Lecture Notes in Electrical Engineering 89

Juan Andrade Cetto
Jean-Louis Ferrier
Joaquim Filipe
Editors

Informatics in Control, Automation and Robotics

Revised and Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2010

Springer

Lecture Notes in Electrical Engineering

Volume 89

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

Informatics in Control, Automation and Robotics

Revised and Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2010

Springer

Prof. Juan Andrade Cetto
Univ. Politecnica Catalunya
Institut Robotica i Informatica
Industrial
Llorens i Artigas, 4-6
Edifici U
08028 Barcelona
Spain
E-mail: cetto@iri.upc.edu

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

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

ISBN 978-3-642-19538-9 e-ISBN 978-3-642-19539-6

DOI 10.1007/978-3-642-19539-6

Lecture Notes in Electrical Engineering ISSN 1876-1100

Library of Congress Control Number: 2011925362

© 2011 Springer-Verlag Berlin Heidelberg

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

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

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

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

The present book includes extended and revised versions of a set of selected papers from the Seventh International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010), held in Funchal, Madeira - Portugal, from 15 to 18 June 2010. The conference was organized in three simultaneous tracks: Intelligent Control Systems and Optimization, Robotics and Automation, and Systems Modeling, Signal Processing and Control. The book is based on the same structure.

ICINCO 2010 received 320 paper submissions, from 57 countries in all continents. From these, after a blind review process, only 27 were accepted as full papers, of which 20 were selected for inclusion in this book, based on the classifications provided by the Program Committee. The selected papers reflect the interdisciplinary nature of the conference. The diversity of topics is an important feature of this conference, enabling an overall perception of several important scientific and technological trends. These high quality standards will be maintained and reinforced at ICINCO 2011, to be held in Noordwijkerhout, The Netherlands, and in future editions of this conference.

Furthermore, ICINCO 2010 included 6 plenary keynote lectures given by 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, USA) and John Hollerbach (University of Utah, USA). We would like to express our appreciation to all of them and in particular to those who took the time to contribute with a paper to this book.

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

October 2010

Juan Andrade-Cetto
Jean-Louis Ferrier
Joaquim Filipe

Conference Committee

Conference Chair

Joaquim Filipe	Polytechnic Institute of Setúbal / INSTICC, Portugal

Program Co-chairs

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

Organizing Committee

Patrícia Alves	INSTICC, Portugal
Sérgio Brissos	INSTICC, Portugal
Marina Carvalho	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
Daniel Pereira	INSTICC, Portugal
Filipa Rosa	INSTICC, Portugal
José Varela	INSTICC, Portugal

Program Committee

Arvin Agah, USA	Francesco Amigoni, Italy
Alexande Poznyak, Mexico	Peter Arato, Hungary
Andrew Adamatzky, UK	Alejandro Hernandez Arieta, Switzerland
Eugenio Aguirre, Spain	Marco Antonio Arteaga, Mexico
Hesham Alfares, Saudi Arabia	Vijanth Sagayan Asirvadam, Malaysia
Adel Al-Jumaily, Australia	T. Asokan, India

Ruth Bars, Hungary
Adil Baykasoglu, Turkey
Karsten Berns, Germany
Mauro Birattari, Belgium
Christian Blum, Spain
Patrick Boucher, France
Bernard Brogliato, France
Kevin Burn, UK
Clifford Burrows, UK
Dídac Busquets, Spain
Javier Fernandez de Canete, Spain
Giuseppe Carbone, Italy
J.L. Martins de Carvalho, Portugal
Alessandro Casavola, Italy
Riccardo Cassinis, Italy
Ratchatin Chanchareon, Thailand
Antonio Chella, Italy
Wen-Hua Chen, UK
Graziano Chesi, China
Sung-Bae Cho, Korea, Republic of
Ryszard S. Choras, Poland
Carlos Coello Coello, Mexico
António Dourado Correia, Portugal
José Boaventura Cunha, Portugal
Mingcong Deng, Japan
Guilherme DeSouza, USA
Denis Dochain, Belgium
Tony Dodd, UK
Venky Dubey, UK
Frederick Ducatelle, Switzerland
Ashish Dutta, India
Petr Ekel, Brazil
Atilla Elci, Turkey
Ali Eydgahi, USA
Jean-marc Faure, France
Paolo Fiorini, Italy
Georg Frey, Germany
Wai-Keung Fung, Canada
Dragan Gamberger, Croatia
Andrea Garulli, Italy
Ryszard Gessing, Poland
Lazea Gheorghe, Romania
Paulo Gil, Portugal
Alessandro Giua, Italy
Luis Gomes, Portugal
Dongbing Gu, UK
Kevin Guelton, France
Maki K. Habib, Egypt
Wolfgang Halang, Germany
Onur Hamsici, USA
John Harris, USA
Inman Harvey, UK
Dominik Henrich, Germany
Suranga Hettiarachchi, USA
Victor Hinostroza, Mexico
Wladyslaw Homenda, Poland
Guoqiang Hu, USA
Joris Hulstijn, The Netherlands
Fumiya Iida, USA
Atsushi Imiya, Japan
Giovanni Indiveri, Italy
Mirjana Ivanovic, Serbia
Sarangapani Jagannathan, USA
Masoud Jamei, UK
Ping Jiang, UK
Graham Kendall, UK
DaeEun Kim, Korea, Republic of
Won-jong Kim, USA
Israel Koren, USA
Gerhard K. Kraetzschmar, Germany
Mianowski Krzysztof, Poland
H.K. Lam, UK
Alexander Lanzon, UK
Kathryn J. De Laurentis, USA
Kauko Leiviskä, Finland
Zongli Lin, USA
Guoping Liu, UK
Jing-Sin Liu, Taiwan
José Tenreiro Machado, Portugal
Anthony Maciejewski, USA
Frederic Maire, Australia
Om Malik, Canada
Hervé Marchand, France
Philippe Martinet, France
Rene V. Mayorga, Canada
Seán McLoone, Ireland
Prashant Mehta, USA
Carlo Menon, Canada
António Paulo Moreira, Portugal
Vladimir Mostyn, Czech Republic
Rafael Muñoz-salinas, Spain
Kenneth Muske, USA
Andreas Nearchou, Greece
Luciana Nedel, Brazil

Sergiu Nedevschi, Romania
Monica N. Nicolescu, USA
Klas Nilsson, Sweden
Urbano Nunes, Portugal
Manuel Ortigueira, Portugal
Selahattin Ozcelik, USA
Igor Paromtchik, France
Mario Pavone, Italy
Claudio de Persis, Italy
D.T. Pham, UK
Michael Piovoso, USA
Raul Marin Prades, Spain
José Ragot, France
Jerzy Respondek, Poland
A. Fernando Ribeiro, Portugal
Robert Richardson, UK
Mihailo Ristic, UK
Juha Röning, Finland
Agostinho Rosa, Portugal
Danilo De Rossi, Italy
Fariba Sadri, UK
Mehmet Sahinkaya, UK
Priti Srinivas Sajja, India
Abdel-badeeh Salem, Egypt
Marcello Sanguineti, Italy
Jurek Sasiadek, Canada
Sergio M. Savaresi, Italy
Carla Seatzu, Italy
Michael Short, UK
Silvio Simani, Italy
Dan Simon, USA
Olivier Simonin, France
Joaquin Sitte, Australia
Adam Slowik, Poland
Andrzej Sluzek, Singapore
Safeeullah Soomro, Saudi Arabia
Stefano Squartini, Italy
Burkhard Stadlmann, Austria
Chun-Yi Su, Canada
Ryszard Tadeusiewicz, Poland
Kazuya Takeda, Japan
Daniel Thalmann, Switzerland
N.G. Tsagarakis, Italy
Avgoustos Tsinakos, Greece
Antonios Tsourdos, UK
Angel Valera, Spain
Eloisa Vargiu, Italy
Annamaria R. Varkonyi-koczy, Hungary
Ramiro Velazquez, Mexico
Damir Vrancic, Slovenia
Bernardo Wagner, Germany
Dianhui Wang, Australia
James Whidborne, UK
Sangchul Won, Korea, Republic of
Qishi Wu, USA
Marek Zaremba, Canada
Janan Zaytoon, France
Primo Zingaretti, Italy

Auxiliary Reviewers

Lounis Adouane, France
Iman Awaad, Germany
Yao Chen, China
Rohit Chintala, India
Jonathan Courbon, France
Kun Deng, USA
Anibal Ferreira, Portugal
Leonardo Fischer, Brazil
Fernando Fontes, Portugal
Ronny Hartanto, Germany
Laurent Harter, France
Nico Hochgeschwender, Germany
Ana Lopes, Portugal
Jan Paulus, Germany
Cristiano Presmebida, Portugal
Michael Reckhaus, Germany
Lynda Seddiki, France
Renato Silveira, Brazil
Yu Sun, USA
C. Renato Vázquez, Spain
Huibing Yin, USA

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, USA
John Hollerbach	University of Utah, USA

Contents

Part II: Robotics and Automation

Part III: Signal Processing, Systems Modeling and Control

Invited Papers

Dynamic Modeling of Robots Using Newton-Euler Formulation

Wisama Khalil

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

Abstract. This paper presents the use of recursive Newton-Euler formulation to model different types of robots. The main advantages of this technique are the facility of implementation and obtaining models with reduced number of operations. The following structures are treated: rigid tree structure robots, closed loop robots, parallel robots, robots with lumped elasticity, robots with moving base and wheeled robots.

Keywords: Dynamic modeling, Newton-Euler, Recursive calculation, Tree structure, Parallel robots, Flexible joints, Mobile robots.

1 Introduction

The dynamic modeling 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 is easy to develop and implement. The proposed algorithm can be extended to many types of structures. In section 2 we recall the method used to describe the kinematics of the structure, and then in section 3 we present the inverse and direct dynamic models of tree structure rigid robots. The following sections present the generalization to the other systems.

2 Description of the Robots

The structures of robots will be described using Khalil and Kleinfinger notations [1]. This method can take into account tree structures and closed loop robots. Its use facilitates the calculation of minimum number of inertial parameters [2]…[4], which reduce the number of operations of the dynamic models and are needed in dynamic identification and adaptive control laws.

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 revolute or pris-matic, rigid or elastic. The links are numbered consecutively from the base, to the terminal links. Joint j connects

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 3–20.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

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 (Fig. 1):

- z_i is along the axis of joint i;
- x_i is along the common normal between z_i and one of the succeeding joint axes.

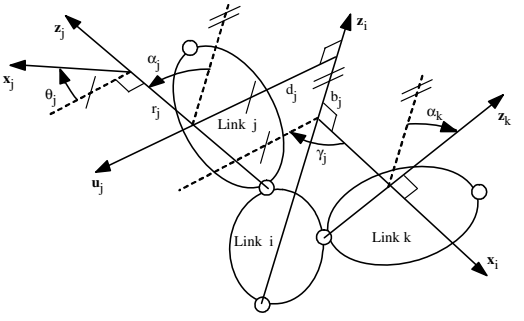


Fig. 1. Geometric parameters for frame j

The transformation matrix ${}^{i}h_j$, defining frame R_j with respect to (wrt) frame R_i is obtained as a function of six geometric parameters $(\gamma_j, b_j, \alpha_j, d_j, \theta_j, r_j)$ such that:

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

After developing, this matrix can be written as follows:

$$ {}^{i}h_j = \begin{bmatrix} {}^{i}R_j & {}^{i}P_j \\ 0_{1x3} & 0 \end{bmatrix} = \begin{bmatrix} C\gamma C\theta - S\gamma C\alpha S\theta & -C\gamma S\theta - S\gamma C\alpha C\theta & S\gamma S\alpha & dC\gamma + rS\gamma S\alpha \\ S\gamma C\theta + C\gamma C\alpha S\theta & -S\gamma S\theta + C\gamma C\alpha C\theta & -C\gamma S\alpha & dS\gamma - rC\gamma S\alpha \\ S\alpha S\theta & S\alpha C\theta & C\alpha & rC\alpha + bj \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Where ${}^{i}R_j$ defines the (3×3) rotation matrix and ${}^{i}P_j$ defines the (3×1) vector defining the position of the origin of frame j with respect to frame i.

If x_i is along the common normal between z_i and z_j, both γ_j and b_j will be zero.

The type of the joint is identified by σ_j where $\sigma_j = 0$ if j is revolute, $\sigma_j = 1$ if j is prismatic, and $\bar{\sigma}_j = 1 - \sigma_j$. The joint variable, denoted as q_j, is θ_j if j is revolute and r_j if j is prismatic.

A serial structure is a special case of a tree structure where $a(j) = j-1,\ \gamma_j = 0,\ b_j = 0$ 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 location of all the links can be determined as a function of the active variables. The geometric parameters of the mechanism can be determined 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) Number the cut joints $k= n+1,\ldots,L$,

c) For each cut joint define two frames on one of the links connected by this joint.

Assuming a cut joint k, and that the links connected by it are link i and link j, the frames are defined as follows (Fig. 2):

- frame R_k is fixed on link j such that $a(k)=i$, the axis z_k is along the axis of joint k, and x_k is along the common normal between z_k and z_j. The matrix ${}^{i}h_k$ is determined using the parameters $(\gamma_k,\ b_k, \alpha_k, d_k, \theta_k, r_k)$.

- frame R_{k+B} is aligned with R_k, but $a(k+B)= j$. The matrix ${}^{j}h_{k+B}$ is constant.

The joint variables are denoted as:

$$q = \begin{bmatrix} q_{tr} \\ q_c \end{bmatrix}, q_{tr} = \begin{bmatrix} q_a \\ q_p \end{bmatrix} \tag{2}$$

- q_{tr} tree structure joint variables, q_c the cut joint variables,
- q_a, q_p active and passive joint variables of the tree structure.

Since R_k and R_{k+B} are aligned, the geometric constraint equations for each loop, can be written as:

$$ {}^{k+B}h_j \ \ldots \ {}^{i}h_k \ = \ I_4 \tag{3}$$

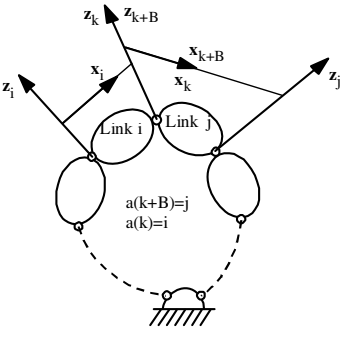


Fig. 2. Frames around a cut joint k

The kinematic constraint equations are given by:

$$\begin{aligned} {}^0V_k &= {}^0V_{k+B} \\ \mathrm{J}_k \dot{q}_{b1} &= \mathrm{J}_{k+B} \dot{q}_{b2} \end{aligned} \tag{4}$$

Where V_k defines the (6×1) kinematic screw vector of frame k, given by:

$$V_k = \begin{bmatrix} v_k^T & \omega_k^T \end{bmatrix}^T \tag{5}$$

v_k linear velocity of the origin of frame $\mathrm{R_k}$, ω_k angular velocity of frame k;
J_k the kinematic Jacobian matrix of frame k;
$\dot{q}_{b1}, \dot{q}_{b2}$ joint velocities through the two branches of the loop.

3 Dynamic Modeling of Tree Structure Robots

3.1 Introduction

The most common methods used to calculate the dynamic models are the Lagrange equations and the Newton Euler Equations [5],…[7]. The Lagrange model is given as:

$$\Gamma = A(q)\ddot{q} + H(q,\dot{q}) \tag{6}$$

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

Calculating Γ in terms of $(\mathrm{q},\dot{\mathrm{q}},\ddot{\mathrm{q}})$ is known as the inverse dynamic problem, and calculating $\ddot{q}$ in terms of $(q,\dot{q},\Gamma)$ is known as the direct dynamic model. The inverse dynamic model is obtained from (6), whereas the direct model is calculated as:

$$\ddot{q} = (A)^{-1}(\Gamma - H) \tag{7}$$

The recursive Newton-Euler algorithms have been shown to be the most efficient technique to model rigid robots [8]…[11]. In [12] a recursive Lagrange algorithm is presented but without achieving better performances than that of Newton-Euler.

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

$$\mathrm{j}\Phi_{\mathrm{j}} = {}^jJ_j\,{}^j\dot{V}_j + \begin{bmatrix} {}^j\omega_j \times \left({}^j\omega_j \times {}^jMS_j \right) \\ {}^j\omega_j \times \left({}^jI_{oj}\,{}^j\omega_j \right) \end{bmatrix} \tag{8}$$

where

ω_{j} angular velocity of link j; $\dot{V}_j$ contains the linear acceleration of the origin of frame j, denoted $\dot{v}_j$, and the angular acceleration $\dot{\omega}_{\mathrm{j}}$.

Φ_{j} total wrench (forces and moments) on link j at origin of frame R_j;

$J_j\ (6\times6)$ inertia matrix of link j, which is given as:

$$ {}^{j}J_j = \begin{bmatrix} M_j I_3 & -{}^{j}M\hat{S}_j \\ {}^{j}M\hat{S}_j & {}^{j}I_{oj} \end{bmatrix} \tag{9} $$

Where M_j, MS_j and I_{oj} are the standard inertial parameters of link j. They are respectively, the mass, the first moments, and the inertia matrix at the origin.

3.2 Calculation of the Inverse Dynamics Using Recursive NE Algorithm

The algorithm consists of two recursive computations: forward and backward [8]. The forward equations, from 1 to n, compute the velocities, accelerations and the dynamic wrench of links. The backward equations from n to 1, provide the joint torques. The algorithm will be denoted by:

$$ \Gamma = \mathrm{NE}(q, \dot{q}, \ddot{q}, F_e) \tag{10} $$

Where F_e is the external wrench exerted by the links of the robot on the environment. The forward equations for $j = 1,...,n$ are as follows:

$$ {}^{j}\omega_j = {}^{j}R_i\, {}^{i}\omega_i + \bar{\sigma}_j\, {}^{j}z_j\, \dot{q}_j \tag{11} $$

$$ {}^{j}\dot{V}_j = {}^{j}T_i\, {}^{i}\dot{V}_i + \ddot{q}_j\, {}^{j}a_j + \begin{bmatrix} {}^{j}R_i\left[{}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_j \right) \right] + 2\sigma_j ({}^{j}\omega_i \times \dot{q}_j\, {}^{j}z_j) \\ \bar{\sigma}_j\, {}^{j}\omega_i \times \dot{q}_j\, {}^{j}z_j \end{bmatrix} \tag{12} $$

$$ {}^{j}\Phi_j = {}^{j}J_j\, {}^{j}\dot{V}_j + \begin{bmatrix} {}^{j}\omega_j \times \left({}^{j}\omega_j \times {}^{j}MS_j \right) \\ {}^{j}\omega_j \times \left({}^{j}I_{oj}\, {}^{j}\omega_j \right) \end{bmatrix} \tag{13} $$

Where

- $i = a(j)$, ${}^{j}z_j = [0\ 0\ 1]^T$, ${}^{j}a_j = [0\ 0\ \sigma_j\ 0\ 0\ \bar{\sigma}_j]^T$, $\hat{w}$ defines the skew matrix of vector product associated to the (3×1) vector w such that $w \times v = \hat{w}\, v$.
- ${}^{j}T_i$ the (6×6) screw transformation matrix:

$$ {}^{j}T_i = \begin{bmatrix} {}^{j}R_i & {}^{j}\hat{P}_i\, {}^{j}R_i \\ 0_{3x3} & {}^{j}R_i \end{bmatrix} \tag{14} $$

These equations are initialized by ${}^{0}\omega_0 = 0, {}^{0}\dot{\omega}_0 = 0$. The gravity effect on all the links is taken into account by putting $\dot{v}_0 = -g$.

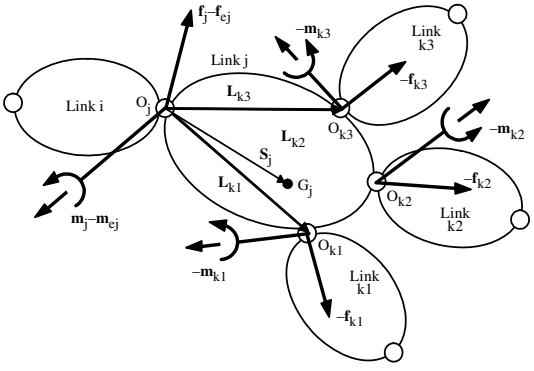


Fig. 3. Forces and moments acting on link j

The backward recursive equations are deduced from the total forces and moments on link j around its origin (Fig. 3). They can be calculated for $j = n,...,1$

$$
\begin{aligned}
{}^{j}F_j &= {}^{j}\Phi_j + \sum_k {}^{k}T_j^{T}\,{}^{k}F_k + {}^{j}F_{ej} \\
\Gamma_j &= {}^{j}F_j^{T}\,{}^{j}a_j + I_{aj}\,\ddot{q}_j + F_{sj}\,\mathrm{sign}(\dot{q}_j) + F_{vj}\,\dot{q}_j
\end{aligned}
\tag{15}
$$

Where $a(k) = j$, F_j is the reaction wrench (forces and moments) of link i on link j, F_{ej} is the wrench (force and moment) exerted by link j on the environment, Ia_j is the gear and rotor inertia of actuator j and F_{sj} , F_{vj} are the coulomb and viscous friction parameters.

The computational cost of this algorithm is linear in 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 and the customized symbolic calculation can be used [9]…[11].

3.3 Computation of the Direct Dynamic Model

Two methods based on Newton-Euler methods can be used to obtain the dynamic model: the first is proposed by Walker and Orin [13], it is based on calculating the A and H matrices, defined in (6), using Newton-Euler inverse dynamic model and to calculate the joint accelerations by (7); the second method is based on a recursive algorithm [14] [16].

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

From (6) and (10) we deduce that $H(q,\dot{q})$ is equal to Γ *if* $\ddot{q} = 0$, and that the ith column of $A(q)$ is equal to $\mathbf{\Gamma}$ if: $\ddot{q} = u_i, \dot{q} = 0, g = 0, F_{ej} = 0$ **,** where u_i is the (nx1) unit vector whose ith element is equal to 1, and the other elements are equal to zero.

3.3.2 Recursive NE Computation of the Direct Dynamic Model

Using (13) and (15) the equilibrium equations of link j can be written as:

$$ {}^{\mathrm{j}}J_{\mathrm{j}}\,{}^{\mathrm{j}}\dot{V}_{\mathrm{j}} = {}^{\mathrm{j}}F_{\mathrm{j}} + {}^{\mathrm{j}}\beta_{\mathrm{j}} - \sum_{k} {}^{\mathrm{k}}T_{\mathrm{j}}^{\mathrm{T}}\,{}^{\mathrm{k}}F_{\mathrm{k}} \tag{16} $$

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

$$ {}^{j}\beta_{j} = -{}^{j}F_{ej} - \begin{bmatrix} {}^{j}\omega_{j}\times\left({}^{j}\omega_{j}\times {}^{j}MS_{j}\right) \\ {}^{j}\omega_{j}\times\left({}^{j}Io_{j}\,{}^{j}\omega_{j}\right) \end{bmatrix} \tag{17} $$

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 ${}^{j}T_{i}$, the link angular velocities ${}^{j}\omega_{j}$ and ${}^{j}\gamma_{j}, {}^{j}\beta_{j}$ vectors, which appear in the link accelerations equation (12), and the link equilibrium equation (17);

$$ {}^{j}\gamma_{j} = \begin{bmatrix} {}^{j}R_{i}\left[{}^{i}\omega_{i}\times\left({}^{i}\omega_{i}\times {}^{i}P_{j}\right)\right] + 2\sigma_{j}({}^{j}\omega_{i}\times\dot{q}_{j}\,{}^{j}z_{j}) \\ \bar{\sigma}_{j}\,{}^{j}\omega_{i}\times\dot{q}_{j}\,{}^{j}z_{j} \end{bmatrix} \tag{18} $$

ii) backward recursive computation [6]: in this step we calculate the elements ${}^{j}H_{j}$, ${}^{j}J_{j}$, ${}^{j}\beta_{j}$, ${}^{j}K_{j}$, ${}^{j}\alpha_{j}$ which express $\ddot{q}_{j}$ and ${}^{i}F_{j}$ in terms of ${}^{i}\dot{V}_{i}$ in the third recursive equations. Thus for j = n, ... ,1 compute:

$$ \begin{aligned} H_{j} &= ({}^{j}a_{j}\,{}^{j}J_{j}^{*}\,{}^{j}a_{j} + Ia_{j}) \\ {}^{j}K_{j} &= {}^{j}J_{j}^{*} - {}^{j}J_{j}^{*}\,{}^{j}a_{j}\,H_{j}^{-1}\,{}^{j}a_{j}^{T}\,{}^{j}J_{j}^{*} \\ {}^{j}\alpha_{j} &= {}^{j}K_{j}\,{}^{j}\gamma_{j} + {}^{j}J_{j}^{*}\,{}^{j}a_{j}\,H_{j}^{-1}(\tau_{j} + {}^{j}a_{j}^{T}\,{}^{j}\beta_{j}^{*}) - {}^{j}\beta_{j}^{*} \end{aligned} \tag{19} $$

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

$$ \begin{aligned} {}^{i}\beta_{i}^{*} &= {}^{i}\beta_{i}^{*} - {}^{j}T_{i}^{T}\,{}^{j}\alpha_{j} \\ {}^{i}J_{i}^{*} &= {}^{i}J_{i}^{*} + {}^{j}T_{i}^{T}\,{}^{j}K_{j}\,{}^{j}T_{i} \end{aligned} \tag{20} $$

These equations are initialized by: ${}^{j}J_{j}^{*} = {}^{j}J_{j}$, ${}^{j}\beta_{j}^{*} = {}^{j}\beta_{j}$ for $j=1,\ldots,n$

iii) third recursive equations. Since $({}^{0}\dot{v}_{0} = -g, {}^{0}\dot{\omega}_{0} = 0)$, the third recursive equations give ${}^{j}\dot{V}_{j}$ and ${}^{j}F_{j}$ (if needed) for j = 1… n as follows:

$$
\begin{aligned}
&\ddot{q}_j = H_j^{-1}[-{}^j a_j \, {}^j J_j^* ({}^j T_i \, {}^i \dot{V}_i + {}^j \gamma_j) + \tau_j + {}^j a_j^T \, {}^j \beta_j^*] \\
&{}^j F_j = {}^j K_j \, {}^j T_i \, {}^i \dot{V}_i + {}^j \alpha_j \\
&{}^j \dot{V}_j = {}^j T_i \, {}^i \dot{V}_i + {}^j a_j \, \ddot{q}_j + {}^j \gamma_j \\
&\text{where } \tau_j = \Gamma_j - F_{sj} \, sign(\dot{q}_j) - F_{vj} \, \dot{q}_j
\end{aligned}
\tag{21}
$$

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 Γ_c are obtained by solving the following equation:

$$
\begin{bmatrix} \Gamma_c \\ 0 \end{bmatrix} = \Gamma_{\mathrm{tr}}(q_{tr}, \dot{q}_{tr}, \ddot{q}_{tr}) + \mathrm{W}^{\mathrm{T}} \lambda \tag{22}
$$

Where λ is the vector of Lagrange multipliers and $W \dot{q}_{tr} = 0$ is the kinematics constraint equations between the velocities of the passive and active joints of the tree structure. The matrix W can be obtained from (4).

After developing we obtain:

$$
\Gamma_c = G^{\mathrm{T}} \Gamma_{\mathrm{tr}}(q_{tr}, \dot{q}_{tr}, \ddot{q}_{tr}) = \Gamma_a + \frac{\partial \dot{q}_p}{\partial \dot{q}_a} \Gamma_p \tag{23}
$$

where:

$$
G = \frac{\partial q_{tr}}{\partial q_a} = \frac{\partial \dot{q}_{tr}}{\partial \dot{q}_a} \tag{24}
$$

Γ_a and Γ_p are the actuated and passive joint torques of the tree structure.

From equation (23) we see that the active joints of the closed loop structure is calculated by projecting the tree structure torques Γ_{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) [6],[17].

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 A_c and H_c such that:

$$
\Gamma_c = A_c(q_{tr}) \ddot{q}_a + H_c(q_{tr}, \dot{q}_{tr}) \tag{25}
$$

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 m 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 present the case of parallel robots with six degrees of freedom. Examples concerning reduced mobility robots are given in [18].

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

Decomposing the robot into two subsystems: the platform and the legs. The dynamics of the platform is calculated by Newton-Euler equation as a function of the Cartesian variables (location, velocity and acceleration of the platform), whereas the dynamics of each leg i is calculated as a function of its joint variables $(q_i, \dot{q}_i, \ddot{q}_i)$ for $i=1,\ldots,m$. The active joint torques are obtained by projecting these two dynamics 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, and to project the leg dynamics on the active joint space we use the Jacobian matrix between these two spaces. Thus the dynamic model of the parallel structure is given by the following equation:

$$\Gamma = \mathrm{J}_P^T(\boldsymbol{\Phi}_P + F_{ep}) + \sum_{i=1}^{m} \left(\frac{\partial \dot{q}_i}{\partial \dot{q}_a} \right)^T \Gamma_i \tag{26}$$

Where

$\boldsymbol{\Phi}_P$ is the wrench required to achieve the motion of the platform, calculated by Newton- Euler equation (13) with the platform inertial parameters.

F_{ep} external forces on the platform.

J_P is the (6×6) kinematic Jacobian matrix of the robot, which gives the platform screw V_P (translational and angular velocities) as a function of the active joint velocities:

$$V_P = \mathrm{J}_P \dot{q}_a \tag{27}$$

Γ_i is the inverse dynamic model of leg i, it is a function of $(q_i, \dot{q}_i, \ddot{q}_i)$, which can be obtained in terms of the platform location, velocity and acceleration, using the inverse kinematic models of the legs [19]. We note that q_i does not include the passive joint variables connecting the legs to the platform.

The calculation of J_p is obtained by inverting $\mathrm{J}_\mathrm{p}^{-1}$, which is easy to obtain for most parallel structures. The detailed calculation of $\partial \dot{q}_i / \partial \dot{q}_a$ is given in [18].

6 Inverse Dynamic Modeling of Robots with Elastic Joints

A tree structure robots with lumped elastic joints can be described using the method presented in section 2. Each joint could be elastic or rigid.

6.1 Lagrange Dynamic Form

The general form of the dynamic model of a system with flexible joints is the same as (6), it can be partitioned as follows:

$$\Gamma = \begin{bmatrix} \Gamma_r \\ \Gamma_f \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} H_r(q,\dot{q}) \\ H_f(q,\dot{q}) \end{bmatrix} \tag{28}$$

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

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

$A(q)$ is the (n×n) inertia matrix of the system,

Γ_r is the vector of rigid joint torques, Γ_f is the vector of elastic joint torques.

If joint j is flexible:

$$\Gamma_j = -\Delta q_j \, k_j = -(q_j - q_{rj}) \, k_j \tag{29}$$

where k_j is the stiffness of the elastic joint, Δq_j is the elastic deformation, q_{rj} is the reference joint position corresponding to zero elastic deformation.

The inverse dynamic model calculates the input torques and the elastic accelerations as a function of the joint positions, velocities and rigid joint accelerations. The definition of the direct model definition is similar to that of the rigid robots; it gives the joint accelerations in terms of the input torques.

Eq. (28) can be used to calculate the inverse model, we have first to calculate the elastic accelerations from the second row, and then we 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. We note that the recursive algorithms of sections 6.2 and 6.3 can be used in case of passive joints without actuators by putting the torque of these joints equal to 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. But 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 section 6.1 for the calculation of $\ddot{q}_f$ and Γ_r.

To solve this problem we use three recursive steps [20].

i) The first forward step is 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 $\ddot{q}_e$ and ${}^{j}F_j$ in terms of ${}^{i}\dot{V}_i$. These matrices can be calculated for $j = n...1$ as follows:

- if joint j is elastic:

$$\begin{aligned} H_j &= {}^{j}a_j \, {}^{j}J_j^* \, {}^{j}a_j \\ {}^{j}K_j &= {}^{j}J_j^* - {}^{j}J_j^* \, {}^{j}a_j \, H_j^{-1} \, {}^{j}a_j^T \, {}^{j}J_j^* \\ {}^{j}\alpha_j &= {}^{j}K_j \, {}^{j}\gamma_j + {}^{j}J_j^* \, {}^{j}a_j \, H_j^{-1} (-k_j \Delta q_j + {}^{j}a_j^T \, {}^{j}\beta_j^*) - {}^{j}\beta_j^* \end{aligned} \tag{30}$$

- If joint j is rigid:

$$\begin{aligned} {}^{j}K_j &= {}^{j}J_j^* \\ {}^{j}\alpha_j &= {}^{j}K_j ({}^{j}\gamma_j + {}^{j}a_j \, \ddot{q}_j) - {}^{j}\beta_j^* \end{aligned} \tag{31}$$

if $a(j) \neq 0$, calculate:

$$\begin{aligned} {}^{i}\beta_i^* &= {}^{i}\beta_i^* - {}^{j}T_i^T \, {}^{j}\alpha_j \\ {}^{i}J_i^* &= {}^{i}J_i^* + {}^{j}T_i^T \, {}^{j}K_j \, {}^{j}T_i \end{aligned} \tag{32}$$

The previous equations are initialized by: ${}^{j}J_j^* = {}^{j}J_j$, ${}^{j}\beta_j^* = {}^{j}\beta_j$.

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 equations:

$${}^{j}F_j = {}^{j}K_j \, {}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}\alpha_j \tag{33}$$

- if j is elastic:

$$\begin{aligned} \ddot{q}_j &= H_j^{-1} [-{}^{j}a_j \, {}^{j}J_j^* ({}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}\gamma_j) - \text{kj} \, \Delta \, \text{qj} + {}^{j}a_j^T \, {}^{j}\beta_j^*] \\ {}^{j}\dot{V}_j &= {}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}a_j \, \ddot{q}_j + {}^{j}\gamma_j \end{aligned} \tag{34}$$

- if j is rigid

$$\Gamma_j = {}^{\text{j}}F_{\text{j}}^T \, {}^{\text{j}}a_{\text{j}} + I_{aj} \, \ddot{q}_j + F_{sj} \, sign(\dot{q}_j) + F_{vj} \, \dot{q}_j \tag{35}$$

7 Dynamic Modeling of Robots with a Mobile Base

This section treats mobile robots which are composed of a tree structure with a moving base. It includes a big number of systems such as: cars, mobile manipulators, walking robots, Humanoid robots, eel like robots [21], snakes like robots, flying

robots, spatial vehicles, 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 acceleration of the base is known as equal to zero, whereas in the case of moving base the acceleration of the base must be determined in both direct and inverse dynamic models. 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 ${}^{w}h_0$. 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 V_0 and $\dot{V}_0$ respectively.

7.1 General Form of the Dynamic Model

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

$$\begin{bmatrix} 0_{6x1} \\ \Gamma \end{bmatrix} = A \begin{bmatrix} {}^{0}\dot{V}_0 \\ \ddot{q} \end{bmatrix} + H = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} \begin{bmatrix} {}^{0}\dot{V}_0 \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \tag{36}$$

Γ $(n \times 1)$ vector of joint torques, q $(n \times 1)$ vector of joint positions,

A is the $(6+n) \times (6+n)$ inertia matrix of the robot,

H is the $(n+6) \times 1$ vector representing the Coriolis, centrifugal, gravity and external forces effect on the robot.

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. At first, the inverse dynamic model is solved by using the first row of (36) to obtain the base acceleration:

$${}^{0}\dot{V}_0 = -(A_{11})^{-1} (H_1 + A_{12}\ddot{q}) \tag{37}$$

Then the second row of (36), can be used to find the joint torques:

$$\Gamma = A_{12}^T \, {}^{0}\dot{V}_0 + A_{22}\, \ddot{q} + H_2 \tag{38}$$

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 (36), the direct dynamic model is solved as follows:

$$\begin{bmatrix} \dot{V}_0 \\ \ddot{q} \end{bmatrix} = A^{-1} \begin{bmatrix} -H_1 \\ \Gamma - H_2 \end{bmatrix} \tag{39}$$

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 mobile base can be taken into account as follows:

- the initial values of velocities and accelerations are calculated from those of the base as follows: ${}^0V_0 = {}^0V_b$, ${}^0\dot{v}_0 = {}^0\dot{v}_0 - g$, ${}^0\dot{\omega}_0 = {}^0\dot{\omega}_b$, where the subscript b indicates the base.
- The backward recursive equations must continue to $j = 0$, where this new iteration will obtain the 6 equations of Newton-Euler equations of the base.

Solving the inverse and direct dynamic problems using A and H is 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 in the number of degrees of freedom.

7.2 Recursive NE Calculation of the Inverse Dynamic Model

The inverse dynamic algorithm in this case consists of three recursive equations (a forward, then a backward, then a forward) [21].

i) Forward recursive calculation:

In this step we calculate the screw transformation matrices, the link rotational velocities, and the elements of the accelerations and external wrenches on the links which are independent of the acceleration of the robot base ($\dot{v}_0, \dot{\omega}_0$). Thus we calculate for $j = 1,...,n$ the following: ${}^{j}T_{i}$, ${}^{j}\omega_j$ as given in section (3.2). We calculate also ${}^{j}\beta_{j}, {}^{j}\gamma_{j}$ using (17) and (18) they represent the elements of total forces and acceleration which are independent of the base acceleration. We define also:

$$ {}^{j}\zeta_{j} = {}^{j}\gamma_{j} + \ddot{q}_j \, {}^{j}a_j \tag{40} $$

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 (17), giving the equilibrium equation of link j, can be rewritten as:

$$ {}^{j}F_{j} = {}^{j}J_{j} \, {}^{j}\dot{V}_{j} - {}^{j}\beta_{j} + \sum_{k} {}^{k}T_{j}^{T} \, {}^{k}F_{k} \tag{41} $$

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

$$ {}^{j}F_j = {}^{j}J_j \, {}^{j}\dot{V}_j - {}^{j}\beta_j + \sum_{s(j)} {}^{s(j)}T_j^T \left({}^{s(j)}J_{s(j)} \, {}^{s(j)}\dot{V}_{s(j)} - {}^{s(j)}\beta_{s(j)} \right) \tag{42} $$

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

Substituting for ${}^{s(j)}\dot{V}_{s(j)}$ in terms of ${}^{j}\dot{V}_{j}$ using (12), we obtain:

$$ {}^{s(j)}\dot{V}_{s(j)} = {}^{s(j)}T_j \, {}^{j}\dot{V}_j + \sum_r {}^{s(j)}T_r \, {}^{r}\zeta_r \tag{43} $$

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

Thus we obtain:

$$ {}^{j}F_j = {}^{j}J_j^c \, {}^{j}\dot{V}_j - {}^{j}\beta_j^c \tag{44} $$

with:

$$ \begin{aligned} {}^{j}J_j^c &= {}^{j}J_j^c + \sum_k {}^{k}T_j^T \, {}^{k}J_k^c \, {}^{k}T_j \\ {}^{j}\beta_j^c &= {}^{j}\beta_j^c - \sum_k {}^{k}T_j^T \, {}^{k}\beta_k^c + {}^{k}T_j^T \, {}^{k}J_k^c \, {}^{k}\zeta_k \end{aligned} \tag{45} $$

${}^{j}J_j^c$ is the inertial matrix of the composite link j.

For j = 0, and since ${}^{0}F_0$ is equal to zero, we obtain using (44):

$$ {}^{0}\dot{V}_0 = \left({}^{0}J_0^c \right)^{-1} {}^{0}\beta_0^c \tag{46} $$

To conclude, the recursive equations of this step consist of initializing ${}^{j}J_j^c = {}^{j}J_j$, ${}^{j}\beta_j^c = {}^{j}\beta_j$ and then calculating (45) *for j = n,…, 0.* At the end ${}^{0}\dot{V}_0$ is calculated by (46).

iii) Forward recursive equations:

After calculating ${}^{0}\dot{V}_0$, the wrench ${}^{j}F_j$ and the joint torques are obtained using equations (12) and (45) for j= 1,…, n as:

$$ \begin{aligned} {}^{j}\dot{V}_j &= {}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}\zeta_j \\ {}^{j}F_j &= {}^{j}J_j^c \, {}^{j}\dot{V}_j - {}^{j}\beta_j^c \end{aligned} \tag{47} $$

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

$$ \Gamma_j = {}^{j}F_j^T \, {}^{j}a_j + I_{aj} \, \ddot{q}_j + F_{sj} \, sign(\dot{q}_j) + F_{vj} \, \dot{q}_j \tag{48} $$

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 forward, backward and forward:

i) Forward recursive equations:

We calculate for *j = 1, ..., n* the link rotational velocities using (11) and the terms ${}^{j}\gamma_{j}$ and ${}^{j}\beta_{j}$ of Cartesian accelerations and equilibrium equations of the links that are independent of the accelerations of the base and the joints.

ii) Backward recursive equations:

In this second step, we initialize ${}^{j}J_{j}^{*} = {}^{j}J_{j}$, ${}^{j}\beta_{j}^{*} = {}^{j}\beta_{j}$ and then we calculate for j = n,…,1 the following elements, which permit to calculate ${}^{j}F_{j}$ and $\ddot{q}_j$ in terms of ${}^{i}\dot{V}_i$ and will be used in the third recursive equations:

$$
\begin{aligned}
H_j &= {}^{j}a_j^T \, {}^{j}J_j^* \, {}^{j}a_j + Ia_j \\
{}^{j}K_j &= {}^{j}J_j^* - {}^{j}J_j^* \, {}^{j}a_j H_j^{-1} \, {}^{j}a_j^T \, {}^{j}J_j^* \\
{}^{i}J_i^* &= {}^{i}J_i^* + {}^{j}T_i^T \, {}^{j}K_j \, {}^{j}T_i \\
{}^{j}\alpha_j &= {}^{j}K_j \, {}^{j}\gamma_j + {}^{j}J_j^* \, {}^{j}a_j H_j^{-1} \left(\Gamma_j + {}^{j}a_j^T \, {}^{j}\beta_j^* \right) - {}^{j}\beta_j^* \\
{}^{i}\beta_i^* &= {}^{i}\beta_i^* - {}^{j}T_i^T \, {}^{j}\alpha_j
\end{aligned}
\tag{49}
$$

iii) Forward recursive equations:

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

$$
{}^{0}\dot{V}_0 = \left({}^{0}J_0^* \right)^{-1} \, {}^{0}\beta_0^* \tag{50}
$$

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

$$
\begin{aligned}
\ddot{q}_j &= H_j^{-1} \left[- {}^{j}a_j^T \, {}^{j}J_j^* \left({}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}\gamma_j \right) + \tau_j + {}^{j}a_j^T \, {}^{j}\beta_j^* \right] \\
{}^{j}F_j &= {}^{j}K_j \, {}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}\alpha_j
\end{aligned}
\tag{51}
$$

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

$$
{}^{j}\dot{V}_j = {}^{j}T_i \, {}^{i}\dot{V}_i + {}^{j}a_j \ddot{q}_j + {}^{j}\gamma_j \tag{52}
$$

8 Dynamic Modeling of Wheeled Mobile Robots

The robot is composed of a cart and classical wheels. The wheels could be fixed, steering or casters. Each of these wheels has a rotational variable φ_j. The steering and castor wheels have also an orientation variable β_j. The vector of configuration variables is composed of the wheel variables φ_j, β_j and the posture variables ζ giving the position and orientation of the cart. To simplify the presentation we consider that the motion of the cart lies in a plane thus the posture is composed of the x,y position coordinates and the orientation is the angle around the z axis. For more details on these robots the reader can consult [22]. These mobile robots contain non-holonomic constraint equations between the configuration velocities and have some joints which are not motorized. The constraint equations can be written as:

$$W\dot{q} = 0, and\ \dot{q} = S\dot{q}_m \tag{53}$$

Where $q = \begin{bmatrix} \zeta^T & \beta^T & \varphi^T \end{bmatrix}^T$, q_m is the motorized joint variables among β and φ.

