) = \mathcal{Z}_k^i \cap \mathcal{Z}_k^j \qquad (28)$$

were used.

The local estimates equal to centralised estimates with delayed measurements. The delays are given by the length of the path between the respective sensors decreased by one. Note that the division by the channel filter density in the equations (25) and (27) is easily tractable for Gaussian densities only.

4.2 Information Measure Approach

In general networks, the optimality cannot be reached without inadequate effort. It can be impossible to decide which measurements have been used to compute the estimates. And even if this is possible, the common information in the denominator of (24) is too complicated to find and to compute with. Multiple processing of the same measurements, with the illusion that the errors are independent, is inevitable. Therefore to not underestimate the estimate error, some bounds must be used.

The idea of the Covariance Intersection method, see (Julier, 2009) for example, arises from the geometrical interpretation of the estimates. The fused estimate $\{\hat{\mathbf{x}}, \mathbf{P}\}$ is required to be consistent, i.e. the error covariance must not be underestimated, $\mathbf{P} - \mathrm{E}[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \geq 0$, where $\mathbf{x}$ denotes the true state. Assuming the local estimates $\{\hat{\mathbf{x}}_1, \mathbf{P}_1\}$, $\{\hat{\mathbf{x}}_2, \mathbf{P}_2\}$ are consistent, the convex combination of them

$$\mathbf{P}^{-1}\hat{\mathbf{x}} = \omega\mathbf{P}_1^{-1}\hat{\mathbf{x}}_1 + (1-\omega)\mathbf{P}_2^{-1}\hat{\mathbf{x}}_2, \qquad (29)$$

$$\mathbf{P}^{-1} = \omega\mathbf{P}_1^{-1} + (1-\omega)\mathbf{P}_2^{-1}, \qquad (30)$$

where $\omega \in [0,1]$, leads to consistent estimate $\{\hat{\mathbf{x}}, \mathbf{P}\}$ for arbitrary cross-covariance $\mathbf{P}_{12} = \mathrm{E}[(\mathbf{x} - \hat{\mathbf{x}}_1)(\mathbf{x} - \hat{\mathbf{x}}_2)^{\mathrm{T}}]$, i.e. for arbitrary common information.

The weight ω can be chosen in order to minimise various criteria. The usual criterion is the determinant of the fused error covariance matrix,

$$\omega^* = \underset{\omega \in [0,1]}{\arg\min}(\det \mathbf{P}), \qquad (31)$$

but the trace $\mathrm{tr}(\mathbf{P})$ is also used. The optimal weight ω^* can be approximated by the use of fast algorithms, see (Fränken and Hüpper, 2005). Special covariance consistency methods can be found in (Uhlmann, 2003).

(Hurley, 2002) generalise the Covariance Intersection method to the combination of probability density functions. The geometrical combination

$$p_\omega(\mathbf{x}) = \frac{p_1^\omega(\mathbf{x}) p_2^{1-\omega}(\mathbf{x})}{\int_R p_1^\omega(\mathbf{x}) p_2^{1-\omega}(\mathbf{x}) \mathrm{d}\mathbf{x}} \tag{32}$$

is used and the criterion of entropy, i.e. the Shannon information, of the fused density

$$\mathcal{H}(p_\omega) = -\int_R p_\omega(\mathbf{x}) \ln p_\omega(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{33}$$

that corresponds to the determinant criterion of the fused estimate of Gaussian density, can be applied. Other proposed criterion is the Chernoff information $\mathcal{C}(p_1,p_2) = -\min_{0\le\omega\le1}\left(\ln\int_R p_1^\omega p_2^{1-\omega}(\mathbf{x})\mathrm{d}\mathbf{x}\right)$. The optimal density is equally distant from the local densities in the Kullback-Leibler divergence sense, $\mathcal{D}(p_{\omega^*} \parallel p_1) = \mathcal{D}(p_{\omega^*} \parallel p_2)$, where the Kullback-Leibler divergence is defined as $\mathcal{D}(p_1 \parallel p_2) = \int_R p_1(\mathbf{x}) \ln\left(\frac{p_1(\mathbf{x})}{p_2(\mathbf{x})}\right) \mathrm{d}\mathbf{x}$. (Julier, 2006) studies the Chernoff fusion approximation for Gausian-mixture models, (Farrell and Ganesh, 2009) and (Wang and Li, 2009) consider fast convex combination methods.

5 NUMERICAL ILLUSTRATION

In this section, the fusion approaches will be illustrated by a numerical example. Let the system (3), (4) with three sensors be t-invariant and given by

$$\mathbf{F} = \mathbf{I}_2, \quad \mathbf{G} = \mathbf{I}_2, \quad \mathbf{Q} = \begin{bmatrix} 1.44 & -1.2 \\ -1.2 & 1 \end{bmatrix}, \tag{34}$$

$$\begin{cases} \mathbf{H}^{(1)} = \begin{bmatrix} 1 & 0 \end{bmatrix}, & \mathbf{R}^{(11)} = 1, \\ \mathbf{H}^{(2)} = \begin{bmatrix} 1 & -1 \end{bmatrix}, & \mathbf{R}^{(22)} = 2, \\ \mathbf{H}^{(3)} = \begin{bmatrix} 0 & 1 \end{bmatrix}, & \mathbf{R}^{(33)} = 1, \end{cases} \tag{35}$$

where the measurement errors are independent, $\mathbf{R}^{(12)} = \mathbf{R}^{(13)} = \mathbf{R}^{(23)} = 0$, and the initial condition is given by $p(\mathbf{x}_0) = \mathcal{N}([0,0]^\mathrm{T}, \mathbf{I}_2)$.

The used hierarchical and decentralised networks are shown on the Fig. 1, the numbers denote the respective estimators. The data links $1 \leftrightarrow 2$ and $2 \leftrightarrow 3$ are considered in the decentralised network.

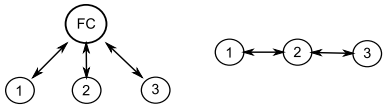


Figure 1: Hierarchical (left) and decentralised network (right), FC = fusion centre.

The centralised fusion (11), will be compared with the maximum likelihood (21), (22), (23), distributed Kalman filter (13)-(15), channel filters (25), (27) and the information measure approaches (29)-(31). The 1-σ bounds, i.e. the multidimensional parallels of the standard deviation, will show the uncertainty of the fused estimates. The bounds will be centred to zero to allow a better graphical comparison and are given by $\{\mathbf{x} : \mathbf{x}^\mathrm{T}\mathbf{P}^{-1}\mathbf{x} = 1\}$, where $\mathbf{P}$ is the estimate covariance and the $\mathbf{x} = [x_1, x_2]^\mathrm{T}$.

All estimators, including the fusion centre of the distributed Kalman filter and the channel filters, have the same initial condition $p(\mathbf{x}_0)$. The system is simulated and the 1-σ bounds at the times $k = 1$, $k = 5$, and $k = 20$ are shown in the Fig. 2 for the hierarchical and decentralised estimators.

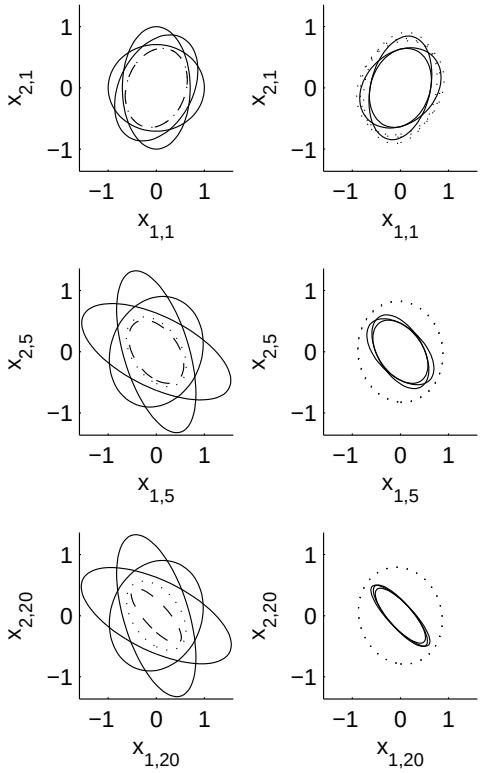


Figure 2: A comparison of the 1-σ bounds of the hierarchical and decentralised estimates at times $k = 1, 5, 20$.

The left half of the Fig. 2 shows the optimal centralised estimator (dashed line), the distributed Kalman filter (with the same estimate - dashed line), local Kalman filters (solid lines), and the fusion by the maximum likelihood at the fusion centre (dotted line). At the time $k = 1$ (top), the maximum likelihood estimate and the centralised estimate have equal covariances, the lines seem to be dash-dotted. At the times $k = 5$ and $k = 20$ (middle and bottom), the influence of not incorporating the prior information is evident, the covariance of the maximum likelihood estimate is greater than that of the centralised estimate. The local filters are the least accurate.

The right half of the Fig. 2 shows the local estimates with the channel filter fusion (solid lines) and the Covariance Intersection fusion (dotted lines). The estimate of the estimator 2 with the channel filter fusion is equal to the centralised estimate in this case. The one-step delay of the measurement exploitation in the estimators 1 (which measures x_1) and 3 (which measures x_2) is visible, there is greater uncertainty in the x_2 and x_1 axis, respectively. The the local estimates which use the Covariance Intersection get close to each other after a few steps. In this example, the estimates 2 and 1 are fused first and the result is fused with the estimate 3. The estimates overestimate the error covariance, but at least they are not worse than the estimates that use local measurements only without any fusion (compare with the solid lines on the left half of the figure). The information measure approaches are useful for more complex networks.

6 SUMMARY

Main approaches to the state estimate fusion for the linear stochastic systems were introduced. The principles and algorithms of hierarchical and decentralised fusion were presented and discussed. Contrary to the standard estimation problem, which is based on using all measurements simultaneously, the estimate fusion allows to respect an alternative technical specification concerning the measurement location and to prefer local information processing. The hierarchical fusion is more suitable for systems with a small number of sensors. In the case of general network with many sensors, the decentralised fusion based on information measures should be preferred due to its simplicity and modest assumptions.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic, project no. 1M0572, and by the Czech Science Foundation, project no. 102/08/0442.

REFERENCES

Bar-Shalom, Y. (2002). Update with out-of-sequence measurements in tracking: Exact solution. *IEEE Transactions on AES*, 38(3):760–778.

Berg, T. M. and Durrant-Whyte, H. F. (1992). Distributed and decentralized estimation. In *SICICI '92 proceedings*, volume 2, pages 1118–1123.

Challa, S., Evans, R. J., and Wang, X. (2003). A bayesian solution and its approximations to out-of-sequence measurement problems. *Information Fusion*, 4(3):185–199.

Chang, K.-C., Sahat, R. K., and Bar-Shalom, Y. (1997). On optimal track-to-track fusion. *IEEE Transactions on AES*, 33(4):1271–1276.

Chong, C.-Y., Mori, S., Chang, K.-C., and H., B. W. (1999). Architectures and algorithms for track association and fusion. In *Proceedings of the 2nd International Conf. on Information Fusion*.

Farrell, W. J. and Ganesh, C. (2009). Generalized chernoff fusion approximation for practical distributed data fusion. In *Proceedings of the 12th International Conf. on Information Fusion*.

Fränken, D. and Hüpper, A. (2005). Improved fast covariance intersection for distributed data fusion. In *Proc. of the 8th Int. Conf. on Inf. Fusion*.

Grime, S. and Durrant-Whyte, H. F. (1994). Data fusion in decentralized sensor networks. *Control Engineering Practice*, 3(5):849–863.

Hashemipour, H. R., Roy, S., and Laub, A. J. (1988). Decentralized structures for parallel kalman filtering. *IEEE Transactions on AuC*, 33(1):88–94.

Hurley, M. B. (2002). An information theoretic justification for covariance intersection and its generalization. In *Proceedings of the 5th International Conference on Information Fusion*.

Julier, S. J. (2006). An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models. In *Proceedings of the 9th Int. Conf. on Information Fusion*.

Julier, S. J. (2009). Estimating and exploiting the degree of independent information in distributed data fusion. In *Proceedings of the 12th International Conference on Information Fusion*.

Li, X.-R., Zhu, Y., Wang, J., and Han, C. (2003). Optimal linear estimation fusion—part i: Unified fusion rules. *IEEE Transactions on Information Theory*, 49(9):2192–2208.

Simon, D. (2006). *Optimal State Estimation*. Wiley.

Uhlmann, J. K. (2003). Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion*, 4(3):201–215.

Wang, Y. and Li, X.-R. (2009). A fast and fault-tolerant convex combination fusion algorithm under unknown cross-correlation. In *Proceedings of the 12th Int. Conf. on Information Fusion*.

REAL-TIME CONTROL OF REWINDING MACHINE
Comparison of Two Approaches

Karel Perutka
Faculty of Applied Informatics, Tomas Bata University in Zlin, nam. T.G.M. 5555, Zlin, Czech Republic
KPerutka@fai.utb.cz

Keywords: Nonlinear control, MATLAB, Off-line identification, On-line identification, Real-time control, Self-tuning control.

Abstract: The paper deals with two simple approaches applied to the real-time control of rewinding machine and their comparison. In brief, the comparison of results obtained by nonlinear real-time control with pre-identification, and by adaptive real-time control with on-line identification was performed. The rewinding machine was controlled by PC from MATLAB's Real-Time Toolbox using technological card, terminal board and wires. Each of two used approaches has its advantages and its drawbacks, which was proven, and nonlinear control seemed to be more suitable for the rewinding machine, minimally because of the action signal history from the nonlinear control, the action is more consistent.

1 INTRODUCTION

As is stated in abstract, the comparison of two approaches of rewinding machine's real-time control was performed. Firstly, let us provide introduction to the control methods which were used.

Many processes can be marked as multivariable systems. For such processes, the centralized controller is commonly used because it provides the best closed loop performance. However, the centralized controller is less fault tolerant than the decentralized controller. This is the main reason why decentralized control strategy is often preferred. The used strategy is based on the linear model of the nonlinear plant and the design of a decentralized controller for this linear model (Li, *et al.*, 2000). Li, *et al.*, (2000) mentions that the plant decomposition is crucial for decentralized design and it is not always possible to obtain satisfactory decentralized control systems using a simple physical decomposition. However, the decentralized approach has one big disadvantage due to the decomposition, the reduction of control performance due to the restricted controller structure (Cui and Jacobsen, 2002). But decentralized control is popular in practice, see (Balachandran and Chidambaram, 1997).

Many nonlinear systems can be identified and controlled as linear systems around the steady state or working points. Nice application of feedback control was performed by Cottenceau et al. (Cottenceau et al., 2001). When nonlinear control is used, it possible to enlarge the working interval even in the case the linear control does not guarantee the sufficient quality of control. Moreover, some systems have nonlinearities, which cannot be linearly approximated, for instance friction, etc. Therefore, the necessity of nonlinear control occurs.

Nonlinear system is a set of elements of system, in which at least one of the elements is nonlinear (Modrlak, 2008).

Some nonlinear systems can be approximated by linear systems within the defined range and when specific conditions hold on. In practice, such systems can be divided into linear and nonlinear part. The dynamics of system can be approximated by linear model and its nonlinear part by the nonlinear characteristics. The superposition is not valid for the nonlinear systems, the output of Hammerstein model is different from the output of Weiner model (Lin, 1994).

This paper uses the simple nonlinear control introduced by Chen et al. (Chen et al., 2006) for nonlinear real-time control of rewinding machine. The simple nonlinear control was applied for instance by Perutka and Dostalek (2009), the application is in MATLAB because it is nice tool for Control Engineering at universities (Perutka, Hezcko, 2007).

2 THEORETICAL BACKGROUND

2.1 Simple Nonlinear Controller

This method was introduced by Chen et al. (Chen, et al., 2006) and verified by Perutka and Dostalek (Perutka and Dostalek, 2009). The controller consists of three parts, "the pure controller" and generator giving together the nonlinear controller and the system model inversion (Chen, et al., 2006).

2.2 Pre-identification

Suppose the existence of continuous-time multivariable N x N system $\mathbf{S}(t)$. Moreover, let as assume the vector of reference signals $\mathbf{R}(t)$, its values are send to the input of the system $\mathbf{S}(t)$. They are same and for the same time as those one which are going to be used during the control. Each time interval of history of control of the system $\mathbf{S}(t)$, where all reference signals have the constant value, is identified separately. Every identification element is identified several times, every time with different identification algorithm, and the obtained model is verified with the measured data. The obtained model which gives the best agreement with the measured data is used for control.

2.3 Self-tuning Control

Self-tuning controllers (STC) belong to the class of adaptive control systems. Self-tuning controllers are based on on-line identification and on tuning the controller parameters with respect to identified changes in controlled systems (Bobal et al., 2005).

2.4 On-line Identification

The action (input) signal $u(t)$ is continuously approximated by Lagrange regression polynomial at the interval of given length during entire control. After the polynomial approximation, the approximating polynomial derivation $u^{(i)}_L(t)$ is counted. It is sampled in purpose to count the values of subsystem parameters using recursive identification algorithm.

2.4.1 Recursive Least-squares and Recursive Instrumental Variable

Least squares method is generally known, for instance presented by Bobal et al. (Bobal et al., 2005). Instrumental variable method is a modification of the least squares method. It does not allow us to obtain the properties of noise, but it has inferior presumptions than the least square method (Zhu & Backx, 1993).

2.5 Suboptimal Linear Quadratic Tracking Controller

Usage of adequate method of controller parameters computation is crucial for control. Linear quadratic control is a reliable method verified by many publications, for instance by Casavola et al. (Casavola et al., 1991), the used suboptimal method was introduced by Dostal (Dostal, 1997).

3 SHORT DESCRIPTION OF USED APPROACHES

The overall controlled system was controlled in the view of decentralized control. Nice paper useful to decentralized control was written by Seatzu and Usai (Seatzu and Usai, 2002).

3.1 Approach 1

This approach is a combination of simple nonlinear control (chap. 2.1) and pre-identification (chap. 2.2). Firstly, the pre-identification run and it provided the initial parameters estimates for the model used during nonlinear control.

3.2 Approach 2

Approach 2 is de facto self-tuning control in real-time. The controller parameters were counted according to the suboptimal linear quadratic method (chap. 2.5), identification was realized using least squares and instrumental variable (chap. 2.2.1).

4 MACHINE DESCRIPTION

The real-time control was realized on CE108 Coupled Drives Apparatus, see figure 1, which is manufactured by TecQuipment Ltd., United Kingdom. The rewinding machine is adapted for its usage in the laboratory. The properties of the apparatus had been studied in detail and its model in MATLAB – SIMULINK environment was created (Perutka, Dolezel 2009). The speed and tension of thread during spooling is an example of rewinding process. This situation is modified for laboratory

experiments where the flexible belt is fastened on three wheels. Speed of two wheels is directly proportional to the number of revolutions of the servo-motors. Third wheel may move, because it is fixed on the moving jib which is hung on the spring. The measurement of speed and tension is indirect via the angle of the moving jib, from -10 deg to 10 deg, which correspond the voltage from -10 V to 10 V. The control voltages of the amplifiers of the servo-motors, which are bi-directional, are the inputs. The outputs are four, the voltage of the speed of two servo-motors, or two wheels respectively, and the voltage of the tension and the speed of the belt, or angular deflection and speed of 3rd wheel respectively. The apparatus is connected to PC via technological card Advantech. The control is realized in MATLAB using Real Time Toolbox.

Figure 1: CE108 Rewinding Machine.

5 REAL-TIME TOOLBOX DESCRIPTION

Real Time Toolbox is used for real-time control and it is based on a high performance real-time kernel and drivers for popular A/D and D/A boards, the toolbox includes drivers for more than 300 industry-standard data acquisition boars. The real-time kernel allows us to use sampling frequencies up to 66 kHz with no external clock source required. Besides standard analog and digital I/O many specialized devices are also supported. Multiple boards of the same or different type can be used simultaneously to offer sufficient I/O even for complex industrial applications (Real-time Toolbox: Introduction, 2010).

6 EXPERIMENTAL PART

In figures 2-5 there are obtained results of real-time control of rewinding machine. In these figures, the meaning of the symbols is following: w_1 – set-point of first subsystem, u_1 – action signal of first subsystem, y_1 – output signal of first subsystem, w_1 – set-point of second subsystem, u_1 – action signal of second subsystem, y_1 – output signal of second subsystem.

Figures 2 and 3 provide the results obtained by adaptive real-time control. It was the self-tuning control with online identification using least squares (figure 2) and instrumental variable method (figure 3). The suboptimal linear quadratic tracking was used as the method of controller parameters tuning.

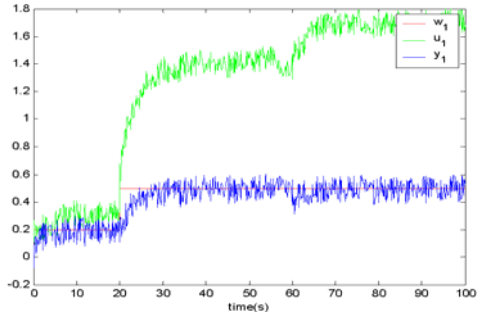


Figure 2: Adaptive real-time control – 1st subsystem.

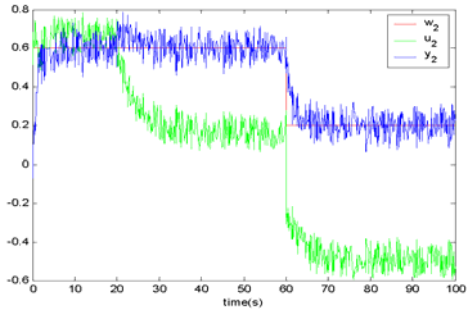


Figure 3: Adaptive real-time control – 2nd subsystem.

Figures 4 and 5 provide the results obtained by nonlinear real-time control, the combination of simple nonlinear controller (Chen et al., 2006) with pre-identification. The pre-identification provided the initial estimates of the used model's parameters.

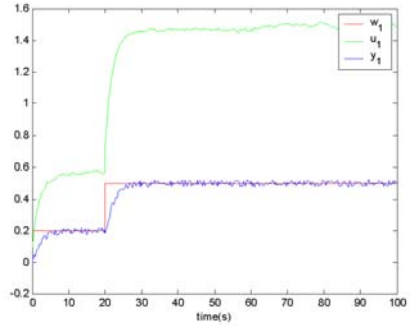


Figure 4: Nonlinear real-time control – 1st subsystem.

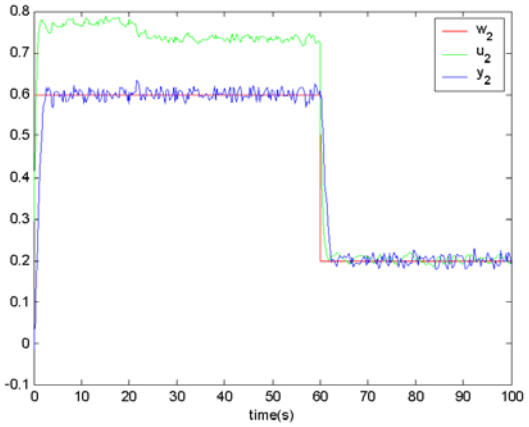


Figure 5: Nonlinear real-time control – 2nd subsystem.

Both used methods of real-time control provided the satisfactory results and they can be used for this machine, but there are some differences which should be mentioned. Nonlinear real-time control is less biased and seemed to be more suitable. The usage of pre-identification decreased the unwanted overshooting caused by interactions. Moreover, the adaptive real-time control is notably more sensitive to the changes of model parameters, whilst the used nonlinear real-time control does not need the change of model parameters.

7 CONCLUSIONS

The paper presented results of real-time control of rewinding machine by two approaches together with the necessary theoretical background. The nonlinear real-time control seems to be more suitable, but adaptive real-time control is also possible to use, because it is more sensitive on the changes during control.

ACKNOWLEDGEMENTS

The author would like to mention MSM7088352102 grant, from which the work was supported.

REFERENCES

Balachandran, R., Chidambaram, M., 1997. Decentralized control of crude unit distillation towers. In *Computers and Chemical Engineering*, 21, pp. 783-786.

Bobal, V., Böhm, J., Fessl, J., Machacek, J., 2005. *Digital Self-tuning Controllers*. Springer-Verlag London Limited.

Casavola, A., Grimble, M. J., Mosca, E., Nistri, P., 1991. Continuous-time LQ regulator design by polynomial equations. In *Automatica*, 27, pp. 555-558.

Chen, Ch.-T.; Chuang, Y.-Ch., Hwang, Ch., 2006. A Simple Nonlinear Control Strategy for Chemical Processes. In *Proc. of the 6th Asian Control Conference*, Inna Grand Bali Beach Hotel, Bali, Indonesia, ISBN 979-15017-0.

Cottenceau, B., Hardouin, L., Boimond, J.-L., Ferrier, J.-L., 2001. Model reference control for time event graphs in diods. In *Automatica*, 37, pp. 1451-1458.

Cui, H., Jacobsen, E.W., 2002. Performance limitations in decentralized control. In *Journal of Process Control*, 12, pp. 485-494.

Dostal, P., 1997. An approach to control of processes of chemical technology. Inaugural dissertation. TU Brno, Brno.

Li, H., Lee, P. L., Bahri, P., Cameron, I.T., 2000. Decentralized control design for nonlinear plants: ν–metric approach. In *Computers and Chemical Engineering*, 24, pp. 273-278.

Lin, C.-F., 1994. *Advanced Control System Design*. New Jersey, USA: Prentice Hall, 1994. ISBN 0-13-006305-3.

Modrlak, O.: Theory of automatic control II: Nonlinear systems. Lectures notes. [on line] *Technical University of Liberec*, [cit. 02-02-2010], available at http://www.fm.vslib.cz/~krtsub/fm/modrlak/pdf/tar2_nel.pdf.

Perutka, K., Dolezel, K., 2009. Simulation Model of CE108 Coupled Drives Apparatus. In *MATHMOD 2009, 6th Vienna Conference on Mathematical Modelling, Vienna, Austria.* ARGESIM.

Perutka, K., Dostalek, P., 2009. Simple Decentralized Autonomous Adaptive Nonlinear Real-time Controller with Controller Source Code Optimization: Case Study. In *ISADS 2009, 9th IEEE International Symposium on Autonomous Decentralized Systems, Athens, Greece.* IEEE.

Perutka, K., Heczko, K., 2007. Teaching of MATLAB Programming Using Complex Game. In *FIE2007, 37th IEEE/ASEE Frontiers in Education Conference, Milwaukee, WI, USA.* IEEE.

Real-time Toolbox: Introduction. [on line] *Humusoft*, [cit. 03-30-2010], available at http://www.humusoft.cz/produkty/rtt/.

Seatzu, C., Usai, G., 2002. A decentralized volume variations observer for open channels. In *Applied Mathematical Modelling*, 26, pp. 975-1001.

Zhu, Y., Backx, T., 1993. *Identification of Multivariable Industrial Processes for Simulation, Diagnosis and Control*. Springer-Verlag Ltd., London, United Kingdom

A SUB-OPTIMAL KALMAN FILTERING FOR DISCRETE-TIME LTI SYSTEMS WITH LOSS OF DATA

Naeem Khan, Sajjad Fekri and Dawei Gu
Department of Engineering, University of Leicester, Leicester LE1 7RH, U.K.
{*nk118, sf111, dag*}*@le.ac.uk*

Keywords: Linear prediction coefficients, Open loop estimation, Autocorrelation, Optimization.

Abstract: In this paper a sub-optimal Kalman filter estimator is designed for the plants subject to loss of data or insufficient observation. The methodology utilized is based on the closed-loop compensation algorithm which is computed through the so-called Modified Linear Prediction Coefficient (MLPC) observation scheme. The proposed approach is aimed at the artificial observation vector which in fact corrects the prediction cycle when loss of data occurs. A non-trivial mass-spring-dashpot case study is also selected to demonstrate some of the key issues that arise when using the proposed sub-optimal filtering algorithm under missing data.