The difference with respect to the previous sections is that the coupling between the configuration variables is kinematically defined.

In this case we apply the procedure of closed loop structure of section 4 such that:

$$\begin{bmatrix} \Gamma_m \\ 0 \end{bmatrix} = \Gamma(q,\dot{q},\ddot{q}) + \mathrm{W}^{\mathrm{T}}\lambda$$
$$\Gamma_m = \left(\frac{\partial \dot{q}}{\partial \dot{q}_m} \right)^T \Gamma = S^T \Gamma \tag{54}$$

Γ is the torques corresponding to the configuration variables without taking into account the constraints, it can be calculated using the method of section 3.2, Γ_m correspond to the real motor torques.

9 Conclusions

This paper presents the inverse and direct dynamic modeling of different types of robots. 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 rigid manipulators to the other cases.

The proposed methods have been applied on more complicated systems such as: flexible link robots [23], Micro Continuous system [24] and Hybrid structure, where the robot is composed of parallel modules, which are connected in series [25].

References

1. Khalil, W., Kleinfinger, J.-F.: A new geometric notation for open and closed-loop robots. In: Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, pp. 1174–1180. IEEE Press, New York (1986)
2. Gautier, M., Khalil, W.: Direct calculation of minimum set of inertial parameters of serial robots. IEEE Trans. on Robotics and Automation RA-6(3), 368–373 (1990)
3. Khalil, W., Bennis, F.: Comments on Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. IEEE Trans. on Rob. & Automation RA-10(1), 78–79 (1994)
4. Khalil, W., Bennis, F.: Symbolic calculation of the base inertial parameters of closed-loop robots. International Journal of Robotics Research 14(2), 112–128 (1995)
5. Craig, J.J.: Introduction to robotics: mechanics and control. Addison Wesley Publishing Company, Reading (1986)
6. Khalil, W., Dombre, E.: Modeling identification and control of robots. Hermes, Penton-Sciences, London (2002)
7. Angeles, J.: Fundamentals of Robotic Mechanical Systems, 2nd edn. Springer, New York (2002)
8. Luh, J.Y.S., Walker, M.W., Paul, R.C.P.: On-line computational scheme for mechanical manipulators. Trans. of ASME, J. of Dynamic Systems, Measurement, and Control 102(2), 69–76 (1980)
9. Khosla, P.K.: Real-time control and identification of direct drive manipulators. Ph. D. Thesis, Carnegie Mellon (1986)
10. Khalil, W., Kleinfinger, J.-F.: Minimum operations and minimum parameters of the dynamic model of tree structure robots. IEEE J. of Robotics and Automation RA-3(6), 517–526 (1987)
11. Khalil, W., Creusot, D.: SYMORO+: a system for the symbolic modelling of robots. Robotica 15, 153–161 (1997)
12. Hollerbach, J.M.: An iterative lagrangian formulation of manipulators dynamics and a comparative study of dynamics formulation complexity. IEEE Trans. on Systems, Man, and Cybernetics SMC-10(11), 730–736
13. Walker, M.W., Orin, D.E.: Efficient dynamic computer simulation of robotics mechanism. Trans. of ASME, J. of Dynamic Systems, Measurement, and Control 104, 205–211 (1982)
14. Armstrong, W.W.: Recursive solution to the equation of motion of an N-links manipulator. In: Proc. 5th World Congress on Theory of Machines and Mechanisms, pp. 1343–1346
15. Brandl, H., Johanni, R., Otter, M.: 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, pp. 365–370 (1986)
16. Featherstone, R.: The calculation of robot dynamics using articulated-body inertias. The Int. J. of Robotics Research 2(3), 87–101 (1983)
17. Nakamura, Y., Ghodoussi, M.: Dynamic computation of closed-link robot mechanism with nonredundant and redundant actuators. IEEE Trans. Robotics and Automation (5), 294–302 (1989)
18. Khalil, W., Ibrahim, O.: General solution for the Dynamic modeling of parallel robots. Journal of Intelligent and Robotic Systems 49, 19–37 (2007)
19. Khalil, W., Guegan, S.: Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. IEEE Transactions on Robotics and Automation 20(4), 754–762 (2004)
20. Khalil, W., Gautier, M.: Modeling of mechanical systems with lumped elasticity. In: Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, pp. 3965–3970 (2000)

21. Khalil, W., Gallot, G., Boyer, F.: Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot. IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and reviews 37(6), 1259–1268 (2007)
22. Campion, G., Chung, W.: Wheeled Robots. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, pp. 391–410. Springer, Heidelberg (2008)
23. Boyer, F., Khalil, W.: An efficient calculation of flexible manipulator inverse dynamic. Int. Journal of Robotics Research 17(3), 282–293 (1998)
24. Boyer, F., Porez, M., Khalil, W.: Macro-continuous torque algorithm for a three-dimensional eel-like robot. IEEE Robotics Transaction 22(4), 763–775 (2006)
25. Ibrahim, O., Khalil, W.: Inverse and direct dynamic models of Hybride robots. Mechanism and Machine Theory 45(4), 627–640 (2010)

Emotive Driver Advisor System (EDAS)

Oleg Gusikhin, Erica Klampfl, Dimitar Filev, and Yifan Chen

Ford Research & Advanced Engineering
RIC Bldg, 2101 Village Rd., Dearborn, MI 48124, U.S.A.
{ogusikhi,eklampfl,dfilev,ychen1}@ford.com

Abstract. This paper describes the Emotive Driver Advisory System (EDAS), Ford Research & Advanced Engineering's project on next generation driver assistance. EDAS integrates several emerging technologies, focusing on personalization and adaptive and intelligent behavior. We will provide a high-level overview of the EDAS architecture, focusing on novel consumer-facing features, such as the emotive spoken dialogue system and an Avatar as an automotive human machine interface. Furthermore, we will discuss the benefits of cloud-based vehicle infotainment and decision support and how this can be integrated in a vehicle environment. The system concept was revealed at the 2009 Consumer Electronics Show and the 2009 North American International Auto Show.

Keywords: Driver aware vehicle, Computational intelligence, Spoken dialogue system, Embodied conversational agents, Affective computing, Cloud-based infotainment.

1 Introduction

This paper introduces a research project, the Emotive Driver Advisory System (EDAS), which explores next-generation driver/vehicle interaction with a focus on intelligent and emotive behavior. The goal of EDAS is to leverage recent advances in affective computing, embodied conversational agents, intelligent vehicle controls, and internet personalized recommendation services to create a new generation of driver assistance systems: we want provide an enhanced level of personalization and customization leading to an emotive bond between the driver and vehicle.

In recent years, vehicle telematics and connectivity have started to play a very important role for automotive brand differentiation and competitive advantage. In 2007, Ford, in collaboration with Microsoft, introduced an in-car communication and entertainment system, SYNC [1], which enables Bluetooth and USB connectivity for consumer phones and MP3 players and allows hands-free voice-activated control of brought-in devices. Leveraging consumer phone connectivity, subsequent SYNC versions encompass a set of cloud-based information services including vehicle health reports, traffic, turn-by-turn directions, weather, and news. The latest SYNC release enables seamless voice/button interface and control of smart phone mobile applications.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 21–36.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

To date, SYNC has focused on delivering functionality and usability. According to Maslow's hierarchy, consumers' needs from a product can be characterized using a three-level hierarchy from functionality and usability to pleasure [2]. Products that invoke a positive emotive response have a competitive edge in the marketplace. Classic examples are Harley-Davidson motorcycles or IPod MP3 players. As a result, efforts to incorporate emotive aspects in functionality and services are actively being pursued to enhance product positioning.

There are currently two main engineering disciplines that address human emotions: Kansei Engineering and affective computing. The design discipline to incorporate a user's emotive response in a product, Kansei engineering, was initiated in Japan in the 1970s: it is also referred to as "emotive" or "affective" design [3]. Affective computing is an emerging area that explores methods and tools to enable an HMI to recognize and understand human emotions, as well as the ability to express emotions [3]. Affective computing is based on recent technological advances in artificial intelligence, biomechanical sensing, computer vision, and speech recognition [4]. The rise of affective computing is closely tied to the recent development and popularity of humanoid robots. In 1995, SONY launched an entertainment robot called "AIBO," which has, among other features, the ability to recognize certain situations and to respond with pre-programmed emotions [5], [6], [7], and [8]. Affective computing can complement Kansei engineering, as it provides the methods and tools to directly capture emotive response from the user.

Automotive companies are displaying a heightened interest in bringing emotions into the vehicle: this is because in addition to aesthetic value, emotions can directly impact driving safety and performance. For example, Toyota developed a concept vehicle call the POD in 2001, which can detect driver's emotion and express the driver's or its own feelings using multi-color light patterns on the front facial panel of the car [9]. Nissan developed a PIVO robot in 2005 and a PIVO 2 in 2007 that reside in concept vehicles: the robot is a dashboard talking buddy that can chat with the driver, provide various information services, and express emotions [10] [11].

The major motivation of the EDAS work described in this paper and the goal of our system was to bring the vehicle interface to a completely new level: from simple voice control to a personal assistant that can understand the driver and driving environment. This work was a research and development effort that falls into the broad scope of affective computing, intelligent control, and machine intelligence but with the focus on an automobile as the product platform. We demonstrate this in a functional prototype vehicle that offers the driver the experience of riding inside a car that is connected, intelligent, can understand your natural speech, talk back naturally, recognize and express emotions, and can offer personalized driving advice and assistance.

The rest of the paper is devoted to the major elements of the EDAS system and its implementation. Specifically, Section 2 provides an overview of the EDAS architecture, Section 3 discusses how we use an Avatar as an advanced HMI, Section 4 describes the emotive spoken dialogue system, and Section 5 provides an overview of cloud-based infotainment. The paper concludes with a summary of the EDAS system and its impact.

2 EDAS Architecture

The EDAS architecture (see Fig. 1) is designed to facilitate personalization of the interaction between the driver and vehicle through customization and context-dependent learning and supports a mixed-initiative multi-task environment. The main elements of the EDAS system are an Intelligent Vehicle Agent, an emotive Spoken Dialogue System (SDS) supplemented by an Avatar, a Message Dispatcher, and a Task Manager.

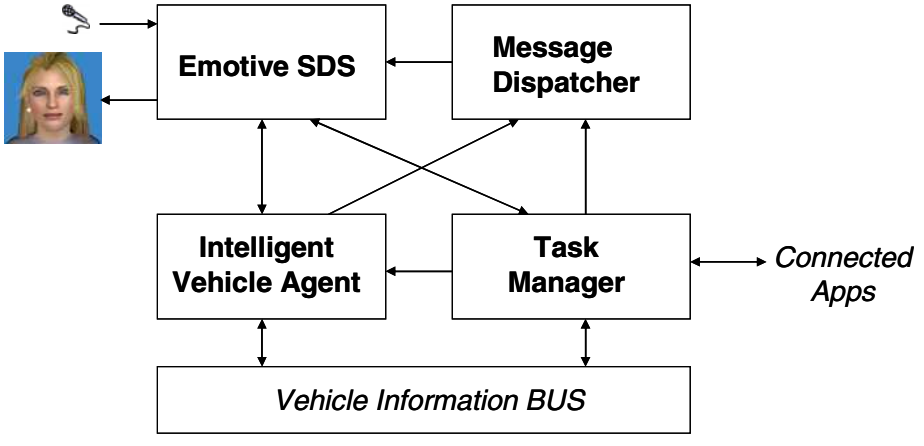


Fig. 1. Diagram of EDAS architecture.

The Intelligent Vehicle Agent aggregates and interprets vehicle data into driving context parameters and implements algorithms for perception and learning driver's actions and making intelligent decisions. The driving context includes parameters such as driving style (e.g. Sporty, Normal, or Relaxed) and workload (e.g. high, low). The driving style is estimated in real time by classifying the accelerator and brake pedal patterns ([12] and [13]) and the speed of the vehicle. The workload is estimated by a set of algorithms that continually evaluates the driving demand based on the driver control activity, traffic density, and road conditions [14]. The learning mechanism of the intelligent vehicle control system estimates the preferences of the driver under different situations and states (e.g. performance, fuel economy, and comfort) and summarizes them in families of models that characterize the driver's actions under different conditions. The driver's models are developed over time and are dependent on the environment, driver actions, and the operation of the vehicle. The models are later used to "invert" the mapped relationships and advise the driver on the most appropriate actions under specific circumstances. They can provide a recommendation for the optimal maximal speed under specific conditions or modes of operation (i.e., comfort vs. sporty, performance vs. improved fuel economy).

The SDS manages the communication between the driver and the vehicle. The importance of an in-vehicle speech interface is related to requirements for non-destructive hands-free control due to the ever increasing number of auxiliary

functions offered in vehicles, such as telephones, entertainment, navigation, and climate control systems. Most of these voice interface systems available today are based on a single utterance command-and-control paradigm. Such systems typically require memorization of all commands from a manual to be effective, not allowing natural language expression. Moreover, they require the user to interact in a hierarchical, menu-driven fashion, which does not allow simultaneous processing of tasks. To address these limitations, automotive companies and suppliers have been actively pursuing research and development of next generation, in-vehicle intelligent dialogue systems [15]. The EDAS SDS provides a mixed-initiative multimodal SDS that allows for many different types of input from speech and button presses to external barge-in messages. Our SDS is further enhanced by including emotion recognition and emotive text-to-speech. The EDAS emotive SDS is presented in more detail later in the paper.

The EDAS SDS is further supplemented by an Embodied Conversational Agent (ECA), or Avatar, as a point of reference for vehicle intelligence. The EDAS ECA is a three dimensional digital cockpit assistant that can provide information and offer driving assistance through human-like interaction, including verbal (e.g. conversation) or non-verbal (e.g. expressions used to represent the status of active tasks) communication. Our ECA offers several conduits for customization, including the Avatar's appearance, voice, and behavior (e.g. personality).

The Message Dispatcher queues messages that arrive asynchronously from vehicle subsystems or external sources and forwards them to the SDS based on the current status of the system, priority of the message, and driving environment. Safety related or urgent navigation messages have higher priority and will be rendered to the SDS immediately. Informational messages, whether from the vehicle or from external sources, will be sent to the SDS only when the system is in idle mode and when the driver's workload is not high.

The Task Manager provides an interface between the SDS and in-vehicle subsystems and/or connected applications from brought-in devices or from the cloud. It is responsible for translating and routing requests to the appropriate application, maintaining the status of the active request, receiving the response or proactive notification from the application, and transferring the message back to the dispatcher.

In the rest of the paper, we focus on the following novel consumer-facing elements: Avatar, emotive SDS, and cloud-based infotainment. Other EDAS components are outside the scope of this paper and are more appropriate for an in-depth discussion within their respective fields, such as Intelligent Controls or Software Engineering.

3 Avatar as an Advanced HMI

There is a growing body of research that demonstrates the benefits of utilizing ECAs and specifically animated humanoid Avatars in human-machine interfaces [16]. Using Avatars has recently been recently entertained within automotive applications. For instance, Volkswagen implemented an Avatar as a digital assistant that can give animated narratives of various functions/features of the vehicle’s instructional panel, such as its built-in navigation and climate control systems [17]. Mercedes-Benz developed a similar application employing an Avatar to give the driver valuable vehicle information [17].

The EDAS ECA is a three dimensional (3D) Avatar that can be efficiently implemented in a cockpit environment (through an LCD display) and can accommodate legal and/or safety requirements: for example, the Avatar could be frozen or removed entirely from the screen when the vehicle is in motion. A 3D Avatar allows for realistic rendering using the latest computer graphics technologies and modern GPU hardware shader capability.

There are several Avatar design options including a human, a cartoon character, an animal, or an object (see Fig. 2 for an example human and cartoon Avatar). Other options are to animate facial expressions only or to include body motions.

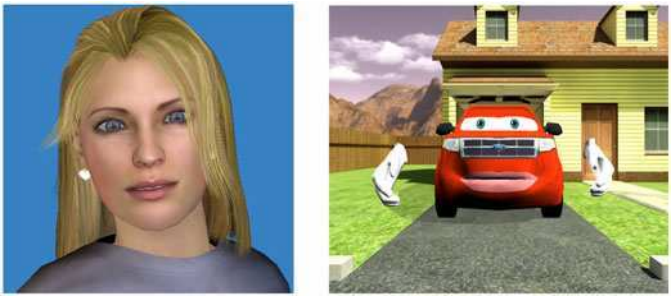


Fig. 2. a) Human Avatar, Jackie; b) Cartoon Avatar, Toby.

To gain insight into human interaction with computer-generated characters in a driving environment, we conducted a customer clinic to understand the following user preferences:

- Are life-like or cartoon-like Avatars preferred?
- Is a partial representation (head, neck and shoulder) Avatar sufficient?
- Are human characters or animal/objects preferred?

Thirty-five subjects participated in our survey: most were Ford Motor Company employees located in Southwest Michigan, USA, approximately 38% of the subjects were under 40 years old, and three quarters of the subjects were male. Subjects were given a brief introduction of the EDAS concept and the role of an Avatar in the cockpit. They were also shown some video clips of the early EDAS prototype system. We then asked them to answer a number of questions concerning their Avatar preferences.

Fig. 3 provides a summary of the major results. We can see that the survey confirms that the human character is the most preferred Avatar form, a partial representation is sufficient for most people, and most subjects prefer a life-like Avatar to a cartoon-like one. As a result, we adopted a partial (head and shoulders), life-like human character as the EDAS Avatar.

An important consideration in the Avatar design is a phenomenon called the "uncanny valley" [18]. The Japanese roboticist, Masahiro Mori, suggested in his 1970 paper that a human's positive emotional reaction to lifelike robots will suffer a setback (the uncanny valley) when the lifelikeness of the robot approaches a certain level and

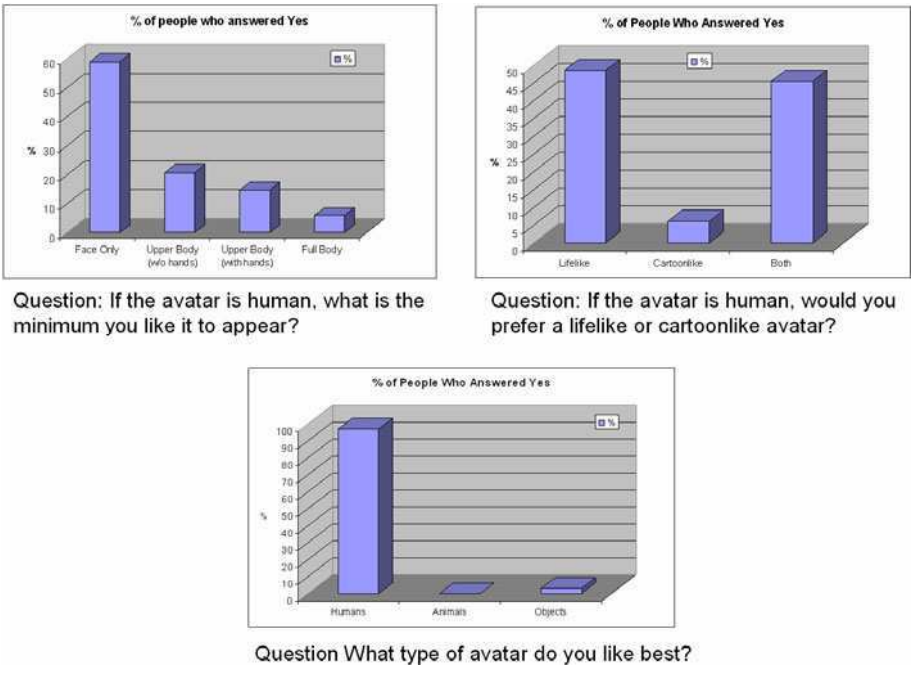


Fig. 3. Avatar survey results.

will continue to do so until, eventually, the robot and real-human become undistinguishable [19]. The uncanny valley hypothesis has since been extended to computer-animated realistic 3D characters widely used in video gaming software.

The development of our Avatar was guided by the uncanny valley principle, as well as the requirement to balance realism and computational limitations of an embedded cockpit environment. Fig. 2 shows our EDAS Avatar, Jackie, and our cartoon Avatar, Toby, that were created in collaboration with Zabaware [20] and implemented using a commercial animation package Haptek [21].

Fig. 4. Several facial emotions of Jackie: listening, thinking/searching, and confused.

As a result of our customer clinic, the uncanny valley principle, and computational requirements, we focused on Jackie Avatar. The Avatar model automatically performs lip syncing and allows for animation of facial emotions through a set of customizable state values. These values can be set in real-time in response to system needs, such as the emotive tag sent by the SDS. Fig. 4 shows facial expressions corresponding to different states of the system, such as listening, searching, and confusion in the case of low confidence speech recognition.

4 Emotive Spoken Dialogue System

A good SDS is the key to a driver's positive in-vehicle communication experience. If the driver cannot communicate his/her wants naturally and with low cognitive demand, this could have serious in-vehicle safety implications. Additionally, low quality recognition hinders the driver's emotive and practical experience and can lead to frustration. In this section, we will provide an overview of the EDAS SDS.

The EDAS SDS is based on four main principles: 1) "open-mic" technology where the system is continually listening; 2) a non-hierarchical interface so that the user does not need to go through a multi-level menu structure; 3) a natural language speech engine so that the user can speak naturally to the system and not have to remember the right words in order to effectively command; and 4) a multimodal and mixed initiative framework that allows other inputs equivalent to speech, such as a button press or proactive massages from external sources. We developed the EDAS SDS using a similar framework as the one used for the International Space Station [22] and [23], which uses the open source Regulus development environment [24] and [25].

The EDAS SDS architecture is detailed in Fig. 5. The system is multimodal, and so we take inputs such as speech, button presses, or external messages filtered through dispatcher. In the case of speech, we utilize an open mic speech identifier that is continually listening. This continual listening is activated by either a key word or an equivalent button press. After this is initiated, the system will be continually listening until the system is deactivated by a command word, a specific button command, or a system time-out. Speech, as a .wav file, is then simultaneously passed to a speech recognizer (e.g., Nuance [26]) and the emotion recognizer (e.g., Affective Media [27]). The speech recognizer has acoustic models that convert what is heard to a logical form, which is then passed to the input manager. For emotion recognition, we used a voice-based recognizer from AffectiveMedia, Ltd. to processes the emotion detected in the .wav file. The recognizer analyzes acoustic features of the speech, such as pitch, volume, and rate and classifies them into different emotions [28]. The EDAS emotion recognizer was trained to classify boredom, happiness, surprise, hot anger, and sadness. We combined boredom and sadness into a "bad" mood and happiness and surprise into a "good" mood by assigning a numerical value that was mapped to the correct mood: this value is then used in SDS output manager to create an appropriate emotive response.

The input manager takes the logical form and the current information state and creates a dialogue move: this provides a representation that can be understood by the dialogue manager. The dialogue manager receives the dialogue move from the input manager and the information state and creates an abstract action and a new information state: this new information state is passed back. The abstract action is then passed

to the output manager, which converts the abstract action to a concrete action, also taking into consideration the emotion. For example, if the driver asked, "What is the time?", then the abstract action might be something that is passed as (time, 8:30 pm) to the output manager. The concrete action created by the output manager would combine the abstract action with information on the personality, driving context, aggressiveness, and workload to create an emotive Text to Speech (TTS) and corresponding Avatar facial response.

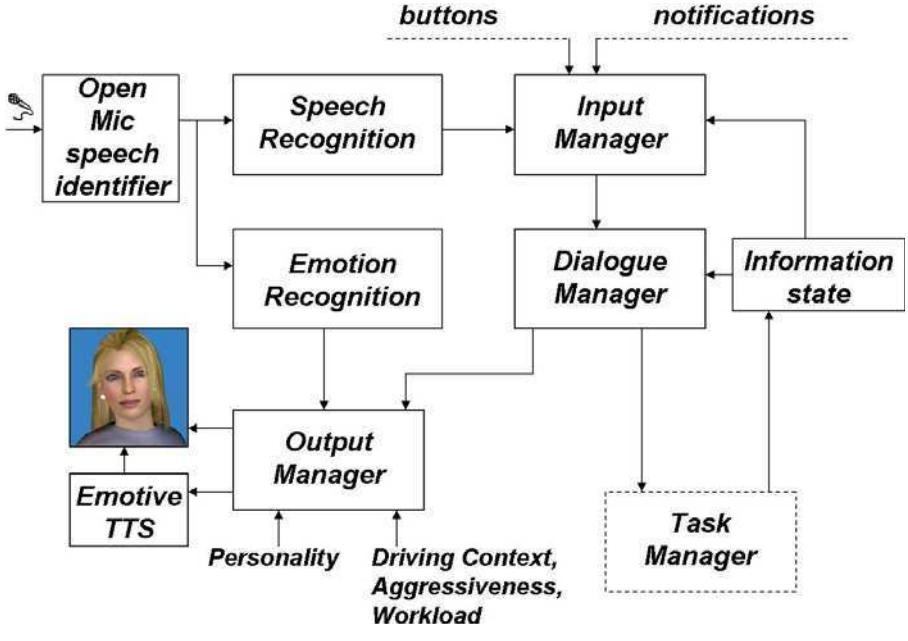


Fig. 5. EDAS Spoken Dialogue System.

The emotive output is based on the following three elements: TTS, emotive tags to control the TTS (i.e., how it is said), and the script to generate the parameters for the facial mapping. The TTS created by the Output Manager is based on a personality, workload, driver's mood, driver aggressiveness, and driving context. The personality is predefined and can include personalities such as polite, humorous, normal, or rude: it is one of the custom configuration parameters. The driver's mood can be gauged from emotion recognition software, such as Affective Media. The driving context can be generated by the driving style and workload. We define the driving style as sporty, normal, or relaxed and workload as high or low.

Emotive tags can be used to control how the text is said. To generate emotive text-to-speech, EDAS used the CereProc speech synthesis system [29] with a custom built voice. The CereProc speech synthesizer provides rich XML control of speech, including several predefined emotive tags such as happy, calm, cross, and sad that can be used to specify the desired emotive modulation of the speech on the fly [30].

The third emotive output element is the script that generates the parameters for the facial mapping. These parameters are used to set the Avatar emotion (i.e., facial

expression) state variables and are sent to Avatar animation engine for real-time emotion rendering.

Table 1 provides an example of how the mapping can be done based on a subset of the inputs: workload, driver mood, and Avatar personality. So, if the driver had selected a polite personality, and asked the question, "What time is it?", the driver might get the following response under high workload: "It's 8:30." Under a low workload, the response might be, "I am happy to tell you that the time is 8:30." Under a high workload the responses are concise in order not to add to the driver's workload; under a low workload, the responses can be more extensive or provide additional related information. If the driver's mood is bad, the output emotion might be neutral, whereas if the mood is good, the output emotion could be happy or playful.

Table 1. Example of input mapping to emotive TTS and Avatar emotion.

Inputs			**Outputs**	
Workload	Driver Mood	Avatar Personality	TTS	Emotion
Low	Good	Polite	I'd be happy to tell you the time. It is 8:30.	Happy
Low	Bad	Polite	The current time is 8:30.	Neutral
High	Good	Polite	It's 8:30.	Happy
High	Bad	Polite	It is 8:30.	Neutral
Low	Good	Humorous	I'm glad you're asking and not looking down at your watch. It's 8:30.	Playful
High	Good	Humorous	In this time zone it is 8:30.	Playful

The EDAS SDS was further enhanced with dynamic grammar capability (i.e., grammar updated from a database at runtime) to support the customized cloud-based infotainment modules, such as personalized news or music. For example, the user could create news folders or music stations with any name they wanted and that name would be added to the recognition vocabulary at runtime.

Although we implemented the EDAS SDS entirely as an on-board system, we recognize the opportunity to leverage seamless integration of on-board and off-board speech recognition capabilities (see Fig. 6). In this hybrid implementation, it would be desirable that content pertaining to in-vehicle information and services be self-contained using on-board speech recognition and not rely on internet connection: requests for content or services residing on the cloud that already require internet connection could utilize powerful, off-board speech recognition to support potentially large vocabulary and grammar. Furthermore, this would enable new cloud-based services and their corresponding vocabulary grammar to be added anytime.

5 Cloud-Based Infotainment and Decision Support

One of the objectives of the EDAS prototype was to investigate new possibilities enabled by advances in connectivity to integrate internet-based resources to create a new class of personalized, interactive vehicle entertainment and decision support functionality. First of all, we wanted to leverage the growing number of recommender-based entertainment services, including news and music that are synergistic

with EDAS personalization goals. In addition, introduction of new services can happen outside of the typical vehicle design cycle, and maintenance and upgrades of infotainment applications are simplified. Furthermore, the emerging cloud-based paradigm of software and platform as a service enables an economical and scalable approach to incorporate computationally intensive support. This section discusses the EDAS approach for cloud-based infotainment and provides an implementation overview of some services for common in-vehicle activities.

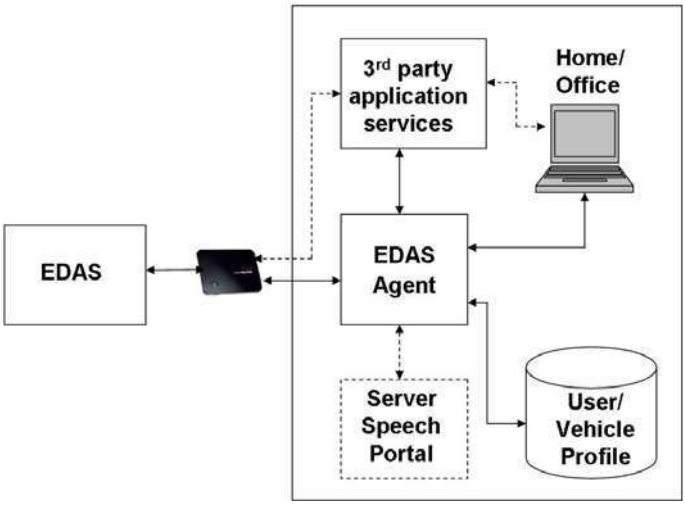


Fig. 6. EDAS Cloud-based infotainment integration.

Fig. 6 shows the EDAS cloud-based integration infrastructure. EDAS is connected to the internet using a 3G network card or cell phone with appropriate data plan. The vehicle has an associated agent that performs internet search tasks on behalf of the driver and exchanges data with the on-board system using an XML format. The agent manages connections to third party applications and services based on the user profile and the current context. The services are invoked either on-demand from the driver or by the internal logic that activates tasks at scheduled times or based on certain conditions. The context is defined by data that the vehicle periodically sends to the agent, including location, direction, speed, intended route, fuel level, fuel economy, driver operating pattern, OBD codes, etc.: this data updates the customer/vehicle profile. The agent also has access to other personal applications, such as calendars or social network sites. It can use this additional information to set up parameters in requests to other applications or to filter, sort, or modify data received from applications before sending it to the vehicle.

In general, the agent can perform sophisticated data aggregation and inference to preprocess data before sending it to the vehicle, substantially enhancing decision support in a driving environment. For example, the driver may be looking for restaurants within an hour drive along the given route. The system could estimate an approximate location where the driver should begin looking for restaurants, filter

restaurants that will be open at the estimated time of arrival, look for restaurants that match the driver's profile food preferences, and use social network collaborative filtering to sort the results according to social network rating. The resulting XML file could contain up-to date and detailed data for each restaurant including general business attributes, such as name, description, address and location, average price of meal for the given time of arrival (e.g. breakfast, lunch or dinner if applicable), as well as non-formal reviews from members of their social network or general customers.

Computationally intensive models can be run on the cloud, providing an economical way to run these models without increasing the software and hardware requirements of the embedded vehicle platform. The Intelligent Refueling Advisor is one such application that demonstrates the functionality within EDAS and is described in detail in [31]. The Refueling advisor integrates the expected routes over time, estimated fuel consumption, current fuel level, and current and forecasted gas prices at individual gas stations along the given route, and generates a recommendation on the cheapest refueling option (i.e., when, where, and how much fuel to get). The system can provide recommendations both upon request, as well as proactively based on known and predicted behavior and preferences. For example, if driver's current fuel level is within a small range of the model's expected fuel level for the day, if the gas price today is cheaper than tomorrow's forecast, and if the driver has less than half a tank but more than the low fuel level, then suggest: "Would you like to stop for gas today since prices are predicted to go up tomorrow?" In addition to recommendations, the driver can query the system regarding the logic behind the given recommendation. For example, the driver could ask, "Why are gas prices forecasted to go up tomorrow?" The system could respond, "This effect is due to wholesale prices rising".

Ubiquitous internet connectivity allows us to also prototype new forms of entertainment. Until now, the in-vehicle entertainment options were either 1) broadcast (i.e., radio, satellite) providing new content catering to groups and not to individuals or 2) personal media collections (i.e., MP3, CDs, tapes) that are limited to a preferred but already known context. The proliferation of web-based recommender systems and personalized entertainment services enable taking advantage of both approaches to provide new context to personal preferences of the given individual. In addition, cloud-based infotainment allows for a continuum of personalized information and entertainment services between in-vehicle and off-vehicle activities.

The EDAS Personalized News Services is an attempt to create a next generation news radio with added personalization and interactivity. The personalized news concept has been widely explored by internet news providers, such as Google [32]. The EDAS prototype leverages Really Simple Syndication (RSS) aggregator services to bring in news articles arranged by different categories. EDAS loads blocks of 6 articles at a time to be announced in the vehicle using the on-board TTS engine. The articles are represented in an XML format that includes the title and body of the article; it also may include additional metadata such as the author, source, time, keywords, and references to other relevant sources. The format of the news delivery is similar to many traditional broadcast radio services: first, the headlines of all articles are introduced followed by the articles in full. However, the EDAS news service allows the listeners to interact and guide news delivery: this allows the driver to skip articles outside of listeners' interests or change the order in which they want to hear it. Furthermore, the user can request specific clarification on certain terms/names in the

article, provided that the item of interest is specified in the keyword metadata: the system can access additional sources of information, such as Wikipedia, to provide details on such requests. Additionally, the user can bookmark, share the article, or vote on their interest in the given article. Although, the current implementation of the news source uses an RSS aggregator, this could be expanded to the other forms of internet news services.

Music is another common form of in-vehicle entertainment that was prototyped in EDAS. In the last decade, there have been a growing number of web-based personalized music services and their mobile versions: Pandora [33], Last.fm [34], Musicovery [35], Slacker [36], etc. These services stream music from their vast music libraries based on the user's profile. These services allow users to create personalized music stations by first providing an example of their music interests (e.g. provide specific seed song name, artist, or genre) and then providing interactive feedback on each song selection. Specifically, the user may vote the song up or down or skip the song. These systems use the feedback to modify the user's profile, further improving the music selection. Each of the services may utilize different methods of selecting songs, such as collaborative filtering or nearest neighbor search within music parameter space.

The EDAS prototype implemented music services using Pandora, which is based on the music genome database, where all the songs are associated with a set of around 300-400 parameters that are set up by musicologists. Song selection is implemented using the closest neighbor algorithm from the user's profile. The EDAS Pandora interface (see Fig. 7) allows the user to select a specific station, vote up and down, or bookmark the song using either voice or button presses as input methods.

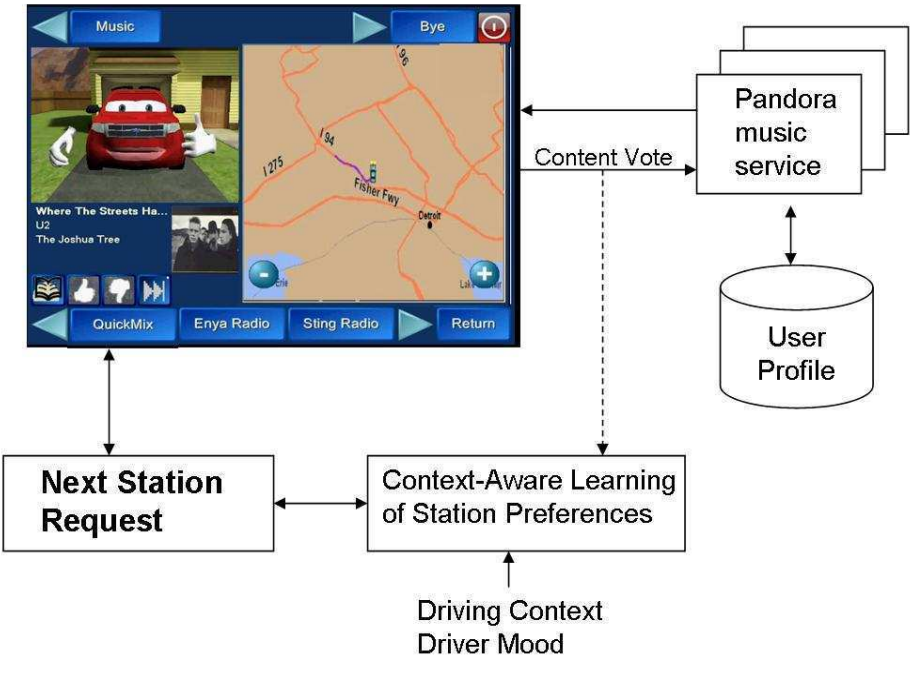


Fig. 7. EDAS context-aware infotainment control.

The personalization of cloud-based infotainment can be further enhanced by exploring context-aware selection of infotainment content. EDAS keeps track of the driver's infotainment content selection, such as the given Pandora station or news category under different conditions and uses this information in an intelligent shuffle mode to provide automatic infotainment content selection (i.e., source and station) corresponding to the summarized driver's preferences and to the current conditions. This context-aware infotainment selection is based on a learning and reasoning algorithm that uses the Markov Chain probabilistic model [37]. The automatic selection feature is designed to work with the Entertainment Application when the infotainment system is in a smart shuffle mode. That is, the selection is driven by the system based on the driver's previously learned infotainment preferences, and the input from the driver is used only to reinforce / reject the recommended station selection (music, talk show, news etc.). When infotainment selection is controlled by the driver, the learning algorithm uses the driver's selections to update the transition probability model.

6 Conclusions

An increased level of electronics, software, computational and communication power in current vehicles enables new types of intelligent features that are based on the ability of the vehicle to estimate road and traffic conditions and driver preferences and to adapt accordingly. This tendency of growing sophistication, autonomy, and intelligence of vehicle control systems marks a trend towards the creation of a “driver-aware” vehicle that appeals to the driver by offering multiple features that maximize driver’s safety, performance, and comfort, while leaving full responsibility and control of the vehicle to the driver. The goal is to not only satisfy the hypothetical nominal driver, but to also meet the requirements of as many individual members of a customer group that is targeted by a specific vehicle model. Examples of such flexible features are the multiple powertrain and suspension performance modes, customizable and adaptable HMI, and vehicle infotainment.

The EDAS system shows a new emotive and intelligent way for drivers and vehicles to communicate with each other. EDAS integrates recent advances in intelligent controls, SDS, embodied conversational agents, and affective computing in a vehicle environment. This system has demonstrated a new level of personalization and learning that can lead to an emotive bond between the driver and the vehicle.

We first demonstrated EDAS in a buck version (see Fig. 8) and subsequently implemented EDAS in a 2006 Edge. Implementation of a full featured EDAS system within the vehicle allowed us to experiment and experience the different features in a real driving environment. It also helped us to better understand what worked and what did not work, allowed us to demonstrate and convey the benefits of the features to the business more effectively, and to prepare more reliable specifications for production implementation.

The EDAS concept was incorporated into the future SYNC vision unveiled as EVA (see Fig. 9) at the 2009 Consumer Electronics Show (CES) and included in the Lincoln C Concept Car at the 2009 North American International Auto Show (NAIAS) [38]. Furthermore, a number of features that EDAS pioneered are being released within upcoming versions of SYNC.

Fig. 8. EDAS in a buck environment.

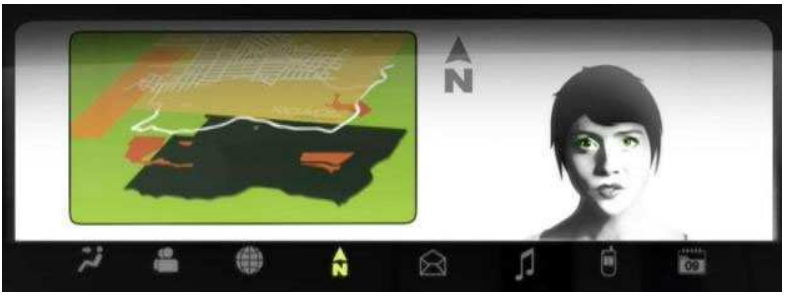


Fig. 9. EVA that was show in 2009 CES and NAIAS.

Acknowledgements. The authors would like to thank Stefano Di Cairano, John Ginder, Thomas Giuli, Perry MacNeille, Jianbo Lu, John Michelini, Kwaku Prakah-Asante, Nestor Rychtyckyj, Mark Schunder, Fazal Syed, Steve Szwabowski, Gary Strumolo, Chris Teslak, Kacie Theisen, Basavaraj Tonshal, Finn Tseng, Xiaoming Wang, and Ilya Kolmanovsky for their valuable ideas and contributions to the development of the EDAS prototype system.

References

1. Ford SYNC, `http://www.fordvehicles.com/technology/sync/`
2. Maslow, A.: A Theory of Human Motivation. Psychological Review 50(4) (1943)
3. Nagamachi, M.: Kansei Engineering. Kaibundou, Tokyo (1989)
4. Picard, R.: Affective computing. The MIT Press, Cambridge (1997)

5. Kaplan, F.: Free creatures: The role of uselessness in the design of artificial pets. In: Christaller, T., Indiveri, G., Poigne, A. (eds.) Proceedings of the 1st Edutainment Robotics Workshop. GMD-AiS (2000)
6. Kaplan, F.: Talking Aibo: First experimentation of verbal interactions with an autonomous four-legged robot. In: Nijholt, A., Heylen, D., Jokinen, K. (eds.) Learning to Behave: Interacting agents CELE-TWENTE Workshop on Language Technology, pp. 57–63 (2000)
7. Steels, L., Kaplan, F.: Aibo's first words, the social learning of language and meaning. In: Evolution of Communication, vol. 4(1). John Benjamins Publishing Company, Philadelphia (2002)
8. SONY. Sony Launches Four-Legged Entertainment Robot, http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046/
9. Mori, K., Kitagawa, N., Inukai, A., Humphries, S.: Vehicle expression control system, vehicle communication system, and vehicle which performs expression operation, US Patent #6,757,593 (2004)
10. Wikipedia. Nissan Pivo, http://en.wikipedia.org/wiki/Nissan_Pivo
11. Nissan. Concept Car Pivo 2, http://www.nissan-global.com/EN/PIVO2/
12. Filev, D., Lu, J., Prakah-Asante, K., Tseng, F.: Real-time Driving Behavior Identification Based on Driver-in-the-loop Vehicle Dynamics and Control. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas (2009)
13. Lu, J., Filev, D., Prakah-Asante, K., Tseng, F., Kolmanovsky, I.: From vehicle stability control to intelligent personal minder: Real-time vehicle handling limit warning and driver style characterization. In: Proc. of IEEE Multiconference on Computational Intelligence, CIVVS 2009, Nashville, TN, pp. 43–50 (2009)
14. Prakah-Asante, K., Filev, D., Lu, J.: Hybrid Intelligent System for Driver Workload Estimation for Tailored Vehicle-Driver Communication and Interaction. In: Proc. IEEE Int. Conference on Systems, Man, & Cybernetics, Istanbul, Turkey, pp. 1178–1184 (2010)
15. Pieraccini, R., Dayanidhi, K., Bloom, J., Dahan, J., Phillips, M., Goodman, B., Prasad, K.: Multimodal Conversational Systems for Automobiles. Communications of the ACM (CACM) 47, 47–49 (2004)
16. Waytz, A., Epley, N., Cacioppo, J.: Social Cognition Unbound: Insights Into Anthropomorphism and Dehumanization. Current Directions in Psychological Science 19(1), 58–62 (2010)
17. Charamel. Automotive Show-Cases, http://www.charamel.com/en/showroom/automotive_show_cases.html
18. Wikipedia. Uncanny Valley, http://en.wikipedia.org/wiki/Uncanny_Valley?phpMyAdmin=fcoQR3VqERTajB4P3dbitaPqsvf
19. Mori, M.: Bukimi no tani - The uncanny valley. In: MacDorman, K.F., Minato, T. (trans.) Energy, vol. 7(4), pp. 33–35 (1970) (Originally in Japanese)
20. Zabaware, http://www.zabaware.com/
21. haptek, http://www.haptek.com/
22. Clarissa, http://www.ic.arc.nasa.gov/projects/clarissa/
23. Clarissa: a HAL type computer for the ISS? http://chasblogspot.blogspot.com/2008/03/clarissa-hal-type-computer-for-iss.html
24. Rayner, M., Hockey, B., Bouillon, P.: Putting Linguistics into Speech Recognition. CSLI Publications, Stanford (2006)

25. Regulus, http://sourceforge.net/projects/regulus/
26. Nuance, http://www.nuance.com/
27. Affective Media, http://www.affectivemedia.com/
28. Jones, C.M., Jonsson, I.-M.: Performance Analysis of Acoustic Emotion Recognition for In-Car Conversational Interfaces. In: Stephanidis, C. (ed.) UAHCI 2007 (Part II). LNCS, vol. 4555, pp. 411–420. Springer, Heidelberg (2007)
29. CereProc, http://www.cereproc.com/
30. Aylett, M., Pidcock, C.: The CereVoice characterful speech synthesiser SDK. In: The 2007 Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, pp.174–178 (2007)
31. Klampfl, E., Gusikhin, O., Theisen, K., Liu, Y., Giuli, T.: Intelligent Refueling Advisory System. In: Proceedings of the 2008 IEEE International Conference on Informatics in Control, Automation & Robotics, Madeira, Portugal (2008)
32. Das, A., Datar, M., Garg, D., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, pp. 271–280 (2007)
33. Pandora, http://www.pandora.com/
34. Last.fm, http://www.last.fm/
35. Musicovery, http://musicovery.com/
36. Slacker, http://www.slacker.com/
37. Filev, D., Kolmanovsky, I.: A Generalized Markov Chain Modeling Approach for On Board Applications. In: Proc. IEEE World Congress on Computational Intelligence – WCCI 2010, Barcelona, Spain (2010)
38. CES. Ford turns hip with Eva, http://blogs.reuters.com/mediafile/2009/01/10/ces-ford-turns-hip-with-eva/

Force Prediction Using Fingernail Imaging: An Overview

Tom Grieve, Yu Sun, John M. Hollerbach, and Stephen A. Mascaro

University of Utah, Salt Lake City, UT 84112, U.S.A.
University of South Florida, Tampa, FL 33620-5399, U.S.A.
tom.grieve@utah.edu, yusun@cse.usf.edu,
jmh@cs.utah.edu, smascaro@mech.utah.edu

Abstract. This paper presents a method for automatically calibrating a fingernail imaging system used to detect arbitrary three-dimensional force on the human fingerpad. The calibration is accomplished using a Magnetic Levitation Haptic Device modified to apply calibrated forces through a flat plate to the fingerpad.

The technique explained here is shown to accurately predict arbitrary shear force with RMS error of 0.3 N (which is 3% of the full range of ± 5 N) and normal force with RMS error of 0.5 N (which is 5% of the full range of $0 - 10$ N). The paper also demonstrates the model and explains some of the methods used to compensate for nonlinearities in the fingernail coloration response.

Keywords: Force sensing, Haptics, Grasping.

1 Introduction

Fingernail imaging as a method for detecting force on the human fingerpad is based on the coloration effect illustrated in Figure 1. As the finger is pressed against a surface, the interactions between the bone and nail compress regions of the finger pulp and thereby constrict blood flow through the capillaries. This causes blood to pool in some regions while evacuating other areas, creating a band near the distal end of the fingernail which begins to whiten. Simultaneously, the rest of the nail except the lunula reddens. Additionally, areas of the flesh alongside the nail whiten. If, in addition to normal force, shear force is applied to the fingerpad, different patterns appear on the nail.

This effect is fast, requiring less than one second to reach steady-state. The coloration response described here is broadly similar in the general population [1] and so studies of this effect are applicable to a wide range of people. Surprisingly, when individual calibration is applied, it is possible not simply to qualitatively predict the direction of the force, but also to quantitatively estimate the magnitude with a reasonable degree of accuracy. This paper presents the current method of calibration for fingernail imaging used to create a model for predicting fingertip force.

Mascaro and Asada [2] first proposed fingernail imaging as a method for detecting force on the human fingerpad. That work used a combination of photodetectors and LEDs embedded in an on-nail sensor to illuminate and perceive the coloration changes. These wearable sensors, with appropriate calibration, were able to predict shear force with an RMS error of 0.5 N over a range of ± 2.25 N and normal force with an RMS

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 37–48.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

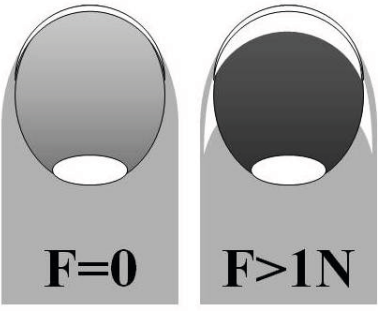


Fig. 1. Illustration of Coloration Effect: (Left) Finger with no Contact (Right) Finger with 1N Normal Contact Force.

error of 1.0 N over a range of $0 - 3$ N [3]. There are, however, two limitations in these sensors. Space restrictions prevent more than a few photodetectors being placed on the nail, which limits the amount of data that may be recorded from the technique. Additionally, the sensors must be custom manufactured for each test subject.

To bypass these restrictions, the development of a high-resolution imaging technique was created [4]. This involves capturing images of the fingernail and the surrounding skin with a digital camera. This method was able to predict normal force with an RMS error of 0.3 N over a range of $0-10$ N. Later experiments improved this prediction error to just 0.1 N when simultaneously predicting normal forces over a range of $0-10$ N and shear forces oriented either laterally (across the finger) or longitudinally (along the finger) over a range of ± 2 N [5]. Another important result of these initial experiments was that the green channel in an RGB image showed the largest response to the force. In addition, it was found that the direction of the force could even be qualitatively estimated without individual calibration with 90% accuracy [6]. These levels of accuracy were found to hold even as the resolution of the images was reduced to 10×10, indicating that extremely high-resolution images were not necessary for accurate measurements. However, this method introduced new challenges, most notably positioning the finger relative to the camera and adjusting for differences in lighting. For the time being, these have been carefully controlled, but they will need to be addressed at a future time.

Both methods have been shown to be accurate in predicting both the magnitude and direction of force, but require individual calibration. The automated calibration technique presented here is able to predict arbitrarily-directed shear force with an RMS error of ± 0.3 N over a range of ± 2.5 N and normal force with an RMS error of ± 0.3 N over a range of $0 - 10$ N [7]. Section 2 of this paper demonstrates the automated calibration procedure. Section 3 contains a discussion of the analysis techniques used on the calibration data. This includes the method used for image registration as well as the model developed to represent the effect of force on the coloration effect. The least-squares method used to generate the values used in the model is discussed. Section 4 presents some results from the model and outlines possibilities for future work.

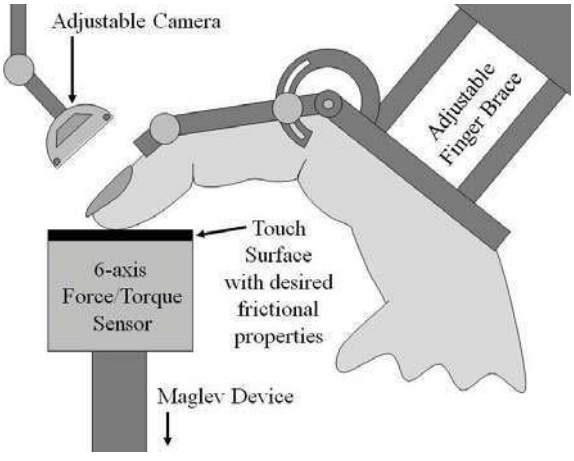


Fig. 2. Diagram of Calibration Setup.

2 Calibration Procedure

To achieve the stated accuracy of force prediction, individual calibration is necessary. This calibration procedure follows a general form: A finger is placed on a force sensor with a camera placed above. A series of forces is exerted between the finger and the sensor while the camera records images. Once the calibration is finished, these images are compared to the data obtained from the force sensor to form a model for prediction. Although in the past, manual calibration has been performed, requiring the test subject to exert forces, the fatigue that resulted limited a 20-minute session to capturing fewer than 100 images. To allow for faster calibration and reduce the fatigue on the test subject, an automated calibration procedure has been developed.

Automated calibration requires a special setup such as that shown in Figure 2. The test subject's finger is placed in a passive restraint that restricts movement and adjusts the finger joint angles without affecting blood flow. The force sensor is attached to a force-controlled Magnetic Levitation Haptic Device (MLHD) [8]. A camera is placed above the finger to record the images. The MLHD applies the desired force to the test subject and records both image and force automatically. The characteristics of the MLHD are such that the calibration routine is capable of recording one image every 0.5 seconds. In a 20-minute session with a few short breaks, more than 1,500 images may be collected. The actual calibration setup is shown in Figure 3.

The MLHD is controlled by means of a standard PIV force feedback controller with a feed-forward term. The signal from the controller is sent to the MLHD as a force command. This controller has a relatively slow step response, with a 98% settling time of 0.2 seconds. Since the calibration trajectories do not use steps in force but ramps, this is not a problem during calibration.

The desired force space to be covered during calibration is a cone with a slope equal to the coefficient of static friction μ between the finger and the contact surface. Different calibration trajectories have been designed to cover the force space in different ways.

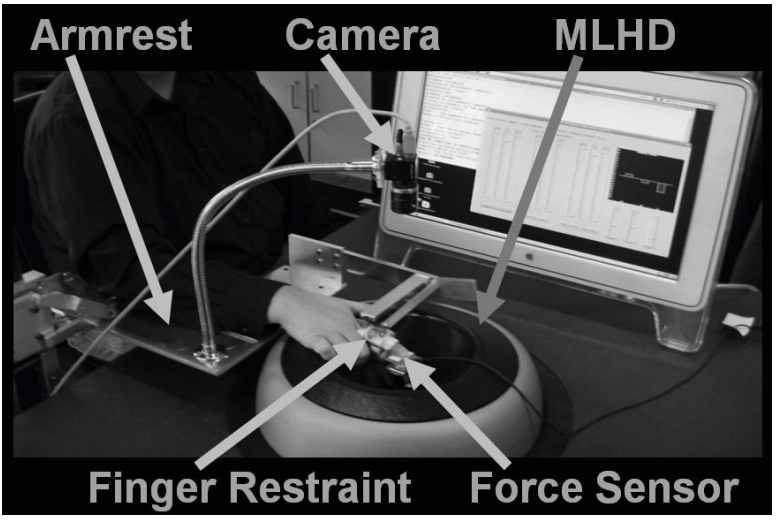


Fig. 3. Photo of Calibration Setup.

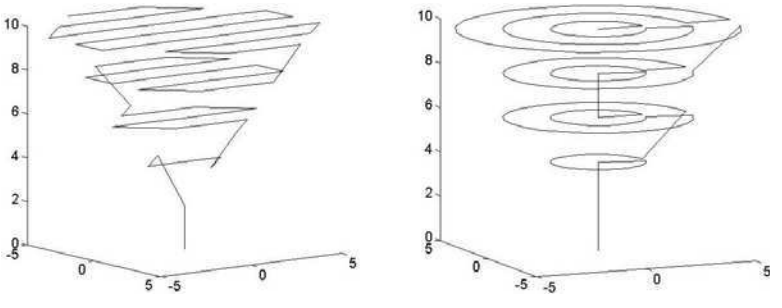


Fig. 4. Example Calibration Trajectories.

Two sample calibration trajectories are shown in Figure 4. Some use a Cartesian (x-y-z) grid, while others use cylindrical coordinates (r-θ-z). The distance between adjacent points on the grid, ΔF varies, as does the value used for μ, the slope of the outside edge of the cone. During calibration, each force level is held for 0.4 seconds to allow the camera time to record an image. Then a 0.1-second ramp is used to transition to the next force level.

3 Data Analysis

Once collected, the calibration data must be processed to develop a model that may then be used to predict force. The processing takes place in two steps: Registration and Model Formulation.

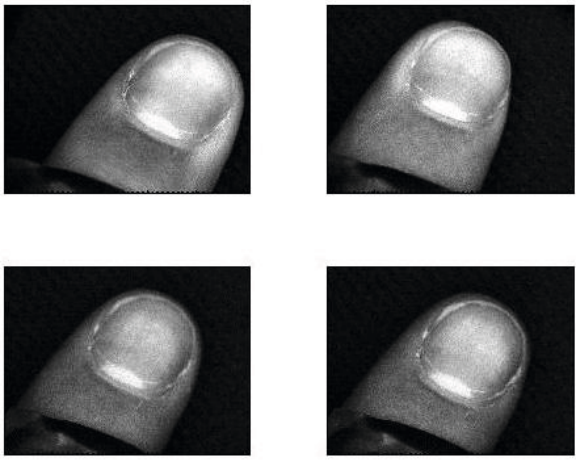


Fig. 5. Finger Images Demonstrating the Need for Registration.

3.1 Registration

The problem of registration is evident in Figure 5. Here, four calibration images are shown. Although the viewing plane of the camera is still parallel to the plane of the fingernail, the position and orientation of the finger has shifted in each of the images. Before these images can be used to generate a model relating pixel intensity to force, they must be registered so that all images have the finger in the same position. One challenge of this task is that the finger is relatively featureless compared to typical registration subjects. Additionally, the calibration itself varies the intensities within the finger, potentially changing some features that might otherwise be used to register the nail.

Although several registration methods have been explored, the current procedure uses what is known as the Scaled Rigid-Body Transformation. This takes advantage of the knowledge that the image plane is always parallel to the finger during calibration. Since the finger only rotates in-plane, it is can be shown that proper registration requires only translations and rotations about the z-axis of the finger. The process is illustrated in Figure 6. An original image is shown in part (a). The finger is found in the image through a connected component analysis and the major axis of the finger is determined (b). This axis is rotated to a predetermined orientation (c) and the image is cropped to include only the largest connected component (d). The image may then be resized as needed to correspond to the target resolution.

3.2 Finger Model

A model of the coloration response of the finger related to force on the fingerpad was first explored in [2]. This initial model was highly nonlinear and took into account the dynamic behavior of the fingernail and surrounding skin. Later experiments [9] have

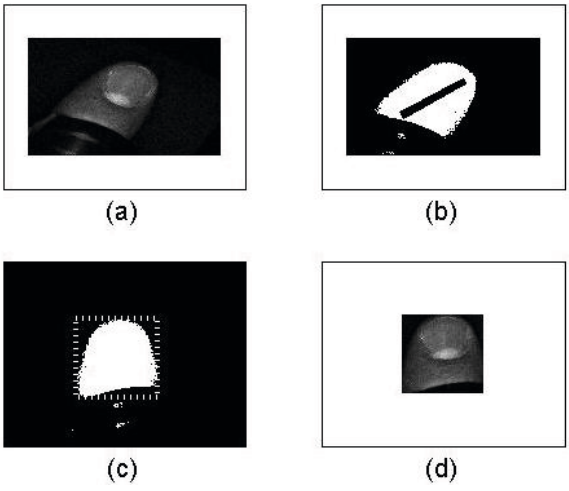


Fig. 6. Registration Procedure (a) Original Image (b) Largest Connected Component with Major Axis (c) Rotated Image with Bounding Box (d) Cropped Image.

shown that the dynamic response of the fingernail is fast enough that it is not necessary for most purposes to account for this, and so the dynamic response is not considered here.

Independence. The static nonlinear behavior of the fingernail may be categorized in one of two general types: independence and saturation. Independence refers to the way that certain regions' coloration response is independent of force in one, two or all three directions of force. For example, the lunula (the light-colored crescent-shaped area at the base of the nail) does not change color in response to any force. Other regions respond only to force in one direction. Research into methods for correcting for this type of nonlinearity is currently ongoing.

Saturation. Certain regions of the fingernail and surrounding skin saturate at relatively low levels of force, while other areas continue to change color even as the normal force exceeds 10 N. When considering only one direction of force, it is relatively easy to see a sigmoid shape appear in the data. As multiple directions of force are added, the response becomes more difficult to visualize and it is more challenging to recognize patterns of saturation.

The procedure to compensate for saturation is illustrated in Figure 7. First, the data is sorted to reveal the saturation. A fourth-order polynomial is fitted to the data. The maximum value of the gradient of that polynomial is found. The data points above and below that maximum value where the gradient reaches 20% of the maximum are found. The intensities of those two points are identified as the upper and lower saturation threshhold, respectively. If no upper or lower saturation limit is found, the minimum or maximum intensity of the pixel is chosen for the respective saturation limit. The data between these two limits is then chosen and fit using the linear model detailed next.

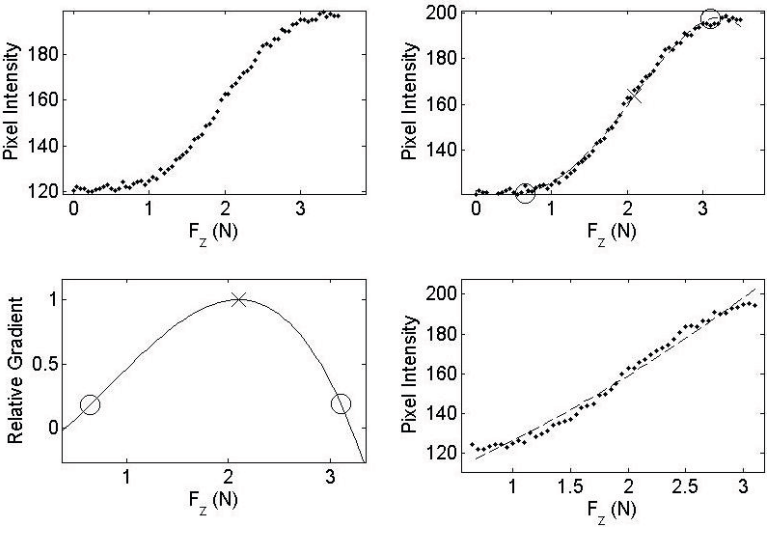


Fig. 7. Saturation Detection in One Force Dimension.

Linear Model. The linear model of the finger is based on the assumption that the intensity of a given pixel (p_i) is a linear combination of the applied forces (f_x, f_y and f_z) with some offset. This results in an equation for the ith pixel:

$$p_i = a_i + b_i f_x + c_i f_y + d_i f_z \tag{1}$$

An $m \times n$ image represented by a pixel vector $\mathbf{p} = \begin{bmatrix} p_1 \; p_2 \cdots p_{mn} \end{bmatrix}^T$ with a corresponding force vector $\mathbf{f} = \begin{bmatrix} f_x \; f_y \; f_z \end{bmatrix}^T$ may then be represented by a single matrix equation:

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{mn} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_{mn} & b_{mn} & c_{mn} & d_{mn} \end{bmatrix} \begin{bmatrix} 1 \\ f_x \\ f_y \\ f_z \end{bmatrix} \tag{2}$$

To form a matrix equation representing an entire calibration set of k images, first the transpose is taken:

$$\begin{bmatrix} p_1 \; p_2 \cdots p_{mn} \end{bmatrix} = \begin{bmatrix} 1 \; f_x \; f_y \; f_z \end{bmatrix} \begin{bmatrix} a_1 \; a_2 \cdots a_{mn} \\ b_1 \; b_2 \cdots b_{mn} \\ c_1 \; c_2 \cdots c_{mn} \\ d_1 \; d_2 \cdots d_{mn} \end{bmatrix} \tag{3}$$

The k equations are then stacked:

$$\begin{bmatrix} {}^1p_1 & {}^1p_2 & \cdots & {}^1p_{mn} \\ {}^2p_1 & {}^2p_2 & \cdots & {}^2p_{mn} \\ \vdots & \vdots & \ddots & \vdots \\ {}^kp_1 & {}^kp_2 & \cdots & {}^kp_{mn} \end{bmatrix} = \begin{bmatrix} 1 & {}^1f_x & {}^1f_y & {}^1f_z \\ 1 & {}^2f_x & {}^2f_y & {}^2f_z \\ \vdots & \vdots & \vdots & \vdots \\ 1 & {}^kf_x & {}^kf_y & {}^kf_z \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_{mn} \\ b_1 & b_2 & \cdots & b_{mn} \\ c_1 & c_2 & \cdots & c_{mn} \\ d_1 & d_2 & \cdots & d_{mn} \end{bmatrix} \tag{4}$$

forming a matrix equation $\mathbf{P} = \mathbf{FA}$. These equations are then solved using standard Ordinary Least Squares (OLS):

$$\mathbf{A} = \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{P} \tag{5}$$

The next section explains how the matrix $\mathbf{A}$ is used to predict force.

3.3 Prediction Model

Different least-squares methods have been considered for use in analyzing the calibration data. Although a Total Least Squares approach might seem appropriate, since there is error in both the force sensor readings and the camera intensity data, the force sensors used in these experiments are so accurate (± 0.8mN) compared to the cameras that this method increases the prediction error. Currently, OLS is used to determine the model and Weighted Ordinary Least Squares (WLS) is used to validate the model and predict forces.

Since it would be undesirable to train the model to artifacts in the data, it is expected that many more images will be recorded than the minimum required to exactly determine all of the coefficients. In other words, $k \gg m \times n$, where $m \times n$ represents the resolution after the image has been registered and cropped. This ensures that the situation is more like curve fitting and less like polynomial interpolation.

To generate the model, the $\mathbf{A}$ matrix is partitioned as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}^T \\ \mathbf{B}^T \end{bmatrix} \tag{6}$$

where

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{mn} \end{bmatrix} \tag{7}$$

and

$$\mathbf{B} = \begin{bmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ \vdots & \vdots & \vdots \\ b_{mn} & c_{mn} & d_{mn} \end{bmatrix} \tag{8}$$

In addition to these two submatrices, the prediction model requires the covariance matrix of the calibration images $\boldsymbol{\Sigma} = \mathbf{cov}(\mathbf{P})$. This prediction model may then be expressed as $\mathbf{f} = \mathbf{C}\left(\mathbf{p} - \mathbf{a}\right)$, or:

$$\mathbf{f} = \left(\mathbf{B}^T \boldsymbol{\Sigma}^{-1} \mathbf{B}\right)^{-1} \boldsymbol{\Sigma}^{-1} \mathbf{B}^T \left(\mathbf{p} - \mathbf{a}\right) \tag{9}$$

This is recognizable as a weighted least squares solution to the equation $\mathbf{p} = \mathbf{B}\mathbf{f} + \mathbf{a}$ with weighting matrix $\boldsymbol{\Sigma}^{-1}$. The matrix $\mathbf{C} = \left(\mathbf{B}^T \boldsymbol{\Sigma}^{-1} \mathbf{B}\right)^{-1} \boldsymbol{\Sigma}^{-1} \mathbf{B}^T$ and the vector $\mathbf{a}$ are stored after calibration is completed for future use in force prediction.

4 Results

4.1 Model Validation

Verification of the model is achieved using a set of images collected using the procedure detailed in Section 2. The Measured Force $\mathbf{f}_m$ is stored at the time the images

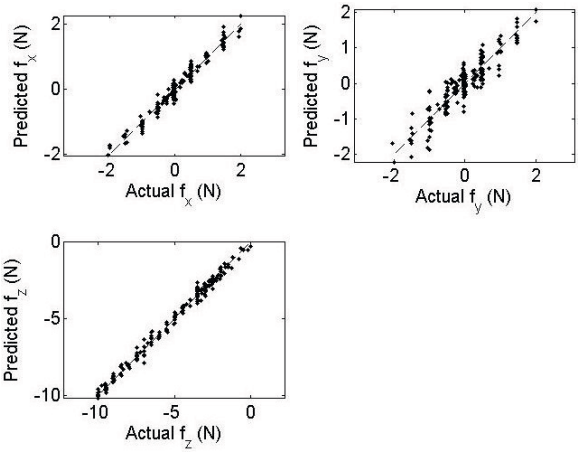


Fig. 8. Validation Error for Subject 5.

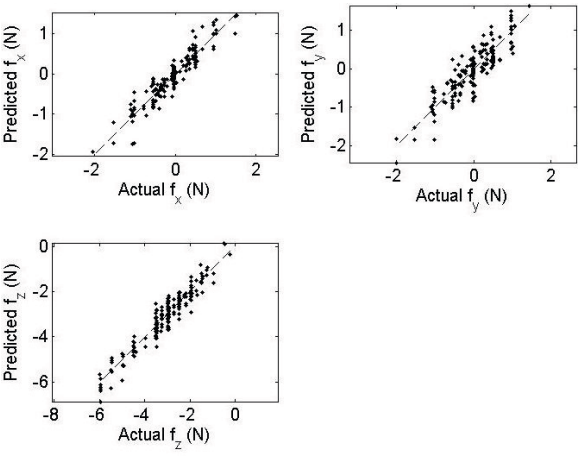


Fig. 9. Validation Error for Subject 6.

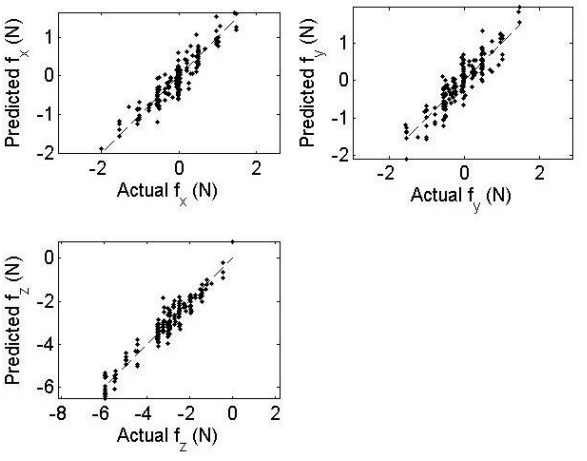


Fig. 10. Validation Error for Subject 7.

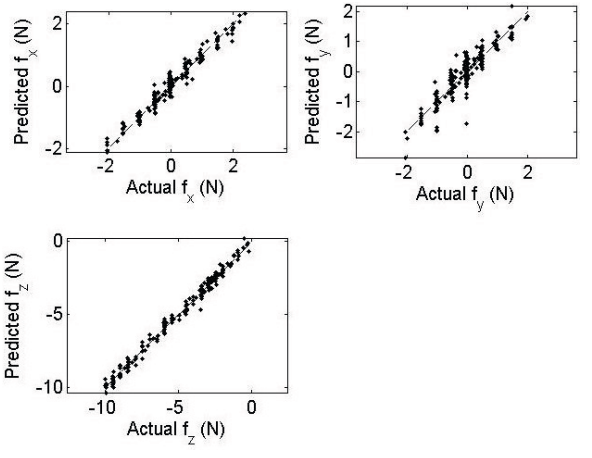


Fig. 11. Validation Error for Subject 8.

are collected. The images are registered using the procedure explained in Section 3.1. Equation 9 is used to obtain the Predicted Force $\mathbf{f}_P$ for each one. This force is compared to $\mathbf{f}_m$ and the RMS error in each direction for the set of images is determined. This error demonstrates the validity of the model in predicting forces when presented with new images. A plot of $\mathbf{f}_m$ vs. $\mathbf{f}_P$ for test subject 5 is shown in Figure 8. For this subject, 1300 calibration images were collected and another 200 validation images were set aside. The 1300 were used to form the model. In this case, the force was predicted on the validation images with an RMS error of 0.2 N in the x-direction, 0.3 N in the y-direction and 0.3 N in the z-direction.

The validation results for subject 6 are shown in Figure 9. For this subject, only 900 training images were collected for use in forming the model. Another 200 validation images were collected. The predicted force on these validation images had an RMS error of 0.2 N in the x-direction, 0.3 N in the y-direction and 0.4 N in the z-direction.

Subject 7's validation results are shown in Figure 10. 750 calibration images were recorded for this subject and 200 validation images tested the accuracy of the model. The RMS error of the validation image prediction was 0.2 N in the x-direction, 0.3 N in the y-direction and 0.4 N in the z-direction.

The validation results for subject 8 are shown in Figure 11. For this subject, 1200 calibration images were collected with 200 in the validation set. The x-, y- and z-direction RMS errors were 0.2 N, 0.3 N and 0.3 N, respectively.

In total, 20 individuals have been tested using this automated calibration routine. The validation results are similar to those described above. The average RMS error for shear force (x- and y-direction) is 0.3 N (which is 3% of the full range of ± 5 N). The average RMS error for normal force (z-direction) is 0.5 N (which is 5% of the full range of $0 - 10$ N).

5 Conclusions

This paper has shown a method of predicting force using fingernail imaging. The technique demonstrated is capable of predicting forces in three dimensions with RMS error less than 0.5 N. Future work includes the application of this method to human grasping exeriments. Success in this endeavor would allow grasping experiments to proceed without the need to instruct test subjects to precisely grasp objects exactly where the test object has been instrumented. Instead, a more natural grasp could be implemented, allowing the subject to comfortably position the hand in whatever manner desired. This would allow for the removal of one more potentially confounding variable in grasping studies, increasing their reliability.

Acknowledgements. This work was supported by NIH Grant 1R21EB004600-01A2 and IGERT Grant DGE-0654414.

References

1. Mascaro, S.A., Asada, H.H.: The common patterns of blood perfusion in the fingernail bed subject to fingertip touch force and finger posture. Haptics-e 4(3), 1–6 (2006)
2. Mascaro, S., Asada, H.: Photoplethysmograph Fingernail Sensors for Measuring Finger Forces without Haptic Obstruction. IEEE Trans. Robotics and Automation 17(5), 698–708 (2001)
3. Mascaro, S.A., Asada, H.H.: Measurement of Finger Posture and Three-Axis Fingertip Touch Force Using Fingernail Sensors. IEEE Trans. Robotics Automation 20, 26–35 (2004)
4. Sun, Y., Hollerbach, J.M., Mascaro, S.A.: Measuring fingertip forces by imaging the fingernail. In: 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 125–131 (2006)
5. Sun, Y., Hollerbach, J.M., Mascaro, S.A.: Predicting fingertip forces by imaging coloration changes in the fingernail and surrounding skin. IEEE Trans. Biomedical Engineering 55(10), 2363–2371 (2008)

6. Sun, Y., Hollerbach, J., Mascaro, S.: Estimation of Finger Force Direction with Computer Vision. IEEE Trans. Robotics 25(6), 1356–1369 (2009)
7. Grieve, T., Lincoln, L., Sun, Y., Hollerbach, J.M., Mascaro, S.A.: 3D Force Prediction Using Fingernail Imaging with Automated Calibration. In: IEEE Haptics Symposium, pp. 113–120 (2010)
8. Berkelman, P., Hollis, R.: Lorentz magnetic levitation for haptic interaction: device design, function, and integration with simulated environments. Intl. J. Robotics Research 9, 644–667 (2000)
9. Sun, Y., Hollerbach, J., Mascaro, S.: Dynamic Features and Prediction Model for Imaging Fingernail to Measure Finger Forces. In: IEEE Intl. Conf. Robotics and Automation, pp. 2813–2818 (2006)

Part I

Intelligent Control Systems and Optimization

Design and Experimental Validation of a Combined PI and Bi-Positional Laws Controller for Delaying the Transition from Laminar Flow to Turbulent Flow over a Morphing Wing

Teodor Lucian Grigorie[1], Andrei Vladimir Popov[1], Ruxandra Mihaela Botez[1], Mahmoud Mamou[2], and Youssef Mébarki[2]

[1] École de Technologie Supérieure, Montréal, Québec, H3C 1K3, Canada
[2] National Research Council, Ottawa, Ontario, K1A 0R6, Canada
lgrigore@elth.ucv.ro, andrei-vladimir.popov.1@ens.etsmtl.ca, ruxandra.botez@etsmtl.ca
{Mahmoud.Mamou,Youssef.Mebarki}@nrc-cnrc.gc.ca

Abstract. This chapter presents the design and the validation of the actuators control system for a morphing wing application. Some smart materials, like Shape Memory Alloy (SMA), are used as actuators to modify the upper surface of the wing made of a flexible skin. The finally adopted control law is a combination of a bi-positional law and a PI law. The controller is validated in two experimental ways: bench test and wind tunnel test. All optimized airfoil cases, used in the design phase, are converted into actuators vertical displacements which are used as inputs reference for the controller. In the wind tunnel tests a comparative study is realized around of the transition point position for the reference airfoil and for each optimized airfoil.

Keywords: Morphing wing, Shape memory alloy actuators, Bi-positional and PI control design, Numerical simulations, Experimental validation.

1 Introduction

Today, aeronautical transport is evolving at a very fast pace, as compared to the beginning of the aviation era; aeronautical traffic tripled during the last fifteen years, and by 2025, is projected to double from today's levels. This traffic is expected to see an estimated +3.0% increase in the number of passengers per year, to approximately 1 billion by 2016; will be accompanied by a load factor increase of 81.7% as compared to today's values by 2025 [1]. This evolution will need the new technologies development in the design and building of modern aircraft equipped with active control systems [2].

During the same time period, fuel cost increases will lead to a slowdown in the aerospace industry, which in turn will stimulate research to find technological solutions; this will specifically involve designing new fuel economy consumption methods. A new green trend has indeed started to spread out from the automobile industry

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 51–76.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

into the aircraft industry, in which research is being carried out to reduce fuel consumption by reducing drag, which is directly related to the airflow type around the aerodynamic aircraft body design. The drag reduction concept is connected to the laminar flow and to the displacement of the transition point between laminar and turbulent flows towards the trailing edge [2].

Numerous studies show that the transition between laminar and turbulent flows is influenced by the shape of the wing airfoil. Aerodynamic studies from the beginning of the aviation history show that for a certain flight condition characterized by a given Mach number and a given Reynolds number, the airflow around a wing airfoil is laminar at the leading edge, but becomes turbulent at a certain point due to air viscosity. A turbulent flow is not desired because of its negative effect in terms of drag increase, which, over time, leads to high fuel consumption, and consequently, increased operating costs [2].

Many researches are made around the world in the new challenge field related to the morphing aircraft, with the purpose to improve operational efficiency, particularly by reducing fuel consumption ([3]÷[9]). Therefore, a lot of architecture were and are still imagined, designed, studied and developed, for this new concept application. One of these is our team project including the numerical simulations and experimental multidisciplinary studies using the wind tunnel for a morphing wing equipped with a flexible skin, smart material actuators and pressure sensors. The aim of these studies is to develop an automatic system that, based on the information related to the pressure distribution along the wing chord, moves the transition point from the laminar to the turbulent regime closer to the trailing edge in order to obtain a larger laminar flow region, and, as a consequence, a drag reduction.

The objective of here presented research work was to develop an actuation control concept for a new morphing mechanism using smart materials, like Shape Memory Alloy (SMA), as actuators. These actuators modify the flexible upper surface of the wing, changing the airfoil shape. The morphing wing project was developed by Ecole de Technologie Supérieure in Montréal, Canada, in collaboration with Ecole Polytechnique in Montreal and the Institute for Aerospace Research at the National Research Council Canada (IAR-NRC).

To achieve the aerodynamic imposed purpose, a first phase of the studies involved the determination of some optimized airfoils available for 35 different flow conditions (five Mach numbers and seven angles of attack combinations). The optimized airfoils were derived from a laminar WTEA-TE1 reference airfoil [10], [11], and used as a starting point for the actuation system design. The transition point position estimation is made using the information received from a pressure system sensors (optical and Kulite types) equipping the upper face of the wing. Two architectures were developed for morphing system: open loop and closed loop. The difference between the two architectures is given by using or not using the position of transition point as a feedback signal for the actuation lines control. Here described work was developed in the open loop phase; in this phase were made numerical and experimental studies related to the aerodynamics of the morphed wing, to the flexible skin realization, to the actuation system, to the control of the actuation system, and, also, to the real-time determination and visualisation of the transition point using the pressure sensors system. Here, the pressure sensors using is limited to the monitoring of the pressure distribution and of the RMS pressure distribution in the boundary layer [12].

2 Structure of the Morphing Wing System

The chosen wing model was a rectangular one, with a reference airfoil WTEA-TE1, a chord of 0.5 m and a span of 0.9 m. The model was equipped with a flexible skin made of composite materials (layers of carbon and Kevlar fibers in a resin matrix) morphed by two actuation lines (Fig. 1). Each actuation line uses SMA wires as actuators. In the same time, 32 pressure sensors (16 optical sensors and 16 Kulite sensors), were disposed on the flexible skin in different positions along of the chord. The sensors are positioned on two diagonal lines at an angle of 15 degrees from centerline. The rigid lower structure was made from Aluminum, and was designed to allow space for the actuation system and wiring [12].

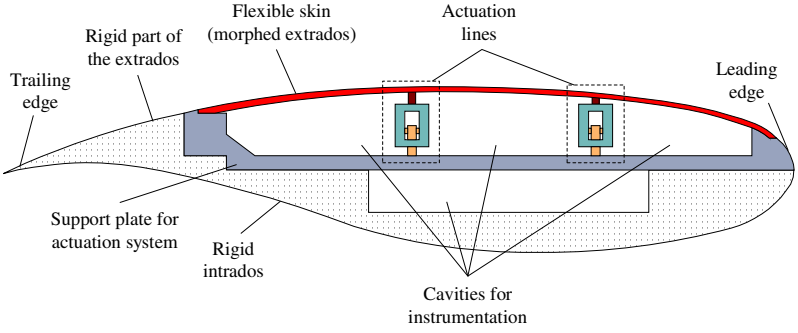


Fig. 1. General architecture of the mechanical model.

Starting from the reference airfoil, depending on different flow conditions, 35 optimized airfoils were calculated for the desired morphed positions of the airfoil. The flow conditions were established as combinations of seven incidence angles (-1˚,-0.5˚, 0˚, 0.5˚, 1˚, 1.5˚, 2˚) and five Mach numbers (0.2, 0.225, 0.25, 0.275, 0.3). Each of the calculated optimized airfoils must be able to keep the transition point as much as possible near the trailing edge [12].

The SMA actuator wires are made of nickel-titanium, and contract like muscles when electrically driven. Also, these have the ability to personalize the association of deflections with the applied forces, providing in this way a variety of shapes and sizes extremely useful to achieve actuation system goals. How the SMA wires provide high forces with the price of small strains, to achieve the right balance between the forces and the deformations, required by the actuation system, a compromise must be established. Therefore, the structural components of the actuation system must be designed to respect the capabilities of actuators to accommodate the required deflections and forces [12].

Each of our actuation lines uses three shape memory alloys wires (1.8 m in length) as actuators, and contains a cam, which moves in translation relative to the structure (on the *x*-axis in Fig. 2). The cam causes the movement of a rod related on the roller and on the skin (on the *z*-axis). The recall used is a gas spring. So, when the SMA is heating the actuator contracts and the cam moves to the right, resulting in the rise of the roller and the displacement of the skin upwards. In contrast, the cooling of the

SMA results in a movement of the cam to the left, and thus a movement of the skin down. The horizontal displacement of each actuator is converted into a vertical displacement at a rate 3:1 (results a cam factor c_f=1/3). From the optimized airfoils, an approximately 8 mm maximum vertical displacement was obtained for the rods, so, a 24 mm maximum horizontal displacement must be actuated [12].

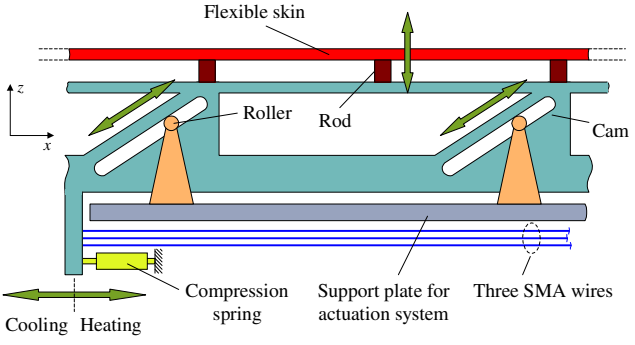


Fig. 2. The actuation mechanism concept.

3 Control Design and Numerical Simulation

The control of SMA actuators can be achieved, in principle, using any method of position control, but the specific properties of SMA actuators, such as hysteresis, the first cycle effect and the long term changes must always be considered. Starting from the established concept of the actuation system the operating schema of the controller can be organized as is presented in Fig. 3. Based on the 35 studied flight conditions a database of the 35 optimized airfoils can be built. Therefore, for each flight condition results a pair of optimal vertical deflections (dY_{1opt}, dY_{2opt}) for the two actuation lines. The SMA actuators morph the airfoil until the obtained vertical deflections of the two actuation lines (dY_{1real}, dY_{2real}) become equals with the required deflections (dY_{1opt}, dY_{2opt}). The morphed airfoil (real airfoil) vertical deflections in the actuation points are measured using two position transducers. The role of the controller is to elaborate an electrical current command signal for the SMA actuators on the base of the error signals (e) between the required vertical displacements and obtained displacements. Because the two actuation lines are identical the designed controller will be valid for both of them.

The first phase of the controller design supposes the numerical simulation of the controlled actuation system. Therefore, a model of SMA actuator was required. In our system a non-linear model was used (a numerical finite element one) build by Prof. P. Terriault using the theoretical model of Lickhatchev [13]. The SMA model has as inputs the initial temperature of the alloy, the electrical current that heats the alloy and the applied force; the outputs are the displacement of the actuator and the temperature of the alloy during functioning. According to this model, to use the shape-changing characteristics the SMA needs to be initialized by an external force, which obliges it to go initially through the transformation phase and further to revert to the initial

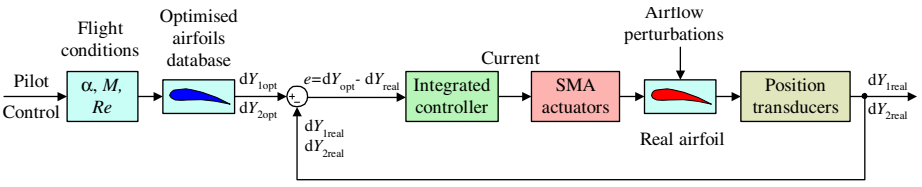


Fig. 3. Operating schema of the SMA actuators control.

phase through the cooling phase. Before these two phases, the control can't be realized, due to the intrinsic behavior of the SMA [13], [14].

Looking the wing as an object moving through the atmosphere, aerodynamic forces are generated between the air and the wing; these forces vary in function of the airflow characteristics (Mach number, Reynolds number and α - angle of attack). Since the aerodynamic forces are suction forces, it tends to lift the skin and to shorten the SMA wire. Against the aerodynamic forces action the elastic force of the flexible skin. A gas spring is needed in order to counteract the aerodynamic forces, so that the resultant force that acts on the SMA wire is given by equation

$$F_{SMA} = F_{spring} + (F_{skin} - F_{aero}) \cdot c_f . \tag{1}$$

To have the premises necessary to initialize the SMA actuators in any conditions, they are loaded by the gas spring even if there are no aerodynamic forces applied on the flexible skin. So, the equation (1) becomes

$$F_{SMA} = (F_{pretension} + k_{spring} \cdot \delta_h) + (k_{skin} \cdot \delta_v - F_{aero}) \cdot c_f , \tag{2}$$

where

$$F_{spring} = F_{pretension} + k_{spring} \cdot \delta_h , \quad F_{skin} = k_{skin} \cdot \delta_v . \tag{3}$$

F_{SMA} is the SMA resultant force, F_{spring} - gas spring elastic force, F_{skin} - elastic force produced by the flexible skin, F_{aero} - aerodynamic force, $F_{pretension}$ - pretension force of the spring, c_f - cam factor (1/3), k_{spring} and k_{skin} are the elastic coefficients of the spring, and of the skin, respectively, δ_h and δ_v are the horizontal and vertical actuated displacements.