1 INTRODUCTION

Loss of observation is a non-trivial case of study in both control and communication systems. Such loss may be due to the faulty sensors, limited bandwidth of communication channels, confined memory space, and mismatching of measurement instruments to name but a few. Overcoming the side effects arose from missing data in control and communication systems are remained as open research problems for researchers during the last decade (Allison, 2001).

Perhaps, the best known tool for the linear estimation problem is Kalman filtering (Khan and Gu, 2009b). However, Kalman filter depends heavily on the plant dynamics, information of unmeasured stochastic inputs, and measured data and hence it is prone to fail if e.g., data is unavailable for measurement update step. To overcome such shortcomings, one approach for state estimation is to utilise the so-called Open-Loop Estimation (OLE) when observations are subjected to random loss, see e.g. (Schenato, 2005; Liu and Goldsmith, 2004; Sinopoli and Schenato, 2007; Schenato et al., 2007). They have studied LOOB cases, while running the Kalman filter in a open loop fashion, i.e. whenever observation is lost, the predicted quantities are processed for next iteration, without any update.

More specifically, in OLE the prediction is based on the system model and processed as state estimation without being updated due to the unavailability of the observed data. Nonetheless, in practice this approach may diverge at the presence of longer loss duration and it is likely that error covariance could exceed the limits if the upper and lower bounds of error covariance are provided (Huang and Dey, 2007). Another shortcoming of the OLE is the sharp spike phenomena when the observation is resumed after the loss. This is because the Kalman filter gain is set to zero at the OLE during the loss time. But when observation is resumed, Kalman gain first surges to the very high gains and then tries to approach the steady state values in order to compensate loss impact. This consequently results in a sudden peak to reach to the normal trajectory of the estimated state which is not a desirable behaviour for a reliable estimation algorithm. Detail stability analysis of OLE can be found in (Li Xie, 2007).

Under loss of observations for a longer period of time, there is a requirement for an advanced estimation technique which could provide superior estimation performance under loss of data so as to maintain the error covariance bounded. Our proposed approach in this paper is based on an artificial optimal observation vector which is computed based on the minimum error generated through the so-called Modified Linear Prediction Coefficient (MLPC). Another advantage of the proposed method is that it eliminates the spike of the OLE technique.

(Micheli, 2001) has considered a delay in the data arrival which may also be translated as lost or inaccurate measured data. In (Schenato, 2005), a system is assumed to be subjected to both LOOB and delay of observation at the same time. All the above works have suggested switching to an OLE estimator when

there is LOOB and a closed loop estimator when the observation arrives at destination. This will aim in fact at designing an estimator which is strongly time-varying and stochastic in nature. In order to avoid random sampling and stochastic behaviour of the designed Kalman filter, (Khan and Gu, 2009b) has proposed a few approaches to compensate the loss of observations in the state estimation through Linear Prediction.

Throughout this paper we shall call the variables in the case of loss of data as "compensated variables", e.g. $P_k^{\{2\}}$ is called the compensated filtered error covariance at time step k with loss of observation. The rest of the paper is organized as follows. The theory of the Linear Prediction Coefficient (LPC) is overviewed in Section II. In Section III we discuss the proposed sub-optimal Kalman filter with loss of data. The mass-spring-dashpot case study is given in Section IV. Simulation results are presented in Section V. Section VI summarizes our conclusions.

2 THEORY OF LINEAR PREDICTION COEFFICIENT

Linear prediction (LP) is an integral part of signal reconstruction e.g. speech recognition. The fundamental idea behind this technique is that the signal can be approximated as a linear combination of past samples, see e.g. (Rabiner and Juang, 1993). Whenever there is the loss of observation, a signal window is selected to approximate the lost-data. The weights assigned to this data are computed by minimizing the mean square error. These weights are termed as Linear Prediction Coefficients. Out of the two leading LPC techniques, (namely Internal and External LPC), we shall develop and employ External LPC for LOOB, which suits to our problem with constraints:

- The signal statistical properties are assumed to vary slowly with time.
- Loss window should not be "sufficiently long", otherwise the prediction performance will be inferior.

In this paper, the LP technique is termed as modified because in conventional LPC there is no defined strategy to account the number of previous data, while have defined several simple-to-implement algorithms to decide that factor. One of it would be explain the subsequent section.

Let us assume that the dynamics of the LTI system is given in discrete time and that the data or observation is lost at time instant k. The LP is performed as:

$$\bar{z}_k = \sum_{i=1}^{n} \alpha_i z_{k-i} \tag{1}$$

where $\bar{z}_k$ is called "compensated observation" and $'\alpha'$'s represent weights of linear prediction coefficients for the previous observations and n denotes the order of the LPC filter. Generally speaking, it depicts the maximum number of previous observations considered for computation of compensated observation vector. Also, n is required to be chosen appropriately - higher value of n does not guaranty an accurate approximation of the signal but rather an optimal value of n decides an efficient approximation and hence prediction, see (Rabiner and Juang, 1993).

3 DESIGN OF SUB-OPTIMAL KF WITH LOSS OF DATA

Let us assume that the process under consideration is to be run by random noise signal whose mean and covariance are independent of time, i.e. wide-sense stationary process, given as

$$x_k = Ax_{k-1} + Bu_{k-1} + L_d\xi_k \tag{2}$$
$$z_k = Cx_k + v_k \tag{3}$$

where A, B and C have appropriate dimensions, and x, u, z, ξ and v are state, input, sensed output, plant disturbance and measurement noise, respectively. The plant noise ξ and sensor noise v are assumed to be zero mean white gaussian noises.

CKF computes the priori state estimation which is solely based on (2). This priori estimation is thereby updated with newly resumed observation at each time instant. In the subsequent section, the performance of CKF is tested and verified in a mass-spring-dashpot system which help illustrate the proposed algorithm. If the observation is not available due to any of the reason mention earlier, the compensated observations are calculated through (1).

The posteriori state estimation using this compensated observation will be

$$\bar{x}_{k|k} = x_{k|k-1} + \bar{K}_k(\bar{z}_k - \hat{z}_k) \tag{4}$$

The corresponding a posterior error for this estimate is

$$\begin{aligned} e_{k|k} &= x_k - \bar{x}_{k|k} = x_k - x_{k|k-1} - \bar{K}_k(\bar{z}_k - \hat{z}_k) \\ &= e_{k|k-1} - \bar{K}_k(\bar{z}_k - \hat{z}_k) \end{aligned} \tag{5}$$

where x_k is the actual state of the system. Conservatively, the cost function of the Kalman filter is obtained based on this a posterior error of the state estimation.

The optimal values of the modified linear prediction coefficients (MLPC) are computed based the residual vector as follows.

$$e_z = \bar{z_k} - \hat{z}_k \tag{6}$$

For the compensated estimation algorithm in MLPC, our goal is to minimize the following cost function:

$$\begin{aligned} J_k &= E\left[e_z^T e_z\right] \\ &= E\left[(\bar{z_k} - \hat{z}_k)^T (\bar{z_k} - \hat{z}_k)\right] \end{aligned} \tag{7}$$

The MLPC are computed provided with the minimum cost function i.e.

$$\frac{\partial J_k}{\partial \alpha_j} = 0 = \frac{\partial J_k}{\partial \bar{z_k}} \cdot \frac{\partial \bar{z_k}}{\partial \alpha_j} \tag{8}$$

Performing simple and straight forward algebra the above equation can be simplified as

$$\begin{aligned} E[\hat{z}_k z_{k-i}] - \sum_{j=1}^{n} \alpha_i E\{z_{k+j} z_{k-i}\} &= 0 \\ \sum_{j=1}^{n} \alpha_i E\{z_{k+j} z_{k-i}\} &= E[\hat{z}_k z_{k-i}] \end{aligned} \tag{9}$$

$$\sum_{j=1}^{n} \alpha_i \gamma_k[i,j] = r_k(i) \tag{10}$$

or

$$\begin{aligned} R_k \cdot A_{\alpha.k} &= r_k \\ A_{\alpha.k} &= r_k \cdot R_k^{-1} \end{aligned} \tag{11}$$

where

$$R_k = \begin{bmatrix} \gamma_k(0,0) & \gamma_k(0,1) & \cdots & \gamma_k(0,n-1) \\ \gamma_k(1,0) & \gamma_k(1,1) & \cdots & \gamma_k(1,n-1) \\ \gamma_k(2,0) & \gamma_k(2,1) & \cdots & \gamma_k(2,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_k(n-1,0) & \gamma_k(n-1,1) & \cdots & \gamma_k(n-1,n-1) \end{bmatrix} \tag{12}$$

$$A_{\alpha.k} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{13}$$

and

$$r_k = \begin{bmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(n) \end{bmatrix} \tag{14}$$

where $E(z_{k-i} z_{k-j}) = \gamma_k(i,j)$ and $E(z_k z_{k-j}) = \gamma_k(j)$ is the autocorrelation function, which will be explain shortly. Equation (11) requires inverting the matrix of R_k which may be increasingly difficult due to computational demanding, especially at large orders. To get rid of such burdensome calculations, several attempts have been introduced in the literature. Through Levinson Durbon or Leroux-Gueguen algorithm the so-called "Reflection Coefficients (RCs)" are computed, which represent one-to-one linear prediction coefficients. We shall explore and focus how to calculate the optimal values of α_i and n, when the measurement contains a solid deterministic input along with the unmeasured stochastic inputs. In practice, computing the autocorrelation coefficients need extra attention. Generally, the autocorrelation coefficients are represented as

$$\gamma_m = \frac{C_m}{C_0} \tag{15}$$

where C_m is the auto-covariance of y at lag m which is

$$C_m = \frac{1}{n-m} \sum_{j=1}^{n-m} (z_j - \bar{z})(z_{m+j} - \bar{z}) \tag{16}$$

where $\bar{z} = \frac{1}{n} \sum_{j=1}^{n} z_j$ i.e. mean of the data for the selected window. Without loss of generality, we shall assume that that $E(z_k) = CE(x_k) = D_k$.
A straightforward calculation would lead to the result

$$C_m = \frac{1}{n-m} \sum_{j=1}^{n-m} (D_j D_{m+j}) + \bar{D}^2 - \bar{D}\bar{D}_m - \bar{D}\bar{D}_M \tag{17}$$

and

$$C_0 = \frac{1}{n} \sum_{j=1}^{n} (D_j^2) - \bar{D}^2 + \frac{1}{n} \sum_{j=1}^{n} (v_j)^2 \tag{18}$$

where $m \leq \frac{n}{2}$ and

$$D_j = E(z_j) = CE(x_j) \qquad \bar{D}_M = \frac{1}{n-m} \sum_{j=m+1}^{n} D_j$$

$$\bar{D}_m = \frac{1}{n-m} \sum_{j=1}^{n-m} D_j \qquad \bar{D} = \frac{1}{n} \sum_{j=1}^{n} D_j$$

Clearly, one can observe that $\gamma_0 = \frac{C_0}{C_0} = 1$. And $\gamma_1 = \frac{C_1}{C_0} < 1$. Therefore, we can write

$$\gamma_0 \geq \gamma_1 \geq \gamma_2 \geq \cdots \gamma_m \tag{19}$$

The inequality of (19) is an important equation which helps in deciding the order of the LP filter as shown in Algorithm-2. For better understanding of the descriptive design, the measurement vector is written as

$$z_k = \eta_k(Cx_k) + v_k \tag{20}$$

where η_k is a random variable, such that

$$\eta_k = \begin{cases} 1, & \text{if there is no LOOB at time step k} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Therefore, the prediction step for the normal operation is as follows

$$\begin{aligned} x_{k|k-1} &= Ax_{k-1|k-1} + Bu_{k-1} \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + L_d Q_{k-1} L_d^T \end{aligned} \quad (22)$$

The above predicted state and predicted state covariance are achievable and remain unaffected with loss of data. The conventional Kalman filter will update the state and covariance on the arrival of observation vector.This updated state and updated state covariance are valid when there is no loss of data, i.e. system is running in the normal operation. However, in the presence of loss of measured data, the above standard technique is failed. Toward this end, we have proposed the closed-loop base MLPC algorithm, which can also tackle the issues arising from data loss for long period of time.

The Open loop estimator propagates the predicted state and covariance without any update due to the unavailability of the measurements as

$$\begin{aligned} x_k^{\{2\}} &= x_k^{\{1\}} = Ax_{k-1}^{\{2\}} + Bu_{k-1} \\ P_k^{\{2\}} &= P_k^{\{1\}} = AP_{k-1}^{\{2\}}A^T + L_d Q_{k-1} L_d^T \end{aligned} \quad (23)$$

While, in the proposed MLCP, the compensated observations are computed through 1 the modified linear prediction scheme providing minimum error production. The estimation produced by compensated observation is very comprehensive than those of open-loop algorithms discussed earlier. The compensated observation are used to calculate compensated innovation vector. Thereafter, the compensated Kalman gain is computed as follows.

$$\bar{K_k} = \bar{P_k^{\{1\}}} C^T (C\bar{\bar{P}}_k^{\{1\}} C^T + \bar{R}_k)^{-1} \quad (24)$$

Hence, the predicted state and covariance are updated using this gain as

$$\begin{aligned} x_k^{\{2\}} &= x_k^{\{1\}} + \bar{K_k}(\bar{z_k} - Cx_k^{\{1\}}) \\ P_k^{\{2\}} &= P_k^{\{1\}} - \bar{P_k^{\{1\}}} C^T (C\bar{\bar{P}}_k^{\{1\}} C^T + \bar{R}_k)^{-1} C\bar{P_k^{\{1\}}} \end{aligned} \quad (25)$$

The closed loop Kalman filtering algorithm is summarized in Algorithm 1. There are various ways to choose the value of the order of LP filter, n. Alternatively among these methods, we have found Algorithm 2 very practical to be implemented in a number of applications.

Algorithm 1: The proposed closed-loop estimation algorithm using MLPC.