Implementing the SMA actuators model in a Matlab S-function, the simulation model in Fig. 4 was obtained. As can be observed, to control the SMA actuators, an adequate electrical current must supply it. The length of the SMA wires is a complex function of the SMA load force and temperature, the last one being influenced by the supplying current in time and by the interaction of the wires with the environment in theirs cooling phase (when the electrical supply is removed) [15].

The block "Mechanical system" in Fig. 4 was modeled accordingly with the equations (1) to (3). As is shown in Fig. 2, the horizontal and the vertical actuated distances are correlated by using the "cam factor" c_f=1/3. Therefore, the aerodynamic and the skin forces (F_{aero} and F_{skin}) are reflected in the SMA load force (F_{SMA}) with the same rate. The gas spring has a preloaded force of 1500 N and a linear elastic coefficient of 2.95 N/mm. In simulations a linear elastic coefficient of approximate 100 N/mm was considered for the skin.

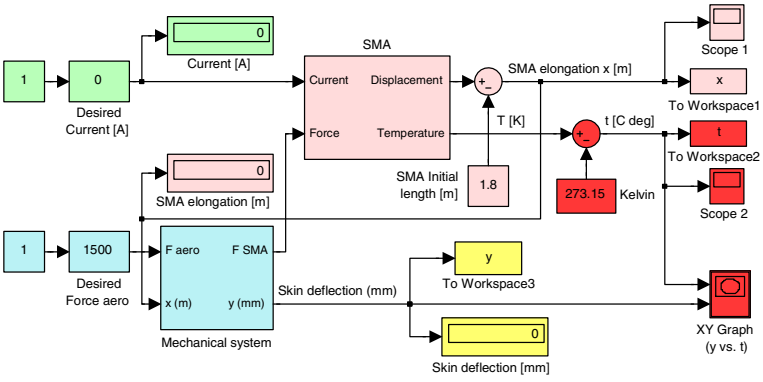


Fig. 4. SMA actuators Simulink model.

The envelope of the SMA actuator, obtained through numerical simulation for different aerodynamic load cases, is shown in Fig. 5.

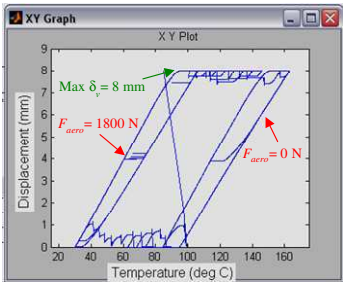


Fig. 5. Simulated envelope of the SMA actuator.

As can be observed from Fig. 5, to obtain a skin maximum vertical displacement (8 mm) in absence of aerodynamic force, it is required a high temperature (approximately 162˚C) in order to counteract the spring force. Because the ability of the SMA wires to contract is dependent upon Joule heating to produce the transformation temperature required, the need in higher temperature is reflected by a need in higher electrical current. Due to the fact that the aerodynamic forces reduce the actuators load the required current and temperature values are decreased; i.e. for F_{aero}=1800 N the need in temperature for the maximum vertical displacement obtaining is approximately 90˚C. From other point of view, the ability of the SMA wires to return to their original configuration is dependant upon the ability of the system to cool the wires. The simulated SMA model offers just summary information about this subject, the proper heating and cooling of the wires being observed only in the moment of a thermodynamic analysis of the physical morphing wing. The system architecture play a big role in the wire cooling by the convection process, and also the performances of the system can be negatively influenced by heat transfer from actuators to the other components.

According with Fig. 3, the integrated controller purpose is to control the SMA actuators in terms of supply electrical current so that to cancel the deviation *e* between the required values for vertical displacements (corresponding to the optimized airfoils) and the real values, obtained from position transducers. As mentioned previously, the design of such controller is difficult considering the strong nonlinearities of the SMA actuators characteristics, nonlinearities significantly influenced by the forces with which they are tense. The chosen design procedure consisted of the following steps:

Step 1: numerical simulation of the SMA model actuators for certain values of the forces in the system;

Step 2: approximation of the system with linear systems in the heating and cooling phases using the System Identification Toolbox of Matlab and the numerical values obtained at the Step 1;

Step 3: the choice of the controller type and its tuning for each of the two SMA actuators phase – heating and cooling;

Step 4: the integration of the two obtained controller in a single one followed by its validation for the general model of the system (non-linear).

Because the team that established the actuation line architecture ([16]) suggested that the pretension force of the gas spring must have the value $F_{pretension}$=1500 N, F_{aero}= 1500 N value was chosen in numerical simulations for the aerodynamic force. Simulating a cooling phase followed by a heating phase with the model in Fig. 4, the blue characteristics depicted in Fig. 6 were obtained. In the first graphical window of the figure is presented the SMA wire length changing in time (δ_h), while in the second window the SMA wire temperature values in the two phases are shown. One observes that a SMA wire dilatation results in the cooling phase, and a wire contraction is obtained in the heating phase. For a horizontal actuation distance of approximately 24 mm the wire temperature reaches a value near by 100˚C. Note are the transient time to reach the steady-state values for the two phases: approximate 60 s for the cooling phase and approximate 40 s for the heating phase. For the steady-state, after the cooling phase, the numerical simulation obtained forces were: F_{SMA}=1000 N, F_{skin}=0 N and F_{spring}=1500 N. In this steady-state the system is relaxed in terms of mechanical and the vertical displacement of the actuator is null. For the steady-state, after the cooling phase, the numerical simulation obtained forces were: F_{SMA}=1337 N, F_{skin}=266.1 N and F_{spring}=1571 N. This steady-state corresponds to the actuation system maximal vertical displacement of approximately 8 mm.

Using the System Identification Toolbox of Matlab and the numerical values characterizing the δ_h response at a series of successive step inputs, two transfer functions (TF) were found for the two SMA phases:

$$H_h(\mathrm{s}) = \frac{0.0177388 \cdot \mathrm{s}^2 + 0.004017 \cdot \mathrm{s} + 0.0241958}{\mathrm{s}^3 - 1.43582 \cdot \mathrm{s}^2 + 0.64742 \cdot \mathrm{s} - 0.001018},$$
$$H_c(\mathrm{s}) = \frac{0.3535 \cdot \mathrm{s} + 0.2672}{\mathrm{s}^2 - 1.9386 \cdot \mathrm{s} + 0.011242}, \tag{4}$$

where H_h(s) and H_c(s) are the transfer functions for heating and cooling phases. The displacements δ_h, corresponding to the linear systems obtained through the two phases

identification, are depicted with red line in Fig. 6. A very good approximation can be also observed for the two phases through the identification in simulated conditions. The previously established transfer functions help to the controller type choice for each phase and to its tuning.

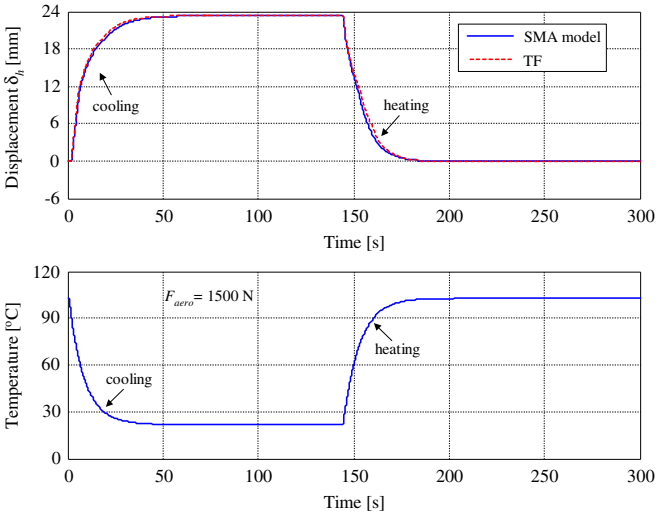


Fig. 6. Actuator displacement and temperature vs. time.

Considering the significance of physical controlled phenomenon, that the SMA wire must be heated to contract and then cooled to dilate by providing an appropriate electrical current by the control block, it is normal that in the cooling phase the actuators not be powered. This phase of cooling may occur in controlling not only a long-term phase, when it ordered a switch between two values of the actuator displacements, but also as a short-lived phase, which occurs when the real value of the deformation exceeds its desired value and is need to cool the actuator wires. On the other hand, it is imperative that in the heating phase actuators to be controlled so that the stationary error of the automatic system to be zero. Therefore, for this phase one opted for a simple architecture of the controller of PI type (proportional-integral). It combines the advantages of proportional type controller, which reduces substantially the overshoot and lead to a short transient time, with the benefits of the integral controller, which cancels the steady-state system error. As a consequence, the resulted controller must behave like a switch between cooling phase and heating phase, situations where the output current is 0 A, or is controlled by a law of PI type. The two phase's interconnection leads to an integrated controller, which can be viewed as a combination of a bi-positional controller (an on-off one) and a PI (proportional-integral) controller.

The input-output characteristic of a bi-positional (on-off) controller can be described by the equation

$$i(t) = \begin{cases} -i_m, \text{ if } e \le 0, \\ i_m, \text{ if } e > 0, \end{cases} \tag{5}$$

where $i(t)$ is the command variable (electrical current in our case) in time, i_m reflects the value of the command and e is the operating error (Fig. 3). The PI controller law is given by

$$i(t) = K_P \cdot e(t) + K_I \cdot \int e(t) \cdot \mathrm{d}t, \tag{6}$$

with K_P - the proportional gain, and K_I - the integral gain. Combining the two controllers in a single one, based on the rules previously mentioned results the control law of the integrated controller as the form

$$i(t) = \begin{cases} 0, \text{ if } e \le 0, \\ K_P \cdot e(t) + K_I \cdot \int e(t) \cdot \mathrm{d}t, \text{ if } e > 0. \end{cases} \tag{7}$$

The optimal tuning of the controller in heating phase was realized using an integral criterion, the error minimum surface criterion, very well known in the literature as Ziegler-Nichols criterion [17]; the tuning methodology is: a) the regulator is considered as a proportional one (P) and it is tuned with respect to the K_P parameter; b) the amplification factor K_P is increased until the response of the automatic system will be self-sustained oscillatory. One memorizes the value K_{P0} of K_P for which the system has an oscillatory behavior and the value of oscillations semi-period (T_0). The optimal values for the parameters of the PI regulator are determined using the relations [17]:

$$K_P = 0.45 \cdot K_{P0}, \ K_I = K_P / (0.85 \cdot T_0). \tag{8}$$

Follows the controller tuning steps the next numerical values for the PI controller parameters were obtained and/or were calculated:

$$\begin{aligned} &K_{P0} = 3984, T_0 = 2.68\text{s}, \\ &K_P = 1792.8, K_I = 787.0061. \end{aligned} \tag{9}$$

As a consequence, the controlled system in heating phase can be modeled with an approximate linear system with the block schema in Fig. 7.

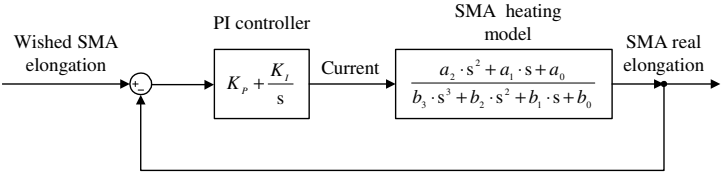


Fig. 7. The block schema with transfer functions of the heating phase linear model.

The parameters $a_0 \div a_2$ and $b_0 \div b_3$ in the schema are the coefficients of the $H_h(s)$ transfer function nominator and denominator in ascending power of s (eq. (4)). The open loop transfer function of the controlled heating phase is

$$H_{ol}(\mathrm{s}) = C_{PI}(\mathrm{s}) \cdot H_h(\mathrm{s}) = \frac{q_3\mathrm{s}^3 + q_2\mathrm{s}^2 + q_1\mathrm{s} + q_0}{b_3\mathrm{s}^4 + b_2\mathrm{s}^3 + b_1\mathrm{s}^2 + b_0\mathrm{s}}, \tag{10}$$

while the closed loop transfer function is

$$H_{cl}(\mathrm{s}) = C_{PI}(\mathrm{s}) \cdot H_h(\mathrm{s}) = \frac{q_3\mathrm{s}^3 + q_2\mathrm{s}^2 + q_1\mathrm{s} + q_0}{r_4\mathrm{s}^4 + r_3\mathrm{s}^3 + r_2\mathrm{s}^2 + r_1\mathrm{s} + r_0}. \tag{11}$$

The included coefficients are

$$\begin{aligned} &q_3 = K_P a_2 = 31.8021,\ \ q_2 = K_P a_1 + K_I a_2 = 21.1622, \\ &q_1 = K_P a_0 + K_I a_1 = 46.5396,\ q_0 = K_I a_0 = 19.0422, \end{aligned} \tag{12}$$

respectively

$$\begin{aligned} &r_4 = b_3 = 1,\ \ r_3 = b_2 + K_P a_2 = 30.3663, \\ &r_2 = b_1 + K_P a_1 + K_I a_2 = 21.8096, \\ &r_1 = b_0 + K_P a_0 + K_I a_1 = 46.5386,\ r_0 = K_I a_0 = 19.0422. \end{aligned} \tag{13}$$

$C_{PI}(\mathrm{s})$ is the transfer function of the PI controller. The poles of the close loop transfer function $H_{cl}(\mathrm{s})$ result with the values

$$\begin{aligned} &p_1 = \text{-}29.6837,\ \ p_1 \in \mathbf{R}_-,\ \ p_4 = \text{-}0.4453,\ \ p_1 \in \mathbf{R}_- \\ &p_2 = \text{-}0.1187 + 1.1943 \cdot \mathrm{i},\ \ p_2 \in \mathbf{C},\ \mathrm{Re}(p_2) \in \mathbf{R}_-, \\ &p_3 = \text{-}0.1187 - 1.1943 \cdot \mathrm{i},\ \ p_3 \in \mathbf{C},\ \mathrm{Re}(p_3) \in \mathbf{R}_-. \end{aligned} \tag{14}$$

One can observe that all poles of the transfer function are placed in the left-hand side of the s-plane, and the obtained system is stable.

In the state-space representation

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t), \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t), \end{aligned} \tag{15}$$

the state matrix A, the input matrix B, the output matrix C and the feed-forward matrix D, were obtained by the forms

$$\begin{aligned} &A = \begin{bmatrix} -30.3663 & -21.8096 & -46.5386 & -19.0422 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\ &B^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},\ C = \begin{bmatrix} 31.8 & 21.1 & 46.5 & 19 \end{bmatrix},\ D = 0. \end{aligned} \tag{16}$$

Evaluating the controllability and observability of the system (P and Q matrices) results

$$P = \begin{bmatrix} 1 & -30.3663 & 900.3025 & -26723.119 \\ 0 & 1 & -30.3663 & 900.3025 \\ 0 & 0 & 1 & -30.3663 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{17}$$

$$Q = \begin{bmatrix} 31.8 & 21.1 & 46.5 & 19 \\ -944.5 & -647 & -1460.9 & -605.5 \\ 28035.4 & 19139.3 & 43352.4 & 17986.3 \\ -832193.6 & -568090.6 & -1286744.7 & -533857.8 \end{bmatrix}; \tag{18}$$

$$\text{rank}(P) = \text{rank}(Q) = \text{system order} = 4. \tag{19}$$

As a consequence, the system is completely controllable and observable.

Based on the previously considerations, the final form of the integrated controller law is

$$i(t) = \begin{cases} 0, \text{ if } e \le 0, \\ 1792.8 \cdot e(t) + 787.0061 \cdot \int e(t) \cdot \mathrm{d}t, \text{ if } e > 0. \end{cases} \tag{20}$$

Introducing the controller in a general block schema, with the non-linear SMA model, the Simulink model in Fig. 8 was obtained for the SMA actuators control (see Fig. 3). The input variable of the schema is the desired skin deflection and the output is the real skin deflection.

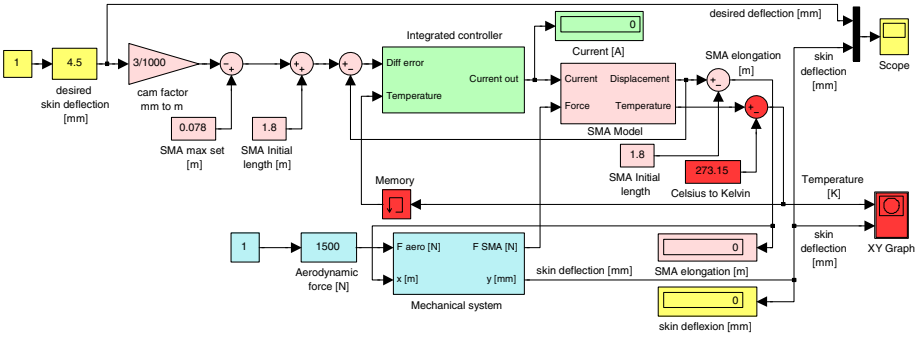


Fig. 8. The simulation model for the controlled SMA actuator with non-linear model.

The "Integrated controller" block contains the implementation of the law described by equation (20) and of the preliminary observations related to the SMA actuators physical limitations in terms of temperature and supplying currents. Its schema is shown in Fig. 9. The inputs of the block are the control error (difference between the desired and the obtained displacements) and the wires temperature, and the output is controlled electrical current applied on the SMA actuators. There are two switches in the schema; the first one chooses one of the two options in the control law (20) and

the second one switching the electrical current value to 0A when the SMA temperature value is over the imposed limit. Also, a current saturation block is used to prevent the current increase over the physical limit supported by the actuation SMA wires.

Loading the simulated model with aerodynamic force F_{aero} = 1500N, the characteristics in Fig. 10 are obtained for a 6 mm step desired skin deflection (δ_v). First of all, can be observed that the controller work good, the transition to the desired steady-state being significantly improved through the integration of the two control law in the equation (20): 1) the amplitudes of oscillations were reduced and the observed oscillations in the SMA temperatures around the steady-state are due only to the thermal inertia of the smart material; 2) the values of the transition time from 0mm to the steady-state values decrease from 20÷25 to approximate 5 s.

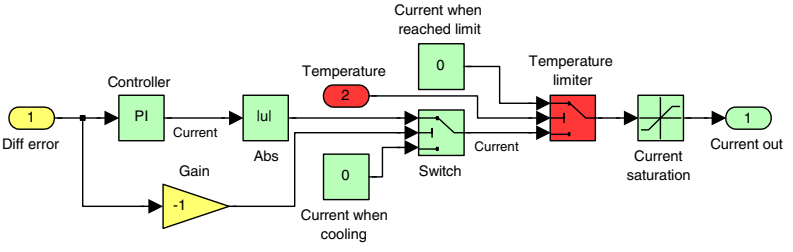


Fig. 9. "Integrated controller" block in Simulink.

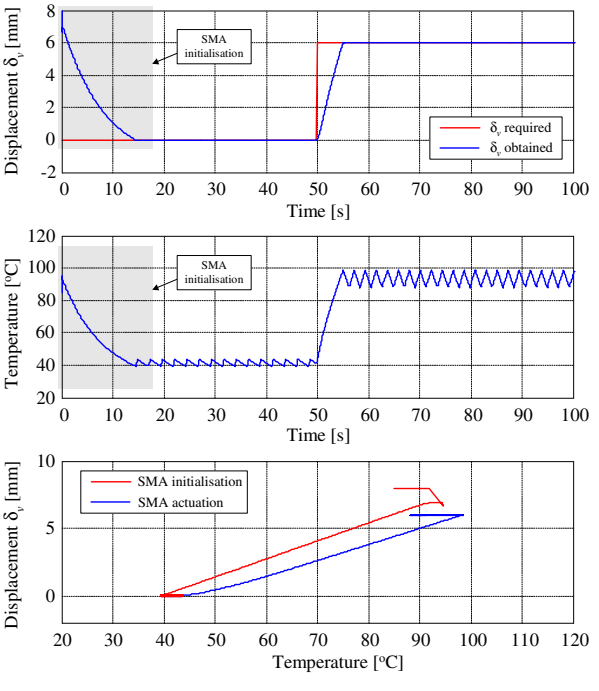


Fig. 10. Response for a step input and F_{aero}=1500 N.

4 Experimental Validation of the SMA Actuators Control for Open Loop Architecture of the Morphing System

Starting from the theoretical and numerical simulation resulted considerations, to implement the controller on the physical model two Programmable Switching Power Supplies AMREL SPS100-33, controlled by Matlab through a Quanser Q8 data acquisition card, were used (Fig. 11) [18]÷[21].

The power supplies have RS-232 and GPIB IEEE-488 as standard features and the technical characteristics: Power 3.3kW, Voltage (dc) 0÷100 V, Current (dc) 0÷33 A. The Q8 data acquisition card has 8 single-ended analog inputs with 14-bit resolution. All 8 channels can be sampled simultaneously at 100 kHz, with A/D conversion times of 2.4 μs/channel, simultaneous sampling and sampling frequencies of up to 350 kHz for 2 channels. Also, the Q8 card is equipped with 8 analog outputs, with software programmable voltage ranges and simultaneous update capability with an 8 μs settling time over full scale (20V) [21].

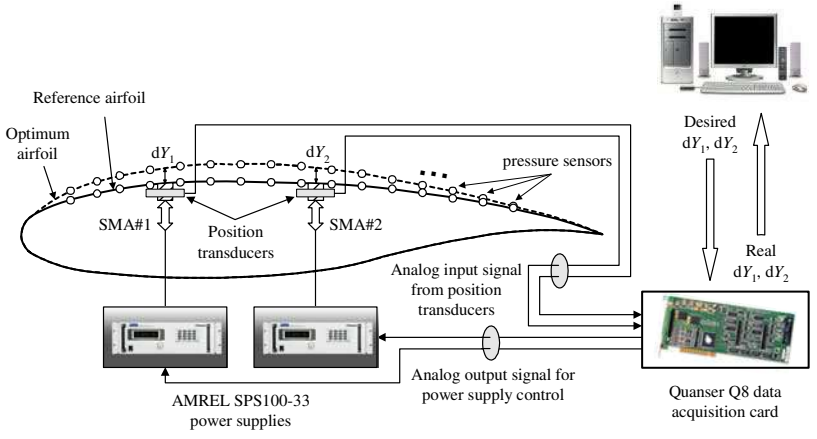


Fig. 11. Physical model operating schema.

The acquisition board was connected to a PC and programmed through Matlab/Simulink R2006b and WinCon 5.2 (Fig. 12).

The input signals were two signals from Linear Variable Differential Transformer potentiometers that indicate the positions of the SMA actuators, and six signals from thermocouples installed on all the SMA wires components. The acquisition sampling time was set to 0.01 second. The outputs channels of the acquisition board were used to control each power supply through analog/external control by use of a DB-15 I/O connector. The current supplied to the actuator was set to be limited at 10 A, and the control signal was set to be 0÷0.6061V (maximum voltage for the power supply is 2 V for 33 A current supply).

The operation principle of the physically implemented controller is relative simple. The initial input, which is the optimized airfoil for any flow condition, is chosen manually by the operator from the computer database through a user interface. Then the displacements (dY_1, dY_2) that are required to be reproduced by the two control

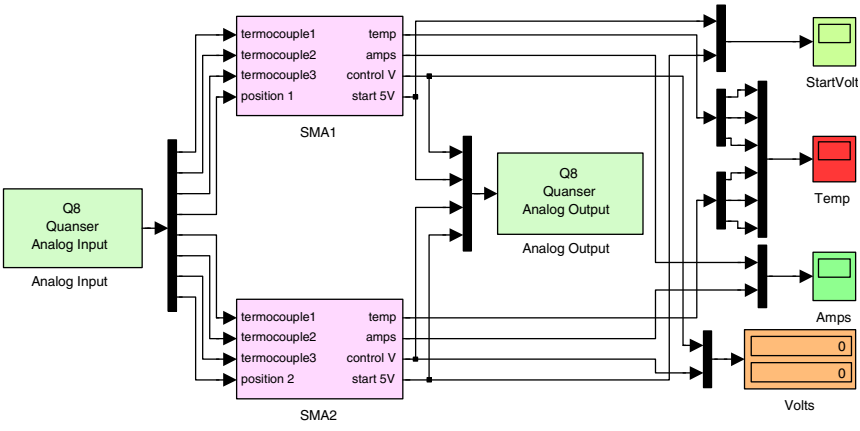


Fig. 12. Simulink actuators control.

points on the flexible skin are sent to the controller. This controller sends an analog signal 0 – 2 V to the power supply that provides a current to the SMA. The SMA will respond accordingly and change its length according to the temperature of the wire. This will result in a change of the actuators positions, which are sensed by the linear variable differential transducer (LVDT). The signal position received from the LVDT is compared to the desired position and the error obtained is fed back to the controller. If the realized position is greater than the desired position the controller will disconnect the control current letting the SMA wire to cool down. During the cooling down process the SMA will maintain its length due to the hysteretic behavior. This effect is taken into account for actuators displacement. Also the controller uses three thermocouples signals from each SMA wire to monitor the temperature of the wires and maintain it below 130˚C, as an upper limit.

4.1 SMA Actuators Control Bench Test Validation

The morphing wing system in the bench test runs is shown in Fig. 13. The gas springs that maintain the SMA wires in tension had a preloaded value of 225 lbs (1000 N) since in the laboratory condition there is no aerodynamic force.

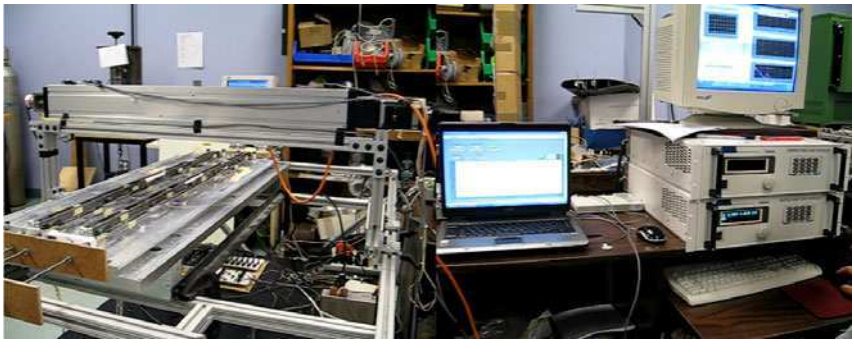


Fig. 13. Morphing wing system in the bench test runs.

After an initial calibration test the calibration gains and constants were established for the two LVDT potentiometers and for the six thermocouples. The calibration test for LVDT potentiometers consisted of several scans of airfoil using a laser beam. On the calibration, the SMA actuators were in "zero setting position" with no power supplied and the skin coordinates were measured using the laser beam that scanned the center line of the wing model. The laser was set to scan the chord of the model on a 370 mm length with a speed of 5 mm per second [21].

In the bench test, the 35 optimized airfoil cases were converted into SMA actuator #1 and #2 vertical displacements which were used as inputs reference for the controller. A typical test run history is shown in Fig. 14 for α=1°, Mach=0.3 flight condition (dY_1=5.22 mm, dY_2=7.54 mm – vertical displacements of the skin in the actuation points).

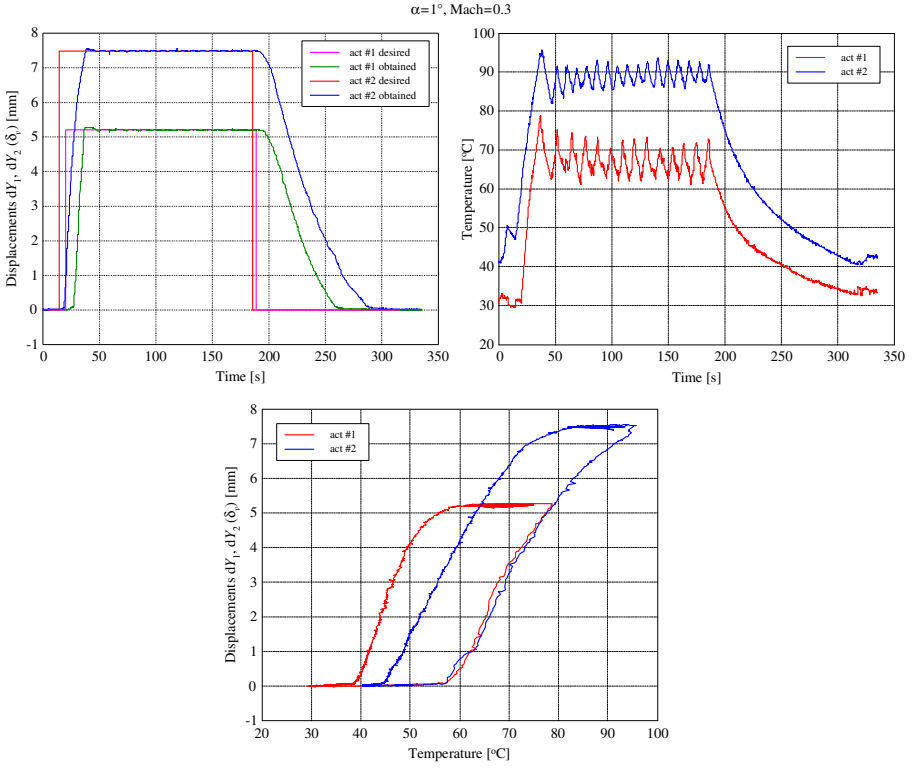


Fig. 14. Bench test for α=1°, M=0.3 flight condition.

On observe that the controllers, in the two actuation lines, work even in zero values of the desired signal because of the gas springs pretension. Also, small oscillations of the obtained deflection are observed around the desired values of the deflections. The amplitude of the oscillations in this phase is due to the LVDT potentiometers mechanical link and to the inertia of the SMA wires, being smallest than 0.05 mm. The heating phase is approximately 9 times more rapidly than the cooling phase; heating

time equals 8 s while the cooling time equals 70 s. There can be observed the differences between the numerical model of the SMA actuators and the physical model.

The bench test results confirmed that the experimental version of the designed integrated controller woks well even in the lab conditions, where no aerodynamic forces are loaded and the preloaded gas springs force is 1000N.

4.2 SMA Actuators Control Validation in Wind Tunnel Tests

Once confirmed the well working of the designed integrated controller through bench test, the next step in our morphing wing project was to validate the controller in a wind tunnel test simultaneously with the transition point real time detection and visualization for all 35 optimized airfoils. The model was tested for all 35 theoretical studied flight conditions, a comparative study being realized around of the transition point position for the reference airfoil and for each optimized airfoil. So, simultaneously with the controller testing, a validation study for the aerodynamic part of the project was realized [21].

The morphing wing system in the wind tunnel runs is shown in Fig. 15.

Fig. 15. Wind tunnel morphing wing model.

The transition detection was made real time using the pressure data obtained from the 32 Kulite and optical pressure sensors. The pressure data acquisition was performed using the IAR-NRC analog data acquisition system which was connected to the 32 sensors. The sampling rate of each channel was 15 kS/s, which allowed a boundary layer pressure fluctuations FFT spectral decomposition up to 7.5 kHz for all channels. The signal was processed by use of Simulink and visualized in real time on the computer screen in dedicated windows.

The pressure signals were analyzed through Fast Fourier Transforms (FFT) decomposition in order to detect the magnitude of the noise in the surface air flow. Subsequently, the data is filtered by means of high-pass filters and processed by calculating the Root Mean Square (RMS) of the signal in order to obtain a plot diagram of the noise in the air flow. This signal processing is necessary to disparate the inherent

electronically induced noise, by the Tollmien-Schlichting that are responsible for triggering transition from laminar flow to turbulent flow. The measurements showed that in processed data the transition appeared at frequencies between 3kHz - 5kHz and the magnitude of pressure variations in the laminar flow boundary layer are of the order 5e-4 Pa (7.25e-8 psi). The transition between laminar flow and turbulent flow was shown by an increase of the pressure variations, reflected also by a strong variation of the pressure signal RMS.

For the wind tunnel test the preloaded forces of the gas springs were reconsidered to the 1500 N because of the presence of the aerodynamic forces on the flexible skin of the wing. In Fig. 16 are presented the control results for test run α=2°, Mach=0.225 (dY_1=5.56 mm, dY_2=7.91 mm).

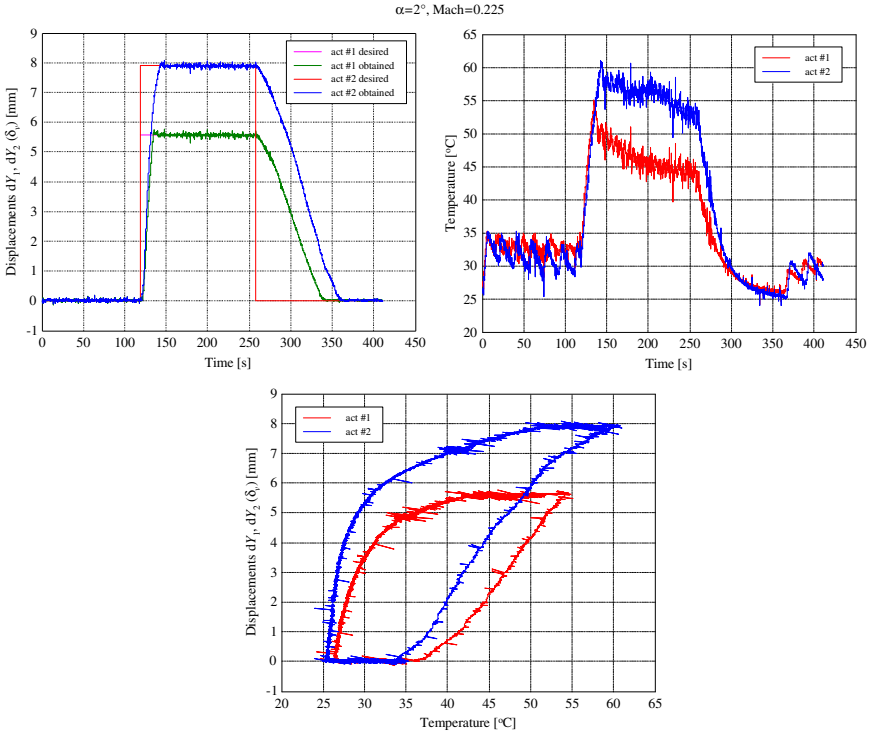


Fig. 16. Wind tunnel test for α=2°, M=0.225 flight condition.

The experimental results show a decrease of the SMA wires work temperatures vis-à-vis of numerically simulated and bench tested cases. An explanation can be the appearing of the aerodynamic forces with particular values for each flight condition. The decrease of these temperatures is a beneficial one taking into account the negative impact of a strong thermal field on the other component of the system, especially on the flexible skin and on the pressure sensors. Also, from the experimental results a high frequency noise influencing the LVDT sensors and the thermocouples instrumentation amplifiers can be observed. The noise sources are the wind tunnel vibrations and instrumentation electrical fields. With this noise, the amplitude of the

actuation error (difference between the realized deflections and desired deflections) is under 0.07 mm, but this don't affecting the transition, which is stable on a sensor with a high RMS spike; Fig. 17 presents the results obtained on the transition monitoring for the run test in Fig. 16.

The actuation line control obtained precision can have some influence in the transition point position detection only if the density of the chord disposed pressure sensors becomes bigger; from the experimental data evaluation one concluded that, even the value of the error is 1 mm around the optimized values, the transition point position on the airfoil surface is not significantly changed.

In Fig. 17 are presented the outputs of the Kulite pressure sensors in leading edge – trailing edge sense of positioning (3 sensors are broken and was not considered in the monitoring phase) and the real time pressure signals RMS for each of these sensors. The left hand column presents the results for reference airfoil, and the right hand column the results for optimized airfoil. The spike of the RMS suggests that we have turbulence on the sensor no. 13, near the trailing edge.

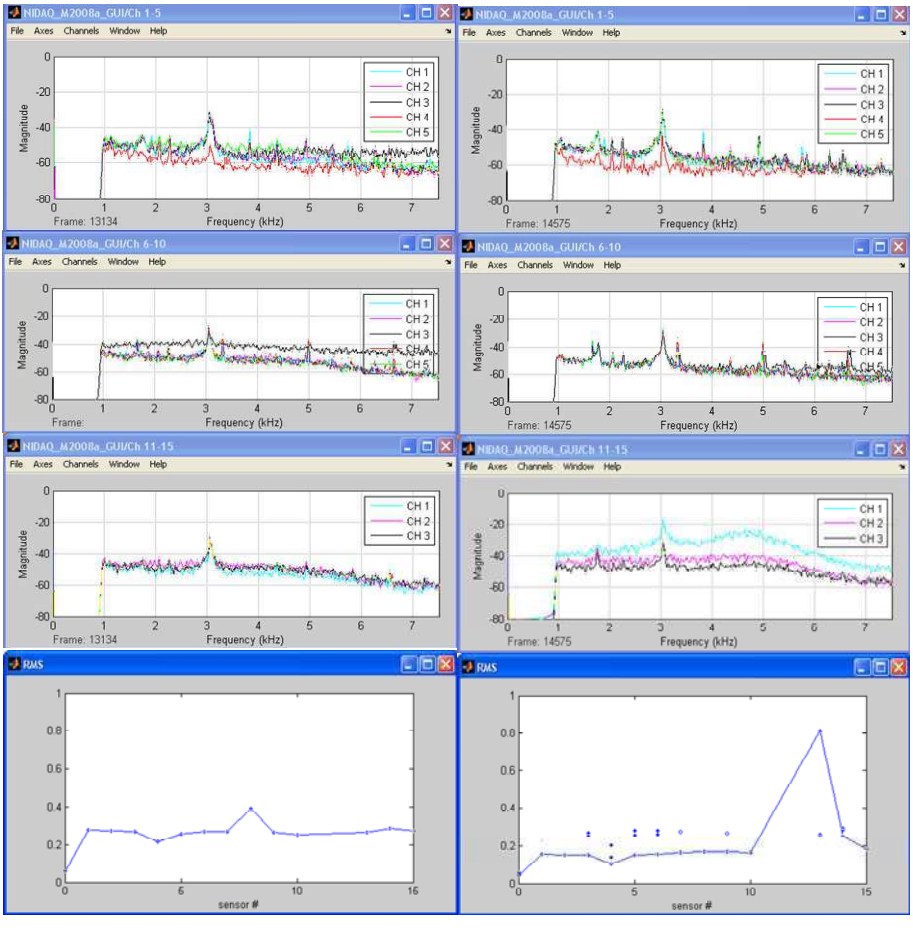


Fig. 17. Results obtained on the transition monitoring for the run test in Fig. 16.

So, the results obtained for the actuators control are very good, the controller fully satisfying the requirements imposed for the project purpose achievement.

5 Wind Tunnel Experimental Validation of the SMA Actuators Control for Closed Loop Architecture of the Morphing System

The next step of the work on the project supposed the development of the closed loop control, based on the pressure information received from the sensors and on the transition point position estimation. Evidently, the closed loop control included, as an internal loop, the actuation lines previous presented controller [22]-[24].

The pressure data acquisition was performed using a NI-DAQ USB 6210 (Fig. 18) card with 16 analog inputs (16 bits, and a total sampling rate of 250 kS/s), 4 digital inputs and 4 digital outputs.

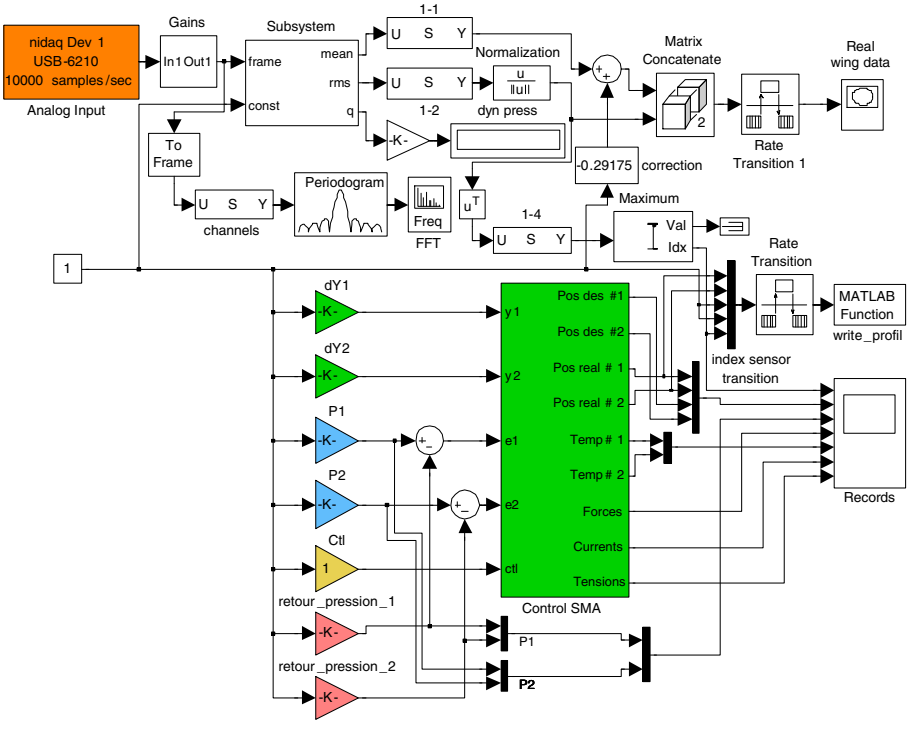


Fig. 18. The closed lop architecture of the morphing wing.

The basic idea of this developed closed loop is to by-pass the necessity of a previously calculated optimized airfoils database, and to generate the optimized airfoils in real time starting from the pressure information received from the sensors

and from the morphing wing main goal (the improvement of the laminar flow over the wing upper surface). In this way, a mixed optimization method was used, between „the gradient ascent" or „hill climbing" method and the „simulated annealing" method. Two variants were tested for the starting point on the optimization map control: 1) dY_1=4 mm, dY_2=4 mm (Fig. 19), and 2) dY_{1opt}, dY_{2opt} of the theoretically obtained optimized airfoil [22]-[24].

In order to obtain the closed loop morphing wing system in this architecture, the software application was developed in Matlab/Simulink and two National Instruments Data Acquisition Cards were used: NI-DAQ USB 6210 and NI-DAQ USB 6229 (Quanser Q8 data acquisition card was removed from this configuration). Also, to realize the control of the morphing wing in closed loop, only the signals from the Kulite pressure sensors were used. In the beginning of wind-tunnel tests, a number of sixteen Kulite sensors were installed, but due to their removal and re-installation during the next two wind tunnel tests, four of them were found defective, therefore a number of twelve sensors remained to be used during the last third wind tunnel tests. So, only twelve Kulite sensors were used for plotting the RMS [22]-[24].

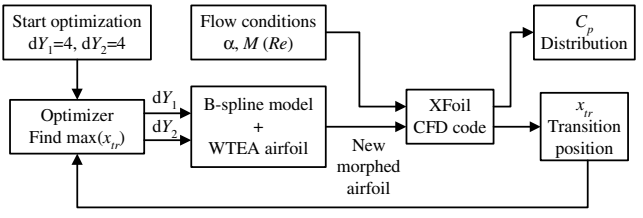


Fig. 19. Optimization logic scheme.

The input channels of the NI-DAQ USB 6210 data acquisition card were connected directly to the IAR-NRC analog data acquisition system, which was connected to the 12 Kulite sensors. The IAR-NRC served as an amplifier and conditioner of the signal at a sampling rate of 15 kS/s. One extra channel was used for the wind-tunnel dynamic pressure q acquisition to calculate the pressure coefficients C_p from the pressure values measured by the 12 pressure sensors. The signal was acquisitioned at a sampling rate of 10 kS/s in frames of 1024 points for each channel, which allowed a boundary-layer pressure fluctuation fast Fourier transform spectral decomposition up to 5 kHz for all channels, at a rate of 9.77 kS/s using Matlab/Simulink software. The plot results were visualized in real time on the computer screen [22]-[24].

The transition between laminar and turbulent flow was detected, as we already said, by means of each pressure signal's RMS. For example, for incidence angle of 0°, and Mach number 0.3 flow condition, the FFT of the pressure signals resulted on the form in Fig. 20. The results in the left side of the figure were obtained for the un-morphed wing, while the results in the right side were obtained for the morphed wing in closed loop on the base of the pressure information received from the sensors. For morphed wing, a detachment of the pressure signals on the sensor number 10

(corresponding to sensor number 13 in the previous configuration, presented at the point 4) was observed, with a peak around of 3 kHz frequency.

To visualize in real time the performances of the morphed wing, a Graphical User Interface (GUI), in which all the aerodynamical and morphing shape information were centralized together with the control buttons of the controlling software, was buildt (Fig. 21). The window shows information about the Mach number, the angle of attack, the airfoil shape of the morphing wing, and the two actuators vertical displacements needed to obtain the desired airfoil shape. In the two sides of the Fig. 20 the GUI shows the coefficients pressure distribution *Cp*'s of the twelve Kulite sensors, and the noise of the signal (RMS) of each pressure signal, for the wing un-morphed position (left side), and for the wing under its morphed position (right side), for α=0°, M=0.3. The lowest RMS plot given in Fig. 21 shows the quantity of the pressure signal noise from each Kulite sensor (red star curve). The RMS red plot in the un-morphed configuration (left side of the Fig. 21) does not show any transition due to the fact that all twelve sensors show the turbulent flow [22]-[24].

As we can see from the „Control SMA" block scheme (Fig. 22), the two SMA actuators had six wires each, which were supplied with power by the two AMREL SPS power supplies, controlled through analog signals by the NI-DAQ USB 6229 data acquisition card. The NI-DAQ, wich have 32 analog inputs (16-bit, 250 kS/s), 4 analog outputs (16-bit, 833 kS/s), 48 digital I/O (32 at up to 1MHz), and 32-bit counters, was connected to the control laptop through an USB connection. The control program implemented in Simulink, used as feedback three temperature signals coming from three thermocouples installed on each wire of the SMA actuator, and a position signal from a LVDT sensor connected to the oblique cam sliding rod of each actuator. The temperature signals served in the overheat protection system that disconnected the current supply to the SMA in case of wire temperature passed over the set limit of 120°C. The position signals served as feedback for the actuator desired position control. Also, the controller activates the power supplies with the needed SMA current values through an analog signal. The control signal of 2 V corresponded to a SMA supplied current of 33 A [22]-[24].

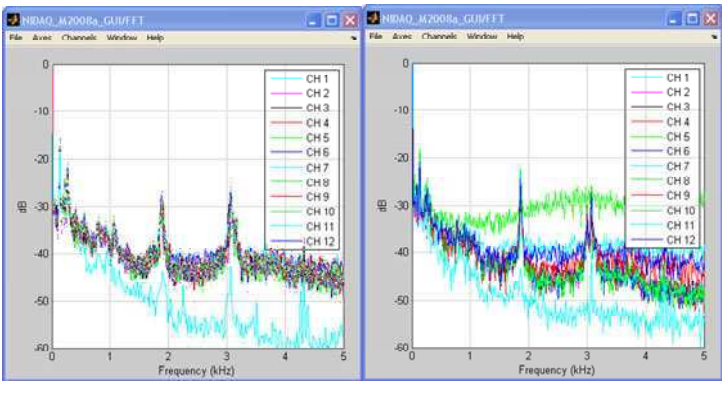


Fig. 20. Pressure signals FFT for un-morphed and morphed wing, for α=0°, M=0.3.

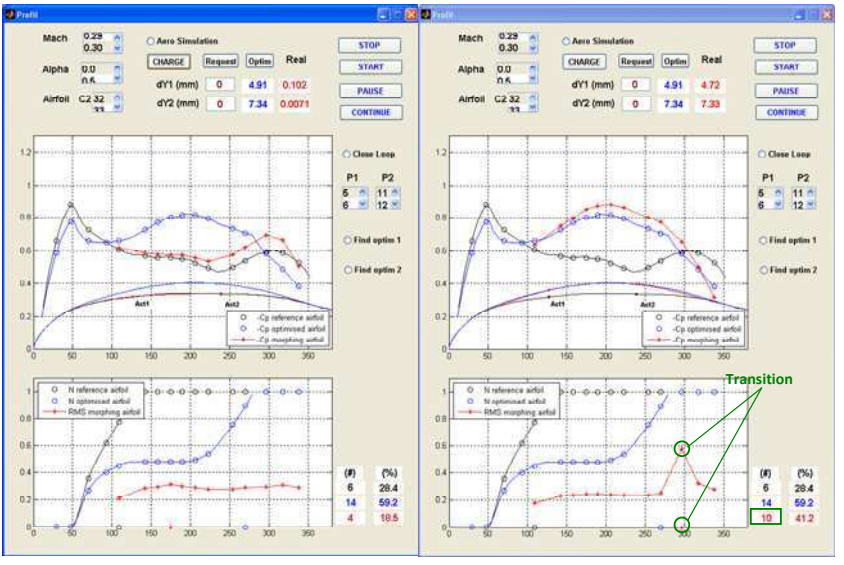


Fig. 21. GUI for un-morphed and morphed wing, for α=0°, M=0.3.

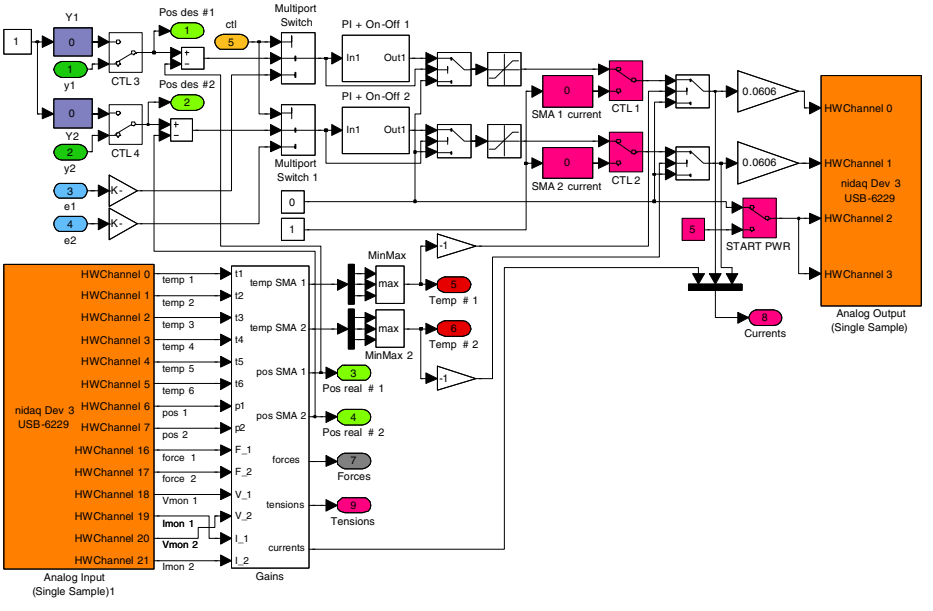


Fig. 22. Control SMA" block scheme.

This new conceived Simulink scheme for closed loop allows users to choose between the manual deflections setting dY_1, dY_2 (open loop), and automatic control of the morphing wing (closed loop). Also, the software allows users to make different

functioning tests on the morphing wing system components (for example, on the power supply).

Our designed SMA controller was embedded in the "PI + On-Off" blocks for first and second actuators, and was used, as an internal loop, for the closed loop control of the morphing wing system. The closed loop results presented in Fig. 23 for α=0°, M=0.3 flow case, confirmed once again that the designed SMA actuators controller worked very well. Also, the results validated the theoretical optimized airfoil obtained by Ecole Polytechnique in Montreal for this case, taking into account that optimization method implemented in the closed loop conducted to a morphed airfoil almost identical with the firs one, and the transition was detected on the same pressure sensor with the open loop case.

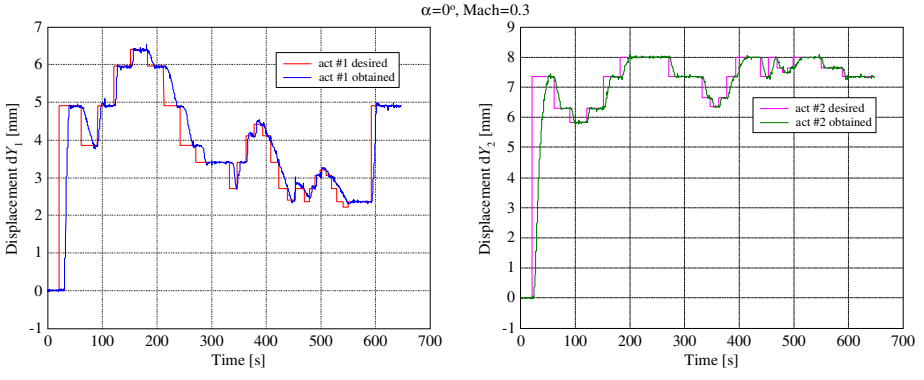


Fig. 23. Wind tunnel test for α=0°, M=0.3 flow condition.

6 Conclusions

The objective of here presented research work is to develop an actuation control concept for a new morphing mechanism using smart materials, like Shape Memory Alloy (SMA), as actuators. These smart actuators modify the upper surface of a wing made of a flexible skin so the laminar to turbulent transition point moves close to the wing airfoil trailing edge.

The designed controller must controls the SMA actuators in terms of supply electrical current so that to cancel the deviation between the required values for vertical displacements (corresponding to the optimized airfoils) and the real values, obtained from position transducers. The envelope of the SMA actuator in Fig. 5, obtained through numerical simulation using model in Fig. 4 for different aerodynamic load cases, confirms that the length of the SMA wires is a complex function of the SMA load force and temperature, the last one being influenced by the supplying current in time and by the interaction of the wires with the environment in theirs cooling phase (when the electrical supply is removed).

As can be observed from Fig. 5, to obtain a skin maximum vertical displacement (8 mm) in absence of aerodynamic force, it is required a high temperature (approximately 162˚C) in order to counteract the spring force. Because the ability of the SMA

wires to contract is dependent upon Joule heating to produce the transformation temperature required, the need in higher temperature is reflected by a need in higher electrical current. Due to the fact that the aerodynamic forces reduce the actuators load the required current and temperature values are decreased; i.e. for F_{aero}= 1800 N the need in temperature for the maximum vertical displacement obtaining is approximately 90˚C.

The final configuration of the integrated controller was a combination of a bi-positional controller (particularly an on-off one) and a PI (proportional-integral) controller, due to the two phases (heating and cooling) of the SMA wires interconnection. The resulted controller must behave like a switch between cooling phase and heating phase, situations where the output current is 0 A, or is controlled by a law of PI type.

Using an integral criterion, the error minimum surface criterion (Ziegler-Nichols), the PI controller for the heating phase was optimal tuned, the resulted values are K_P=1792.8 and K_I=787.0061. Evaluating the systems' performances one observed that the poles of closed loop transfer function of the controlled heating phase resulted with the values (14) are all placed in the left-hand side of the s-plane, so the obtained system is stable. On the other way, the system was found to be completely controllable and observable based on the values established in equations (17)÷(19). So, the final form of the integrated controller law was (20).

Loading the numerically simulated general model (the non-linear one with $F_{pretension}$=1500N) in Fig. 8 with aerodynamic force F_{aero} = 1500N, the obtained characteristics in Fig. 10 confirm that the controller works good, the transition to the desired steady-state being significantly improved through the integration of the two control law in the equation (20): 1) the amplitudes of oscillations were reduced and the observed oscillations in the SMA temperatures around the steady-state are due only to the thermal inertia of the smart material; 2) the values of the transition time from 0mm to the steady-state values decrease from 20÷25 to approximate 5 s.

As second and third validation methods a bench test and a wind tunnel test were performed.

In the bench test phase, the 35 optimized airfoil cases were converted into SMA actuator #1 and #2 vertical displacements which were used as inputs reference for the controller. The characteristics in Fig. 14 (α=1°, Mach=0.3 flight condition) show that the controllers, in the two actuation lines, work even in zero values of the desired signal because of the gas springs pretension. Also, small oscillations of the obtained deflection are observed around the desired values of the deflections. The amplitude of the oscillations in this phase is due to the LVDT potentiometers mechanical link and to the inertia of the SMA wires, being smallest than 0.05 mm. The heating phase is approximately 9 times more rapidly than the cooling phase; heating time equals 8 s while the cooling time equals 70 s.

For the final experimental validation test (wind tunnel test), with real aerodynamic forces load, the 1500 N preloaded forces of the gas springs was reconsidered. From Fig. 16 (α=2°, Mach=0.225) a decrease of the SMA wires work temperatures vis-à-vis of numerically simulated and bench tested cases is observed. The decrease of these temperatures is a beneficial one taking into account the negative impact of a strong thermal field on the other component of the system, especially on the flexible skin and on the pressure sensors. Also, a high frequency noise influencing the LVDT sensors and the thermocouples instrumentation amplifiers can be observed. The noise sources

are the wind tunnel vibrations and instrumentation electrical fields. With this noise, the amplitude of the actuation error (difference between the realized deflections and desired deflections) is under 0.07 mm, but this doesn't affecting the transition, which is stable on a sensor with a high RMS spike like in Fig. 17 and Fig. 21 (for closed loop).

So, the results obtained for the actuators control are very good, the controller fully satisfying the requirements imposed for the project purpose achievement.

Acknowledgements. We would like to thank the Consortium of Research in the Aerospatial Industry in Quebec (CRIAQ), Thales Avionics, Bombardier Aerospace, and the National Sciences and Engineering Research Council (NSERC) for the support that made this research possible. We would also like to thank George Henri Simon for initiating the CRIAQ 7.1 project and Philippe Molaret from Thales Avionics and Eric Laurendeau from Bombardier Aeronautics for their collaboration on this work.

References

1. Shellabarger, N.: National Forecast Overview 2008-2025. Director Aviation Policy and Plans, Federal Aviation Administration (2008), http://www.faa.gov/news/conferences_events/aviation_forecast_2008/agenda_presentation/media/nan_shellabarger.pdf
2. Popov, A.V.: Design of an Active Controller for Delaying the Transition from Laminar Flow to Turbulent Flow over a Morphing Wing in Wind Tunnel. PhD Thesis, Ecole de Technologie Supérieure, Montréal (2009)
3. Chang, P., Shah, A., Singhee, M.: Parameterization of the Geometry of a Blended Wing Body Morphing Wing, Project report, Georgia Institute of Technology, Atlanta, Georgia, USA (April 2009)
4. Smith, K., Butt, J., Spakovsky, M.R., Moorhouse, D.: A Study of the Benefits of Using Morphing Wing Technology in Fighter Aircraft Systems. In: 39th AIAA Thermophysics Conference, Miami, Forida, USA (2007)
5. Hinshaw, T.L.: Analysis and Design of a Morphing Wing Tip using Multicellular Flexible Matrix Composite Adaptive Skins. Master of Science Thesis, Virginia Polytechnic Institute and State University, Virginia (2009)
6. Gonzalez, L.: Morphing Wing Using Shape Memory Alloy: a concept proposal, Final research paper, Texas A&M University, College Station, Texas, USA (2005)
7. Namgoong, H., Crossley, W.A., Lyrintzis, A.S.: Aerodynamic Optimization of a Morphing Airfoil Using Energy as an Objective. In: 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA (2006)
8. Majji, M., Rediniotis, O.K., Junkins, J.L.: Design of a Morphing Wing: Modeling and Experiments. In: AIAA Atmospheric Flight Mechanics Conference and Exhibit, Hilton Head, South Carolina, USA (2007)
9. Ruotsalainen, P., et al.: Shape Control of a FRP Airfoil Structure Using SMA-Actuators and Optical Fiber Sensors. Journal of Solid State Phenomena 144, 196–201 (2009)
10. Khalid, M., Jones, D.J.: Navier Stokes Investigation of Blunt Trailing Edge Airfoils using O-Grids. AIAA Journal of Aircraft 30(5), 797–800 (1993)

11. Khalid, M., Jones, D.J.: A CFD Investigation of the Blunt Trailing Edge Airfoils in Transonic Flow. In: Inaugural Conference of the CFD Society of Canada (1993)
12. Grigorie, T.L., Popov, A.V., Botez, R.M., Mamou, M., Mebarki, Y.: A morphing wing used shape memory alloy actuators new control technique with bi-positional and PI laws optimum combination. Part 1: design phase. In: 7th International Conference on Informatics in Control, Automation and Robotics ICINCO 2010, Funchal, Madeira, Portugal, June 15–18 (2010)
13. Terriault, P., Viens, F., Brailovski, V.: Non-isothermal Finite Element Modeling of a Shape Memory Alloy Actuator Using ANSYS. Computational Materials Science 36(4), 397–410 (2006)
14. Popov, A.V., Labib, M., Fays, J., Botez, R.M.: Closed-Loop Control Simulations on a Morphing Wing. Journal of Aircraft 45, 1794–1803 (2008)
15. Georges, T., Brailovski, V., Morellon, E., Coutu, D., Terriault, P.: Design of Shape Memory Alloy Actuators for Morphing Laminar Wing With Flexible Extrados. Journal of Mechanical Design 31(9) (2009)
16. Mihoc, D.: Teoria si elementele sistemelor de reglare automata. Editura Didactica si Pedagogica, Bucuresti, Romania (1980)
17. Grigorie, T.L., Botez, R.M.: Adaptive neuro-fuzzy inference system-based controllers for smart material actuator modeling. Journal of Aerospace Engineering 223(6), 655–668 (2009)
18. Kirianaki, N.V., Yurish, S.Y., Shpak, N.O., Deynega, V.P.: Data Acquisition and Signal Processing for Smart Sensors. John Wiley & Sons, Chichester (2002)
19. Park, J., Mackay, S.: Practical data acquisition for instrumentation and control systems. Elsevier, UK (2003)
20. Austerlitz, H.: Data acquisition techniques using PCs. Elsevier, USA (2003)
21. Grigorie, T.L., Popov, A.V., Botez, R.M., Mamou, M., Mebarki, Y.: A morphing wing used shape memory alloy actuators new control technique with bi-positional and PI laws optimum combination. Part 2: experimental validation. In: 7th International Conference on Informatics in Control, Automation and Robotics ICINCO 2010, Funchal, Madeira, Portugal, June 15–18 (2010)
22. Popov, A.V., Grigorie, T.L., Botez, R.M., Mamou, M., Mebarki, Y.: Real Time Morphing Wing Optimization Validation Using Wind-Tunnel Tests. AIAA Journal of Aircraft 47(4), 1346–1355 (2010)
23. Popov, A.V., Grigorie, T.L., Botez, R.M., Mamou, M., Mebarki, Y.: Closed-Loop Control Validation of a Morphing Wing Using Wind Tunnel Tests. AIAA Journal of Aircraft 47(4), 1309–1317 (2010)
24. Popov, A.V., Grigorie, T.L., Botez, R.M., Mamou, M., Mebarki, Y.: Controller Optimization in Real Time for a Morphing Wing in a Wind Tunnel. In: 15th IEEE Mediterranean Electrotechnical Conference (MELECON 2010), Valletta, Malta, April 26-28, pp. 107–112 (2010)

Energy-Efficient Parameter Adaptation and Prediction Algorithms for the Estimation of Temperature Development Inside a Food Container

J. Palafox-Albarrán, R. Jedermann, and W. Lang

Microsystems Center Bremen (MCB), University of Bremen
Otto Hahn Allee NW1, D-28359 Bremen, Germany
jpalafox@imsas.uni-bremen.de

Abstract. This paper presents the development and implementation of energy-efficient parameter adaptation for a grey-box model representing the temperature profile in spatial points of the interior of a refrigerated container with the aim to improving the logistics of perishable goods. A mixed linear / non-linear singe-input-single-output grey-box model was selected for accurate prediction of the temperature behavior of the loaded food products. The algorithms were specially modified to reduce the matrix dimensions, implemented in Matlab, and applied to experimental data for validation. Apart from being highly accurate, the predictions comply with the desired figures of merit for the implementation in wireless sensor nodes, such as high robustness against quantization and environmental noise. The OSGi framework, which allows for easy update of software bundles, was selected as basis of the software implementation on the iMote2 as sensor network platform. Performance measurements have shown that this method provides a fast and accurate prediction with high energy efficiency.

Keywords: System identification, Temperature, Organic heat, Feedback-hammerstein, OSGI, Java.

1 Introduction

Research has been done in the past to estimate the temperature profile inside refrigerated containers. Several options have been investigated: mathematical approaches as presented in [1], K-ε models as proposed in [2], and several numerical models as reviewed in [3]. With the exception of [4], in which the effect of the pallets is considered; usually the focus is put on the cold air flow as the main factor governing the temperature pattern inside a container and the effects due the cargo presence is sub estimated.

Babazadeh [5] suggested an approach that takes the effect of the cargo to the temperature into account. He proposed the use of wireless sensor nodes (WSN) to measure the ambient parameters in the surroundings of a spatial point of interest and the use of system identification to estimate the parameters of a linear Multi-Input

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 77–90.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

Single-Output (MISO) system and concluded that in order to have a good estimation, it is necessary to have a high number of training samples and many inputs to the system.

In this paper an alternative Single-Input Single-Output (SISO) grey-box model is presented to predict the temperature inside the container under the presence of perishable goods with the aim of reducing the complexity and preserving the accuracy. The proposed model provides a meaningful description of the factors involved in the physical system including the effect of transporting living goods such as fruits and vegetables. The starting point is based on the physical relations; subsequently, a tuning parameter for the specific case of bananas is found by simulations.

2 Model of the System

The factors affecting the temperature distribution inside a refrigerated container are illustrated in Figure 1. The cold air flows from bottom to top through the gratings in the floor and through the spaces between the pallets, and eventually the air is drawn off the channel between the pallets and the container ceiling.

A naive representation of the container can be done by a SISO linear dynamic system in which the input is the air supply and the output is the spatial point of interest. However, in reality this is only a simple model of the main contributor to the temperature pattern, the air flow. Several other factors affect the speed of the cooling down.

To improve the accuracy of the model, other contributors are considered as well: first is the heat, produced by respiration of living goods such as fruits and vegetables; second is the thermal loss, affecting the correct cooling of the goods; finally, unpredictable temperature variations due to highly changing external climatic conditions during transportation.

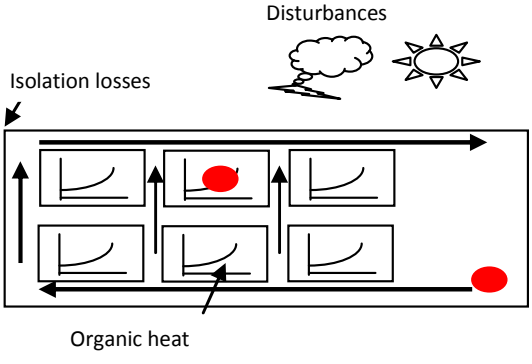


Fig. 1. Factors affecting the temperature inside a refrigerated container.

The linear SISO black-box model which represents the air flow is represented mathematically by a linear dynamic system H, in the discrete domain, given by the Equation 1.

$$H(q^{-1}) = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} \tag{1}$$

Where n_a and n_b are the orders of the system polynomials, $b_1 \dots b_{n_b}, a_1 \dots a_{n_a}$ are the polynomial coefficients, and q is the delay operator in discrete domain.

An attenuator, α ,models the isolation loses of the air supply temperature and is modeled to affect the input of the dynamic system. The external climatic conditions are unknown in advance, therefore considered a statistical process. The output of the Moving Average (MA) process, which is in fact white noise (WN) filtered by the filter C represented in Equation 2 added to the output of the dynamic system, models them.

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \tag{2}$$

To model the organic heat, it is necessary to use experimental data. Figure 2 [6] shows a family of curves for organic heat in the case of bananas. A proportional relationship between of the organic heat and the rippening state is observed.

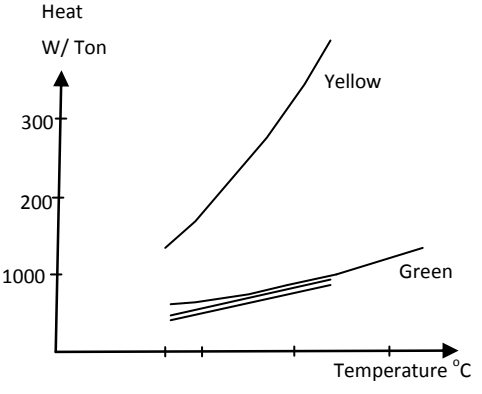


Fig. 2. Heat Production of bananas.

Equation 3 represents the organic heat relation with respect to the temperature. P_{fruit} is the heat production in Watts, γ is a constant which is fixed for a certain type of fruit and rippening-state in $1/^{O}C$, T is the fruit temperature in ^{O}C, and β is a scaling factor which depends of the amount of food and is given in kilograms.

$$P_{fruit} = \beta e^{\gamma T} \tag{3}$$

Finally, the block diagram to represent the input-output relations of all the factors is built. It is shown in Figure 3. The air flow dynamics are represented as a feed-forward block as it is the most important contributor. The isolation losses affect the correct cooling of the goods before the dynamic system and the noise effect has an additive effect on the output.

The contribution of the organic heat depends on the cooling temperature inside the container. Simultaneously, it has a small additive effect in the input of the linear dynamic system as the air flows through the pallets and is slightly warmed. It is

represented by a static exponential feedback. The resulting block diagram, in which a linear dynamic system has a non-linear feedback corresponds to a Feedback-Hammerstein (FH) configuration [7].

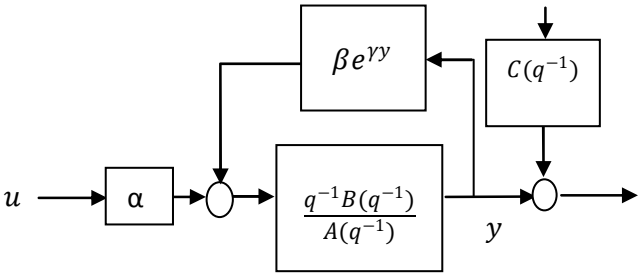


Fig. 3. Model of the system.

3 Parameter Adaptation Algorithm

In [7] a Parameter Adaptation Algorithm (PAA) was developed to identify the parameter-set of a FH system. It uses an intermediate variable $\tilde{y}(t)$ and converts the non-linear system into a pseudo-linear one. Its principal advantage is that the conventional recursive matrix-based linear system identification algorithms as those presented in [9] can be applied to estimate the parameter matrix □. The recursive form of those algorithm is given by Equation 4. Where $\varepsilon(t)$ is the prediction error as described in Equation 5, P(t+1) is an adaptation matrix to perform the minimization of ε using Recursive Least Squares method, and φ(t) is the observation matrix that contains the input and the output data. $\lambda(t+1)$ in Equation 6 is the so called Forgetting Factor (FF).

$$\Theta(\mathrm{t+1})=\Theta(\mathrm{t})+\big(\mathrm{P(t+1)}\varphi(\mathrm{t})\big)^{T}\varepsilon(t) \tag{4}$$

$$\varepsilon(t) = y(t) - \Theta(t)^{T}\varphi(\mathrm{t\text{-}1}) \tag{5}$$

$$\mathrm{P(t+1)}= \frac{P(t)-P(t)\varphi\varphi^{T}\left(\frac{P(t)}{\varphi^{T}P\varphi+\lambda(t+1)}\right)}{\lambda(t+1)} \tag{6}$$

$$\lambda(t+1) = \lambda_{o} * \lambda(t) + 1 - \lambda_{o} \tag{7}$$

Guo [7]considers the non-linearity as a polynomial of order *l* as shown in Equation 8; however, the dimensions of the matrices in the algorithm would be significantly too large to be applied in platforms where power consumption is an important figure of merit.

$$\eta\big(\mathrm{y}(t)\big) = \sum_{k=0}^{l}\mu_{k}\mathrm{y^{k}(t)} \tag{8}$$

In order to reduce the dimensions of the matrices, it was proposed to use the exponential term in Equation 3 instead. γ is to be determined and it remains constant, while β is a parameter to be identified as it depends on the amount of fruit being transported.

The linear term of the Equation 8 needs to be extracted to be included in the polynomial $A^*(q^{-1})$ of the equivalent SISO pseudo-linear system. Expanding it into a Taylor series and rearranging, the summation of the non-linear coefficients of the exponential function can be calculated using Equation 9. The non-linear coefficients and an offset are on the left hand of the equation.

$$\sum_{k=2}^{\infty} \frac{(\gamma y(t))^k}{k!} + 1 = e^{\gamma y(t)} - \gamma y(t) \tag{9}$$

The equivalent pseudo-linear system for an exponential non-linearity is shown in Equation 10.

$$A^*(q^{-1})\mathrm{y(t)} = b_1 \alpha u(t) + b_1 \beta e^{\gamma y(t)} - b_1 \beta y(t) + \frac{B^*(q^{-1})}{b_1} \tilde{y}(t) + \mathrm{C}(q^{-1}) e(t) \tag{10}$$

The resulting coeficients of the polynomials $A^*(q^{-1})$ and $B^*(q^{-1})$ are given by Equation 11 and 12.

$$a_k^* = a_k - (\beta\gamma) b_k \tag{11}$$

$$B^*(q^{-1}) = b_2 q^{-2} + \ \ldots + b_{n_b} q^{-n_b} \tag{12}$$

And the intermediate variable is shown by Equation 13.

$$\tilde{y}(t) = b_1 \left[\alpha u(t) + \ \beta (e^{\gamma y(t)} - \gamma y(t)) \right] \tag{13}$$

The choice of the forgetting factor in the algorithm is often critical. In theory, it must be one that converges. On the other hand, if it is less than one the algorithm becomes more sensitive and the estimated parameter changes quickly making the convergence faster. A more complex solution is to allow it to vary with time, lower than one at the beginning but tending to one.

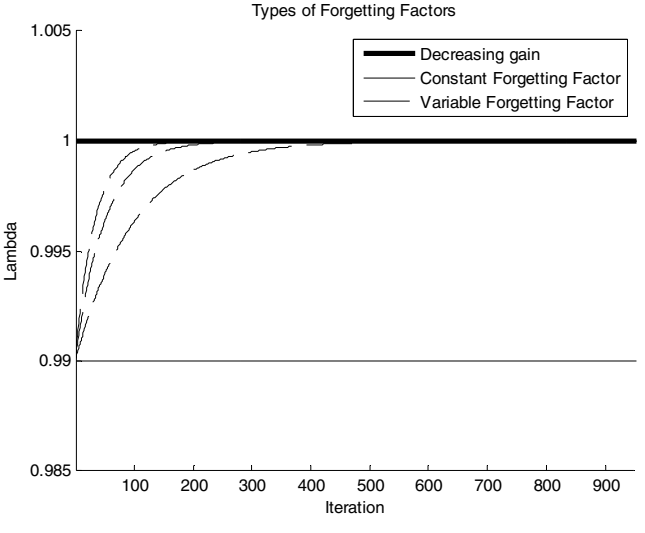


Fig. 4. Types of forgetting factors.

Figure 4 illustrates three different types of FF. The first case is obtained by making λ_o, and $\lambda(t)$ in Equation 7 equal to one. It is called Decreasing Gain (DG). In the second case, the Constant Forgetting Factor (CFF) $\lambda(t)$ is set to a value smaller than one and λ_o set to one. Finally, the Variable Forgetting Factor (VFF) uses a value of λ_o smaller than one and recalculates $\lambda(t)$ for each iteration.

4 Prediction Algorithm

The predictions are made using the estimated parameters in the model. Figure 5 shows experimental data sets from a container transporting bananas. It can be observed how the air supply is kept constant after some days. For the prediction algorithm, $u(t)$ is set to the value of the last sampled input temperature of the parameter adaptation process. Similarly, the initial predicted output value is set to the last acquired value of the output. Equation 14 to 17 describes the prediction algorithm. *m* is the number of iterations used for the PAA.

$$u_{pred}(t) = u(m) \tag{14}$$

$$y_{pred}(m) = y(m) \tag{15}$$

$$y_{pred}(t) = \Theta^T(\mathrm{m})\ \varphi_{pred}\ (\mathrm{t\text{-}1}) \tag{16}$$

$$\tilde{y}_{pred}(t) = b_1\left[\alpha u(m) + \beta(e^{\gamma y_{pred}(t)} - \gamma y_{pred}(t))\right] \tag{17}$$

Table 1. Elements of the elements in the algorithm matrices.

Symbol	Arrangement of the elements into the matrices
$\varphi(\mathrm{t})$	$[-y(t) \cdots -y(t-n_a+1), u(t-1), (e^{\gamma y(t)} - \gamma y(t)), \tilde{y}(t-1) \ldots \tilde{y}(t-n_b), \varepsilon_n(t) \cdots \varepsilon_n(t-n_c+1))]$
$\Theta^T(t)$	$\left[a_1^* \ldots a_{n_a}^*, b_1\alpha, \beta b_1, {b_2}/{b_1} \ldots {b_{n_b}}/{b_1}, c_1 \ldots c_{n_c}\right]$
$\varphi_{pred}(\mathrm{t})$	$[-y_{pred}(t) \cdots -y_{pred}(t-n_a+1), u(m), (e^{\gamma y_{pred}(t)} - \gamma y_{pred}(t)), \tilde{y}_{pred}(t-1) \ldots \tilde{y}_{pred}(t-n_b))]$
$\Theta^T(m)$	$\left[a_1^* \ldots a_{n_a}^*, b_1\alpha, \beta b_1, {b_2}/{b_1} \ldots {b_{n_b}}/{b_1}\right]$

5 Determination of γ

Considering the linear dynamic system H as the most important contributor to the temperature profile, an exponential discrete time decaying system like the one presented in Figure 5 can be described as of the order of one with its unique pole on the real positive axis. The closer the pole to one the higher the delay of the system.

The selection of γ is a key factor in the precision of the algorithms. To find a trustworthy γ parameter that characterizes the respiration heat of bananas. The

presented Feedback-Hammerstein model of linear order one and the FH parameter adaptation and prediction algorithms are run using given experimental data sets. The Mean Squared Error (MSE) of the prediction over *n* samples, equivalent to fifteen days, is stored for several values of γ and fixed number of training days. If the stored values of the MSE are plotted, the local minimums are determined by the observation of the MSE vs. γ curves. In Figure 6, it can be seen that in the above mentioned plot for five days of training and for the data set 1, a local minimum exists for a value γ of 0.0587.

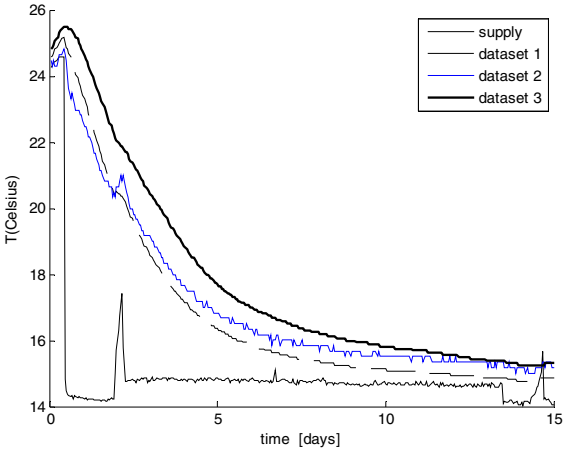


Fig. 5. Banana data sets.

$$MSE = \frac{1}{n}\sum_{t=m}^{n}(y_{real}(t) - y_{pred}(t))^2 \tag{18}$$

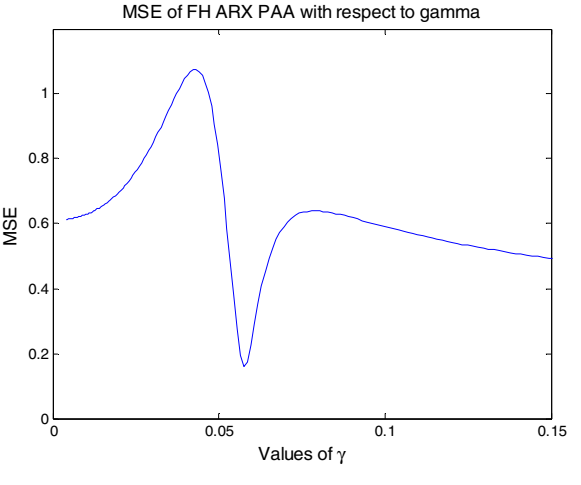


Fig. 6. Prediction accuracy vs. γ.

6 Results

Several figures of merit are considered for validation of the model and algorithms. The accuracy and the speed of convergence are of paramount importance; however, quantization and noise robustness are also highly desirable for implementation in a WSN. Only the linear orders of one and two are considered to avoid computation of complex conjugate poles that would characterize oscillations.

To observe the speed of convergence and the accuracy of the predictions with respect to the number of training days, parameter estimation and a prediction in Matrix form are done (See Table 1) for a fixed number of training days. Subsequently, *MSE vs. Training days* graphs are plotted. Assuming a quantization level of 0.2^{O}C, a Matlab script was written to assign the nearest value of the quantization grid to the input and the output datasets. The results of the predictions using the quantized datasets are overlapped with the results of non-quantized.

Similarly, to determine the noise robustness, MSE versus the signal to noise ratio (SNR) is plotted. Several noise levels of white noise were added to the output of the data set 1, and the resulting signals were applied to PAA and prediction algorithms with fixed number of training days.

$$\mathrm{SNR(dB)}=10\log\left(\frac{Psignal}{Pnoise}\right) \tag{19}$$

Table 2. Summary of simulation results.

	Accuracy		Number of matrix elements	Convergence speed	Quantization Robustness	Critical SNR	Estimation for linear data
	Best Forgetting Factor	Best Linear order					
ARX	CFF	2	3	Bad	Good	47dB	Good
ARMAX	CFF	2	$3+n_c$	Bad	Bad	43 dB	
FH and WN model	DG	1	3	Good	Good	43 dB	Bad
FH and MA model	DG	1	$3+n_c$	Good	Bad	43 dB	

Simulations were done for two types of data sets. First, the experimental data of bananas were used to include the presence of organic heat. Secondly, the data sets corresponding to a cheese experiment, which does not present organic heat, were considered. A summary of all simulation results is presented on Table 2.

6.1 FH vs. Linear Models in the Presence of Organic Heat

From the simulations it is observed in Figures 7(a) and 7(b) that if linear methods are applied to the banana datasets, the accuracy of the results for different sensor positions of are not sufficient. Quantization robustness is improved with the linear order of one and the speed of convergence is better using CFF. Even in the best of cases acceptable prediction accuracy can only be achieved after more than five days of training.

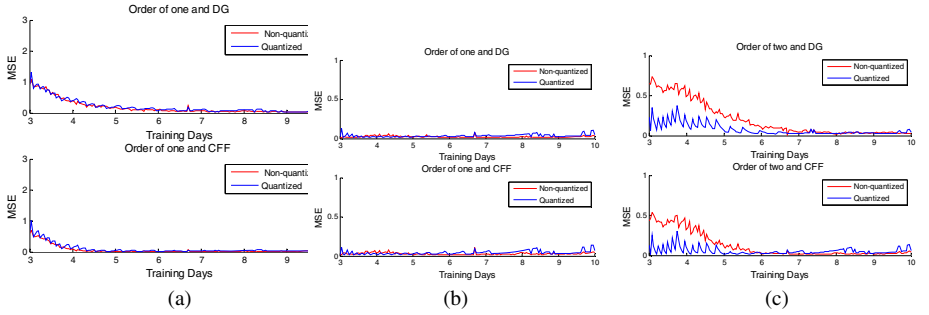


Fig. 7. ARX of order one in the presence of organic heat.

It is also observed in Figures 7(b) and 7(c) that FH identification algorithms are the best to achieve fast convergence speeds. In the best cases, less than 3 days of training is sufficient to achieve good predictions. However, the plots are made for the data from three days onwards to avoid the visualization of the effects in MSE due to the set point variations in the reefer supply temperature. Linear system orders of one are in all cases better than order of two, both in the speed of convergence and the quantization robustness. Decreasing Gain must be optimal to preserve the accuracy and the quantization robustness.

Concerning the noise models, results of the simulation of Feedback-Hammerstein with MA process are worse than when modeled as white noise (WN). It affects the accuracy and the quantization robustness.

6.2 FH vs. Linear Models in the Absence of Organic Heat

In the case of cheese data set, the linear methods accuracy results are better than that of the Feedback-Hammerstein as can be observed in Figure 8. Modeling noise as white gives better quantization robustness than modeling it as MA process.

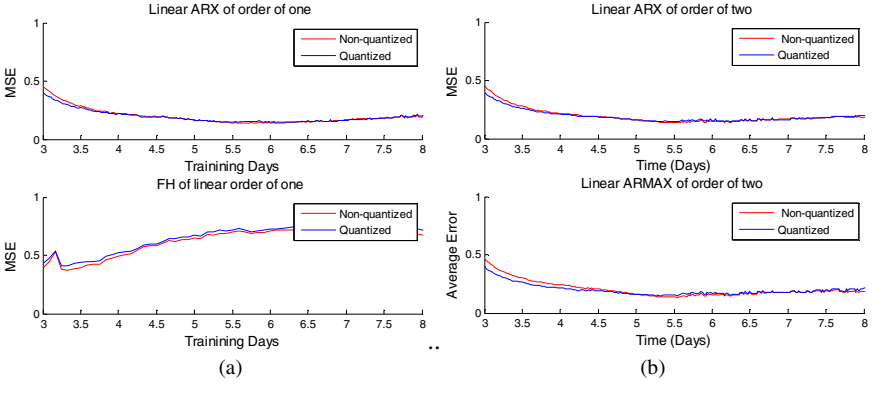


Fig. 8. Comparison of FH and linear methods in the absence of organic heat.

The use of forgetting factors does not have a big impact in the results of ARX predictions; however, Constant Forgetting Factor is slightly better for ARMAX predictions. Linear orders do not affect the simulated predictions, but an order of two is selected because it can model more accurately if the behavior of the system is not purely decaying.

6.3 Noise Robustness

The noise was added to validate FH and linear models; also for both of them the accuracy is compared with and without the MA model. Maximum Signal-to-Noise Ratio to obtain a good prediction is observed to be around 43 dB for all of them with the exception of ARX which has a maximum value of 47 Decibels as shown in Table 2.

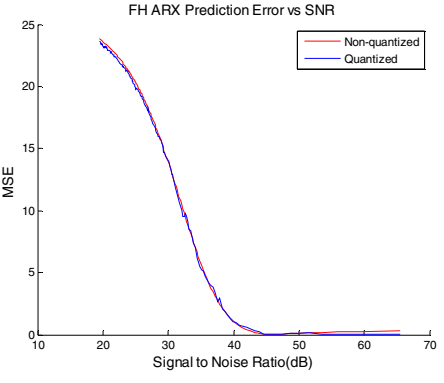


Fig. 9. Noise Robustness for FH method.

7 Prediction Improvement

The described approach was originally developed based on an experiment in 2008 with records for 3 sensors (data set A). Two new data sets with 16 sensors each, which were recorded in 2009 [9] in two separate containers (data set B and C), were used to cross validate the approach.

FH algorithm of linear order of one was applied to all data sets; neither quantization nor forgetting factor is used. For the initial parameter settings, the pole and zero of the feed-forward linear system was set to 0.9 and 0.0; β was set to 2.

The previously obtained value of γ equal to 0.0587 is used to predict the temperature inside the containers for many spatial positions. The results are compared to the predictions for the datasets shown in Figure 5 and resumed in Table 3. A good average is observed for the three containers; however, in some positions the predictions are not as accurate as is observed in the Maximum column.

A second approach is to select γ according to the position of the pallets inside the container. The method to find γ, described previously, is applied to all the new container datasets.