1: At time step: $k-1$, **Prediction** is carried out as $x_k^{\{1\}} = Ax_{k-1}^{\{2\}} + Bu_{k-1}$, and $P_k^{\{1\}} = AP_{k-1}^{\{2\}}A^T + L_d Q_k L_d^T$
2: **Check:** Status of η_k
   if $\eta_k = 1$
   Run normal Kalman filter (obtain Filtered Response i.e. $x_{k|k}$ and $P_{k|k}$)
   Else Obtain compensated filtered response ($x_k^{\{2\}}$ and $P_k^{\{2\}}$) as mentioned below.
3: **Select** a suitable size for window (n) (No. of previous observations) and LP filter order (m) with the constraint $m < n/2$
4: **Construct** autocorrelation matrix R_k.
5: **Construct** modified residual matrix r_k.
6: **Compute** MLPC through $A_{\alpha.k} = R_k^{-1} \cdot r_k$
7: **Calculate** compensated measurement vector as $\bar{z_k} = \sum_{j=1}^{n} \alpha_j z_{k-j}$
8: **Obtain** compensated residual vector
9: **Calculate** Compensated Kalman gain $\bar{K_k}$
10: Measurement update step is carried out as: $x_k^{\{2\}} = x_k^{\{1\}} + \bar{K_k}(\bar{z_k} - Cx_k^{\{1\}})$: and $P_k^{\{2\}} = P_k^{\{1\}} - P_k^{\{1\}} C^T (CP_k^{\{1\}} C^T + \bar{R_k})^{-1} CP_k^{\{1\}}$:
11: **Return** to Step 1, i.e. repeat prediction cycle;

Algorithm 2: Selection of LP filter order.

1: Select γ_{th}.
2: **Compute** $\gamma_i = \frac{C_i}{C_0} \qquad i = 1, 2, \cdots m$
3: **Check**: Is $\gamma_i < \gamma_{th}$,
   Yes Stop further computation of γ_i
   $m \leftarrow i$ and select order of LP filter as $n = 2m+1$.
   Else
4: **Compute** $i \longleftarrow i+1$
5: **Repeat** Step 2

4 THE CASE STUDY EXAMPLE

The system under study in this paper is a slightly modified version of a mass-spring-dashpot (MSD) system given in (Fekri et al., 2007) as shown in Fig 1 which is a continues time system with dynmics as follows.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + L\xi(t) \\ y(t) &= Cx(t) + \theta(t) \end{aligned} \quad (26)$$

where the state vector is defined as

$$x^T(t) = [\ x_1(t) \quad x_2(t) \quad \dot{x}_1(t) \quad \dot{x}_2(t)\ ] \quad (27)$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_1}{m_1} & \frac{k_1}{m_1} & -\frac{b1}{m1} & \frac{b1}{m1} \\ \frac{k_1}{m_2} & -\frac{k_1+k_2}{m_1} & \frac{b_1}{m_2} & -\frac{b_1+b_2}{m_2} \end{bmatrix} \quad (28)$$

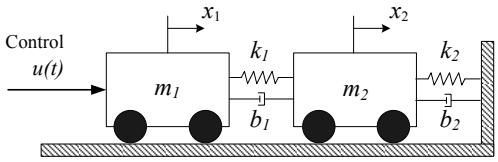


Figure 1: MSD two cart system.

$$B^T = [\ 0 \quad 0 \quad \tfrac{1}{m_1} \quad 0\ ] \tag{29}$$

$$C = [\ 0 \quad 1 \quad 0 \quad 0\ ] \tag{30}$$

$$L = [\ 0 \quad 0 \quad 0 \quad 3\ ] \tag{31}$$

The known parameters are $m_1 = m_2 = 1$, $k_1 = 1$, $k_2 = 0.15$ and $b_1 = b_2 = 0.1$ and the sampling time is $T_s = 1msec$. Plant disturbance and sensor noise dynamics are characterized as

$$E\{\xi(t)\} = 0,\ E\{\xi(t)\xi(\tau)\} = \Xi\delta(t-\tau),\ \Xi = 1 \tag{32}$$

$$E\{\theta(t)\} = 0, \quad E\{\theta(t)\theta(\tau)\} = 10^{-6}\delta(t-\tau) \tag{33}$$

After substituting the above known values the matrices will be as follows:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -0.1 & 0.1 \\ 0.1 & -1.15 & 0.1 & -0.2 \end{bmatrix} \tag{34}$$

and

$$B^T = [\ 0 \quad 0 \quad 1 \quad 0\ ] \tag{35}$$

In subsequent section, we will apply the proposed MLPC algorithm to the above MSD system and show some of the representative results. Many others were also done but are not shown in this paper due to lack of space.

5 SIMULATION RESULTS

Here we implement the above closed-loop MLPC algorithm to the MSD system as discussed in Section IV. For the purpose of our study, the continuous-time dynamics of the MSD system is transformed to an appropriate discrete-time model. Results depict the performance of the Kalman filter when it is running under the open loop i.e. during the period of unavailability of observation, the prediction is not updated and the predicted state and covariance are propagated for the next time instant, see also (Khan and Gu, 2009a). Figure 3 shows the performance of conventional kalman filter via plotting the measured signal; x_2, the position of Mass 2 with no loss of data.

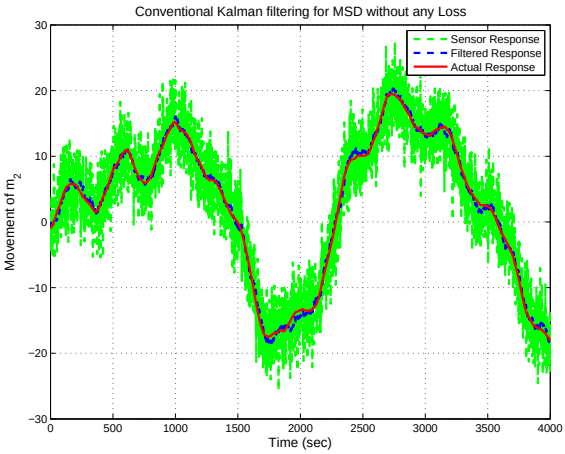


Figure 2: Performance of CKF without data loss.

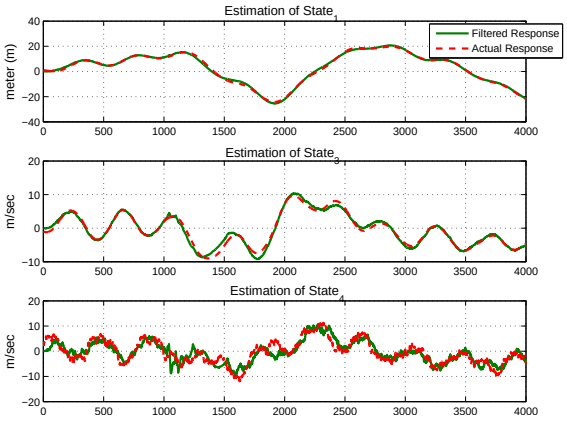


Figure 3: Other plant states.

Figure 4 shows three other states of the MSD plant which again depicts the performance of conventional Kalman filter when it is running normally, i.e. when there is no data loss, for the rest three states (x_1, $x_3 = v_1$ and $x_4 = v_2$), the position of Mass 1, the velocity of Mass 1 and velocity of Mass 1, respectively. Figure 5 shows the comparison analysis of the existing open loop Kalman filtering and the proposed closed loop MLPC algorithm based on compensated observation Kalman filtering. The sensor failure, namely the loss of data, is introduced at 10–15 Secs. Figure 5 shows that the Open-Loop based estimation algorithm diverges shortly and the estimation performance is extremely inferior while the compensated closed-loop observations generate satisfactory results and better estimations. Figure 6 represents the Open-Loop Kalman filtering along with measurements and true sketch. During the loss, the observation value is zero, and the predicted state which is taken as measurement updated state is not following the true state trajectory properly. Also, Figure 7 shows state esti-

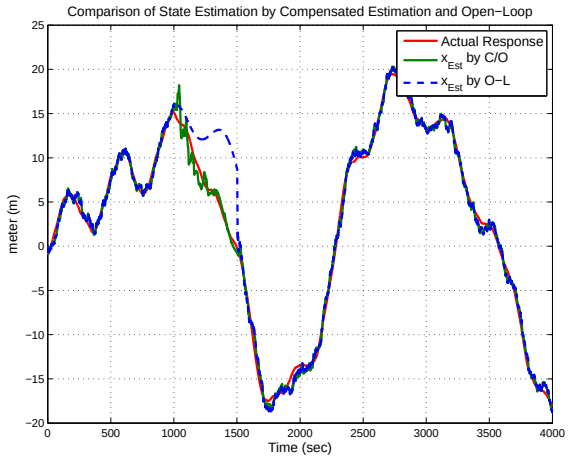


Figure 4: Comparison to two Estimation method.

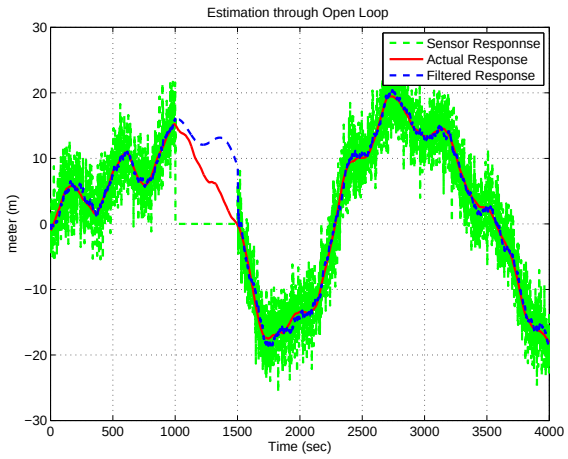


Figure 5: State Estimation through Open-Loop.

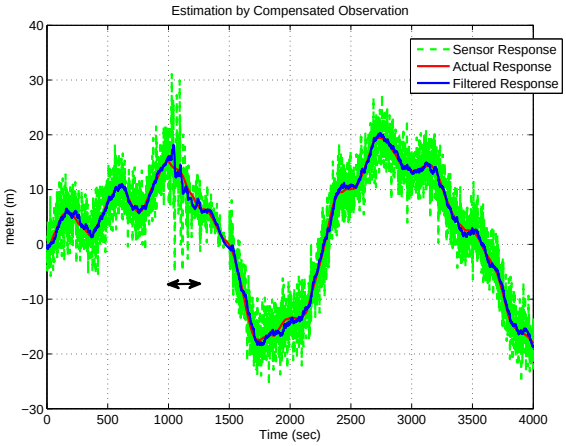


Figure 6: State Estimation through Closed-Loop.

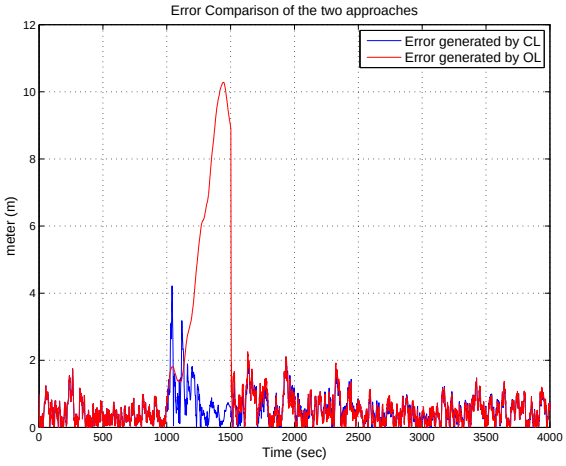


Figure 7: Error Comparison.

mation through the proposed closed-loop. The measurement vector is of higher magnitude but the update state based on this higher value observation are much better and comprehend. As a brief comparison, the absolute error signals are shown in Figure 8. This error plots depict that priority of the proposed closed-loop Kalman filtering MLPC over the previous open-loop Kalman filter with loss of data. It is also true that by providing the upper limit on the error bound, one can notice that the data loss in the open-loop manner will be very conservative than that of the close-loop Kalman filtering.

6 CONCLUSIONS

We have presented a novel approach for state estimation problem in discrete-time LTI systems subject to loss of data. The approach exploits the artificial observation vector which in fact corrects the prediction cycle when loss of data occurs, in order not to allow the estimation error bounds to exceed the desired limits. The resulting closed-loop Kalman filtering also avoids the spike generated in OLE. The performance of the proposed closed-loop Kalman filter approach, when the prediction is updated with compensated observations, was illustrated via a mass-spring-dashpot case study example.

REFERENCES

Allison, P. D. (2001). *Missing Data*. Sage Publications, 1st edition.

Fekri, S., Athans, M., and Pascoal, A. (2007). Robust multiple model adaptive control (RMMAC): A case study. *International Journal of Adaptive Control And Signal Procession*, 21:1–30.

Huang, M. and Dey, S. (2007). Stability of kalman filtering with markovian packet loss. *Automatica*, 43:598–607.

Khan, N. and Gu, D.-W. (2009a). "Properties of a robust kalman filter". In *The 2nd IFAC Conference on Intelligent Control System and Signal Processing*. ICONS, Turkey.

Khan, N. and Gu, D.-W. (2009b). "State estimation in the case of loss of observations". In *ICROS-SICE International Joint Conference*. ICCAS-SICE, Japan.

Li Xie, L. X. (2007). "Peak covariance stability of a random riccati equation arising from kalman filtering with observation losses". *Journal System Science and Complxity*, 20:262–279.

Liu, X. and Goldsmith, A. (2004). "Kalman filtering with partial observation loss". In *IEEE Conference on Decision and Control*, 43.

Micheli, M. (2001). "Random sampling of a continuous time stochastic dynamical system: analysis, state estimation and applications". Master's thesis,, University of Calfornia, Barkeley.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall International, Inc.

Schenato, L. (2005). "Kalman filtering for network control system with random delay and packet loss". In *European Community Research Information Society Technologies*, volume MUIR, Italy.

Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S. S. (2007). Foundation of control and estimation over lossy network. *Proceeding of The IEEE*, 95(1):163–187.

Sinopoli, B. and Schenato, L. (2007). "Kalman filtering with intermittent observations". *IEEE transactions on Automatic Control*, 49(9).