It is observed that an improvement in the accuracy of the predictions can be made if two different values of γ are selected: one for pallets close to the door-end, and one for pallets close to the reefer supply. In Table 3(b) it is resumed the prediction results if values of 0.0525 and 0.055 are set respectively.

Table 3. MSE prediction results.

	MSE prediction results for a unique value of γ			MSE prediction results for values of γ according to the position inside the container		
Container/Result	**Maximum**	**Minimum**	**Average**	**Maximum**	**Minimum**	**Average**
Data set A	0.1893	0.0173	0.0778	0.1893	0.0173	0.0778
Data set B	1.4558	0.0550	0.4130	0.4767	0.0279	0.0946
Data set C	0.8888	0.0101	0.2798	0.5747	0.0201	0.1743

8 Software Bundle Implementation and Energy Consumption Measurements

In a container scenario, energy consumption turns out to be the most limiting factor, and therefore, a priority consideration. Furthermore, the system should be able to install the tuned algorithm according to spatial positions and/or update it according to the new knowledge obtained from experimental results.

The chosen hardware platform is Imote2 [11]. At the core of it is a PXA271 Intel processor, integrated with volatile and non-volatile memory, a power management IC to go to deep-sleep mode, a transceiver, and an antenna. Furthermore, it allows stack ability of additional modules to interconnect additional devices, such as, temperature sensor cards.

Linux operating system is installed, and on it the Java based OSGi [12] framework (formerly Open Source Gateway Initiative) to enable features such as dynamic software updates. OSGi can update and install the so-called software bundles during runtime without interrupting the execution of the remainder of the system.

For the evaluation of power consumption of the iMote2 the supply current was measured over a 1 Ohm resistor in series with the power supply wire, with a test probe connected in parallel to it. Additional hardware modules were detached one by one to measure their individual power consumption. Imote is programmed to run infinitely with the Feedback-Hammerstein training algorithm and the radio is powered at the end of each iteration to observe the last period of training. Regarding the prediction algorithm, it is considered that the iMote2 would only perform the training process locally and send the obtained parameters to a remote server. The server calculates the model prediction. But because the server has only very relaxed energy-constraints compared to the iMote2, an energy analysis is not required.

The amount of *Work* consumed by the algorithms is calculated by equation 20.

$$W = P * t \tag{20}$$

The required CPU time for the algorithm to perform one parameter adaptation iteration was measured. Measurements showed that each iteration takes aproximately 5 ms and consumes 0.9 mJ [13]; according to simulations, it is possible to predict the temperature after 3 days with a sampling rate of one hour that equals to 72 iterations the equivalent of only 64.8 mJ.

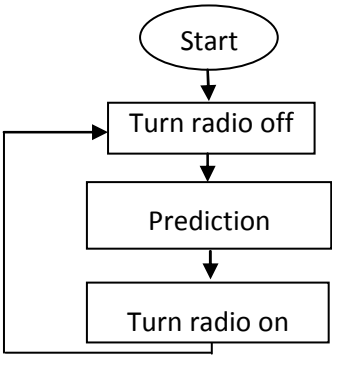


Fig. 10. Program flow diagram to measure FH training time.

9 Conclusions

A model to represent the factors affecting the temperature inside a refrigerated container transporting perishable goods was proposed. It models the effect of organic heat using a static non-linear feedback system, the refrigeration by a linear dynamic feed-forward system, and the disturbances by stochastic processes. This complex model can provide an accurate description of the factors involved in the physical system.

The selected identification method was adapted specifically to reduce the dimensions of the matrices. The non-linear exponential function is used instead of a polynomial to preserve the simplicity of the parameter adaptation and the prediction algorithms. The disadvantage of the simplification is that depending on the kind of fruits to be transported, it is required to tune the algorithm by a correct selection of γ which has to be known in advance. An improvement can be observed in the accuracy of the predictions if γ is set according to the position of the pallets inside the container.

Results concludes that the FH identification algorithm is efficient when the cargo emits organic heat. The method of FH of order 1 is optimal to achieve all figures of merit. It makes accurate predictions only after three days of training and maintains low dimensions of matrices.

However, if the linear method is applied to the banana datasets, a comparable accuracy can only be achieved after more than five days of training. Also, results graphs evidence that when the goods to transport are free of organic heat, such as in the case of cheese, it is preferable to use a linear system instead.

Three days of Feedback-Hammerstein training, which is the minimum to achieve a good prediction, requires in total 64.8 mJ of energy on an iMote2 platform using Linux as Operating System and OSGi as software framework. The latter one allows dynamic software updates and tuning of the algorithm according to the spatial position of a mote in a container or the installation of a linear parameter adaptation algorithm if the cargo does not produce organic heat.

Acknowledgements. The authors would like to express their gratitude to Prof. Rainer Laur, Dirk Hentschel, Mehrdad Babazadeh, and Chanaka Lloyd for all their help. This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes". Further project information can be found at www.intelligentcontainer.com.

References

1. Shaikh, N.I., Prabhu, V.: Mathematical modeling and simulation of cryogenic tunnel freezers. Journal of Food Engineering 80(2), 701–710 (2007)
2. Rouaud, O., Havet, M.: Computation of the airflow in a pilot scale clean room using K-ε turbulence models. International Journal of Refrigeration 25(3), 351–361 (2002)
3. Smale, N.J., Moureh, J., Cortella, G.: A review of numerical models of airflow in refrigerated food applications. International Journal of Refrigeration 29(6), 911–930 (2007)
4. Moureh, J., Flick, D.: Airflow pattern and temperature distribution in a typical refrigerated truck configuration loaded with pallet. International Journal of Refrigeration 27(5), 464–474 (2004)
5. Babazadeh, M., Kreowski, H.-J., Lang, W.: Selective Predictors of Environmental Parameters in WirelessSensorNetworks. International Journal of Mathematical Models and Methods in Applied Sciences 2, 355–363 (2008)
6. Mercantila. Guide to food transport: fruit and vegetables. Mercantila Publ. Copenhagen (1989)
7. Guo, F.: A New Identification Method for Wiener and Hammerstein System. Ph.D. Thesis, Institut für Angewandte Informatik Forschungzentrum Karlsruhe, Karlsruhe (2004)
8. Jedermann, R., Becker, M., Görg, C., Lang, W.: Field Test of the Intelligent Container. In: European Conference on Wireless Sensor Networks EWSN 2010, Coimbra, Portugal, February 16-19 (2010)
9. Landau, I.D., Zito, G.: Digital Control Systems: Design, identification and implementation, 1st edn. Springer, London (2006)
10. Palafox, J.: Prediction of temperature in the transport of perishable goods based in On-line System Identification. M.Sc. Thesis, University of Bremen, Bremen (2009)
11. Crossbow. IMote2 - High-performance Wireless Sensor Network Node datasheet (2007), `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf`
12. Wütherich, G., Hartmann, N., Kolb, B., Lübken, M.: Die OSGi-Service-Platform: eine Einführung mit Eclipse Equinox. dpunkt.verlag, Heidelberg (2008)
13. Jedermann, R., Palafox-Albarrán, J., Jabbari, A., Lang, W.: Energy requirements of intelligent algorithms on wireless platforms. Technical Report Version 1.1 (2010)

List of Abreviations

ARMAX	Auto Regressive Moving Average with External input.
ARX	Auto Regresive with External input.
CFF	Constant Forgetting Factor
DG	Decreasing Gain
FF	Forgetting Factor
FH	Feedback Hammerstein
MA	Moving Average
MISO	Multiple-Input and Single-Output
MSE	Mean Squared Error
OSGi	Open Services Gateway initiative
PAA	Parameter Adaptation Algorithm
SISO	Single-Input and Single-Output
SNR	Signal-to-Noise Ratio
VFF	Variable Forgetting Factor
WN	White Noise
WSN	Wireless Sensor Node

Distributed Network Protocol Parameter Adaptation in Mobile Ad-Hoc Networks

Sven Tomforde, Björn Hurling, and Jörg Hähner

Leibniz Universität Hannover, Institute of Systems Engineering
Appelstr. 4, 30167 Hannover, Germany
{tomforde,hurling,haehner}@sra.uni-hannover.de

Abstract. The performance of a network protocol depends highly on the configuration of its parameter setting. Typical installations consider just a static setup that works well on average. The Organic Network Control (ONC) system provides a possibility to adapt the configurations of network protocols to environmental changes dynamically at runtime. Such rapidly changing conditions can be found e.g. in mobile ad-hoc networks (MANets), since nodes that form the direct neighbourhood are moving and the sending distance is restricted. Thus, MANets are a promising application domain for the ONC system. This chapter describes how ONC is applied to the control of MANet protocols and demonstrates the positive effect caused by the on-line adaptation. Furthermore, a collaboration mechanism is introduced that has been developed to alleviate the impact of some high-effort tasks of the system. Instead of fulfilling tasks on their own, neighboured nodes can share their knowledge and take over some responsibilities from each other.

1 Introduction

Data communication is characterised by a steadily increasing number of interconnected devices and the corresponding transfer load in networks – some of these already reach their limits. Hence, the currently used techniques (e.g. protocols and infrastructure) will not be able to cope with the demand in the near future [6]. Due to these upcoming demands, researchers started to develop new concepts (e.g. for the Internet [19]). Until they will be commonly used, the current problems have to be alleviated and progress towards new paradigms of networking is needed. One approach that optimises the performance of today's technologies is the Organic Network Control (ONC) system [24], which adapts the – usually static – protocol configurations dynamically to environmental changes.

A demanding challenge for the ONC system is the control and adaptation of mobile ad-hoc network (MANet) protocols as they are used in highly dynamic environments. The possible movement of nodes leads to a continuous change of the situation: Neighbours are getting out of reach or joining the sending distance. This does not only lead to complex problems of how to configure the protocol, it also offers high potential for an improvement of the system performance. Within this chapter, we explain how the ONC system is applied to a MANet broadcast protocol and how the overall performance of the MANet system can be increased using ONC.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 91–104.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

ONC combines the advantages of on-line learning with safety aspects by providing a sandbox-learning component to extend the behavioural repertoire of the system in cases of unforeseen situations. This safety concept and the restrictions applied to the learning component are characterised by drawbacks in terms of resource usage and the duration until a new rule is available. To alleviate this effect, a new collaboration mechanism has been developed that allows for knowledge exchange and dynamic load balancing between neighbouring nodes. After discussing the basic application of ONC to the control of MANets, this collaboration mechanism is explained followed by an analysis of its behaviour.

This chapter is based on the initial concept as presented in [23] which demonstrates the application of ONC to MANet-based broadcast protocols. Compared to the previous paper, the approach is extended by introducing a collaboration mechanism that reduces the ONC-implied overhead significantly. The chapter is organised as follows: Section 2 describes the related work and gives an overview of approaches that aim at adapting network protocols dynamically to changing environments. Afterwards, Sec. 3 briefly presents the ONC system and its architectural design. Furthermore, the adaptations of the framework to allow for the control of MANet protocols are described and a short evaluation is used to outline the positive effect of ONC control in MANets. As extension to the previous paper, Section 4 introduces a new collaboration mechanism that increases the learning performance of ONC. Finally, Sec. 5 summarises the presented system and names further research to be done.

2 Related Work

During the last decade, researchers developed several approaches to automatically adapt network protocols to some environmental conditions. In this context, IBM's Autonomic Computing (AC) initiative [8] has been the most prominent platform. New demands in networking and reaching the limits of existing networks brought up challenges that can be covered by adaptation and more dynamic solutions. Besides the ONC system as discussed in this chapter, different directions of research are known to cope with the problem: adaptive protocols, composition of protocol stacks, centralised solutions to adapt protocol configurations, or self-parametrising systems.

2.1 Protocol Composition

The most popular way of enabling adaptivity aspects for data communication protocols is to integrate them into the protocol logic. Prominent examples are the congestion avoidance in TCP [12] and the collision detection on the link layer (e.g. Ethernet-protocols [9]). Considering adaptivity aspects within the protocol logic has drawbacks: The logic has to be changed and a co-operation with standard non-adaptable protocols might not be possible (besides further aspects like limited re-usability, limitation to only one purpose, etc.). Consequently, a research domain called *protocol stack composition* emerged that provides a more generalised solution. Instead of interfering with the protocol logic, it covers the upcoming tasks by exchanging protocols and stacks dynamically [17]. Several different systems have been developed through the years – the most popular representatives might be *Appia* [13], *Cactus* [7], and *Horus* [15]. In contrast to the

ONC system, all the named approaches do not adapt the protocols and their configurations, instead they select protocols with static settings on demand from a pre-defined set. Thus, the behavioural repertoire is limited to this fixed set of actions without any chance to extend it at runtime (e.g. in cases of unforeseen situations or disturbances). Furthermore, the usage of a centralised element which is responsible for observing the whole network and deciding about a global re-composition for the protocol stack is characterised by typical problems of centralised systems – for instance, an adaptation on a per-node basis is not possible.

2.2 Centralised Protocol Adaptation

Due to the constraints of protocol composition in terms of restricted rule-bases, systems have been developed that focus on adapting protocols by considering their variable parameters at runtime - namely the work presented by *Sudame and Badrinath* [21], *Ye et al.* [26], and *Georganopoulos and Lewis* [5]. The authors of the former approach discussed a first TCP- and UDP-based study and defined the need of dynamic adaptation, but detailed examination and a demonstration of the re-usability for other protocols are currently not addressed. *Ye et al.* proposed a centralised system that tackles the optimisation task for each node using a heuristic approach in combination with a simulation of the complete network at server-side. *Georganopoulos and Lewis* discussed their dynamic optimisation framework for the reconfiguration of network protocols. Based again on a central element, they mainly focus on cross-layer optimisation for the protocol stack, but less on considering environmental conditions. Since all three approaches rely on a client-server concept, problems like e.g. bandwidth usage, single point of failure, or accessing local knowledge have to be covered in order to allow for such a division of work between a central server and the particular network nodes.

2.3 Self-parametrising Systems

Recently, part of the research split from the centralised approaches and presented self-organised systems such as ONC, since the drawbacks of centralised systems cannot be covered in settings like MANets. As one example, *Su et al.* [22] presented their approach for a mobility-adaptive self-parametrisation of different unicast and multicast routing protocols. They calculate link expiration times based on locally measured information (position, velocity, and moving direction) and periodic status messages of the neighbours with the additional need of a positioning system like GPS. In contrast, *Ahn et al.* [1] do not depend on the existence of a positioning system. Instead they use the observed number of changes in the 1-hop neighbourhood of the nodes as mobility metric. Similar concepts have been presented by, e.g., *Stanze et al.* [20] or *Boleng* [3]. All approaches named before are restricted to routing protocols in MANets. They are application-specific and cannot be transferred to enable self-parametrisation of protocols from other domains, since they focus on determining the mobility aspect in MANets and use pre-defined actions for observed situations.

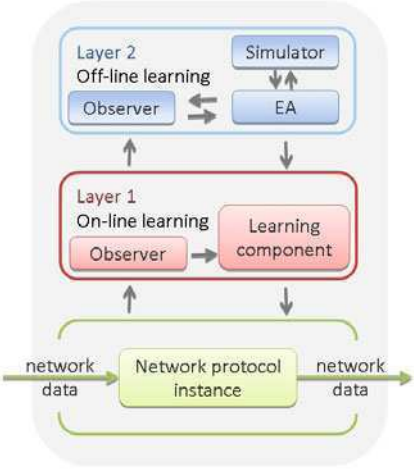


Fig. 1. ONC's architecture.

3 On-Line Adaptation of MANet Protocol Configurations

3.1 Organic Network Control

The Organic Network Control (ONC) system adapts network protocol configurations dynamically at runtime [24]. The system's architecture is founded on the generic Observer/Controller approach as presented by *Richter et al.* in [16] and is organised using three consecutive layers, see Fig. 1. The following part of this chapter describes the particular layers and introduces tasks that have to be fulfilled when applying ONC to a new protocol. The lowest layer (Layer 0) is used to encapsulate an existing network protocol instance, e.g. a broadcast algorithm for mobile ad-hoc networks (MANets) or a Peer-to-Peer protocol. In terms of Organic Computing [18], this controlled network protocol instance is the *System under Observation and Control* (SuOC). ONC provides a basic solution for controlling existing protocols without the need of knowing internals of the protocol or interfering with its logic. However, it is required that the parameters of the protocol can be altered on-line. Additionally, the current status of the protocol instance and its environment have to be observable and accessible.

Layer 1 of the architecture adapts the SuOC dynamically to changes in the environment. It therefore consists of two basic components: an Observer and a Controller. The Observer is responsible for collecting status information about the network protocol instance, its settings, and its environmental conditions. It aggregates the observed figures and augments them with optional further knowledge (prediction values, historic knowledge) and builds a vector describing the current situation at the node. This situation vector serves as input to the Controller which has to fulfil two tasks: evaluate the system's performance within the last evaluation cycle and decide about the next action to be taken. Since ONC shall be able to self-optimise the quality of its adaptation process, the main component of the Controller is a learning technique, which is realised

as an adapted variant of Wilson's Learning Classifier System XCS [25]. This learning component is responsible for choosing the best currently known configuration for the observed situation. The automated learning is achieved by analysing the last actions and updating the measurements for the underlying rules. In cases where no appropriate rules are available, new classifiers need to be created. In contrast to the original XCS algorithm, the ONC-variant is not allowed to create new classifiers (pairs of situation/conditions and parameters/actions) randomly by a Genetic Algorithm. Instead, control is transferred to Layer 2 of the architecture.

Layer 2 is again designed using the Observer/Controller pattern: The Observer monitors Layer 1 and realises the need of a new classifier for a specific situation. The Controller part contains two basic components: a simulator and an Evolutionary Algorithm (EA) [2]. The Controller creates an appropriate simulation scenario from the situation vector and triggers the EA to repeatedly evolve a number of parameter sets for the network protocol. These parameter sets are evaluated in the simulator. This bears the advantage that newly created parameter sets are not directly used in the live system, as this can cause the system to perform badly or even malfunction. Only those parameter sets that qualify in the simulator of Layer 2 are passed back to Layer 1 and may then be applied in the real world – thus, Layer 2 allows for a kind of *sandbox*-learning.

The ONC approach as described before provides a black-box solution to control different types of network protocols. The integration process of a new protocol into the ONC framework requires that an engineer has to fulfil three major tasks: Specify the performance metric, describe the situation (what are the dynamic factors defining the need of an adaptation, e.g. available neighbours and their positions in MANets) accompanied by a distance function between two situations, and provide a simulation model to enable the simulation-based optimisation process of Layer 2. Within the following section, we describe how the ONC system is applied to MANet protocols.

3.2 Application to Mobile Ad-Hoc Networks

The focus of Layer 0 is to integrate a new protocol into the framework. Therefore, the engineer has to describe its observation and control process leading to the need of two interfaces: one for accessing the protocol parameters and one for collecting information about the local status of the system. The former interface enables the framework to adapt the behaviour of the protocol which means the parameter settings can be adapted at runtime. In the latter interface, the engineer has to define what is relevant and influences the protocol's performance - we call this the *situation* the system is in. In a MANet environment, the most important factor influencing the protocol's performance is the distribution of other nodes within its sending and sensing range. Therefore, a sector-based approach as depicted in Fig. 2 has been developed. The radius of the outer circle is equal to the sensing distance (s) of the node, as this is the most remote point where messages of this node can interfere with other ones. Typically, the transmission range for Wireless-LAN based MANets is about 250 meter (half of the sensing distance). The radii of the inner circles have been chosen empirically.

As nodes within the first circle are really close (50m), their exact position does not matter. The second circle (125m) has been partitioned into 4 sectors, the third circle (200m) into 8 sectors, and the forth circle (250m, maximum transmission range) into

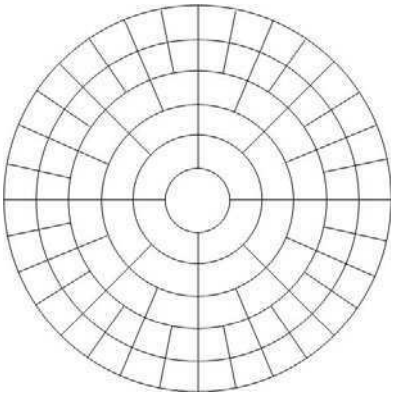


Fig. 2. Environment representation.

16 sectors. The next two circles (375m and 500m) are representing the area within sensing range - with both circles divided into 32 sectors each. We assume that a node is able to determine the current positions of its neighbours in sensing range relative to its own position (e. g. based on GPS, see [14]). Additionally, the node's direction of movement is needed since it has high influence on the best parameter set (e. g. moving to/away from a set of nodes influences the delay). This sector-based approach generalises situations which is necessary to reduce the rule-generation effort.

The Layer 1 component is responsible for the adaptation process and for increasing the system's performance by learning. Again, two aspects have to be considered: a learning feedback and a measurement to compare different situations. To enable the learning feedback, a local fitness or evaluation function is needed. Several metrics have been proposed for analysing MANet protocols at the global scope. Since our current focus is set on MANet-based broadcast algorithms, the standard metrics are *Packet Delivery Ratio* and *Packet Latency* – both cannot be measured locally at each node. But, the goal of the system is the same: we want to reduce the overhead, increase the delivery, and avoid large delays. Based upon the locally available information, this means to reduce the number of forwarded broadcasts and assure the delivery of the broadcast to each node at the same time. Therefore, we introduce the following formula:

$$Fit_x = \frac{\#RecMess}{\#FwMess}$$

The variable x represents the particular network protocol instance. Since a new parameter set has to be applied for a minimum duration to show its performance, we use *evaluation cycles* defining discrete time slots. The duration of the cycle depends on how dynamic an environment is: The faster it changes, the shorter is the cycle to be chosen. For the last evaluation cycle, the function takes into account the sum of all messages being forwarded by all of the neighbours and the node to be evaluated within the last evaluation cycle ($\#FwMess$), and the sum of all messages being received by them ($\#RecMess$).

The second aspect on Layer 1 is the comparison of situations: we need to quantify the distance of two situations. The goal is that more similar situations will receive a low distance value and those having low similarity will receive a high distance value. Based upon the sector-based situation description as introduced before, a measure for the similarity of two entities (A, B) can be defined. To be able to determine the distance, the possible influence of rotation and reflection are deducted initially. Afterwards, the formula for the distance (δ) can be defined with $r \in RADII$ and $s \in SECTORS$ as follows:

$$\delta(A,B) = \sum_{r} \sum_{s} (A_{r,s} - B_{r,s})^2 / r.distance$$

The function $r.distance$ defines the radius size as introduced before (50m, 125m, ...). $A_{r,s}$ gives the number of neighbours within the sector s of radius r for the situation description A. This means that the importance of a node's neighbour decreases if it is situated within an outer radius.

Finally, Layer 2 has to be able to build adequate simulation scenarios out of the information obtained by Layer 1. In ONC, we use the standard network simulation tool *NS-2* [4], but this can easily be exchanged by other solutions. The network simulator *NS-2* has a large set of integrated or available standard protocols, but for recently developed or proprietary protocols a simulation model probably does not exist. The engineer has to provide a realistic model (as it is also used in the development process) which can be adapted to the observed situation by generating an appropriate scenario. The adaptation of the scenario is done using *NS-2*'s configuration interface considering the observed situation at the node. In our case, this means, a randomised instance of the sector-model is created defining the distribution of the neighbouring nodes along with the node's movement direction and transferred to *NS-2* for simulation. After finishing the previously described tasks, ONC is able to control MANet-based protocols.

3.3 Evaluation

The ONC framework is implemented in JAVA. The moving agents communicating via the MANet protocol are simulated using the Multi-Agent Simulation Toolkit *MASON* [11], with each agent's protocol instance representing a SuOC of the architecture as depicted in Fig. 1. The respective Layer 1 Controller is an adapted Learning Classifier System as described in [24]. At Layer 2, the standard network simulation tool *NS-2* [4] is used to evolve new parameter sets in combination with a standard Genetic Algorithm (population size: 15, new children per iteration: 7, mutation rate: 0.2 per child, all children via crossover with fitness-based selection of parents). We use two different simulation tools in order to avoid having exactly the same conditions while optimising rules, since a complete copy of the current situation observed in the real environment within the simulator is not realistic. 100 agents have been created and applied to the simulated area, which has dimensions of 1000 x 1000 meter. The agents move according to a random-waypoint-model. The Physical/Mac layer is an IEEE 802.11 in ad-hoc mode at 2 Mbps.

To demonstrate the performance of ONC controlling MANets, we choose the *Reliable Broadcast Protocol* (R-BCast) as introduced in [10], since this protocol is

representative for the research field of reliable broadcast protocols in MANets. In order to achieve reliability and increase the packet delivery ratio compared to other protocols, additional effort is made by equipping the nodes with extra buffers. These round-robin based buffers are used to store the last p unique packets the particular nodes received. In contrast to other protocols, the R-BCast protocol has significantly more variable parameters and consequently the task to control the protocol is more complex, but it also offers a higher potential benefit due to a dynamic adaptation. The parameters being subject to ONC control actions are: **Delay** (Maximum deceleration time between receiving and forwarding of a message), **Allowed Hello-loss** (Maximum number of Hello-messages, which may be lost until a node is assumed to be out of transmission range), **HelloInterval** (Interval between two Hello-messages), δ **Hello-Interval** (Randomises Hello-Interval), **Packet count** (Number of the last x stored NACK messages), and **Minimum difference** (Minimum difference between NACK messages). Details on the parameters and the protocol can be found in [10].

In order to analyse the performance of the ONC system, the simulation is repeated for two cases under the same restrictions and using the same seeds for the randomised values: a) all nodes are uncontrolled (no ONC system) and use the manually optimised standard configuration of the protocol, and b) all nodes have an own instance of the ONC system to control their protocol configuration. All values presented in the remainder of this section are averaged values received from three runs, where each run has a duration of 10,000 simulated seconds. During one run of the scenario, 17,400 BCast-messages have been simulated. The learning component has been trained using 10 complete runs with different seeds – leading to completely different movements of the nodes and along with these to different situations.

The collected performance figures consist of two basic aspects: the local figures as considered for the learning component at each node (local fitness) and a network-wide view. The latter is the averaged value of local fitness values which is obtained as reference only in the simulator.

Figure 3 plots the system's performance considering only one node. The X-axis describes the simulation time (in simulated seconds) and the Y-axis the measured fitness value. In principle, all simulated nodes show a comparable behaviour; this specific node has been explicitly chosen to demonstrate the typical differences between an ONC-controlled and an uncontrolled node. During the simulation, the node gets separated from the rest of the network (no other nodes within *sending* distance) between simulation seconds 7,350 and 7,700. Within this interval, the fitness is 0 for both cases. But especially these situations demonstrate the benefit of ONC control: The delays have been lengthened so that the node receives more *old* messages when it arrives back in sending distance of another node resulting in a *quicker recovery of the ONC-controlled system*.

Another observation that can be made considering Fig. 3 is the impact of the learning module. To be able to learn, it has to be allowed to try different rules and not to use always the best matching one. E.g. at simulation second 1,800, the learning component tried a rule that results in a performance slightly worse than the standard protocol configuration. These small drawbacks have to be taken into account to achieve an improvement for the system. Averaged over the complete simulation time (10,000

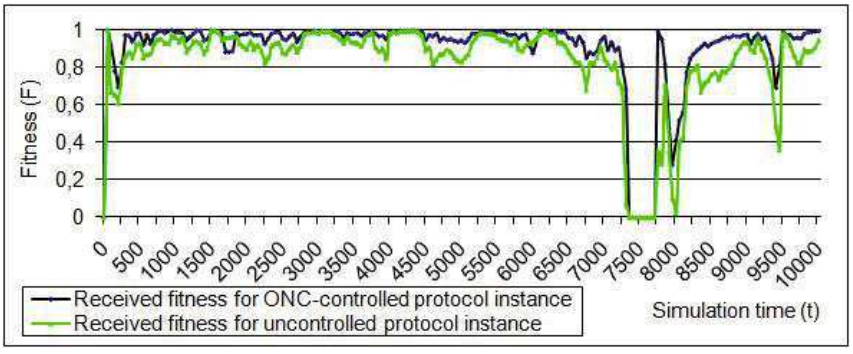


Fig. 3. Comparison for one node: uncontrolled and ONC-controlled.

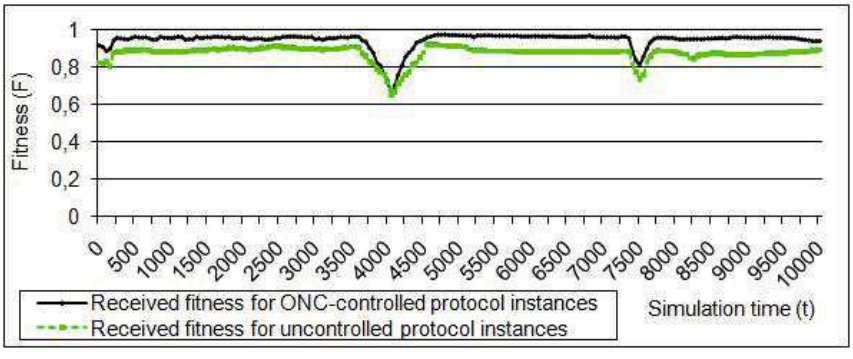


Fig. 4. Network-wide comparison: uncontrolled and ONC-controlled.

simulated seconds), the performance of the protocol instance has been enhanced in terms of the fitness function from 0.8270 (all nodes perform the standard protocol without any adaptation) to 0.8991 which is an increase of 8.71%.

Figure 4 depicts the averaged performance of the network protocol instances on network-level. The averaging leads to the effect that separations of single nodes influences the performance only slightly. Nevertheless, Fig. 4 shows two drops (simulation seconds 3,850 to 4,200 and 7,450 to 7,550). The first drop can be explained by a split of the set of nodes – about 30 nodes are not within sending distance of the rest. Here, two different networks have been established. Within the second drop, again, a larger group (18 nodes) has been separated from the rest of the network. Despite these separation effects, the performance of the system has been increased. When all nodes perform just the standard protocol configuration without any adaptation, the resulting averaged fitness is 0.8760. The same simulation with additional ONC control for all nodes leads to an averaged fitness value of 0.9456 which is an increase of 7.94%. To get back to the intially named typical metrics like *Packet Delivery Ratio* (PDR), *Packet Latency* (PL), and *Overhead*, we can observe a significant improvement. The network-wide measured PDR is slightly increased (about 0.8%), while the PL stays nearly constant. But, the

number of forwarded messages and hence the total overhead in the network has been reduced – which is mainly responsible for the increase in terms of the fitness function.

4 A Collaboration Mechanism for Knowledge Exchange

The performance of the ONC system as described before depends strongly on the existence of *matching* rules. Layer 1's Controller selects rules based on a similarity measurement and applies them to the SuOC. This rule-set is only extended by Layer 2, if no matching rule exists. The creation of new rules using Layer 2 is a time-consuming process (on average $10\ min$ to evolve one rule). Since MANet-nodes are homogeneous and create identical situation descriptions for identical situations, tasks for the Layer 2 component can be transferred to other nodes. Thus, the following part of the chapter introduces a novel collaboration mechanism that allows for knowledge exchange (share rules between neighbours) and dynamic load balancing (share Layer 2 facilities).

4.1 The Collaboration Mechanism

The distributed collaboration mechanism consists of three parts: a local activation (Part 1), a processing of the request by neighbouring nodes (Part 2), and a local analysis that finishes the initial request. For the whole process we assume that all nodes are aware of their direct neighbours and nodes are immediately answering the requests, so that a short delay time after sending the initial request is enough to ensure an accurate feedback. Additionally, it is assumed that all nodes want to collaborate and no bad messages or behaviour occur.

Part 1: Local Activation. The first part of the mechanism is activated if Layer 1's learning component does not contain an adequate classifier for the currently observed situation at one node. The initial version of ONC covers the problem by activating its Layer 2 component to evolve a new rule. The collaboration unit encapsulates this Layer 2 mechanism by involving the neighbours into the rule development process. The main target is to re-use existing knowledge from neighbours. Furthermore, the mechanism provides an instrument to balance the load of Layer 2 tasks within the network. The collaboration mechanism is message-based. Such a message exchanged between the nodes is composed of three parts: (a) the situation description (the condition part of a rule and consequently the corresponding setup information for the simulator at Layer 2), (b) the origin node's current queue length of Layer 2 tasks, and (c) a TimeToLive-value (TTL), indicating whether the message has to be broadcasted to neighbours further away or not:

$$Message : \|HEADER|SituationDesciption|QueueSize|TTL\|$$

In Part 1, the collaboration mechanism generates such a message and sends the request to its direct neighbours, which forward it according to the TTL-value. After initialising the request, the node waits a given delay time (during that the neighbours are processing Part 2) and continues with Part 3. The duration of the delay depends on the channel characteristics and can be estimated as follows: $2 \times (Maximum\ sending\ duration)$

$\times TTL + (processing\ constant)$. Without neighbours in transmission range, the task is directly added to the Layer 2 queue.

Part 2: Process Request. The second part of the process is performed by all neighbours receiving the broadcast message from Part 1 (directly or forwarded). The TTL-part is handled by lower protocol layers – thus, the main task of a node is to check its rule base. Similar to Layer 1's adaptation process, the learning component is asked whether it contains an adequate rule for the submitted situation description or not (currently queued tasks are taken into account). Existing rules are directly send to the inquirer. If no matching rule is available, the received queue-size is compared to the own one: For those cases where the received queue size is higher, the node reserves the spot and answers with an offer reflecting the next possibility for this node to perform the task. Otherwise, a reject is returned to the inquirer.

Part 3: Finalise Mechanism. Finally, the initial node is able to complete the process by analysing the received answers. If the set of answers contains an appropriate rule, this is added to Layer 1's rule-base. Otherwise, the received offers are considered: The best offer will be accepted. If neither a rule nor a suitable offer are available, the task is added to the own Layer 2 queue.

4.2 Evaluation

The evaluation of the collaboration mechanism is based on the NS-2 scenarios developed by *Kunz* in [10] for the initial protocol. The first scenario contains $100\ nodes$ and analyses the load balancing performance: All nodes are using their ONC system, but only $10\ nodes$ are able to evolve new rules by using their Layer 2 (they might have a better power supply). If a node has no "active" node in its direct neighbourhood, it can increase the TTL-value to consider, e.g., a 2-hop neighbourhood. The initial rule base for each node is empty, which means the 10 active nodes have to evolve new rules for all 100 nodes. All results are averages of ten runs. The scenario is processed in three consecutive steps: (1) all 100 nodes determine their current situation (empty rule-base), (2) the collaboration mechanism takes place, and (3) the process is validated, i.e. the experiment is repeated with the newly acquired knowledge provided by the learning mechanism. The learning mechanism was able to respond correctly to a previously learned situation which resulted in the same performance figures at both occasions. Since the population of the classifier system is initially empty, the standard parameter sets are used for the performance evaluation in step 1. In step 2 the queues of all 10 *active* nodes have been processed and the rules distributed. The fitness calculation takes place after processing each step. The overall fitness (same function as in Sec. 3.2) of the system is 0.882 in the first step and increases to 0.926 in the second step (equal to step 3). On average, each active node needs 3 hours and 42 minutes to complete the queue of optimisation tasks (the maximum is 7h and 6min). Due to the abstraction of situations (some situation descriptions are so similar that they have been handled in the same way), an averaged number of 94.3 jobs have been performed in total. Each active node had to process 9.43 jobs on average (the most heavily loaded node had to process 11.33 jobs). The different size of the queue can be explained by the distribution of the

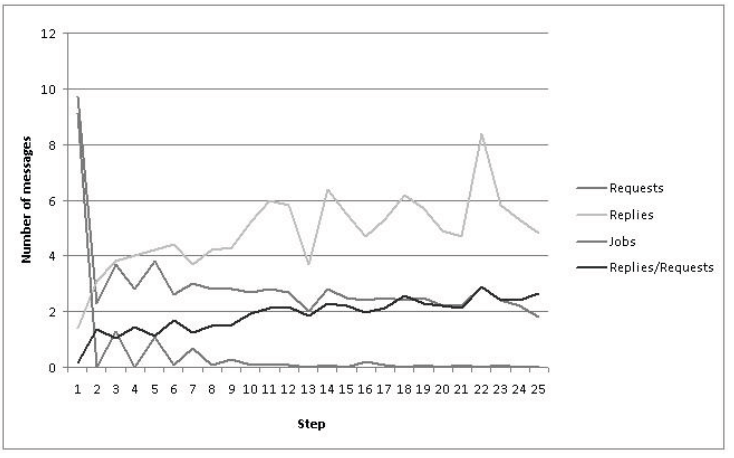


Fig. 5. Messages in the second scenario.

nodes within the environment - two active nodes had only a few neighbours (lowest: 5), while the other eight had to work for comparatively more nodes (up to 14).

The experiment aims at determining the effort in terms of messages that have to be exchanged during the collaboration. The amount of messages sent system-wide is changing during the three consecutive steps: The total number of messages drops from 17922.6 to 17400 on average in step 3. An averaged number of 522.6 messages has been simulated for the collaboration mechanism (Broadcasts 99.6, Unicasts 81.9, and replies 341.1) – which is only a low overhead compared to the 17400 normal BCast messages (2.9%).

In the following part, we investigated the nodes' reactions on abruptly changing situations. Therefore, a static setup of the scenario without node movements has been chosen and nodes were exchanged. Nodes are not moving to avoid the occurrence of new situations and force them to receive the particular rule from their neighbours. The exchange takes place by randomly choosing two nodes and switching their positions. The scenario consists of 25 $steps$ and contains again 100 $nodes$, the broadcast distance is two hops. Initially, all nodes are active nodes (they learn the matching action for their particular situation). Since nodes are not moving, the environment is static and consequently the observed situations stay the same. Within each of the 25 steps, 10 pairs of nodes are randomly selected and their positions exchanged. Probably, these nodes have to receive a matching rule due to the changed situation – although their Layer 2 is deactivated (at least the predecessor at that position has an appropriate rule). Figure 5 demonstrates the message traffic for all steps. The lines for the number of jobs decreases nearly to 0, which means that almost every exchange is followed by a transmission of the rule. In some cases a job is performed, which means that the exchanged pair is not available with a 2-hop broadcast. The number of responses (and also the ratio reply/request) increases continuously as nodes do not delete rules. Hence, the goal of the collaboration mechanism is fulfilled, since the exchange of rules works and the re-creation of rule is avoided.

5 Conclusions

This chapter discussed the application of the Organic Network Control (ONC) system to the control of mobile ad-hoc network (MANet) protocols. Therefore, a brief overview of the system has been given, followed by a description of the adjustments needed to apply ONC to this type of protocols. Afterwards, the benefit of the additional ONC control has been demonstrated using an exemplary MANet-broadcast protocol. We identified the duration of the rule-generation (which is crucial for the safety-based learning approach) as a limiting aspect of the ONC approach. Hence, a dynamic collaboration mechanism has been introduced that decreases the number of performed rule generations significantly and avoids idle times for Layer 2 of the architecture by dynamic load balancing.

Besides more analysis on the behaviour of the ONC system in the presented setup, future work will mainly focus on demonstrating the generic character of the approach by applying it to other types of protocols. Furthermore, the learning and the optimisation component will be considered in detail with a special focus on comparing the currently used techniques to possible alternatives.

References

1. Ahn, S., Shankar, A.U.: Adapting to Route-demand and Mobility in Ad hoc Network Routing. In: 9th Int. Conf. on Network Protocols, pp. 56–66. IEEE, Los Alamitos (2001)
2. Bäck, T., Schwefel, H.P.: Evolutionary Computing: An Overview. In: Proceedings of IEEE Conference of Evolutionary Computing (1996)
3. Boleng, J.L.: Exploiting location information and enabling adaptive mobile ad hoc network protocols. Ph.D. thesis, Golden, CO, USA (2002)
4. Fall, K.: Network Emulation in the Vint/NS Simulator. In: Proc. of the 4th IEEE Symposium on Computers and Communications (ISCC 1999), p. 244 (1999)
5. Georganopoulos, N., Lewis, T.: A framework for dynamic link and network layer protocol optimisation. Mobile and Wireless Communications Summit,16th IST 1–5 (2007)
6. Handley, M.: Why the internet only just works. BT Technology Journal 24(3), 119–129 (2006), http://dx.doi.org/10.1007/s10550-006-0084-z
7. Hiltunen, M.A., Schlichting, R.D., Ugarte, C.A., Wong, G.T.: Survivability through customization and adaptability: the cactus approach. In: Proc. of DARPA Information Survivability Conference and Exposition, vol. 1, pp. 294–307 (2000)
8. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer 36(1), 41–50 (2003)
9. Kleinrock, L., Tobagi, F.A.: Packet Switching in Radio Channels: Part I–Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. IEEE Transactions on Communications 23(12), 1400–1416 (1975)
10. Kunz, T.: Reliable Multicasting in MANETs. Ph.D. thesis, Carleton University (2003)
11. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K.: MASON: A New Multi-Agent Simulation Toolkit. In: Proc. of the 2004 Swarmfest Workshop (2004)
12. Martin, J., Nilsson, A., Rhee, I.: Delay-based congestion avoidance for tcp. IEEE/ACM Trans. Netw. 11(3), 356–369 (2003)
13. Miranda, H., Pinto, A., Rodrigues, L.: Appia: A flexible protocol kernel supporting multiple coordinated channels. In: Proceedings of the 21st Int. Conf. on Distributed Computing Systems, ICDCS 2001 (2001)

14. Pahlavan, K., Krishnamurthy, P.: Principles of Wireless Networks: A Unified Approach. Prentice-Hall, Upper Saddle River (2001)
15. van Renesse, R., Birman, K.P., Maffeis, S.: Horus: a flexible group communication system. Communications of the ACM 39(4), 76–83 (1996)
16. Richter, U., Mnif, M., Branke, J., Müller-Schloer, C., Schmeck, H.: Towards a generic observer/controller architecture for Organic Computing. In: GI Jahrestagung, vol. (1), pp. 112–119 (2006)
17. Rosa, L., Lopes, A., Rodrigues, L.: Appia to R-Appia: Refactoring a Protocol Composition Framework for Dynamic Reconfiguration. Tech. Rep. 1, University of Lisbon, Department of Informatics (1997)
18. Schmeck, H.: Organic Computing – A new vision for distributed embedded systems. In: Proc. of the 8th Int. Symp. on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pp. 201–203 (2005)
19. Siekkinen, M., Goebel, V., Plagemann, T., Skevik, K.A., Banfield, M., Brusic, I.: Beyond the Future Internet–Requirements of Autonomic Networking Architectures to Address Long Term Future Networking Challenges. In: IEEE International Workshop on Future Trends of Distributed Computing Systems, pp. 89–98 (2007)
20. Stanze, O., Zitterbart, M., Koch, C.: Mobility Adaptive Self-Parameterization of Routing Protocols for Mobile Ad Hoc Networks. In: Proceedings of IEEE Wireless Communication and Networking Conference, vol. 1, pp. 276–281 (2006)
21. Sudame, P., Badrinath, B.R.: On providing support for protocol adaptation in mobile wireless networks. Mob. Netw. Appl. 6(1), 43–55 (2001)
22. Sung-Ju, W.S., ju Lee, S., Gerla, M.: Mobility prediction in wireless networks. In: Procceedings of the IEEE MILCOM, pp. 491–495 (2000)
23. Tomforde, S., Hurling, B., Hähner, J.: Dynamic control of mobile ad-hoc networks - network protocol parameter adaptation using organic network control. In: Proc. of the 7th Int. Conf. on Informatics in Control, Automation, and Robotics (ICINCO 2010). INSTICC, pp. 28–35 (2010)
24. Tomforde, S., Steffen, M., Hähner, J., Müller-Schloer, C.: Towards an organic network control system. In: González Nieto, J., Reif, W., Wang, G., Indulska, J. (eds.) ATC 2009. LNCS, vol. 5586, pp. 2–16. Springer, Heidelberg (2009)
25. Wilson, S.W.: Classifier fitness based on accuracy. Evolutionary Computation 3(2), 149–175 (1995)
26. Ye, T., Harrison, D., Mo, B., Sikdar, B., Kaur, H.T., Kalyanaraman, S., Szymanski, B., Vastola, K.: Network Management and Control Using Collaborative On-line Simulation. In: Proceedings of IEEE ICC. IEEE, Helsinki (2001)

Unsupervised Learning of Finite Gaussian Mixture Models (GMMs): A Greedy Approach

Nicola Greggio[1,2], Alexandre Bernardino[1], and José Santos-Victor[1]

[1] Instituto de Sistemas e Robótica, Instituto Superior Técnico
1049-001 Lisboa, Portugal
[2] ARTS Lab - Scuola Superiore S. Anna, Polo S. Anna Valdera
Viale R. Piaggio, 34 - 56025 Pontedera, Italy
nicola.greggio@ieee.org

Abstract. In this work we propose a clustering algorithm that learns on-line a finite gaussian mixture model from multivariate data based on the expectation maximization approach. The convergence of the right number of components as well as their means and covariances is achieved without requiring any careful initialization. Our methodology starts from a single mixture component covering the whole data set and sequentially splits it incrementally during the expectation maximization steps. Once the stopping criteria has been reached, the classical EM algorithm with the best selected mixture is run in order to optimize the solution. We show the effectiveness of the method in a series of simulated experiments and compare in with a state-of-the-art alternative technique both with synthetic data and real images, including experiments with the iCub humanoid robot.