NONLINEAR CONSTRAINED PREDICTIVE CONTROL OF EXOTHERMIC REACTOR

Joanna Ziętkiewicz
Institute of Control and Information Engineering, Poznan University of Technology, Piotrowo 3A, Poznan, Poland
joanna.zietkiewicz@put.poznan.pl

Keywords: Predictive Control, Feedback Linearization, LQ Control.

Abstract: Predictive method which allows applying constraints in the process of designing control system has wide practical significance. The method developed in the article consists of feedback linearization and linear quadratic control applied to obtained linear system. Employment of interpolation method introduces constraints of variables into control system design. The control algorithm was designed for a model of exothermic reactor, results illustrate its operation in comparison with PI control.

1 INTRODUCTION

The predictive algorithms have a wide industrial applications because of the simplicity of its operation and good features of regulation. One of important advantages of the predictive control is the possibility to impose the signal constraints in the process of designing the control law. In the practical applications it is convenient to use the linear models for the theory of them is well known.

First examples of the industrial use of the MPC applications had place in 1970's, but the idea was known earlier (Lee, Markus, 1967). One of the most important algorithms was the Dynamic Matrix Control (Cutler, Ramaker,1980) and Quadratic DMC (Garcia et al.,1989) with linear models. There appeared a number of articles with nonlinear models with the exact and suboptimal algorithms. The use of nonlinear models cause additional problems with finding global minimum and can have an effect on calculation time (Tatjewski, 2002). Adaptation of a controller with linearization around the working point may result in system instability (Dimitar et al., 1991), changes of variables have to be limited.

The aim of the work was to design an application used for control of an exothermic reactor with constraints, to propose use of feedback linearization for this nonlinear plant, present predictive control method solving problem of constraints(Poulsen et al., 2001) and its modification (Ziętkiewicz 2008) for changed reference signal.

2 EXOTHERMIC REACTOR

2.1 CSTR Model

The plant to be controlled is the Continuous Stirred Tank Reactor (CSTR). The structure of reactor is presented on figure 1. It contains tank, cooling jacket, inflow and outflow of both elements. It is assumed that, because of perfect mixing, there are no spatial gradients of parameters in the tank area.

The work of reactor is described by 3 differential equations. First equation (1) illustrates the mass balance,

$$V\frac{dC(t)}{dt} = \phi\left[C_i - C(t)\right] - VR(t), \qquad (1)$$

where $C(t)$ is the concentration of product measured in [kmol/m^3]. The second and the third equations (2,3) represent the balance of energy in the reactor,

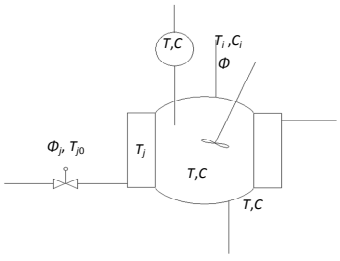


Figure 1: Model of exothermic reactor.

$$V\rho c_p\frac{dT(t)}{dt} = \phi\rho c_p\left[T_i - T(t)\right] - Q(t) + (-\Delta_i)VR(t), \qquad (2)$$

and the balance of energy in the cooling jacket

$$v_j \rho_j c_{pj} \frac{dT_j(t)}{dt} - \varphi_j(t) \rho_j c_{pj} \left[T_{j0} - T_j(t) \right] + Q(t) \tag{3}$$

where T(t) is the temperature inside the reactor and $T_j(t)$ temperature in the cooling jacket, measured in Kelvin. $\phi_j(t)$ [m^3/h] represents cooling flow through the reactor jacket. Remaining equations represent

$$Q(t) = \alpha A_c \left[T(t) - T_j(t) \right] \tag{4}$$

- thermal energy in the process of cooling,

$$R(t) = C(t) k_0 e^{\frac{-E}{RT(t)}} \quad \text{- velocity of reaction.} \tag{5}$$

Constant values used in experiments are placed in the table 1.

Table 1: Constant values of CSTR model.

const.	value	const.	value
ϕ	1.13 [m^3/h]	T_{j0}	294.4 [K]
V	1.36 [m^3]	ρ_j	998 [kg/m^3]
C_i	8 [kmol/m^3]	c_{pj}	4186.8 [J/(kgK)]
ρ	801 [kg/m^3]	k_0	7.08*10^{10} [1/h]
c_p	3140.1 [J/(kgK)]	E	6.96*10^7 [J/kmol]
T_i	294.4 [K]	R	8314.3 [J/(kmolK)]
$(-\Delta_i)$	6.96*10^7 [J/kmol]	α_c	3.07*10^6 [J/(hKm^2)]
v_j	0.109 [m^3]	A_c	23.2 [m^2]

In the further parts of the paper the function of time will be omitted to simplify equations. The control signal will be denoted as $u = \phi_j(t)$ and the state variables x_1=$C(t)$, x_2=$T(t)$, x_3=$T_j(t)$. The system (1-3) can be describe by 3 equations:

$$\begin{aligned} \dot{x}_1 &= A_1 C_i - (A_1 + k_0 e^{-E/Rx_2}) x_1, \\ \dot{x}_2 &= A_1 T_i - (A_1 + B_1) x_2 + B_1 x_3 + C x_1 k_0 e^{-E/Rx_2}, \\ \dot{x}_3 &= B_2 (x_2 - x_3) + \frac{T_{j0} - x_3}{v_j} u, \end{aligned} \tag{6}$$

where $A_1 = \frac{\phi}{V}$, $B_1 = \frac{\alpha_c A_c}{V \rho c_p}$, $B_2 = \frac{\alpha_c A_c}{v_j \rho_j c_{pj}}$, $C = \frac{(-\Delta_i)}{\rho c_p}$.

2.2 Formulation of Control Problem

The objective of control is to make the temperature inside the reactor $T(t)$ track a desired trajectory $w(t)$ using the control signal u. The complete model with output signal can be described by (6) with defined output signal

$$y = x_2. \tag{7}$$

Furthermore the control signal is constrained

$$0m^3/h \le \phi_j \le 2.5m^3/h \tag{8}$$

3 FEEDBACK LINEARIZATION

The functions describing the considered system are smooth and have continuous derivatives of any required order in region $\Omega=\{(x_1, x_2, x_3) \in R^2 | x_2 > T_{j0}, x_3 > T_{j0}\}$, which is the normal area of reactor operation. Since the relative degree is equal to 2 and the system order was equal to 3, the system has internal dynamic described by one equation. From (6) it takes form:

$$\dot{x}_1 = A_1 C_i - (A_1 + k_0 e^{-E/Ry}) x_1. \tag{9}$$

Parameters E and R are positive (tab.1). The output signal y is also positive. If we assume, that control law provides, that signal y is bounded ($y(t)=e(t)+w(t)$, where $e(t)$ is the tracking error), then the internal dynamic of the system is stable.

The system (6,7) can be described in a the form

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}) + g(\mathbf{x}) u \\ y &= h(\mathbf{x}). \end{aligned} \tag{10}$$

There exists a diffeomorphism $\mathbf{z}=\varphi(\mathbf{x})$ in region Ω,

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \varphi(\mathbf{x}) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \eta(x) \end{bmatrix} \tag{11}$$

which conditions normal form of transformed system. $L_f h(\mathbf{x})$ is the Lie derivative of $h(\mathbf{x})$ with respect to $f(\mathbf{x})$. All variables of vector z have to be independent, therefore $\eta(\mathbf{x})$ should satisfy $L_g \eta(\mathbf{x})=0$. One of solutions is $\eta(\mathbf{x})=x_1$. The feedback law is defined as

$$u = \psi(v, \mathbf{x}) = \frac{v - L_f^2 h(\mathbf{x})}{L_g L_f h(\mathbf{x})}, \tag{12}$$

where v is the new input signal. The feedback linearization method is illustrated in fig.3.
The system with new coordinates takes form

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} z_2 \\ v \\ A_1 C_i - (A_1 + k_0 e^{-E/Rz_1}) z_3 \end{bmatrix}, \tag{13}$$

$$y = z_1, \tag{14}$$

for which the mapping $\mathbf{z}=\varphi(\mathbf{x})$:

$$\varphi(\mathbf{x}) = \begin{bmatrix} x_2 \\ A_1T_1 - (A_1 + B_1)x_2 + B_1x_3 + Cx_1k_0e^{-E/Rx_2} \\ x_1 \end{bmatrix},$$

and the inverse mapping $\mathbf{x}=\varphi^{-1}(\mathbf{z})$:

$$\phi^{-1}(\mathbf{z}) = \begin{bmatrix} z_3 \\ z_1 \\ \dfrac{(A_1 + B_1)z_1 + z_2 - Cz_3k_0e^{-E/Rz_1} - A_1T_1}{B_1} \end{bmatrix}. \quad (15)$$

The transformed system is linearized partly, the third equation is nonlinear. However, the relation between input and output signal is linear, which will be used in control algorithm.

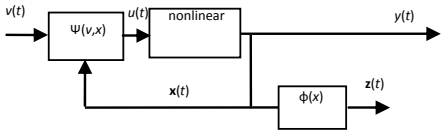


Figure 2: Feedback linearization.

4 PREDICTIVE CONTROL

To design the control algorithm we will use linear model obtained in previous section

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ v \end{bmatrix}$$
$$y = z_1. \quad (16)$$

Third equation of (13) will be used only to calculate successive variables of vector **z**, and then from (15) vector **x**. After discretisation of the linear model with T_s=60s and adding reference signal w_k which is imposed by using an additional variable

$$p_{k+1} = p_k + w_k + y_k, \quad (17)$$

we obtain a discrete model

$$\begin{bmatrix} z \\ p \end{bmatrix}_{k+1} = \begin{bmatrix} A_d & 0 \\ -C_d & 1 \end{bmatrix}\begin{bmatrix} z \\ p \end{bmatrix}_k + \begin{bmatrix} B_d \\ 0 \end{bmatrix}v_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix}w_k,$$
$$y_k = \begin{bmatrix} C_d & 0 \end{bmatrix}\begin{bmatrix} z \\ p \end{bmatrix}_k, \quad (18)$$

where A_d, B_d, and C_d denote matrices of discrete model.

4.1 Linear Quadratic Control

The predictive control algorithm for the system without constraints and infinite horizon can be designed by LQ control method (Maciejowski, 2002). The cost function which prevents too large deviation from equilibrium point is given by:

$$J_t = \sum_{k=t}^{\infty} \begin{bmatrix} z_k - z_k^0 \\ p_k - p_k^0 \end{bmatrix}^T Q \begin{bmatrix} z_k - z_k^0 \\ p_k - p_k^0 \end{bmatrix} + R(v_k - v_k^0)^2, \quad (19)$$

with $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and R=0.1. The optimal gain L is obtained from LQ method. Then the control law describes

$$\hat{u}_{k|t} = Mw_t - L\begin{bmatrix} \hat{z} \\ \hat{p} \end{bmatrix}_{k|t}, \quad (20)$$

where M is the first element of L, because the output is the first element of the state vector **z**. The index $k|t$ denotes the sample of variable predicted for the moment t and calculate in the instant k.

4.2 Constrained Predictive Control

In order to include the constraints to the control problem, there will be applied the interpolation technique (Poulsen et al., 2001). It consists in using the LQ method for a system with so changed required output trajectory $\tilde{w}_{k|t}$ that the obtained variables fulfil the constraints. The changed trajectory is defined by

$$\tilde{w}_{k|t} = w_t + \hat{s}_{k|t}, \quad (21)$$

then the control law

$$\hat{v}_{k|t} = M\tilde{w}_{k|t} - L\hat{z}_{k|t}. \quad (22)$$

The so called perturbation trajectory $\hat{s}_{k|t}$ calculated in the instant k for successive steps $k \le t \le H$ is obtained from

$$\hat{s}_{k|t} = \alpha_k \hat{s}_{k-1|t}, \quad (23)$$

where $0 \le \alpha_k \le 1$.

It can be seen from (21) and (23), that α_k=0 corresponds to the unconstrained LQ control. To find proper $\hat{s}_{k|t}$ assuring feasibility of $\tilde{w}_{k|t}$ we use the initial perturbation trajectory $\hat{s}_{0|t}$, which ensures fulfilling the constraints. One of solution is to chose the $\hat{s}_{0|t}$ so it maintains trajectory $\tilde{w}_{k|t}$ unchanged for future t, therefore every variable in model is unchanged (assuming that initial condition is stable and fulfil given constraints).

With above reasoning the objective of control is to minimize the parameter α_k with respect to constraints on assumed horizon H. Even though the model (18) is linear, the relation between constrained variable u and α is nonlinear, because it goes through the function $u = \psi(v, \mathbf{x})$. To solve this nonlinear problem it is possible to use simple numeric procedure as bisection.

The above procedure was designed for the instant change of the set point. When desired output trajectory w_k changes in another way the following method of calculation of $\hat{s}_{k|t}$ can be used:

$$\hat{s}_{k|t} = w_{k|t-1} - w_{k|t+}\alpha_k \hat{s}_{k-1|t} \,. \tag{24}$$

Under assumption that initial conditions are stable and then the initial perturbations sequence is stable, because of the constraints values the control law designed on the interpolation algorithm is asymptotically stable.

5 RESULTS

Two experiments were performed in matlab environment. The PI controller tuned experimentally was used as comparison was. In the first experiment the trajectory w_t was suddenly changed from one value to another. In the second experiment w_t was changed along the linear function, which is a proper behaviour of desired temperature in the reactor. In every figures placed below first chart illustrate the desired trajectory w_t and the output y_t, whilst the second chart show the behaviour of constrained input of the reactor u_t.

The results of the first experiment are illustrated below. The desired trajectory was changed from 333 to 338K with jump in t=20min. Figure 4 illustrate the result obtained from use of PI method, figure 5 with predictive algorithm developed in the article.

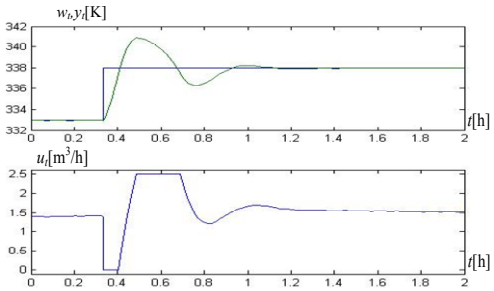


Figure 3: First experiment, PI control.

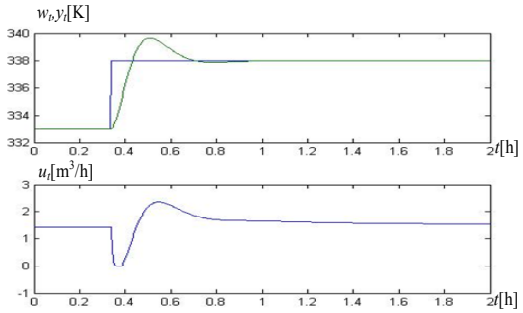


Figure 4: First experiment, predictive control.

In the second experiment trajectory was changed in linear function from 310 to 340K. Results are placed below in a way as in the first experiment.

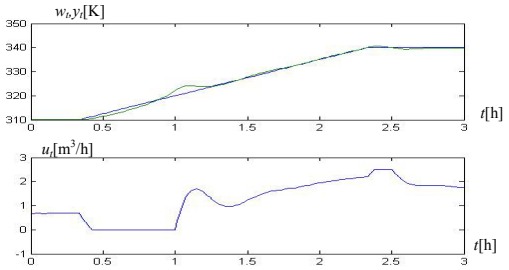


Figure 5: Second experiment, PI control.

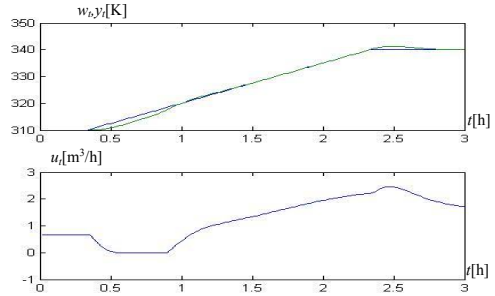


Figure 6: Second experiment, predictive control.

5.1 Conclusions

The operation of predictive method presented in the paper was correct, it fulfils the constraints. In both experiments the use of the algorithm improved the quality of control in comparison with PI control. However the disadvantage of the method is that it relies on feedback linearization, which can be use to limited class of objects.

REFERENCES

Cutler C. R., Ramaker B. L., 1980. Dynamic Matrix Con-

trol – a computer control algorithm, in *Proc. of Joint Automatic Control Conference*, San Francisco.

Dimitar R., Ogonowski Z., Damert K., 1991. Predictive control of a nonlinear open-loop unstable polymerization reactor, in *Chemical Engineering Science,* 46, 2679-2689.

Garcia C. E., Prett D. M., Morari M., 1989. Model Predictive control: theory and practice – a survey, in *Automatica*, 25:335-348.

Lee E. B., Markus L., 1967. *Foundations of Optimal Control Theory*, J. Wiley, New York.

Maciejowski J. M., 2002 *Predictive Control with constraints*, Prentice Hall, Pearson Education Limited, Harlow, UK.

Poulsen N. K., Kouvaritakis B., Cannon M., 2001. Constrained predictive control and its application to a coupled-tanks apparatus, in *International Journal of Control*, Vol.74, 552-564.

Tatjewski P., 2002 *Sterowanie zaawansowane obiektów przemysłowych*, Exit, Warszawa.

Ziętkiewicz J., 2008. Sterowanie predykcyjne z ograniczeniami dla modelu egzotermicznego reaktora chemicznego, in *Proc. of XVI Krajowa Konferencja Automatyki*, Szczyrk.

ANYTIME MODELS IN FUZZY CONTROL

Annamária R. Várkonyi-Kóczy, Attila Bencsik
Institute of Mechatronics, Óbuda University, Népszínház u 8, H-1081 Budapest, Hungary
varkonyi-koczy@uni-obuda.hu, bencsik. attila@bgk.uni-obuda.hu

Antonio Ruano
Faculty of Science & Technology, University of Algarve, Campus de Gambelas, 8005 -139 Faro, Portugal
aruano@ualg.pt

Keywords: Power plant control, Intelligent control, Situational control, Anytime modeling, Fuzzy modeling, SVD based complexity reduction, Time critical systems, Resource insufficiency.

Abstract: In time critical applications, anytime mode of operation offers a way to ensure continuous operation and to cope with the possibly dynamically changing time and resource availability. Soft Computing, especially fuzzy model based operation proved to be very advantageous in power plant control where the high complexity, nonlinearity, and possible partial knowledge usually limit the usability of classical methods. Higher Order Singular Value Decomposition based complexity reduction makes possible to convert different classes of fuzzy models into anytime models, thus offering a way to combine the advantages, like low complexity, flexibility, and robustness of fuzzy and anytime techniques. By this, a model based anytime control methodology can be suggested which is able to keep on continuous operation using non-exact, approximate models of the plant, thus preventing critical breakdowns in the operation. In this paper, an anytime modeling method is suggested which makes possible to use complexity optimized fuzzy models in control. The technique is able to filter out the redundancy of fuzzy models and can determine the near optimal non-exact model of the plant considering the available time and resources. It also offers a way to improve the granularity (quality) of the model by building in new information without complexity explosion.

1 INTRODUCTION

Nowadays, solving control problems model-integrated computing has become very popular. This integration means that the available knowledge is represented in a proper form and acts as an active component of the procedure to be executed during the operation.

For linear problems, well established methods are available and they have been successfully combined with adaptive techniques to provide optimum performance.

In case of nonlinear techniques, fuzzy modeling seems to result in a real breakthrough even when no analytical knowledge is available about the system, the information is uncertain or inaccurate, or when the available mathematical form is too complex to be used. Although, major limitation of fuzzy models is their exponentially increasing complexity. An especially critical situation is, when due to failures or an alarm appearing in the modeled system, the required reaction time is significantly shortened and one has to make decisions before the needed and sufficient information arrives or the processing can be completed.

In such cases, anytime techniques can be applied advantageously to carry on continuous operation and to avoid critical breakdowns. Anytime systems are able to provide short response time and are able to maintain the information processing even in cases of missing input data, temporary shortage of time, or computational power (Zilberstein, 1996). The idea is that if there is a temporal shortage of computational power and/or there is a loss of some data, the actual operations should be continued to maintain the overall performance "at lower price", i.e., information processing based on algorithms and/or models of simpler complexity (and less accuracy) should provide outputs of acceptable quality to continue the operation of the complete system.

There are a few approaches aiming to exploit the advantages of anytime control however mostly in the field of linear control. To mention two of the characteristic approaches, Fontanelli at al. 2008 applies a hierarchical anytime control design strategy with stochastic scheduling conditioning resulting in usually acceptable worst-case execution time and almost sure stability while Battacharya et Balas 2004 uses balanced truncation and residualization of models to generate a set of reduced-order controllers in order to ensure smooth switching between the truncated models.

There are mathematical tools, like Singular Value Decomposition (SVD), which offer a universal scope for handling the complexity problem by anytime operations. SVD proved to be very advantages at different fields of (linear) control, like receding horizon control (RHC) where the application of the technique may offer a sub-optimal control strategy, see e.g. Rojas et al. 2004.

Embedding fuzzy models in anytime systems extends the advantages of the Soft Computing (SC) approach with the flexibility with respect to the available input information and computational power.

In this paper, we deal with the applicability of fuzzy models in dynamically changing, complex, time-critical, anytime systems. The analyzed models are generated by using (Higher Order) Singular Value Decomposition ((HO)SVD). This technique provides a uniform frame for a family of modeling methods and results in low (optimal or nearly optimal) computational complexity, easy realization, and robustness. The accuracy can also easily and flexibly be increased, thus both complexity reduction and the improvement of the approximation can be implemented.

The paper is organized as follows: In Section 2 the generalized idea of anytime processing is introduced. Section 3 summarizes the basics of Singular Value Decomposition. Section 4 is devoted to the SVD based complexity reduction and density improvement of fuzzy models. Section 5 briefly deals with anytime fuzzy control. Finally, in Section 6, conclusions are drown.

2 ANYTIME PROCESSING

In recourse, data, and time insufficient conditions, anytime algorithms, models, and systems (Zilberstein, 1996) can be used advantageously. They are able to provide guaranteed response time and are flexible with respect to the available input data, time, and computational power. This flexibility makes these systems able to work in changing circumstances without critical breakdowns in the performance. The main goal of anytime systems is to keep on the continuous, near optimal operation through finding a balance between the quality of the processing and the available resources.

Iterative algorithms/models are popular tools in anytime systems, because their complexity can easily and flexibly be changed. These algorithms always give some, possibly not accurate result and more and more accurate results can be obtained, if the calculations are continued. When the results are needed, by simply stopping the calculations, the, in the given circumstances best results are got.

Unfortunately, the usability of iterative algorithms is limited. Because of this limitation, a general technique for the application of a wide range of other types of models/ computing methods in anytime systems has been suggested in Várkonyi-Kóczy, et al. 2001, however at the expense of lower flexibility and a need for extra planning and considerations.

3 SINGULAR VALUE DECOMPOSITION

SVD has successfully been used to reduce the complexity of a large family of systems based on both classical and soft techniques (Yam, 1997). An important advantage of the SVD complexity reduction technique is that it offers a formal measure to filter out the redundancy (exact reduction) and also the weakly contributing parts (non-exact reduction). This implies that the degree of reduction can be chosen according to the maximum acceptable error corresponding to the current circumstances. In case of multi-dimensional problems, the SVD technique can be defined in a multidimensional matrix form, i.e. HOSVD can be applied.

The SVD based complexity reduction algorithm is based on the decomposition of any real valued $\underline{\underline{F}}$ matrix:

$$\underline{\underline{F}}_{(n_1 \times n_2)} = \underline{\underline{A}}_{1,(n_1 \times n_1)} \underline{\underline{B}}_{(n_1 \times n_2)} \underline{\underline{A}}^{T}_{2,(n_2 \times n_2)} \quad (1)$$

where $\underline{\underline{A}}_k$, k=1,2 are orthogonal matrices ($\underline{\underline{A}}_k \underline{\underline{A}}_k^T = \underline{\underline{E}}$), and $\underline{\underline{B}}$ is a diagonal matrix containing the λ_i singular values of $\underline{\underline{F}}$ in decreasing order. The maximum number of the

nonzero singular values is $n_{SVD} = \min(n_1, n_2)$. The singular values indicate the significance of the corresponding columns of $\underline{\underline{A}}_k$. The matrices can be partitioned in the following way:

$$\underline{\underline{A}}_k = \left| \underline{\underline{A}}^r_{k,(n_k \times n_r)} \quad \underline{\underline{A}}^d_{k,(n_k \times (n_k - n_r))} \right|$$

$$\underline{\underline{B}} = \left| \begin{matrix} \underline{\underline{B}}^r_{(n_r \times n_r)} & 0 \\ 0 & \underline{\underline{B}}^d_{((n_1 - n_r) \times (n_2 - n_r))} \end{matrix} \right|,$$

where *r* denotes "reduced" and $n_r \le n_{SVD}$. If $\underline{\underline{B}}^d$ contains only zero singular values then $\underline{\underline{B}}^d$ and $\underline{\underline{A}}^d_k$ can be dropped: $\underline{\underline{F}} = \underline{\underline{A}}^r_1 \underline{\underline{B}}^r \underline{\underline{A}}^{rT}_2$. If $\underline{\underline{B}}^d$ contains nonzero singular values, as well, then the $\underline{\underline{F}}' = \underline{\underline{A}}^r_1 \underline{\underline{B}}^r \underline{\underline{A}}^{rT}_2$ matrix is only an approximation of $\underline{\underline{F}}$ and the maximum difference between the values of $\underline{\underline{F}}$ and $\underline{\underline{F}}'$ equals

$$E_{RSVD} = \left| \underline{\underline{F}} - \underline{\underline{F}}' \right| \le \left(\sum_{i=n_r+1}^{n_{SVD}} \lambda_i \right) \underline{1}_{=(n_1 \times n_2)} \qquad (2)$$

For *n*-dimensional cases, HOSVD based reduction $((\underline{\underline{A}}_1, \cdots, \underline{\underline{A}}_n, \underline{\underline{F}}^r) = HOSVDR(\underline{\underline{F}}))$ can be made in *n* steps, in every step one dimension of matrix $\underline{\underline{F}}$, containing the $y_{i_1,\ldots,i_n}$ consequences is reduced.

The first step sets $\underline{\underline{F}}_1 = \underline{\underline{F}}$. In the followings, $\underline{\underline{F}}_i$ is generated by step *i*-1. The *i*-th step of the algorithm (*i*>1) is

1, Spreading out the *n*-dimensional matrix $\underline{\underline{F}}_i$ (size: $n_1^r \times \ldots \times n_{i-1}^r \times n_i \times \ldots \times n_n$) into a two-dimensional matrix $\underline{\underline{S}}_i$ (size: $n_i \times (n_1^r * \ldots * n_{i-1}^r * n_{i+1} * \ldots * n_n)$).

2, SVD based reduction of $\underline{\underline{S}}_i$: $\underline{\underline{S}}_i \approx \underline{\underline{A}}_i \underline{\underline{B}} \underline{\underline{A}}'^T_i = \underline{\underline{A}}_i \underline{\underline{S}}^*_i$, where the size of $\underline{\underline{A}}_i$ is $n_i \times n_i^r$ and the size of $\underline{\underline{S}}^*_i$ is $n_i^r \times (n_1^r * \ldots * n_{i-1}^r * n_{i+1} * \ldots * n_n)$.

3, Re-stacking $\underline{\underline{S}}^*_i$ into the *n*-dimensional matrix $\underline{\underline{F}}_{i+1}$ (size $n_1^r \times \ldots \times n_i^r \times n_{i+1} \times \ldots \times n_n$), and cont. with step 1. for $\underline{\underline{F}}_{i+1}$

The SVD based complexity reduction can be applied to various types of fuzzy systems, see e.g. Takács et Várkonyi-Kóczy, 2002, and Takács et Várkonyi-Kóczy, 2003.