Keywords: Image processing, Unsupervised learning, (Self-adapting) Gaussians mixtures, Expectation maximization, Machine learning, Clustering.

1 Introduction

Nowadays, computer vision and image processing are involved in many practical applications. The constant progress in hardware technologies leads to new computing capabilities, and therefore to the possibilities of exploiting new techniques, for instance considered to time consuming only a few years ago. Image segmentation is a key low level perceptual capability in many robotics related application, as a support function for the detection and representation of objects and regions with similar photometric properties. Several applications in humanoid robots [11], rescue robots [2], or soccer robots [6] rely on some sort on image segmentation [15]. Additionally, many other fields of image analysis depend on the performance and limitations of existing image segmentation algorithms: video surveillance, medical imaging and database retrieval are some examples [4], [12]. Two main principal approaches for image segmentation are adopted: Supervised and unsupervised. The latter one is the one of most practical interest. It may be defined as the task of segmenting an image in different regions based on some similarity criterion among each region's pixels.

One of the most widely used distributions is the Normal distribution. Due to the central limit theorem, any variable that is the sum of a large number of independent

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 105–120.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

factors is likely to be normally distributed. For this reason, the normal distribution is used throughout statistics, natural science, and social science as a simple model for complex phenomena. If we model the entire dataset by a mixture of gaussians, the clustering problem, subsequently, will reduce to the estimation of the gaussians mixture's parameters.

1.1 Related Work

Expectation-Maximization (EM) algorithm is the standard approach for learning the parameters of the mixture model [8]. It is demonstrated that it always converges to a local optimum [3]. However, it also presents some drawbacks. For instance, EM requires an *a-priori* selection of model order, namely, the number of components to be incorporated into the model, and its results depend on initialization. The higher the number of components within the mixture, the higher will be the total log-likelihood. Unfortunately, increasing the number of gaussians will lead to overfitting and to an increase of the computational burden.

Particularly in image segmentation applications, where the number of points is in the order of several hundred thousand, finding the best compromise between precision, generalization and speed is a must. A common approach to choose the number of components is trying different configurations before determining the optimal solution, e.g. by applying the algorithm for a different number of components, and selecting the best model according to appropriate criteria.

Adaptive mixture models can solve the problem of the original EM's model selection. It was originally proposed in 2000 by Li and Barron [10], and subsequently explored in 2003 by Verbeek et al. in [14]. They developed a deterministic greedy method to learn the gaussians mixture model configuration [14]. At the beginning a single component is used. Then, new components are added iteratively and the EM is applied until it reaches the convergence.

Ueda *et Al.* proposed a split-and-merge EM algorithm to alleviate the problem of local convergence of the EM method [13]. Subsequently, Zhang *et Al.* introduced another split-and-merge technique [16]. Merge an split criterion is efficient in reducing number of model hypothesis, and it is often more efficient than exhaustive, random or genetic algorithm approaches. To this aim, particularly interesting is the method proposed by Figueiredo and Jain, which goes on step by step until convergence using only merge operations [5].

1.2 Our Contribution

We propose an algorithm that simultaneously determines the number of components and the parameters of the mixture model with only split operations. In [7] we previously proposed a split and merge technique for learning finite Gaussian mixture models. However, the principal drawbacks were the initialization, with particular regards to the beginning number of mixture classes, and the superimposition of the split and merge operations. The particularly of our new model is that it starts from only one mixture component progressively adapting the mixture by splitting components when necessary.

In a sense, we approach the problem in a different way than [5]. They start the computation with the maximum possible number of mixture components. Although that work is among the most effective to date, it becomes too computationally expensive for image segmentation applications, especially during the first iterations. It starts with the maximum number of components, decreasing it progressively until the whole space of possibilities has been explored, whereas our method starts with a single component and increases its number until a good performance is attained.

1.3 Outline

The paper is organized as follows. In sec. 3 we introduce the proposed algorithm. Specifically, we describe its formulation in sec. 3.1, the initialization in sec. 3.2, the component split operation in sec. 3.4, and the decision thresholds update rules in sec. 3.5. Furthermore, in sec. 4 we describe our experimental set-up for testing the validity of our new technique and in sec. 5 we discuss our results. Finally, in sec. 6 we conclude.

2 Expectation Maximization Algorithm

2.1 EM Algorithm: The Original Formulation

A common usage of the EM algorithm is to identify the *"incomplete, or unobserved data"* $\mathcal{Y} = (\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^k)$ given the couple $(\mathcal{X}, \mathcal{Y})$ - with $\bar{x}$ defined as $\mathcal{X} = \{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^k\}$, also called *"complete data"*, which has a probability density (or joint distribution) $p(\mathcal{X}, \mathcal{Y}|\bar{\vartheta}) = p_{\bar{\vartheta}}(\mathcal{X}, \mathcal{Y})$ depending on the parameter $\bar{\vartheta}$. More specifically, the *"complete data"* are the given input data set $\mathcal{X}$ to be classified, while the *"incomplete data"* are a series of auxiliary variables in the set $\mathcal{Y}$ indicating for each input sample which mixture component it comes from. We define $E^{'}(\cdot)$ the expected value of a random variable, computed with respect to the density $p_{\bar{\vartheta}}(\bar{x}, \bar{y})$.

We define $Q(\bar{\vartheta}^{(n)}, \bar{\vartheta}^{(n-1)}) = E^{'} L(\bar{\vartheta})$, with $L(\bar{\vartheta})$ being the log-likelihood of the observed data:

$$L(\bar{\vartheta}) = \log p_{\bar{\vartheta}}(\mathcal{X}, \mathcal{Y}) \tag{1}$$

The EM procedure repeats the two following steps until convergence, iteratively:

– E-step: It computes the expectation of the joint probability density:

$$Q(\bar{\vartheta}^{(n)}, \bar{\vartheta}^{(n-1)}) = E^{'}[\log p(\mathcal{X}, \mathcal{Y}|\bar{\vartheta}^{(n-1)})] \tag{2}$$

– M-step: It evaluates the new parameters that maximize Q:

$$\bar{\vartheta}^{(n+1)} = arg \max_{\bar{\vartheta}} Q(\bar{\vartheta}^{n}, \bar{\vartheta}^{(n-1)}) \tag{3}$$

The convergence to a local maxima is guaranteed. However, the obtained parameter estimates, and therefore, the accuracy of the method greatly depend on the initial parameters $\hat{\vartheta}^0$.

2.2 EM Algorithm: Application to a Gaussians Mixture

When applied to a Gaussian mixture density we assume the following model:

$$\begin{aligned} p\left(\bar{x}\right) &= \sum_{c=1}^{nc} w_c \cdot p_c\left(\bar{x}\right) \\ p_c\left(\bar{x}\right) &= \frac{1}{\left(2\pi\right)^{\frac{d}{2}}\left|\Sigma_c\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(x-\bar{\mu}_c\right)^T\left|\Sigma_c\right|^{-1}\left(x-\bar{\mu}_c\right)} \end{aligned} \tag{4}$$

where $p_c\left(\bar{x}\right)$ is the component prior distribution for the class c, and with d, $\bar{\mu}_c$ and Σ_c being the input dimension, the mean and covariance matrix of the gaussians component c, and nc the total number of components, respectively.

Consider that we have nc classes C_{nc}, with $p\left(\bar{x}|C_c\right) = p_c\left(\bar{x}\right)$ and $P\left(C_c\right) = w_c$ being the density and the *a-priori* probability of the data of the class C_c, respectively. In this setting, the unobserved data set $\mathcal{Y} = \left(\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^N\right)$ contains as many elements as data samples, and each vector $\bar{y}^i = \left[y_1^i, y_2^i, \cdots, y_c^i, \cdots y_{nc}^i\right]^T$ is such that $y_c^i = 1$ if the data sample x^i belongs to the class C_c and $y_c^i = 0$ otherwise. The expected value of the c^{th} component of the random vector $\bar{y}$ is the class C_c prior probability $E^{'}\left(y_c\right) = w_c$.

Then the E and M steps become, respectively:

<u>*E-step*</u>:

$$\begin{aligned} P\left(y_c^i = 1|\bar{x}^i\right) &= P\left(C_c|\bar{x}^i\right) \\ &= \frac{p\left(\bar{x}^i|C_c\right)\cdot P\left(C_c\right)}{p\left(\bar{x}^i\right)} = \frac{w_c \cdot p_c\left(\bar{x}^i\right)}{\sum_{c=}^{nc} w_c \cdot p_c\left(\bar{x}^i\right)} \\ &\triangleq \pi_c^i \end{aligned} \tag{5}$$

For simplicity of notation, from now on we will refer to $E^{'}\left(y_c|x^i\right)$ as π_c^i. This is probability that the x^k belongs to the class C_c.

<u>*M-step*</u>:

$$\begin{aligned} \bar{\mu}_c^{(n+1)} &= \frac{\sum_{i=1}^N \pi_c^i \bar{x}^i}{\sum_{i=1}^i \pi_c^i} \\ \Sigma_c^{(n+1)} &= \frac{\sum_{i=1}^N \pi_c^i \left(\bar{x}^i - \bar{\mu}_c^{(n+1)}\right)\left(\bar{x}^i - \bar{\mu}_c^{(n+1)}\right)^T}{\sum_{i=1}^N \pi_c^i} \end{aligned} \tag{6}$$

Finally, *a-priori* probabilities of the classes, i.e. the probability that the data belongs to the class c, are reestimated as:

$$w_c^{(n+1)} = \frac{1}{N}\sum_{i=1}^N \pi_c^i, \quad with \;\; c = \{1, 2, \ldots, nc\} \tag{7}$$

3 FASTGMM: Fast Gaussian Mixture Modeling

Our algorithm starts with a single component and only increments its number as the optimization procedure progresses. With respect to the other approaches, our is the one with the minimal computational cost.

The key issue of our technique is looking whether one or more gaussians are not increasing their own likelihood during optimization. Our algorithm evaluates the current likelihood of each single component c as (8):

$$\Lambda_{curr(c)}(\vartheta) = \sum_{i=1}^{N} log\left(w_c \cdot p_c\left(\bar{x}_i\right)\right) \tag{8}$$

In other words, if their likelihood has stabilized they will be split into two new ones and check if this move improves the likelihood in the long run. For our algorithm we need to introduce a state variable related to the state of the gaussian component:

- Its age, that measures how long the component's own likelihood does not increase significantly (see sec. 3.1);

Then, the split process is controlled by the following adaptive decision thresholds:

- One adaptive threshold Λ_{TH} for determining a significant increase in likelihood (see sec. 3.5);
- One adaptive threshold A_{TH} for triggering the split process based on the component's own age (see sec. 3.5);
- One adaptive threshold ξ_{TH} for deciding to split a gaussian based on its area (see sec. 3.4).

It is worth noticing that even though we consider three thresholds to tune, all of them are adaptive, and only require a coarse initialization.

These parameters will be fully detailed within the next sections.

3.1 FASTGMM Formulation

Our algorithm's formulation can be summarized within three steps:

- Initializing the parameters;
- Splitting a gaussian;
- Updating decision thresholds.

Each mixture component c is represented as follows:

$$\bar{\vartheta}_c = \varrho\left(w_c, \bar{\mu}_c, \Sigma_c, \xi_c, \Lambda_{last(c)}, \Lambda_{curr(c)}, a_c\right) \tag{9}$$

where each element is described in tab. I. In the rest of the paper the index notation described in tab. I will be used.

Here, we define two new elements, the area (namely, the covariance matrix determinant) and the age of the gaussians, which will be described later.

Table 1. Symbol notation used in this paper.

Symbol	Element
w_c	*a-priori* probabilities of the class c
$\bar{\mu}_c$	mean of the gaussian component c
Σ_c	covariance matrix of the gaussian component c
ξ_c	area of the gaussian component c
$\Lambda_{last(c)}$	log-likelihood at iteration $t-1$ of the gaussian component c
$\Lambda_{curr(c)}$	log-likelihood at iteration t of the gaussian component c
a_c	*age* of the gaussian component c
c	single mixture component
nc	total number of mixture components
i	single input point
N	total number input points
d	single data dimension
D	input dimensionality

During each iteration, the algorithm keeps memory of the previous likelihood. Once the re-estimation of the vector parameter $\bar{\vartheta}$ has been computed in the EM step, our algorithm evaluates the current likelihood of each single component c as:

If a_i overcomes the age threshold A_{TH} (i.e. the gaussians i does not increase its own likelihood for a predetermined number of times significally - over Λ_{TH}), the algorithm decides whether to split this gaussians depending on whether their own single area overcome ξ_{TH}.

The whole algorithm pseudocode is shown in Algorithm 1.

Algorithm 1. FASTGMM: Pseudocode.

```
1  Parameter initialization;
2  while (stopping criterion is not met) do
3      Λ_curr(c), evaluation, for c = 0, 1, ..., nc;
4      Whole mixture log-likelihood L(ϑ̄) evaluation;
5      Re-estimate priors w_c, for c = 0, 1, ..., nc;
6      Recompute center μ̄_c^(n+1) and covariances Σ_c^(n+1), for c = 0, 1, ..., nc;
       - Evaluation whether changing the gaussians distribution structure;
7      for (c = 0 to nc) do
8          if (a_c > A_TH) then
9              if ((Λ_curr(c) − Λ_last(c)) < Λ_TH) then
10                 a_c+ = 1;
                   - General condition for changing satisfied; now checking
                   those for each component;
11                 if (Σ_c > ξ_TH) then
12                     if (c < maxNumComponents) then
13                         split gaussians → split;
14                         nc+ = 1;
15                         reset ξ_TH ← ξ_TH−INIT / nc;
16                         reset Λ_TH ← L_TH−INIT;
17                         reset a_A, a_B ← 0 - with A, B being the new two
                           gaussians;
18                         return;
19                 ξ_TH = ξ_TH · (1 + α · ξ);
20     Optional: Optimizing selected mixture;
```

3.2 Parameters Initialization

The decision thresholds $(\cdot)_{INIT}$ will be initialized as follows:

$$\begin{aligned} \xi_{TH-INIT} &= \xi_{data}; \\ L_{TH-INIT} &= k_{LTH}; \\ A_{TH-INIT} &= k_{ATH} \end{aligned} \tag{10}$$

with k_{LTH} and k_{ATH} (namely, the minimum amount of likelihood difference between two iterations and the number of iterations required for taking into account the lack of a likelihood consistent variation) relatively low (i.e. both in the order of 10, or 20). Of course, higher values for k_{LTH} and smaller for k_{ATH} give rise to a faster adaptation, however adding instabilities.

At the beginning, before starting with the iterations, ξ_{TH} will be automatically initialized to the Area of the whole data set - i.e. the determinant of the covariance matrix relative to all points, as follows:

$$\begin{aligned} \mu_{data,d} &= \frac{1}{N}\sum_{i}^{N} x_d^N \\ \Sigma_{data,i} &= \langle \bar{x}_i - \bar{\mu}_{data} \rangle \langle \bar{x}_i - \bar{\mu}_{data} \rangle^T \end{aligned} \tag{11}$$

where N is the number of input data vectors $\bar{x}$, and D their dimensionality.

3.3 Gaussian Components Initialization

The algorithm starts with just only one gaussian. Its mean will be the whole data mean, as well as its covariance matrix will be that of the whole data set.

That leads to a unique starting configuration.

3.4 Splitting a Gaussian

When a component's covariance matrix area overcomes the maximum area threshold ξ_{TH} it will split. As a measure of the area we adopt the matrix's determinant. This, in fact, describes the area of the ellipse represented by a gaussian component in 2D, or the volume of the ellipsoid represented by the same component in 3D.

It is worth noticing that the way the component is split greatly affects further computations. For instance, consider a 2-dimensional case, in which an *elongated* gaussian is present. Depending on the problem at hand, this component may approximating two components with diverse configurations: Either covering two smaller data distribution sets, placed along the longer axis, or two ovelapped sets of data with different covariances, etc. A reasonable way of splitting is to put the new means at the two major semi-axis' middle point. Doing so, the new components will promote non overlapping components and, if the actual data set reflects this assumption, it will result in faster convergence.

To implement this split operation we make use of the singular value decomposition. A rectangular n x p matrix A can be decomposed as $A = USV^T$, where the columns

of U are the left singular vectors, S (which has the same dimension as A) is a diagonal matrix with the singular values arranged in descending order, and V^T has rows that are the right singular vectors. However, we are not interested in the whole set of eigenvalues, but only the bigger one, therefore we can save some computation by evaluating only the first column of U and the first element of S.

More precisely, A gaussian with parameters $\bar{\vartheta}_{OLD}$ will be split in two new gaussians A and B, with means:

$$\begin{aligned} \bar{\Sigma}_{OLD} &= USV^T \\ \bar{u}_{MAX} &= U_{*,1}; \quad s_{MAX} = S_{1,1} \\ \bar{\mu}_A = \bar{\mu}_{OLD} + \frac{1}{2} s_{MAX} \bar{u}_{MAX}; &\quad \bar{\mu}_B = \bar{\mu}_{OLD} - \frac{1}{2} s_{MAX} \bar{u}_{MAX} \end{aligned} \tag{12}$$

where $\bar{u}_{MAX}$ is the first column of U, and s_{MAX} the first element of S.

The covariance matrices will then be updated as:

$$S_{1,1} = \frac{1}{4} s_{MAX}; \quad \Sigma_A = \Sigma_B = USV^T \tag{13}$$

while the new *a-priori* probabilities will be:

$$w_A = \frac{1}{2} w_{OLD}; \quad w_B = \frac{1}{2} w_{OLD} \tag{14}$$

The decision thresholds will be updated as explained in sec. 3.5.

Finally, their ages, a_A and a_B, will be reset to zero.

3.5 Updating Decision Thresholds

The decision thresholds are updated in two situations:

A. When a mixture component is split;
B. When each iteration is concluded.

These two procedures will be explained in the following.

- Single Iteration: The thresholds Λ_{TH}, and ξ_{TH} vary at each step with the following rules:

$$\begin{aligned} \Lambda_{TH} &= \Lambda_{TH} - \frac{\lambda}{nc^2} \cdot \Lambda_{TH} = \Lambda_{TH} \cdot \left(1 - \frac{\lambda}{nc^2}\right) \\ \xi_{TH} &= \xi_{TH} - \frac{\alpha_{MAX}}{nc^2} \cdot \xi_{TH} = \xi_{TH} \cdot \left(1 - \frac{\alpha_{MAX}}{nc^2}\right) \end{aligned} \tag{15}$$

with nc is the number of current gaussians, λ, and α_{MAX} are the coefficients for the likelihood and area change evaluation, respectively. Using high values for λ, and α_{MAX} results in high convergence speed. However, a faster convergence is often associated to instability around the optimal point, and may lead to a divergence from the local optimum. We can say that α_{MAX} can be interpreted as the *speed* the mixture components are split. In normal conditions, ξ_{TH} will become closer to the *area* of the bigger component's determinant step-by-step at each iteration. Then, it will approach the split threshold, allowing the splitting procedure.

Following an analog rule, Λ_{TH} will decrease step by step, approaching the current value of the global log-likelihood increment. This will allow the system to avoid some local optima, by varying its configuration whether a stationary situation occurs. Moreover, dividing λ and α_{MAX} by the square of nc consents to reduce the variation of the splitting threshold according to the number of components increases with a parabolic curve. This favorites the splitting when a low number of components is present, while avoiding a diverging behavior in case of an excessive amount of splitting operations.

Finally, every time a gaussians is added these thresholds will be reset to their initial value (see next section).

- After Gaussian Splitting: The decision thresholds will be updated as follows:

$$\xi_{TH} = \frac{\xi_{TH-INIT}}{nc}; \quad \Lambda_{TH} = L_{TH-INIT} \tag{16}$$

where nc_{OLD} and nc are the previous and the current number of mixture components, respectively. Substantially, this updates the splitting threshold to a value that goes linearly with the initial value and the actual number of components used for the computation.

3.6 Optimizing the Selected Mixture

This is an optional procedure. Once the best, or chosen, mixture, is saved, there are two possibilities:

1. Keeping the chosen mixture as the final result;
2. Optimizing the chosen mixture with the original EM algorithm.

The first one is the fastest but less accurate, while the second one introduces new computations ensuring more precision. It may happen that FASTGMM decides to increase the number of components even though the EM has not reached its local maximum, due to the splitting rule. In this case current mixture can still be improved by running the EM until it achieves its best configuration (the log-likelihood no longer increases).

Whether applying the first or second procedure is a matter of what predominates in the *"number of iterations vs. solution precision"* compromise at each time.

3.7 Computational Complexity Evaluation

We refer to the pseudocode in algorithm 1, and to the notation presented in sec. 3.1.The computational burden of each iteration is:

- the original EM algorithm (steps 1 to 1) takes $O(N \cdot D \cdot nc)$ for each step, for a total of $O(4 \cdot N \cdot D \cdot nc)$ operations;
- our algorithm takes $O(nc)$ for evaluating all the gaussians (step 1 to 1);
- our split (step 1) operation requires $O(D)$.
- the others take $O(1)$.
- the optional procedure of optimizing the selected mixture (step 1) takes $O(4 \cdot N \cdot D \cdot nc)$, being the original EM.

Therefore, the original EM algorithm takes:

- $O(4 \cdot N \cdot D \cdot nc)$, while our algorithm adds $O(D \cdot nc)$ on the whole, or $O(4 \cdot N \cdot D \cdot nc)$, giving rise to $O(4 \cdot N \cdot D \cdot nc) + O(D \cdot nc) = O(4 \cdot N \cdot D \cdot nc + D \cdot nc) = (nc \cdot D \cdot (4N + 1))$ in the first case;
- $2 \cdot O(4 \cdot N \cdot D \cdot nc) + O(D \cdot nc) = O(8 \cdot N \cdot D \cdot nc + D \cdot nc) = (nc \cdot D \cdot (8N + 1))$ in the second case, with the optimization procedure.

Considering that usually $D << N$ and $nc << N$, and that the optimization procedure is not essential, our procedure does not add a considerable burden, while giving an important improvement to the original computation in terms of self-adapting to the data input configuration at best. Moreover, it is worth noticing that even though the optimization procedure is performed, this starts very close to the optimal mixture configuration. In fact, the input mixture is the result of the FASTGMM computation, rather than a generic random or k-means initialization (as it happens with the simple EM algorithm, generally).

4 Experiments

Since now we use the following notation:

- FASTGMM: Our algorithm;
- FIGJ: [5].

4.1 Synthetic Data

We tested it by classifying different input data sets randomly generated by a known gaussians mixture. The same input sets have been proposed to [5]. Each distribution has a total of 2000 points, but arranged with different mixture distributions, with 3, 4, 8, and 16 Gaussian components.

The output of the two algorithms is shown in Fig. 1. Each subplot set is composed by the graphical output representation for the 2-D point distribution (top) and the 3-D estimation mixture histogram (bottom). The data plots show the generation mixture (blue) and the evaluated one (red). On the left the data result from our approach is shown, while on the right those of [5], relative to the same input data set. Moreover, the 3D histograms at the bottom in each subfigure represent: The generated mixture, our algorithm's estimated one, and that estimated by [5], respectively.

We can see that our algorithm is capable to learn the input data mixture starting from only one component with an accuracy comparable with those of [5].

4.2 Colored Real Images

We segmented the images as 3-dimensional input in the (R,G,B) space. The color image segmentation results are shown in Fig. 2.The set of images is divided into two groups: Some general images, on the left (from (1) to (6)), and some images taken by the iCub's cameras, on the right (from (7) to (12)). For each group we show the original images, those obtained with [5], and those obtained with our algorithm on the left, in the middle, and on the right, respectively.

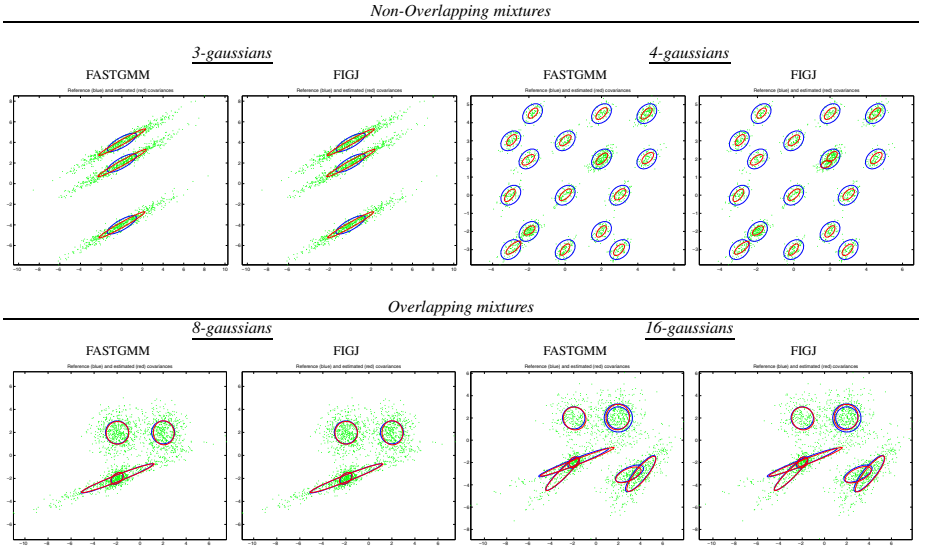


Fig. 1. For each plot set: Generation mixture (blue) and the evaluated one (red) for FASTGMM and FIGJ on the same input sets. Moreover, the 3D histograms at the bottom in each subfigure represent: The generated mixture, our algorithm's estimated one, and that estimated by [5], respectively.

5 Discussion

5.1 Synthetic Data

In table 2 the results of FASTGMM and FIGJ applied to the selected images are shown.

Table 2. Experimental results on 2D synthetic data.

Input	Algorithm	# Initital gaussians	# Detected gaussians	Actual gaussian number	# Iterations	Elapsed Time [s]	Diff time FASTGMM with opt vs FIGJ%	Log-likelihood	Diff lik FASTGMM with opt vs FIGJ %	Normalized L2 Distance without optimization	Normalized L2 Distance with optimization	Crashed
3	FSAEM	1	3	3	76	3.99716	-53.89844289	-8420.917867	0.495867477	5.770135	3.918034	no
	Optimization		3		130	6.151567		-8379.161274				
	FSAEM + Opt.		3		206	10.148727		-8379.161274				
	FIGJ	16	3		277	29.433288		-9524.692099		3.670464	3.670464	no
4	FSAEM	1	4	4	101	5.615204	-123.166389	-7573.101881	2.218687212	10.670613	0.07519	no
	Optimization		4		186	12.531248		-7405.078438				
	FSAEM + Opt.		4		287	18.146452		-7405.078438				
	FIGJ	16	4		205	13.52505		-8729.761818		0.076403	0.076403	no
8	FSAEM	1	9	8	276	5.750431	22.99575458	-8599.51	0.015582283	0.196817	1.971166	no
	Optimization		9		199	4.428076		-8598.17				
	FSAEM + Opt.		9		475	10.178507		-8598.17				
	FIGJ	16	7		333	48.156629		-9798.154848		0.14491	0.14491	no
16	FSAEM	1	16	16	501	26.667825	51.82061529	-8165.436422	0.057038433	0.251515	1.033934	no
	Optimization		16		202	12.848394		-8160.778985				
	FSAEM + Opt.		16		703	39.516219		-8160.778985				
	FIGJ	20	13		363	63.740854		-9540.91802		2.98916	2.98916	no

5.1.1 Evaluated Number of Components

There are substantially no differences in the selected number of components. Both our approach and [5] perform well on low mixture components, while having the tendency of underestimating the best number when it increases, with exception for our approach

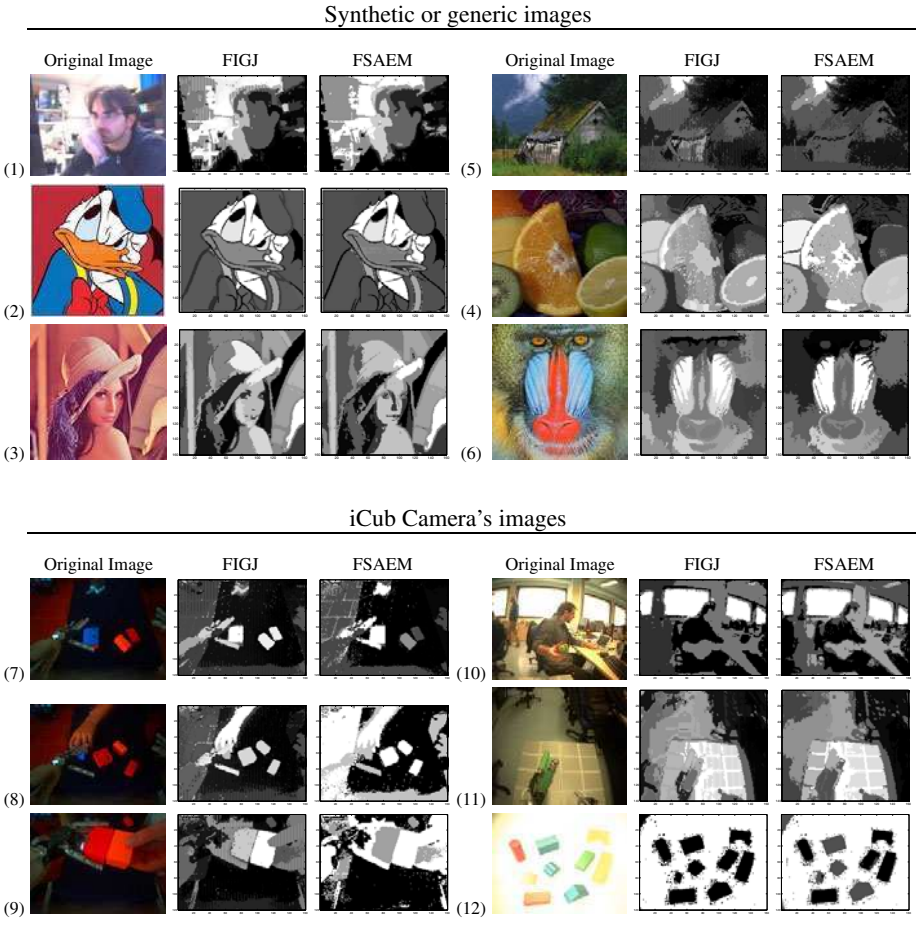


Fig. 2. Color images segmentation. From image (1) to (6) we tested the algorithms on well-known images, or synthetic ones, and from (7) to (12) we exploit the algorithms possibilities on real images captured by our robotic platform iCub's cameras. Note: FIGJ has not been able to segment image (13) also starting with merely 2 components, due to some internal covariances ill-posedness problems.

that overestimates it with the 8-component synthetic data and acting exactly with the 16-components. However, it is worth considering that even though it approaches the actual number correctly, this not necessary means that the components are in the right place. For instance, two components may be regarded as only one, while a single one can be considered as a multiple one. Nonetheless, it happens for both algorithm (see Fig. 1), suggesting that a perfect algorithm is hard to find.

5.1.2 Elapsed Time

It is important to distinguish the required number of iterations from the elapsed time. FASTGMM employs fewer iterations than FIGJ without making use of the optimization

Table 3. Experimental results on real images segmentation.

Input	Algorithm	# Initial gaussians	# Detected gaussians	# Iterations	Elapsed Time [s]	Diff time FASTGMM with opt vs FIGJ %	Log-likelihood	Diff lik FASTGMM with opt vs FIGJ	Diff lik FASTGMM with opt vs FIGJ	Crashed
1	FASTGMM	1	9	551	71.460507	130.9699636	-235130.6216	0.186374662	17.03330409	no
	Optimization		9	23	22.131293		-234692.3977			
	FASTGMM + Opt.		9	700	93.5918		-234692.3977			
	FIGJ	16	16	422	307.454885		-274668.2675			yes, 17
2	FASTGMM	1	9	426	68.31949	2.854809074	-301594.6385	6.29984E-09	39.70671245	no
	Optimization		9	3	1.950391		-301594.6384			
	FASTGMM + Opt.		9	429	70.269882		-301594.6384			
	FIGJ	19	19	308	441.170848		-421347.9543			yes, 20
3	FASTGMM	1	14	426	80.365553	109.5710584	-314931.44	0.120630946	11.16531396	no
	Optimization		14	374	88.057387		-314551.5352			
	FASTGMM + Opt.		14	800	168.422941		-314551.5352			
	FIGJ	30	25	572	1611.816189		-349672.2017			no
4	FASTGMM	1	12	451	75.660802	28.28196296	-235735.0604	0.00245062	28.67314092	no
	Optimization		11	103	21.39836		-235729.2834			
	FASTGMM + Opt.		11	554	97.059162		-235729.2834			
	FIGJ	18	19	329	500.708511		-303320.2731			yes 19
5	FASTGMM	1	13	451	80.165239	72.03688746	-311585.8059	0.134313369	10.60749104	no
	Optimization		13	246	57.748543		-311167.3045			
	FASTGMM + Opt.		13	697	137.913782		-311167.3045			
	FIGJ	30	27	611	2110.020131		-344174.3484			no
6	FASTGMM	1	7	276	34.977055	16.7543208	-226138.1494	3.34203E-05	17.57803149	no
	Optimization		7	36	5.860168		-226138.0738			
	FASTGMM + Opt.		7	312	40.837223		-226138.0738			
	FIGJ	16	16	420	272.227272		-265888.6956			yes, 17
7	FASTGMM	1	7	576	72.744922	3.588007146	-281176.5478	1.30736E-06	15.72288044	no
	Optimization		7	14	2.610093		-281176.5441			
	FASTGMM + Opt.		7	590	75.355015		-281176.5441			
	FIGJ	11	11	267	106.453566		-325385.5959			yes, 12
8	FASTGMM	1	8	426	54.119749	11.97025322	-205718.9337	7.50364E-05	26.47107115	no
	Optimization		8	46	6.478271		-205718.7793			
	FASTGMM + Opt.		8	472	60.59802		-205718.7793			
	FIGJ	11	11	228	90.200195		-260174.7438			yes, 12
9	FASTGMM	1	4	251	25.685892	5.220795914	-211551.64	0	15.41441064	no
	Optimization		4	3	1.341008		-211551.64			
	FASTGMM + Opt.		4	254	27.026901		-211551.64			
	FIGJ	13	13	313	137.77516		-244161.0785			yes, 14
10	FASTGMM	1	3	201	18.822216	34.15089913	-230138.8367	0.000990588	14.70275567	no
	Optimization		3	63	6.427956		-230136.557			
	FASTGMM + Opt.		3	264	25.250172		-230136.557			
	FIGJ	14	13	275	106.626624		-263972.9727			yes, 14
11	FASTGMM	1	11	451	67.534808	47.35502469	-210899.0185	0.114549212	17.68706256	no
	Optimization		10	180	31.981125		-210657.4353			
	FASTGMM + Opt.		10	631	99.515933		-210657.4353			
	FIGJ	24	22	514	624.913104		-247916.5477			no
12	FASTGMM	1	3	151	16.899695	13.4646217	-218447.7912	2.96364E-06	16.0195988	no
	Optimization		3	23	2.27548		-218447.7848			
	FASTGMM + Opt.		3	174	19.943671		-218447.7848			
	FIGJ	12	12	260	130.416222		-253442.2435			yes, 13

process, while more in the other case. At a first glance, this may suggest a whole FASTGMM slower computation than FIGJ. However, the whole elapsed time that occurs for running our procedure is generally less than FIGJ's. Nevertheless, we made FIGJ starting with a reasonable number of components, just a few more than the optimum, so that they do not affect its performance negatively. FASTGMM's better performance is due to the fact that our approach, growing in the number of components, computes more iterations than FIGJ but with a small number of components per iteration. Therefore it runs each iteration faster, while slowing only at the end due to the augmented number of components.

5.1.3 Mixture Precision Estimation

It is possible to see that FASTGMM usually achieves a higher final log-likelihood than FIGJ. This suggests a better approximation of the data mixture. However, a higher log-likelihood does not strictly imply that the extracted mixture covers the data better than another one. This is because it is based on the probability of each component, which

may be more or less exact, being not deterministic. Nevertheless, it is a good index on the probability that such mixture would be better.

A deterministic approach is to adopt a unique distance measure between the generation mixture and the evaluated one. In [9] Jensen *et Al.* exposed three different strategies for computing such distance: The Kullback-Leibler, the Earh Mover, and the Normalized L2 distance. The first one is not symmetric, even though a symmetrized version is usually adopted in music retrival. However, this measure can be evaluated in a close form only with mono-dimensional gaussians. The second one also suffers analog problems of the latter. The third choice, finally is symmetric, obeys to the triangle inequality and it is easy to compute, with a comparable precision with the other two. We then used the last one. Its expression states [1]:

$$\begin{aligned} z_c N_x\left(\bar{\mu}_c, \bar{\Sigma}_c\right) &= N_x\left(\bar{\mu}_a, \bar{\Sigma}_a\right) \cdot N_x\left(\bar{\mu}_b, \bar{\Sigma}_b\right) \\ &\quad where \\ \bar{\Sigma}_c = \left(\bar{\Sigma}_a^{-1} + \bar{\Sigma}_b^{-1}\right)^{-1} &\quad and \quad \bar{\mu}_c = \bar{\Sigma}_c\left(\bar{\Sigma}_a^{-1}\bar{\mu}_a + \bar{\Sigma}_b^{-1}\bar{\mu}_b\right) \\ z_c = |2\pi\bar{\Sigma}_a\bar{\Sigma}_b\bar{\Sigma}_c^{-1}|^{\frac{1}{2}} e^{-\frac{1}{2}(\bar{\mu}_a-\bar{\mu}_b)^T\bar{\Sigma}_a^{-1}\bar{\Sigma}_c\bar{\Sigma}_b^{-1}(\bar{\mu}_a-\bar{\mu}_b)} \\ = |2\pi\left(\bar{\Sigma}_a + \bar{\Sigma}_b\right)|^{\frac{1}{2}} e^{-\frac{1}{2}(\bar{\mu}_a-\bar{\mu}_b)^T\left(\bar{\Sigma}_a+\bar{\Sigma}_b\right)^{-1}(\bar{\mu}_a-\bar{\mu}_b)} \end{aligned} \tag{17}$$

5.2 Colored Real Images

It is salient to report that in [5] it has not been performed any experiment on real images segmentation. In fact, their result only concern different families of synthetic data. Contrariwise, we want to focus more on image processing, due to its relevant importance in several different scientific fields, like robotics and medicine, as mentionned within the introduction.

As we pointed out in the previous section (sec. 4.2), to compare a detected mixture versus a generic image is not possible quantitatively, only qualitatively. This is due to the high number of colors present within the image. It is obvious that with more components the image is better reconstructed. However, it is possible to visually recognize a pattern even with fewer components, although with less accuracy. Therefore, what algorithm gives the best result is again a matter of what compromise is better, in terms of computational complexity and result accuracy.

Moreover, we have to report a problem that has not been addressed in the original work [5]. This crashes with some images when the number of components increases too much. Than means that it is not able to finish the computation. Therefore, as reported on tab. 3 we had to start it with a relatively low number of gaussians. However, even though it is able to finish the computation, it very often returns a mixture having the same starting number of components as the best one. Moreover, it explores the whole solution space, from the input mixture to one with only one element. This makes pointless the usage of such approach: Segmenting an image with the original EM instead of [5] will give the same result in less time.

5.3 FASTGMM Optimization Procedure

We reported our results both using the optimization procedure, and not. Since one of the most prominent key feature of our approach is its fast computation, together with its simple implementation, the optimization process may seem worthless or too much time demanding. However, by comparing its performances against those of [5], our algorithm still remain faster (see sec. 5.1.2). The difference in terms of final mixture precision is not so evident at a first glance, both referring to the final log-likelihood and to the normalized L2 distance, although present. Nevertheless, the required time for the EM optimization step is important, since sometimes it approaches (and overcome) the splitting part. Here, selecting whether optimizing or not is merely a question of performance requiring. If one claims for the fastest algorithm it is advisable to not use the optimization, even though it may lead to some improvements to the final mixture. Otherwise, FASTGMM gives a good precision maintaing a better computational burden than FIGJ.

5.4 Limitation of the Proposed Algorithm

The bigger issue with our approach is the α_{MAX} parameter tuning. This cause FASTGMM being less general than FIGJ in input domain. If α_{MAX} is too small the input description may be underestimated, or overestimated if it is too high. This does not mean that it cannot perform in general purposes, but only that it has to be tuned for getting precise results. However, this makes FASTGMM suitable for a first data description due to its great velocity. Once a first input analysis has been performed, it can be fine tuned to have a better data description. Moreover, we demonstrated that if well tuned, FASTGMM is able to segment the input data even better than FIGJ.

6 Conclusions

In this paper we proposed a unsupervised algorithm that learns a finite mixture model from multivariate data on-line. The algorithm can be applied to any data mixture where the EM can be used. We approached the problem from the opposite way of [5], i.e. by starting from only one mixture component instead of several ones and progressively adapting the mixture by adding new components when necessary. Our algorithm starts from a single mixture component and sequentially *growing* both increases the number of components and adapting their means and covariances. Therefore, its initialization is unique, and it is not affected by different possible starting points like the original EM formulation. Moreover, by starting with a single component the computational burden is low at the beginning, increasing only whether more components are required. Finally, we presented the effectivity of our technique in a series of simulated experiments with synthetic data, artificial, and real images, and we compared the results against [5].

Acknowledgements. This work was supported by the European Commission, Project IST-004370 RobotCub and FP7-231640 Handle, and by the Portuguese Government - Fundação para a Ciência e Tecnologia (ISR/IST pluriannual funding) through the PIDDAC program funds and through project BIO-LOOK, PTDC / EEA-ACR / 71032 / 2006.