4 ANYTIME MODELING: COMPLEXITY REDUCTION AND IMPROVING THE APPROXIMATION

With the help of the SVD-based reduction not only the redundancy of the rule-bases of fuzzy systems can be removed, but further reduction can also be obtained, considering the allowable error. This latter can be done adaptively according to the temporal conditions, thereby offering a way to use fuzzy models in anytime systems.

The method also offers a way for improving the model if later on we get into possession of new information (approximation points) or more resources. An algorithm can be suggested, which finds the common minimal implementation space of the compressed original and the new approximation points, thus the complexity will not explode as we include new information into the model. These two techniques, non-exact complexity reduction and the improvement of the approximation accuracy, ensure that we can always cope with the temporarily available (finite) time/resources and find the balance between accuracy and complexity.

4.1 Reducing the Complexity of the Model

For creating anytime models, first a practically "accurate" fuzzy system is to be constructed. The rule-base can be determined e.g. by using expert knowledge. In the second step, the redundancy of this model is reduced by (HO)SVD. The (non-exact) anytime models can be obtained either by applying the iterative transformation algorithm described in Takács et Várkonyi-Kóczy, 2004 or in the general frame of modular architecture (for details, see Várkonyi-Kóczy et al., 2001).

In the first case, the transformation can be performed off-line and the model evaluation can be executed till the control action/results are needed. The newest output corresponds to the, in the given circumstances obtainable best results.

In the latter case, the models resulted by the HOSVD reduction will differ in their accuracy and complexity. An intelligent expert system, monitoring

the actual state of the supervised system, can adaptively determine and change for the units (rule base models) to be applied according to the available computing time and resources at the moment. These considerations need additional computational time/resources (further reducing the resources).

It is worth mentioning, that the SVD based reduction finds the optimum, i.e., minimum number of parameters which is needed to describe the system.

One can find more details about the intelligent anytime monitor and the algorithmic optimization of the evaluations of the model-chain in Zilberstein, 1993 and Várkonyi-Kóczy et Samu, 2004.

4.2 Improving the Approximation of the Model

The complexity of the control can be tuned both by evaluating only a degraded model (decreasing the granulation) and both by improving the existing model (increasing the granulation) in the knowledge of new information. This latter means the improvement of the density of the approximation points. Here a very important aim is not to let to explode the complexity of the compressed model when the approximation is extended with new points. Thus, if approximation A is extended to B with a new set of approximation points and basis, then the question is how to transform A^r to B^r directly without decompressing A^r, where A^r and B^r are the reduced forms of A and B. In the followings, an algorithm is summarized for the complexity compressed increase of such approximations.

To enlighten more the problem, let us show a simple example. Assume that we deal with the approximation of function $F(x_1, x_2)$ (see Fig. 1). For simplicity, assume that the applied approximation A is a bi-linear approximation based on the sampling of $F(x_1, x_2)$ over a rectangular grid (Fig. 2), so, the bases are formed of triangular fuzzy sets (or first order B-spline functions). After applying SVD based reduction, the minimal dimensionality of the basis is defined. In Fig. 3, as the minimum basis, two basis functions are shown on each dimension instead of the original three as depicted in Fig. 2.

Let us suppose that at a certain stage, further points are sampled (Fig. 4) in order to increase the density of the approximation points in dimension X_1, hence, to improve approximation A to achieve approximation B. The new points can easily be added to approximation A shown in Fig. 2 to yield approximation B with an extended basis, as is shown in Fig. 5. Usually, however, once reduced approximation A^r is found then the new points should directly be added to A^r (where there is no localized approximation point) to generate a reduced approximation B^r (see Fig. 6). Here again, as an illustration, two basis are obtained in each dimension, hence the calculation complexity of A^r and B^r are the same, but the approximation is improved.

In more general, the crucial point is to inject new information, given in the original form, into the compressed one. If the dimensionality of B^r is larger than A^r then the new points and basis lead to the expansion of the basis' dimensionality of the reduced form A^r. On the other hand, if the new points and basis have no new information on the dimensionality of the basis then they are swallowed in the reduced form without the expansion of the dimensionality, however the approximation is improved. Thus, the approximation can get better with new points without increasing the calculation complexity. This implies a practical question, namely: how to apply those extra points taken from a large sampled set to be embedded, which have no new information on the dimensionality of the basis, but carry new information on the approximation?

For **fitting of two approximations into a common basis system,** we use the transformation of the rational general form of PSGS and Takagi-Sugeno-Kang fuzzy systems. The rational general form (Klement et al., 1999) means that these systems can be represented by a rational fraction function

$$y = \frac{\sum_{j_1=1}^{e_1} \cdots \sum_{j_n=1}^{e_n} \prod_{i=1}^{n} \mu_{i,j_i}(x_i)\, f_{j_1,\cdots,j_n}(x_1,\cdots,x_n)}{\sum_{j_1=1}^{e_1} \cdots \sum_{j_n=1}^{e_n} \prod_{i=1}^{n} \mu_{i,j_i}(x_i) w_{j_1,\cdots,j_n}} \tag{3}$$

where $f_{i_1,\cdots,i_n}(x_1,\cdots,x_n) = \sum_{t=1}^{m} b_{i_1,\cdots,i_n,t}\phi(x_1,\cdots,x_n)$.

It can be proved (see e.g. Yam, 1997 and Baranyi et al., 1999) that (3) can always be transformed into the form of

$$y = \frac{\sum_{j_1=1}^{e_1^r} \cdots \sum_{j_n=1}^{e_n^r} \prod_{i=1}^{n} \mu_{i,j_i}^r(x_i)\, f_{j_1,\cdots,j_n}^r(x_1,\cdots,x_n)}{\sum_{j_1=1}^{e_1^r} \cdots \sum_{j_n=1}^{e_n^r} \prod_{i=1}^{n} \mu_{i,j_i}^r(x_i) w_{j_1,\cdots,j_n}^r} \tag{4}$$

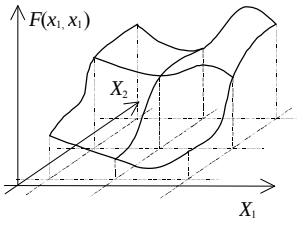


Figure 1: Sampling $F(x_1, x_2)$ over a rectangular grid.

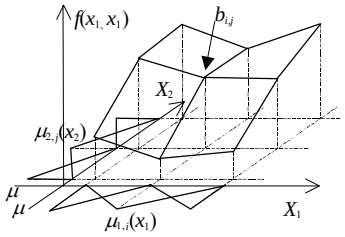


Figure 2: Bi-linear approximation *A* of function $F(x_1, x_2)$.

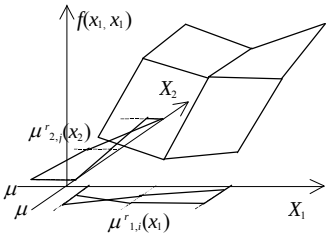


Figure 3: Approximation A^r, which is the reduced form of approximation *A*.

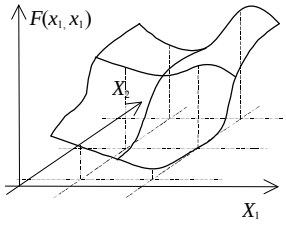


Figure 4: Sampling further approximation points.

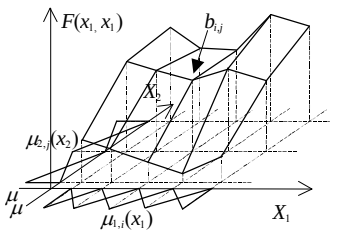


Figure 5: Approximation *B*.

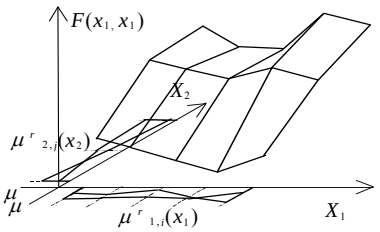


Figure 6: Reduced approximation B^r.

where $f^r_{i_1,\cdots,i_n}(x_1,\cdots,x_n) = \sum_{t=1}^{m} b^r_{i_1,\cdots,i_n,t}\phi(x_1,\cdots,x_n)$ and $\forall i : e^r_i \le e_i$, which is essential in complexity reduction.

Let us suppose that two *n*-variable approximations are defined on the same domain with the same basis functions $\underline{\underline{\mu}}_i$. One is called "original" and is defined by matrix $\underline{\underline{O}}$ of size $e_1 \times \cdots \times e_n \times p$ where p is m or $m+1$ (see (3) and (4)).

The other one is called "additional" and is given by matrix $\underline{\underline{A}}$ of the same size. Let us assume that both approximations are reduced by the HOSVD complexity reduction technique as: $(\underline{\underline{N}}_1, \cdots, \underline{\underline{N}}_n, \underline{\underline{O}}^r) = HOSVDR(\underline{\underline{O}})$ and $(\underline{\underline{G}}_1, \cdots, \underline{\underline{G}}_n, \underline{\underline{A}}^r) = HOSVDR(\underline{\underline{A}})$, where the sizes of matrices $\underline{\underline{N}}_i$, $\underline{\underline{O}}^r$, $\underline{\underline{G}}_i$, and $\underline{\underline{A}}^r$ are $e_i \times r_i^o$, $r_1^o \times \cdots \times r_1^o \times p$, $e_i \times r_i^a$, and $r_1^a \times \cdots \times r_1^a \times p$, respectively, and $\forall i : r_i^o \le e_i$ and $\forall i : r_i^a \le e_i$. This implies that the size of $\underline{\underline{O}}^r$ and $\underline{\underline{A}}^r$ may be different, thus the number and the shape of the reduced basis of the two functions can also be different. The method detailed in the following finds the minimal common basis for the reduced forms. The reduction can be exact or non-exact, the dimension of the minimal basis in the non-exact case can be defined according to a given error threshold like in case of HOSVD.

For finding the minimal common basis $(\underline{\underline{U}}_i, \underline{\underline{\Phi}}^o, \underline{\underline{\Phi}}^a)$ for $(\underline{\underline{N}}_i, \underline{\underline{O}}^r)$ and $(\underline{\underline{G}}_i, \underline{\underline{A}}^r)$, the following steps have to be executed in each $i = 1..n$ dimension $(\forall i : (\underline{\underline{\mathbf{U}}}_i, \underline{\underline{\mathbf{\Phi}}}^o, \underline{\underline{\mathbf{\Phi}}}^a) = unify(i, \underline{\underline{\mathbf{N}}}_i, \underline{\underline{\mathbf{O}}}^r, \underline{\underline{\mathbf{G}}}_i, \underline{\underline{\mathbf{A}}}^r))$:

The first step of the method is to determine the minimal unified basis $(\underline{\underline{U}}_i)$ in the *i*-th dimension. Let us apply $(\underline{\underline{U}}_i, \underline{\underline{Z}}_i) = reduct(i, \lfloor \underline{\underline{N}}_i \quad \underline{\underline{G}}_i \rfloor)$ where

function $reduct(d, \underline{\underline{B}})$ reduces the size of an *n*-dimensional ($e_1 \times \cdots \times e_n$) matrix in the *d*-th dimension. The results of the function are matrices $\underline{\underline{N}}$ and $\underline{\underline{B}}^r$. The size of $\underline{\underline{N}}$ is $e_d \times e_d^r$, $e_d^r \le e_d$; the size of $\underline{\underline{B}}^r$ is $c_1 \times \cdots \times c_n$, where $c_d = e_d^r$ and $\forall i, i \neq d : c_i = e_i$. (The algorithm of the function is similar to the HOSVD reduction algorithm, i.e. the steps are: spread out, reduction, re-stack.) Thus, as a result, we get $\underline{\underline{U}}_i, \underline{\underline{Z}}_i$ where the size of $\underline{\underline{U}}_i$ is $e_i \times r_i^u$ ("u" denotes unified) and the size of $\underline{\underline{Z}}_i$ is $r_i^u \times (r_i^o + r_i^a)$.

The second step of the method is the transformation of the elements of matrices $\underline{\underline{O}}^r$ and $\underline{\underline{A}}^r$ to the common basis:

Let $\underline{\underline{Z}}_i$ be partitioned as $\underline{\underline{Z}}_i = \lfloor \underline{\underline{S}}_i \quad \underline{\underline{T}}_i \rfloor$ where the sizes of $\underline{\underline{S}}_i$ and $\underline{\underline{T}}_i$ are $r_i^u \times r_i^o$ and $r_i^u \times r_i^a$ respectively. $\underline{\underline{\Phi}}^o$ and $\underline{\underline{\Phi}}^a$ are the results of transformations $\underline{\underline{\Phi}}^o = product(i, \underline{\underline{S}}_i, \underline{\underline{O}}^r)$ and $\underline{\underline{\Phi}}^a = product(i, \underline{\underline{T}}_i, \underline{\underline{A}}^r)$ where function $(\underline{\underline{A}}) = product(d, \underline{\underline{N}}, \underline{\underline{L}})$ multiplies the multi-dimensional matrix $\underline{\underline{L}}$ of $e_1 \times \cdots \times e_n$ by matrix $\underline{\underline{N}}$ in the *d*-th dimension. If the size of $\underline{\underline{N}}$ is $g \times h$ then $\underline{\underline{L}}$ must hold $e_d = h$. The size of the resulted matrix $\underline{\underline{A}}$ is $a_1 \times \cdots \times a_n$ where $\forall i, i \neq d : a_i = e_i$, and $a_d = g$.

Let us return to the original aim, which is injecting the points of additional approximation *A* into O^r, the reduced form of the original approximation *O*. According to the problem, the union of *A* and O^r must be done without the decompression of O^r. For this purpose the following method is proposed:

Let us assume that an *n*-variable original approximation *O* is defined by basis functions $\underline{\mu}_i^o$, $i = 1..n$ and matrix $\underline{\underline{O}}$ of size $e_1^o \times \cdots \times e_n^o \times p$ in the form of (3) (see also Fig. 2). Let us suppose that the density of the approximation grid lines is increased in the *k*-th dimension (Figs. 4 and 5). Let the extended approximation *E* be defined by matrix $\underline{\underline{E}}$ whose size agrees with the size of $\underline{\underline{O}}$ except in the extended *k*-th dimension where it equals $e_k^e = e_k^o + e_k^a$ (e_k^a indicates the number of additional basis functions) (Fig. 5). The basis of the extended approximation is the same as the original one in all dimensions except in the *k*-th one, which is simply the joint set of the basis functions of approximations *O* and *A*

$$\underline{\mu}_k^e = \underline{\underline{P}} \begin{bmatrix} \underline{\mu}_k^o \\ \underline{\mu}_k^a \end{bmatrix} \tag{5}$$

$\underline{\mu}_k^a$ is the vector of the additional basis functions. $\underline{\underline{P}}$ stands for a perturbation matrix if some special ordering is needed for the basis functions in $\underline{\mu}_k^e$. The type of the basis functions, however, usually depends on their number due to various requirements of the approximation, like non-negativeness, sum normalization, and normality. Thus, in case of increasing the number of the approximation points, the number of the basis functions is increasing as well and their shapes are also changing. In this case, instead of simply joining vectors $\underline{\mu}_k^o$ and $\underline{\mu}_k^a$, a new set of basis $\underline{\mu}_k^e$ is defined according to the type of the approximation like in Fig. 4. Consequently, having approximation *O* and the additional points, the extended approximation *E* can easily be obtained as $\underline{\underline{E}} = fit(k, \underline{\underline{O}}, \underline{\underline{A}})$ where function $\underline{\underline{A}} = fit(d, \underline{\underline{L}}_1, \cdots, \underline{\underline{L}}_z)$ is for fitting the same sized, except in the *d*-th dimension, matrices in the *d*-th dimension: Matrices $\underline{\underline{L}}_k = [l_{k,i_1,\cdots,i_n}]$ have the size of $e_{k,1} \times \cdots \times e_{k,n}$, $k = 1..z$ to the subject that $\forall k, i, i \neq d : e_{k.i} = e_i$. The resulted matrix $\underline{\underline{A}}$ has the size as $e_1 \times \cdots \times e_n$, where $e_d = \sum_{k=1}^{z} e_{k,d}$ and the elements of $\underline{\underline{A}} = [a_{i_1,\cdots,i_n}]$ are $a_{i_1,\cdots,i_n} = l_{k,j_1,\cdots,j_n}$ where $\forall t, t \neq d : i_t = j_t$, $i_d = j_d + \sum_{s=1}^{k-1} e_{s,d}$, $k = 1..z$. (More precisely, according to the perturbation matrix in (5) $\underline{\underline{E}} = product(k, \underline{\underline{P}}, fit(k, \underline{\underline{O}}, \underline{\underline{A}}))$).

Embedding the New Approximation *A* into the reduced Form of *O*. The steps of the method are as follows:

First, the redundancy of approximation *A* is filtered out by applying $(\underline{\underline{G}}_1, \cdots, \underline{\underline{G}}_n, \underline{\underline{A}}^r) = HOSVDR(\underline{\underline{A}})$. As next, the merged basis of O^r and A^r is defined. The common minimal basis is determined in all, except the *k*-th, dimensions.

Let $\underline{\underline{W}}_{[1]} = \underline{\underline{O}}^r$ and $\underline{\underline{Q}}_{[1]} = \underline{\underline{A}}^r$. Then, for *t*= 1…*n*-1 evaluate $(\underline{\underline{U}}_j, \underline{\underline{W}}_{[t+1]}, \underline{\underline{Q}}_{[t+1]}) = unify(j, \underline{\underline{N}}_j, \underline{\underline{W}}_{[t]}, \underline{\underline{G}}_j, \underline{\underline{Q}}_{[t]})$

where $j = \begin{cases} t & t < k \\ t+1 & t \geq k \end{cases}$. Finally, let $\underline{\underline{\Phi}}^o = \underline{\underline{Q}}_{[n]}$ and $\underline{\underline{\Phi}}^a = \underline{\underline{Q}}_{[n]}$.

For the k-th dimension let $\underline{\underline{M}} = \underline{\underline{P}} \begin{bmatrix} \underline{\underline{N}}_k & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{G}}_k \end{bmatrix}$, where $\underline{\underline{0}}$ contains only zero elements and $\underline{\underline{P}}$ can ensure any special ordering, as used in (5). $\underline{\underline{N}}_k$ and $\underline{\underline{G}}_k$ are full rank matrices which means that no further (exact) reduction of $\underline{\underline{M}}$ can be obtained. According to the basis, matrices $\underline{\underline{\Phi}}^o$ and $\underline{\underline{\Phi}}^a$ are unified as $\underline{\underline{F}} = fit(k, \underline{\underline{\Phi}}^o, \underline{\underline{\Phi}}^a)$.

Finally, the redundancy, i.e., the linear dependence between matrices $\underline{\underline{\Phi}}^o$ and $\underline{\underline{\Phi}}^a$ is filtered out of $\underline{\underline{F}}$ by $(\underline{\underline{K}}, \underline{\underline{E}}^r) = reduct(k, \underline{\underline{F}})$. Thus, $\underline{\underline{U}}_k = \underline{\underline{M}}\,\underline{\underline{K}}$.

(Here we would like to note again that $\underline{\underline{K}}$ is full rank matrix, i.e., no further (exact) reduction of $\underline{\underline{U}}_k$ can be obtained.) Matrix $\underline{\underline{U}}_i$, having the size of $e_i \times r_i^u$, is to transform the basis as $\underline{\mu}_i^u = \underline{\underline{U}}_i^{\mathrm{T}} \underline{\mu}_i^e$. The size of matrix $\underline{\underline{E}}^r$ is $r_1^u \times \cdots \times r_n^u \times p$. (For more details, see Baranyi et Várkonyi-Kóczy, 2002)

5 ANYTIME TS FUZZY CONTROL

There are numerous successful applications of anytime control which affect on the analysis and design of anytime control systems (see e.g. Andoga et al., 2008, Madarasz et al., 2009, and Várkonyi-Kóczy, 2008). The previously discussed ideas can fruitfully be applied in plant control if Takagi-Sugeno (TS) fuzzy modeling and Parallel Distributed Compensation (PDC) (Tanaka et Wang, 2001) based controller design is used (Fig. 7). If the model approximation is given in the form of TS fuzzy model, the controller design and Lyapunov stability analysis of the nonlinear system reduce to solving the Linear Matrix Inequalities (LMI) problem (Tanaka et al., 1999). This means that first of all we need a TS model of the nonlinear system to be controlled. The construction of this model is of key importance. This can be carried out either by identification based on input-output data pairs or we

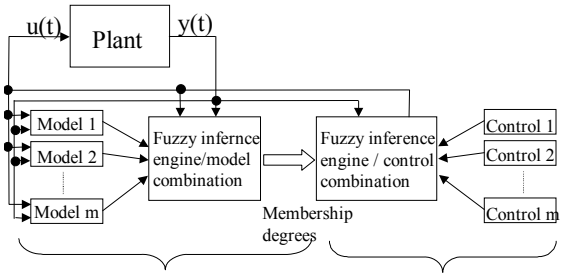


Figure 7: TS fuzzy observer based control scheme.

can derive the model from given analytical system equations.

The PDC offers a direct technique to design a fuzzy controller from the TS fuzzy model. This procedure means that a local controller is determined to each local model. This implies, that the more complex the system model is, the more complex controller will be obtained. According to the complexity problems outlined in the previous sections we can conclude that when theapproximation error of the model tends to zero, the complexity of the controller explodes to infinity. This pushes us to focus on possible complexity reduction and anytime use.SVD based complexity reduction can be applied on two levels in the TS fuzzy controller. First, we can reduce the complexity of the local models (local level reduction). Secondly, it is possible to reduce the complexity of the overall controller by neglecting those local controllers, which have negligible or less significant role in the control (model level reduction). Both can be applied in an anytime way, where we take into account the "distance" between the current position and the operating point, as well. The model granularity or the level of the iterative evaluation can cope with this distance: the further we are, the more rough control actions can be tolerated. Although, approximated models may not guarantee the stability of the system, this can also be ensured by introducing robust control (see e.g. Tanaka et al., 1999).

6 CONCLUSIONS

In this paper, the applicability of (Higher Order) Singular Value Decomposition based anytime fuzzy models in control is analyzed. It is proved that the presented technique can be used for both complexity reduction and for improving the approximation without complexity explosion. The introduced

anytime models can advantageously be used in many types of time critical applications during resource and data insufficient conditions.

ACKNOWLEDGEMENTS

This work was sponsored by the Hungarian National Scientific Fund (OTKA K 78576) and the Hungarian-Portuguese Intergovernmental S&T Cooperation Program.

REFERENCES

Andoga, R., Főző, L., and Madarász, L., 2008 Use of anytime control algorithms in the area of small turbojet engines. *In Proc. of the 6th IEEE Int. Conf. on Computational Cybernetics*, Stará Lesná, Slovakia, Nov 28-30, pp. 33-36.

Baranyi, P., Y. Yam, Y., Yang, Ch-T. and Várkonyi-Kóczy, A.R., 1999. Complexity Reduction of a Rational General Form. *In Proc. of the 8th IEEE Int. Conf. on Fuzzy Systems*, Seoul, Korea, Aug. 22-25, 1, pp. 366-371.

Baranyi P., Lei, K., and Yam, Y., 2000. Complexity reduction of singleton based neuro-fuzzy algorithm. *In Proc. of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 8-11, Nashville, USA, 4, pp. 2503-2508.

Baranyi, P., Várkonyi-Kóczy, A. R, Várlaki, P., Michelberger, P. and Patton, R. J., 2001. Singular Value Based Model Approximation. In N. Mastorakis (ed.) *Problems in Applied Mathematics and Computational Intelligence (Mathematics and Computers in Science and Engineering)*, World Scientific and Engineering Society Press, Danvers, pp. 119-124.

Baranyi, P. and Várkonyi-Kóczy, A. R., 2002. Adaptation of SVD Based Fuzzy Reduction via Minimal Expansion. *IEEE Trans. on Instrumentation and Measurement,* 51(2), pp. 222-226.

Battacharya, R. and Balas, G. J., 2004. Anytime Control Algorithms: Model Reduction Approach. *AIAA Journal of Guidance, Control and Dynamics*, 27(5).

Fontanelli, D., Greco, L., and Bicchi, A, 2008. Anytime Control Algorithms for Embedded Real-Time Systems. In Egerstedt, M, Mishra, B. (eds) *Hybrid Systems: Computation and Control.* Springer-Verlag, Heidelberg, pp. 158-166.

Klement, E. P, Kóczy, L. T., and Moser, B., 1999. Are fuzzy systems universal approximators?. *Int. J. General Systems*, 28(2-3), pp. 259-282.

Madarász, L., Andoga, R., Főző L., and Lazar T., 2009. Situational control, modeling and diagnostics of large scale systems. In Rudas, I., Fodor, J., Kacprzyk, J. (eds.) *Towards Intelligent Engineering and Information Technology*. Springer-Verlag, Heidelberg, pp. 153-164.

Rojas, O. J, Goodwin, G. C., Seron, M. M, and Feuer, A., 2004. An SVD based strategy for receding horizon control of input constrained linear systems. *Int. Journal of Robust and Nonlinear Control.*

Takács, O. and Várkonyi-Kóczy, A. R., 2002. SVD Based Complexity Reduction of Rule Bases with Non-Linear Antecedent Fuzzy Sets, *IEEE Trans. on Instrumentation and Measurement*, 51(2), pp. 217-221.

Takács, O. and Várkonyi-Kóczy, A. R., 2003. SVD-based Complexity Reduction of "Near PSGS" Fuzzy Systems, *In Proc. of the IEEE Int. Symp. on Intelligent Signal Processing*, Budapest, Hungary, Sep. 4-6, pp. 31-36.

Takács, O. and Várkonyi-Kóczy, A. R., 2004. „Iterative Evaluation of Anytime PSGS Fuzzy Systems." In Sincak, P., Vascak, J., Hirota, K. (eds.) *Quo Vadis Machine Intelligence? - The Progressive Trends in Intelligent Technologies,* World Scientific Press, Heidelberg, pp. 93-106.

Tanaka, K., Taniguchi, T., and H. O. Wang, 1999. Robust and Optimal Fuzzy Control: A Linear Matrix Inequality Approach. *In Proc. of the 1999 IFAC World Congress*, Beijing, July 1999, pp. 213-218.

Tanaka K. and Wang, H. O, 2001. Fuzzy Control Systems Design and Analysis, *John Wiley & Sons, Inc*. New York.

Várkonyi-Kóczy, A. R., Ruano, A., Baranyi, P. and Takács, O., 2001. Anytime Information Processing Based on Fuzzy and Neural Network Models. *In Proc. of the 2001 IEEE Instrumentation and Measurement Technology Conf.*, Budapest, Hungary, May 21-23, pp. 1247-1252.

Várkonyi-Kóczy, A. R., and Samu, G., 2004. Anytime System Scheduler for Insufficient Resource Availability. *Int. J. of Advanced Computational Intelligence and Intelligent Informatics*, 8(5), pp. 488-494.

Várkonyi-Kóczy, A. R., 2008. State Dependant Anytime Control Methodology for Non-linear Systems. Int. J.l of Advanced Computational Intelligence and Intelligent Informatics, March, 12(2), , pp. 198-205.

Yam, Y., 1997. Fuzzy Approximation via Grid Sampling and Singular Value Decomposition. *In Proc. of the IEEE Trans. on Systems, Men, and Cybernetics*, 27(6), pp. 933-951.

Zilberstein, S., 1993. Operational Rationality through Compilation of Anytime Algorithms, *PhD Thesis*.

Zilberstein, S., 1996. Using Anytime Algorithms in Intelligent systems. *AI Magazine*, 17(3) , pp. 73-83.

AUTHOR INDEX

Proceedings Indexed by:

Proceedings of ICINCO 2010
International Conference on Informatics in Control, Automation and Robotics
ISBN: 978-989-8425-02-7
http://www.icinco.org