References

1. Ahrendt, P.: The multivariate gaussian probability distribution. Tech. rep. (January 2005), http://www2.imm.dtu.dk/pubdb/p.php?3312
2. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: Bridging the gap between simulation and reality in urban search and rescue. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, pp. 1–12. Springer, Heidelberg (2007)
3. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood estimation from incomplete data via the em algorithm. J. Royal Statistic Soc. 30(B), 1–38 (1977)
4. Dobbe, J.G.G., Streekstra, G.J., Hardeman, M.R., Ince, C., Grimbergen, C.A.: Measurement of the distribution of red blood cell deformability using an automated rheoscope. Cytometry (Clinical Cytometry) 50, 313–325 (2002)
5. Figueiredo, A., Jain, A.: Unsupervised learning of finite mixture models. IEEE Trans. Patt. Anal. Mach. Intell. 24(3) (2002)
6. Greggio, N., Silvestri, G., Menegatti, E., Pagello, E.: Simulation of small humanoid robots for soccer domain. Journal of The Franklin Institute - Engineering and Applied Mathematics 346(5), 500–519 (2009)
7. Greggio, N., Bernardino, A., Santos-Victor, J.: A practical method for self-adapting gaussian expectation maximization. In: International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010), Funchal, Madeira - Portugal, June 15–18 (2010)
8. Hartley, H.: Maximum likelihood estimation from incomplete data. Biometrics 14, 174–194 (1958)
9. Jensen, J.H., Ellis, D., Christensen, M.G., Jensen, S.H.: Evaluation distance measures between gaussian mixture models of mfccs. In: Proc. Int. Conf. on Music Info. Retrieval ISMIR 2007, pp. 107–108 (2007)
10. Li, J.Q., Barron, A.R.: Mixture density estimation. In: NIPS, vol. 11. MIT Press, Cambridge (2000)
11. Montesano, L., Lopes, M., Bernardino, A., Santos-Victor, J.: Earning object affordances: From sensory motor maps to imitation. IEEE Trans. on Robotics 24(1) (2008)
12. Shim, H., Kwon, D., Yun, I., Lee, S.: Robust segmentation of cerebral arterial segments by a sequential monte carlo method: Particle filtering. Computer Methods and Programs in Biomedicine 84(2-3), 135–145 (2006)
13. Ueda, N., Nakano, R., Ghahramani, Y., Hiton, G.: Smem algorithm for mixture models. Neural Comput. 12(10), 2109–2128 (2000)
14. Verbeek, J., Vlassis, N., Krose, B.: Efficient greedy learning of gaussian mixture models. Neural Computation 15(2), 469–485 (2003)
15. Vincze, M.: Robust tracking of ellipses at frame rate. Pattern Recognition 34, 487–498 (2001)
16. Zhang, Z., Chen, C., Sun, J., Chan, K.L.: Em algorithms for gaussian mixtures with split-and-merge operation. Pattern Recognition 36, 1973–1983 (2003)

Timed Specification Patterns for System Validation: A Railway Case Study

Ahmed Mekki[1,2], Mohamed Ghazel[1,2], and Armand Toguyéni[1,3]

[1] Univ Lille Nord de France F-59000 Lille, France
[2] INRETS, ESTAS, F-59666, Villeneuve d'Ascq, France
[3] EC LILLE, LAGIS, F-59651, Villeneuve d'Ascq, France
{ahmed.mekki,mohamed.ghazel}@inrets.fr,
armand.toguyeni@ec-lille.fr

Abstract. The aim of the work discussed in this paper is to introduce a method for verifying temporal requirements of time-constrained systems. The method predates by establishing a generic repository of observation patterns relative to a new time constraint taxonomy that we define. The method allows the automated verification of temporal requirements, initially expressed in a semi-formal formalism -UML State Machines (SM) with time annotations- through model transformation and model-checking technique. In practice, in order to check the temporal aspects of a given specification, the observation patterns relative to the extracted requirements are instantiated to obtain appropriate observers. Then, using a transformation algorithm, the system specification as well as the obtained observers are translated into Timed Automata (TA) models. Thereby, the verification is reduced to a reachability analysis within the global model bringing together the system under study and the obtained observers.

Keywords: Time-constrained System, Verification and validation, UML state machines, Timed automata, Model transformation.

1 Introduction

Given their practical implication on the safety and correctness of critical applications (e.g. transportation systems, nuclear plants), specification and verification are among the most important research topics in critical systems engineering, since such systems must achieve a high level of robustness and reliability. In addition, these systems usually involve time-dependent functionality. Consequently, methods for behavior modeling and verifying (especially temporal requirements) are increasingly important. The approaches most used for specifying timed systems are based on **T**imed **A**utomata (**TA**). TA are well suited for expressing timed behavior and for modeling real-time components. A number of automatic verification tools for TA have been developed and have proven to be efficient e.g., Uppaal [13] and Kronos [24]. Nevertheless, specifying and verifying time constraints are becoming more and more difficult tasks due to the widespread applications and increasing complexity of checked systems. Despite the different advantages proposed by TA, such as parallel composition, users often need to manually manipulate a set of clock variables with complex calculated clock constraints

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 121–134.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

in order to express the time properties. This process is tedious, error-prone and requires sophisticated logical and/or mathematical skills.

On the other hand, in order to cope with the complexity of critical systems engineering, approaches based on **M**odel **D**riven **E**ngineering (**MDE**) seem to be very useful [18]. The aim of this work is to introduce a new temporal requirement verification method based on MDE. First, we define a Pattern Basis for monitoring time constraints. Indeed, based on a new time constraint classification, we developed a set of time observation patterns expressed in **U**nified **M**odeling **L**anguage (**UML**) **S**tate **M**achines (**SM**)[21]: this is expected to be a relatively inexpensive activity, since this procedure is done once and for all. UML has been chosen since it is relatively intuitive, offers a graphic description, is implemented by several tools and, finally, is a standard notation well supported by the **O**bject **M**anagement **G**roup (**OMG**). This set of patterns facilitates high-level system design. These patterns cover a large class of common time constraints.

Since our aim is to keep a high precision level, a subsequent step consists in giving an accurate definition of each developed pattern. Hence, for each pattern, we give (1) a textual definition, (2) a UML SM model, (3) a structured English specification and, finally, (4) a temporal logic expression (**T**imed **C**omputational **T**ree **L**ogic (**TCTL**)) relative to the property concerned.

Concretely, the verification process is based on the set of patterns. The suitable patterns corresponding to the time constraints extracted from the system requirements are picked up and instantiated. This instantiation step generates a set of SM Observers. The SM observers are translated into more formal notation, the TA, which provides support for the properties' verification. The translation is made according to a transformation algorithm that will be discussed later in the paper. In this way, analyzers exploit the benefits of formal notations without having to go through the complex and expensive formal modeling phase. This transformation generates a set of TA Observers. A system's model, which is also generated by the same transformation algorithm, is synchronized with the obtained TA observers to obtain a global model. Hence, the verification task is performed on this obtained model with a reachability analysis while checking whether the observers' forbidden states - corresponding to constraints violation - are reachable.

The paper is organized as follows. In Section 2, we set out the context and we briefly go through some related works. Section 3 describes our first contribution by introducing the new time-constraints taxonomy and the patterns basis. The second contribution of our method is outlined in Section 4 where the translation from UML SM, with time annotations, to TA is described. The method is illustrated using a **L**evel **C**rossing (**LC**) case study in Section 5. Section 6 concludes the paper while outlying some future work.

2 Context and Background

2.1 Related Work

There have been many recent research efforts in the field of time-constrained system validation. Only two of these will be discussed in this section. First, based on the Dwyer [5] pattern basis, Dhaussy [3] defines a textual language, called "CDL", for requirement specification. The requirements are then translated into observer automata.

Furthermore, Dhaussy defines for each requirement a path, called "context" where the requirement should be checked. Finally, the system model, the observer automata and the context are translated into IF notation (Intermediate Format). Then, the verification is carried out using the IFx tool. Second, Nascimento [17] presents an approach for automatic generation of network of timed automata from a functional specification depicted via UML class and sequence diagrams. Nascimento uses UML sequence diagrams for the property specification phase. However, sequence diagrams suffer from a limited expressiveness when dealing with temporal aspects, since they only depict order.

Unlike the above methods, our approach uses TA as target notation; TA are assumed to be more expressive and well supported.

On the other hand, several projects have introduced natural-language-based approaches where natural language is mapped into a more formal specification. Dwyer [5] proposes several patterns applicable to property specification expressed in different formalisms and logics such as LTL, CTL, GIL, and quantified regular expressions (QRE). Konrad [11] proposes an extension to Dwyer's classification and real-time properties are added to the original classification. Moreover, TCTL, MTL and RTGIL are used to specify the added real-time properties.

Compared to the above-mentioned works, our contribution offers the following advantages:

- Our method takes advantage of the flexibility and expressiveness of UML SM in modeling tasks and the precision of TA formalism in the verification tasks. UML SM are also more expressive than UML sequence diagrams or UML collaboration diagrams used in other works,
- TA are well supported,
- Patterns facilitate high-level specification and promote reusability and knowledge capitalization,
- The verification task is reduced to a reachability analysis, this allows us to overcome some limitations met with some existing tools, such as Uppaal.

2.2 Observer Technique

We use observers whenever we set artifacts to monitor system behavior [4]. Let us recall here that the goal of our approach is to check whether the temporal requirements expected from a given system are satisfied. Hence, we make use of observers in order to express the satisfaction vs the violation of the predefined requirements [9]. Typically, checking a given temporal property consists in examining whether the error state of the corresponding observer is reachable.

3 Observation Patterns

In this section, we first propose a classification of all the common temporal requirements one may meet when dealing with critical systems. Then, we develop a structured English grammar that we use to express the predefined properties. Next, we introduce the patterns used in order to monitor the predefined temporal requirements. Finally, a standardized description of these patterns is suggested.

Table 1. Temporal Requirement's Taxonomy Descriptions

Class	Category	Pattern Name	Description
Quantitative	QuantitativeAbsence	Forbidden Before	**R** ensures that an event (E_{mon}) must never occur before a minimum T_{before} (time unit over E_{Ref}). **S**$\models$**R** is true if this event does not occur before T_{before}.
		Forbidden After	**R** ensures that an event (E_{mon}) must never occur after a deadline T_{after} (time unit over E_{Ref}). **S**$\models$**R** is true if this event does not occur after T_{after}.
		Forbidden Between	**R** ensures that an event (E_{mon}) must never occur within a temporal interval $]t_{Begin}; t_{End}[$ (over E_{Ref}). **S**$\models$**R** is true if this event does not occur between temporal interval $]t_{Begin}; t_{End}[$.
	QuantitativePresence	MinimumDelay	**R** ensures that an event (E_{mon}) must occur after a minimum time T_{min} (time unit over E_{Ref}). **S**$\models$**R** is true if this event occurs after T_{min}.
		MaximumDelay	**R** ensures that an event (E_{mon}) must occur before a deadline T_{max} (time unit over E_{Ref}). **S**$\models$**R** is true if this event occurs before T_{max}.
		Punctuality	**R** ensures that an event (E_{mon}) must occur at one punctual date t (time unit over E_{Ref}). **S**$\models$**R** is true if this event occurs at the t date.
	QuantitativeRecurrence	UnlimitedRecurrence	**R** ensures that an event (E_{mon}) must occur infinitely. **S**$\models$**R** is true if this event occurs.
Qualitative	QualitativeRecurrence	LimitedRecurrence	**R** ensures that an event (E_{mon}) must occur k times. **S**$\models$**R** is true if this event occurs k times.
	QualitativePresence	PresenceAfter	**R** ensures that an event (E_{mon}) must occur after E_{Ref} has been detected. **S**$\models$**R** is true if this event occurs at least once after E_{Ref}.
		PresenceBefore	**R** ensures that an event (E_{mon}) must occur before E_{Ref}. **S**$\models$**R** is false if E_{Ref} occurs before E_{mon}.
	QualitativeAbsence	AbsenceAfter	**R** ensures that an event must never occur after E_{Ref}. **S**$\models$**R** is true if this event does not occur.
		AbsenceBefore	**R** ensures that an event (E_{mon}) must never occur before E_{Ref}r. **S**$\models$**R** is true if E_{mon} does not occur before E_{Ref}.

3.1 Main Time-Constraints

We strive to identify all the common temporal requirements one may meet when dealing with critical systems. The main identified requirements are defined and explained in Table 1 (The relation which denotes that a system **S** satisfies a requirement **R** is written **S**$\models$**R**) and are also depicted in the shape of a UML Class diagram (Figure 1). This classification offers the advantage that it deals with requirements on events only, since we express the requirements on states using two events: the first event represents the activation of the state and the second the deactivation.

3.2 Structured English

To facilitate the expression and the formalization of temporal properties, we have developed a structured English grammar (**SEG**) [8]. This grammar supports both qualitative and quantitative properties. The aim of developing such a grammar is to provide the user with simple means for expressing temporal requirements. The SEG is somehow a guiding framework which offers expression means to the user. It also constrains him to make assertions according to a predefined syntax, hence making it possible to automatically recognize the expressed requirements. Concretely, each sentence generated by our grammar describes a temporal property and serves as a handler that aids expressing and

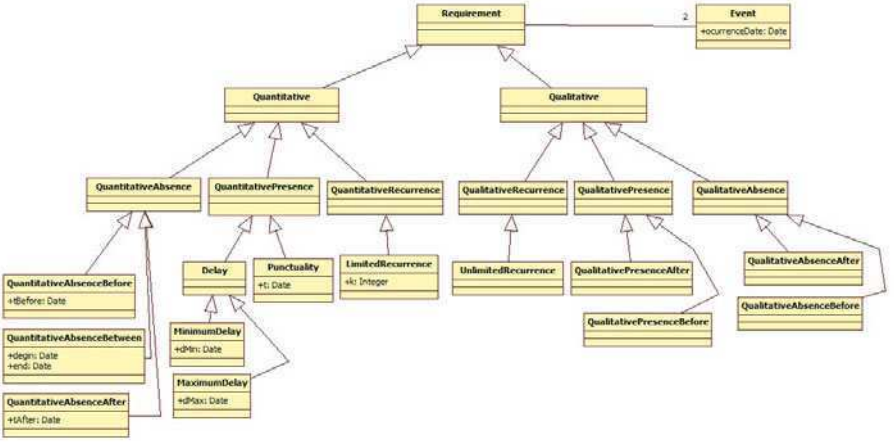


Fig. 1. Temporal Requirements Classification.

understanding the requirement. For more details about the proposed SEG, the reader can refer to [8]. SEG is expressed below using BNF (Backus-Naur Form) notation:

Property = {<*Scope*>**:** <*Specification*>};
Specification = {<*Entity*><*Obligation*>**occur** <*Reference*>} ;
Scope = **Global** |**Before** <*Entity*>|**After** <*Entity*>|**Between** <*Entity*>**and** <*Entity*>;
Entity = "Event" |"Active(State)" |"Desactive(State)";
Obligation = **must** |**cannot**;
Reference = ((**exactly at** <*time*>**over**) |(**After** [**a delay of** <*time*>**over**]) |(<**Before** [**a delay of** <*time*>**over**])) <*Entity*>;
Time = <*Number*>**tu**;
Number = <*Digit*>+ ;
Digit = {**0**|**1**|**2**|**3**|**4**|**5**|**6**|**7**|**8**|**9**} ;

Literal terminals are given in **bold** font, non-literal terminals are delimited by quotation marks (" ") and non-terminals are given in *italics*. The start symbol of the grammar is property and the language $\mathcal{L}$ of the grammar is finite, since the grammar is non-circular and has no repetitions.

3.3 Observation Patterns Basis

A pattern is a commonly reusable model in software systems that guarantees a set of characteristics and functionalities. The identification of a pattern is based on the context in which it is used. The goal behind developing patterns is to offer a support for system design and development. Using patterns helps in keeping design standardized and useful and minimizes reinventing in the design process, since the patterns facilitate reusability and knowledge capitalization [6].

In this work, we define a set of patterns which will serve as a basis to generate observers for all the identified temporal requirements. The notation used is UML State Machines. The basis of patterns is introduced regardless of the systems' specification and is used to model all the common temporal requirement types that one may express. This pattern basis guarantees the reusability and the genericity of the mechanisms developed within our approach.

3.4 Pattern Formalization

We have introduced a new temporal requirement classification that is used in implementing our pattern repository. Additionally, we include a graphical representation of each pattern in the shape of a UML SM diagram [16]. This field will be used later as an input model for the model transformation phase. Each pattern in the repository contains the following fields:

Pattern Name. The pattern name serves as a handle for the pattern's use and describes the type of the pattern.

Pattern Definition. A short description and definition of the requirement for which the pattern is used.

Scoped Structured English Specification. The scoped structured English sentence captures the invoked property using the grammar previously defined. The scope, initially introduced by Dwyer in [5], is used to express the applicability interval (scope) of the property. Four scopes are used in our grammar: *globally*, *before* an event occurs, *after* an event occurs and *between* two events.

Temporal Logic Description. Contains mappings of the property monitored by the pattern to TCTL for each of the four defined scopes. We chose TCTL since it allows quantitative temporal property expression.

4 Transformation Approach

Since the observation pattern basis has been introduced, we will now discuss how to use this basis in the verification process. In practice, once the temporal requirements for the system under study are identified and extracted, the appropriate patterns for these requirements are selected and instantiated with the suitable parameters, thus resulting in some SM observers. Each SM observer monitors an elementary requirement. The SM observers are then translated into TA observers. This transformation will be presented hereafter.

4.1 Transformation Idea

In spite of the number of automated analyzers developed for TA, these tools suffer from two main limitations: the first is that users must be familiar with their formal notations. The second is the lack of patterns for high-level system design (hierarchy notion namely). On the other hand, semi-formal languages, such as UML SM, are suitable for expressing system requirements. However, the automatic verification of these models is unfeasible directly. The temporal requirement verification approach that we propose takes advantage of the expression flexibility of SM and the analysis facilities offered by TA formalism.

The various rules of the transformation algorithm we have defined are expressed according to the **M**odel-**D**riven **A**rchitecture (**MDA**) approach. MDA is an initiative and a standard proposed by the OMG, allowing developers to create systems entirely based on models. It points out the idea of separation of concerns by unlinking/uncoupling the application logic from the implementation platforms technology [23].

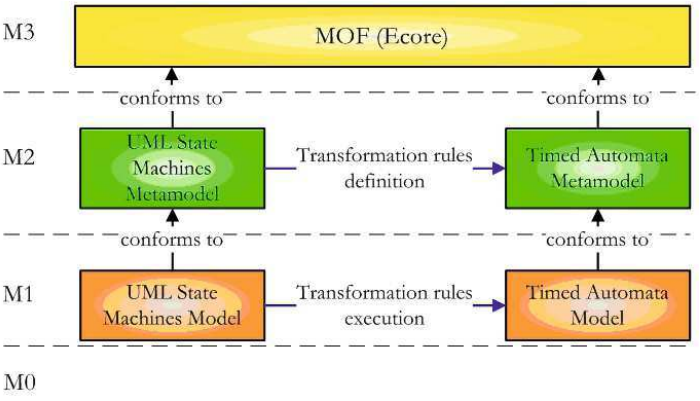


Fig. 2. Transformation Approach.

Figure 2 illustrates the use of the MDA four-layer metamodeling architecture for our transformation;

- The source model (resp. target model) is expressed according to the source metamodel (resp. target metamodel),
- The metamodels are defined and expressed according to the MOF metametamodel (in our transformation, we used the Ecore metametamodel, which is the Eclipse implementation of MOF),
- A metamodel is developed for TA. On the other side, we used the UML metamodel distributed in the Eclipse framework,
- All the rules are introduced at the metamodel level,
- The transformation takes a UML SM model as a source model and generates a TA model with a corresponding formatted code.

4.2 Time Annotations

Here we use SM as a modeling notation to take advantage of the flexibility they offer. However, since we strive to obtain an accurate specification, we should guide the user while introducing the temporal constraints. Concretely, we propose a set of timed annotations in order to express the states' characteristics as well as the transition guards. Table 2 shows some examples of them and defines the signification of each annotation.

4.3 Transformation Algorithm

One of the key parts of our method is the translation of UML SM with time annotations into TA. For the sake of space, we will briefly describe the transformation rules while giving the source and target element for each of them in 3. For more details, the reader can refer to [15].

The main rule of this algorithm is the *FromStateMachine* rule. This rule is the first one carried out by the transformation algorithm. It picks elements in the source model, then calls on other rules to translate the selected elements into TA elements in the target

Table 2. Time annotations used.

	Time annotation	Signification
1	at_most(T_{max})	t$\leq T_{max}$
2	at_least(T_{min})	t$\geq T_{min}$
3	after(d)	t=d
4	between(T_{min}, T_{max})	$T_{min} \leq$t$\leq T_{max}$
5	upper(T_{min})	t$> T_{min}$
6	lower(T_{max})	t$< T_{max}$

Table 3. Transformation Rules.

Rule Name	Source element: UML SM	Target element: TA
FromStateMachine	StateMachine	TA
Simple2Simple	State	State
Final2State	Final Pseudostate	State
OR2Automata	State	Automaton
AND2Automata	State	Automaton
Trans2Trans	Transition	Transition
Entry2State	EntryAction	State
Exit2State	ExitAction	State
Do2State	DoActivity	State

model. Likewise, the rules called on behave in the same way; they select elements in the source model and call on the appropriate rule to transform them. For example, the *FromStateMachine* rule is applied to elements of the "UML::StateMachine" type and translates them into a "TA::AutomataMachine" element. Also, different element types are selected and different rules are called on in this rule. First, the rule selects all the UML states. Then for each selected state, according to its type, one or more of the *Simple2Simple* or *OR2Automata* or *AND2Automata* rules are called on. Secondly, it translates the "UML::Transition" elements by invoking the *Trans2Trans* rule. Also, this rule deals with another element type, the "UML::Pseudostate", by invoking some other rules, such as *Final2State*.

This internal transformation process is the same for all the rules; each rule transforms the source element into the target one. Then, it selects subelements of the source element and calls on the appropriate rule to transform them.

4.4 Verification Process

Once our observation patterns' repository is implemented, we introduce a verification process based on it. This section will outline the global architecture of our approach. The architecture is depicted graphically in Figure 3 [14].

Concretely, our approach is composed of four processes: first, temporal requirements for the system under study are identified and extracted. Second, the appropriate patterns for these extracted requirements are selected and instantiated with the suitable parameters (This second process results in some SM observers. Each SM observer

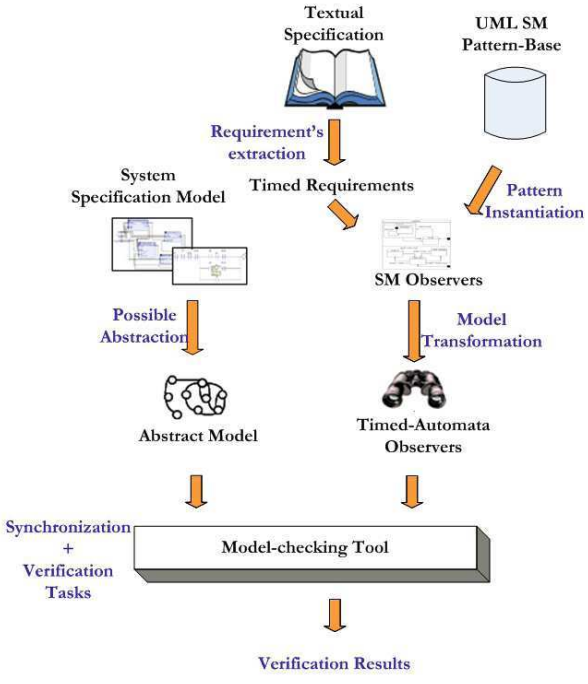


Fig. 3. Method Global View.

corresponds to an elementary requirement). Third, the SM observers are translated into TA observers. In parallel and in the same way, the specification under study (SM model with time annotations) is abstracted and translated into a TA model. The translation from the UML SM to TA is performed using the MDA model transformation technique, as shown previously. Finally, the generated TA are synchronized with the formal system's specification model (TA) to generate a global model holding both the system specification and the requiements' monitoring. Thus, the verification task is reduced to an error-state reachability search on the obtained global model.

5 Case Study

5.1 Case Study Description

A classical automatic level crossing system is composed of several modules. The local **control** system which manages the traffic in the crossing area, a pair of barriers (**gate**), traffic **lights** whose role is to alert and prevent road users from entering the crossing zone, and a **train**-sensing module which monitors trains approaching/leaving [7]. The subsystems mentioned above execute in parallel and synchronize through events.

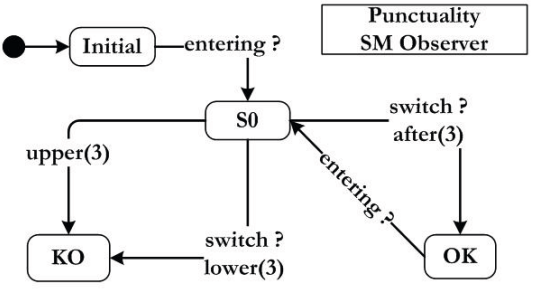
Several requirements [25] are given in textual specification in Figure 4. Next, using this textual specification, we will show how patterns are used to express and monitor requirements.

1. *"...As long as an approaching train runs over the activation sensor the sensor shall generate an occupied signal. When the last axle of the train has traversed the activation sensor it shall generate a free signal again. If the control unit receives an occupied signal from the activation sensor in the unsaved mode, it will enter the saving mode and gives the command to turn on the yellow lamp of the set of lights.* ***Three seconds after entering the saving mode the control unit has to give the command for switching off the yellow lamp and turning on the red lamp****..."*
2. *"...By entering saved mode the controller shall switch on the supervision signal to show signal aspect LC1, which means to turn on the blinking light.* ***Twelve seconds after the system has entered the safed mode it must start to lower the gate. The activity of lowering or raising the gate must not last longer than six seconds from one end position to the other.*** *When the gate has reached the lower end position within the six seconds interval the control unit will enter the mode saved and gates closed..."*

Fig. 4. Specification Example [25].

Punctuality Pattern

Pattern Name: Punctuality
Textual Description: The Punctuality pattern is used to check that a given event execution is delayed by exactly d time units relative to a given reference.
UML SM Diagram:



Structured English: *Scope* "switch" **must occur exactly at** "3" **tu over** "entering".
Temporal Logic description:
Globally: **AG**[receive(entering) $\Rightarrow$ **AF**$_{=d}$ receive(switch)]
After E$_a$: **AG**(E$_a \rightarrow$(**EG**[receive(entering) $\Rightarrow$ **AF**$_{=d}$ receive(switch)]))
Before E$_b$: **A**((**AG**[receive(entering) $\Rightarrow$ **AF**$_{=d}$ receive(switch)]) **U** E$_b$)
Between E$_b$ and E$_e$:

$$\mathbf{AG}((E_b \wedge \neg E_e) \rightarrow (P)\ \mathbf{U}\ E_e)$$

where P=**AG**[receive(entering) $\Rightarrow$ **AF**$_{=d}$ receive(switch)]

Fig. 5. Punctuality Pattern.

5.2 Using Patterns

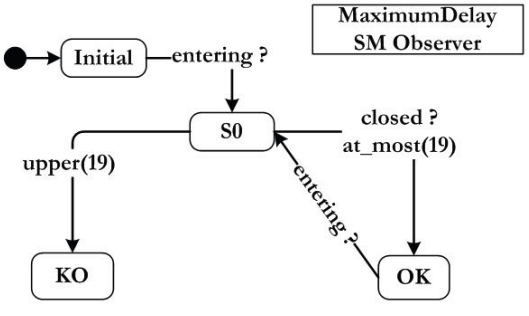
For the sake of space, only two requirements will be checked on the basis of the textual system specification. For each requirement, formalization is introduced using the defined generic template. First, a textual description of the requirement is presented, followed by an intuitive graphical representation in the shape of a UML SM (SM patterns). Then, a definition using our structured English grammar is given and, finally, a temporal logic expression is used to express the requirement formally.

1) The first requirement states that, three seconds after receiving the *entering* signal, the controller should send a command for *switching* lights. This requirement consists in a **punctuality** property (Figure 5) of 3 seconds between the *entering* signal and the *switch* signal.

2) The second requirement states that the command signal *closed* should be detected at most 19 seconds after the *entering* signal. This requirement consists in **maximumdelay** property (Figure 6) of 19 time units between *entering* and *closed* signals.

MaximumDelay Pattern

Pattern Name: MaximumDelay
Textual Description: The MaximumDelay pattern is used to check if event execution is delayed by at most a given value ℓ time units from a given reference. This value ℓ should be in [0, T_{max}] interval.
UML SM Diagram:



Structured English: *Scope* "closed" **must occur Before a delay of** "19" **tu over** "entering".
Temporal Logic description:
Globally: **AG**[receive(entering) $\Rightarrow$ $\neg$closed $U_{\leq 19}$ receive(closed)]
After E_a**:** **AG**($E_a \rightarrow$(**AG**[receive(entering) $\Rightarrow$ ($\neg$closed $U_{\leq 19}$ receive(closed))]))
Before E_b**:** **A**((**AG**[receive(entering) $\Rightarrow$ ($\neg$closed $U_{\leq 19}$ receive(closed))]) **U** E_b)
Between E_b **and** E_e**:**

$$\mathbf{AG}((E_b \wedge \neg E_e) \rightarrow (P)\ \mathbf{U}\ E_e)$$

where P=**AG**[receive(entering) $\Rightarrow$ ($\neg$closed $U_{\leq 19}$ receive(closed))]

Fig. 6. MaximumDelay Pattern.

5.3 Verification

The above step consists in instantiating the appropriate patterns in order to obtain observers for the extracted requirements. These observers are then translated into TA models to be synchronized with the system specification. The verification process consists in examining the reachability of the ***KO***-states within the observers. The verification of our case study is carried out using the UPPAAL model checker.

6 Conclusions

In this paper, we have presented a model-based method applicable to formal specification and validation of time-constrained systems. The approach uses a set of observation patterns that we have established and which act as watch-dogs for the defined temporal requirements. Each pattern has been defined using a standard template we developed. Using patterns offers genericity and reusability.

On the other hand, we have developed a transformation algorithm to translate SM with time annotations into TA, the aim being to make a basis for the verification process. For this purpose, we have introduced a TA metamodel using an extended definition of the original TA definition given by [1] (for reasons of brevity, the TA metamodel description is omitted in this paper). Once the relationships (transformations rules) between UML SM metamodel elements and TA metamodel elements were defined, we expressed them in the **QVT** (**Q**uery/ **V**iew/ **T**ransformation) language, defined by the OMG as the standard for the transformation phase. Then we used QVTo -an Eclipse Plugin- to run the algorithm.

Processing the verification upon the TA observers synchronized with the system specification reduces the verification task to a reachability analysis of the **KO**-nodes within the observers.

One of the main issues that we are focusing on currently is the validation of the model transformation algorithm that we have developed since it is a key in ensuring the correctness of our approach. There are several techniques that aim at validating model transformation [2], [10], [12], [19], [20], [22]. In our case, we elaborate the underlying transition system for both UML SM with time annotations and TA, and we establish equivalence relation (temporal bisimulation) between the obtained transition systems [10], [19]. Despite that this technique is the most time-consumer over other techniques, it remains that it is the best proof to guarantee semantic preserving (fill the semantic gap) through the transformation since it is mathematical-based.

Based on the structured English grammar we have developed, a prototype tool which offers interesting facilities in terms of requirement specification and requirement consistency-check has been implemented [8]. A subsequent step will be to extend this tool with a new module which automatically instantiates observers for the entered requirements using the observation patterns repository.

References

1. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science 126, 183–235 (1994)
2. Anastasakis, K., Bordbar, B., Georg, G., Ray, I.: UML2Alloy: A Challenging Model Transformation. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 436–450. Springer, Heidelberg (2007)
3. Dhaussy, P., Pillain, P.-Y., Creff, S., Raji, A., Le Traon, Y., Baudry, B.: Evaluating Context Descriptions and Property Definition Patterns for Software Formal Validation. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 438–452. Springer, Heidelberg (2009)
4. Dong, J.S., Hao, P., Qin, S., Sun, J., Yi, W.: Timed Automata Patterns. IEEE Transactions on Software Engineering 34(6), 844–859 (2008)
5. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in Property Specifications for Finite-State Verification. In: 21st International Conference on Software Engineering, pp. 411–420. IEEE Computer Society Press, Los Alamitos (1999)
6. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
7. Ghazel, M.: Using Stochastic Petri Nets for Level-Crossing Collision Risk Assessment. IEEE Transaction on Intelligent Transportation Systems 10(4), 668–677 (2009)
8. Ghazel, M., Mekki, A.: Assisting Specification and Consistency-Check of Temporal Requirements for Critical Systems. In: Software Engineering Research and Practice (SERP 2010), Las Vegas, Nevada, USA (2010)
9. Ghazel, M., Toguyéni, A., Yim, P.: State Observer for DES Under Partial Observation with Time Petri Nets. Journal of Discrete Event Dynamic Systems 19(2), 137–165 (2009)
10. Hülsbusch, M., König, B., Rensink, A., Semenyak, M., Soltenborn, C., Wehrheim, H.: Showing full semantics preservation in model transformation - A comparison of techniques. In: Méry, D., Merz, S. (eds.) IFM 2010. LNCS, vol. 6396, pp. 183–198. Springer, Heidelberg (2010)
11. Konrad, S., Cheng, B.H.C.: Real-time Specification Patterns. In: 27th International Conference on Software Engineering (ICSE 2005). St Louis, MO, USA (2005)
12. Küster, J.M.: Definition and validation of model transformations. Software and System Modeling 5(3), 233–259 (2006)
13. Larsen, K., Pettersson, P., Yi, W.: UPPAAL in a nutshells. International Journal of Software Tools for Technology Transfer 1, 134–152 (1997)
14. Mekki, A., Ghazel, M., Toguyéni, A.: Validating time-constrained systems using UML Statecharts Patterns and Timed Automata Observers. In: 3rd International Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS 2009), Rabat, Morroco (2009)
15. Mekki, A., Ghazel, M., Toguyéni, A.: Time-constrained systems validation using MDA model transformation. A railway case study. In: 8th ENIM IFAC International Conference of Modeling and Simulation (MOSIM 2010), Hammamet, Tunisia (2010)
16. Mekki, A., Ghazel, M., Toguyéni, A.: Patterns For Temporal RequireÛments Engineering - A Level Crossing Case Study. In: 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010), Madeira, Portugal (2010)
17. do Nascimento, F.A.M., da Silva Oliveira, M.F., Wagner, F.R.: Formal verification for embedded systems design based on MDE. In: Rettberg, A., Zanella, M.C., Amann, M., Keckeisen, M., Rammig, F.J. (eds.) IESS 2009. IFIP Advances in Information and Communication Technology, vol. 310, pp. 159–170. Springer, Heidelberg (2009)
18. Schmidt, D.C.: Model Driven Engineering. IEEE Computer 23(2), 25–31 (2004)

19. Seidner, C.: Vérification des EFFBDs: Model checking en Ingénierie Système. PhD Report Université de Nantes. France (2009)
20. Steel, J., Lawley, M.: Model-Based Test Driven Development of the Tefkat Model-Transformation Engine. In: International Symposium on Software Reliability Engineering, pp. 151–160. IEEE Computer Society Press, Los Alamitos (2004)
21. UML: Unified Modeling Language Specification, Version 2.2. OMG (2009)
22. Varro, D.: Automated formal verification of visual modeling languages by model checking. Software and Systems Modeling 3(2), 85–113 (2004)
23. Weis, T., Ulbrich, A., Geihs, K.: Model Metamorphosis. IEEE Software 20(5), 46–51 (2003)
24. Yovine, S.: KRONOS: a verification tool for real-time systems. International Journal of Software Tools for Technology Transfer 1, 123–133 (1997)
25. `http://www.dagstuhl.de/fileadmin/redaktion/Programm/Seminar/07241/07241.CaseStudy.pdf`

A Passivity-Based Approach to Group Coordination in Multi-agent Networks

Heath LeBlanc, Emeka Eyisi, Nicholas Kottenstette,
Xenofon Koutsoukos, and Janos Sztipanovits

Vanderbilt University, Nashville, TN 37203, U.S.A.
heath.j.leblanc@vanderbilt.edu

Abstract. Surveillance and convoy tracking applications often require groups of networked agents for redundancy and better coverage. Important goals upon deployment include the establishment of a formation around a target and synchronization of the output (e.g., velocity). Although there exist distributed algorithms using only local communication that achieve these goals, they typically ignore destabilizing effects resulting from implementation uncertainties, such as network delays and data loss. This paper resolves these issues by introducing a discrete-time distributed design framework that uses a compositional, passivity-based approach to ensure l_2^m-stability regardless of overlay network topology, in the presence of network delays and data loss. For the restricted case of a regular overlay network topology, this work shows that asymptotic formation establishment and output synchronization can be achieved. Finally, simulations of velocity-limited quadrotor unmanned air vehicles (UAVs) are presented to show the performance in the presence of time-varying network delays and varying amounts of data loss.

Keywords: Passivity, Compositionality, Deployment, Input-output stability, Overlay network, Output synchronization.

1 Introduction

Modern surveillance and convoy tracking applications often require deploying groups of unmanned aerial vehicles (UAVs). The benefit of using multiple UAVs is redundancy, which reduces the likelihood of missing interesting events on the ground, in the presence of obstructions caused by nonuniform terrain, vegetation, or man-made structures. Further, the additional UAVs provide greater breadth of coverage. A central task for such multi-agent systems is to establish a formation around an area of interest. For example, an n-gon with a target as its center, at the appropriate radius, may simultaneously provide significant redundancy and breadth of coverage. Another problem of interest when controlling autonomous UAVs is to synchronize a subset of the outputs. For example, consider flying the UAVs in a desired formation with identical speed and heading. This problem is known as the output synchronization problem.

Performing coordinated tasks in multi-agent systems using only local information has been studied extensively over the past decade [16], [17], [15]. Typically, in group coordination the desired formation emerges from the design of the control law. In [4],

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 135–149.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

the so-called information filter is used for formation stability of LTI systems. For coordination of nonlinear systems, contraction theory with wave variable communication [20], explicit design of Lyapunov vector fields [12], and passivity [1], [8], [2], and [7], have been used successfully.

Much of the above work—especially the passivity-based methods—has considered continuous-time systems; however, for implementation discrete-time design is needed. In addition, implementation uncertainties such as network delays and data loss must be taken into consideration. This paper focuses on decoupling the control design and discrete-time implementation by using a passivity-based framework inspired by work in telemanipulation [3], port-Hamiltonian systems [19], and network control [11].

In this work, we introduce a compositional network control system (NCS) design approach that guarantees passivity of the networked system. We show that a multi-agent network designed within the constraints of this framework is l_2^m-stable for any bidirectional overlay network with asymmetric delays whenever the input-output mapping of each agent is strictly-output passive. The stability result holds for packet-switched networks using easily enforced constraints. For the single-input, single-output (SISO) case, we perform steady-state analysis to show that the multi-agent network can achieve output synchronization or establish an n-gon upon deployment under appropriate assumptions. Finally, we provide simulations of velocity-limited quadrotor UAVs illustrating how the multi-agent network performs in establishing an n-gon in the presence of time-varying network delays and varying amounts of data loss.

The rest of the paper is organized as follows: Sect. 2 defines the formation establishment and output synchronization problems and provides background material. The distributed NCS design framework is described in Sect. 3. The theoretical results are detailed in Sect. 4. Section 5 presents simulations in Simulink/TrueTime. Concluding remarks are given in Sect. 6.

2 Preliminaries

In this paper we consider two problems: formation establishment and output synchronization. First, consider the problem of n agents establishing a formation around a target in $\mathbb{R}^2$. Assume a global inertial coordinate system and suppose the starting positions of the agents are arbitrary. The goal is to establish an n-gon, where the n agents tend to the coordinates of the vertices asymptotically. Formally, we assign a vertex ν_i of the n-gon to agent i, with position $y_i(k)$, $i = 1, 2, \ldots, n$. Then we require $\lim_{k\to\infty} \|y_i(k) - \nu_i\|_2 = 0$.

For the problem of output synchronization, the outputs y_i and y_j of any two agents i and j ($i \neq j$) must satisfy $\lim_{k\to\infty} \|y_i(k) - y_j(k)\|_2 = 0$.

For these problems, we consider a network of n interacting agents with communication topology described by a connected undirected graph, $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ describes the agents and $E \subset V \times V$ models the bidirectional communication. Additionally, each bidirectional link may have asymmetric, time-varying delays. The delays are denoted $d_{ij}(k)$ for link $(i, j) \in E$.

For the purpose of analysis, it is useful to introduce the *adjacency matrix*, $A = [a_{ij}]$, associated with graph G [5]. For an undirected graph, the adjacency matrix is a symmetric matrix (i.e., $A = A^{\mathsf{T}}$). An entry a_{ij} is equal to one if the edge $(i, j) \in E$ and

zero otherwise. Additionally, we define the set of *neighbors*, N_i, of a node i as those nodes which send messages to i, given by $N_i = \{j \in V \mid a_{ij} \neq 0\}$. Finally, we denote the number of neighbors by $|N_i| = n_i$.

The agents communicate and process signals in the extended l_2-space of functions that maps $\mathbb{Z}^+$ to $\mathbb{R}^m$, denoted l_{2e}^m. These signals are mapped onto l_2^m by the N-truncation operator $(\cdot)_N : l_{2e}^m \to l_2^m$, which simply nullifies the function values for indices strictly larger than $N-1$. Further, for all $f, g \in l_{2e}^m$ define $\langle f, g \rangle_N \triangleq \sum_{k=0}^{N-1} f^{\mathsf{T}}(k) g(k)$ and $\|(f)_N\|_2^2 \triangleq \langle f, f \rangle_N$. We use definitions for l_2^m-stability and passivity for discrete-time systems, which are analogous to the continuous-time counterparts in [18]:

Definition 1. *Given a discrete-time system with input-output mapping, $H : l_{2e}^m \to l_{2e}^m$, the discrete-time system is* l_2^m-stable if $u \in l_2^m \implies H(u) \in l_2^m$.

Definition 2. *Let $H : l_{2e}^m \to l_{2e}^m$. Then, for all $u \in l_{2e}^m$:*

1. *H is* passive if there exists some constant $\beta \in \mathbb{R}$ (the bias) such that $\langle H(u), u \rangle_N \geq -\beta, \forall N \in \mathbb{N}$;
2. *H* is *strictly output passive if there exists some constants $\beta \in \mathbb{R}$ and $\epsilon > 0$ such that $\langle H(u), u \rangle_N \geq \epsilon \|(H(u))_N\|_2^2 - \beta$, $\forall N \in \mathbb{N}$.*

We assume a synchronous network, with period T.[1] Further, each agent shares information only locally (no global shared resources). However, the desired setpoints are calculated prior to deployment. Finally, the agents begin execution at time index $k = 0$.

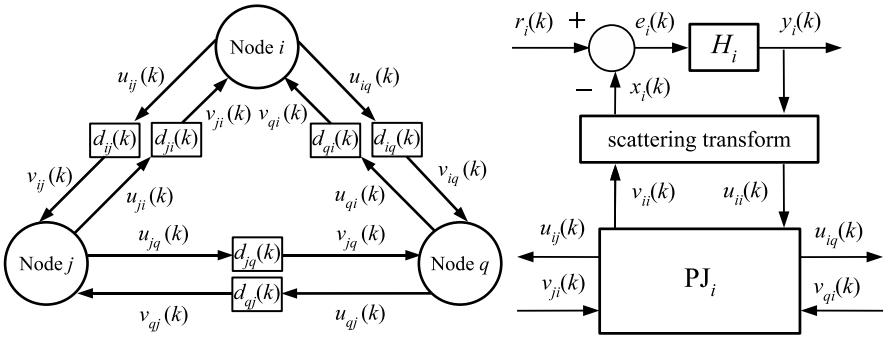
3 NCS Design

This section details the distributed network control system (NCS) design. The objective is to provide a passive-by-construction, discrete-time multi-agent network. In general, the overlay network is bidirectional with asymmetric delays. For simplicity, consider the three node network shown in Fig. 1a. Each node represents a quadrotor UAV, with each edge modeling the communication between UAVs. Realistically, each link in the network is subject to delay imposed by packet handling and transmission delays. This is modeled by the time-varying delays (e.g., $d_{ij}(k)$), shown in Fig. 1a. The u and v variables in the figure are power wave variables, which are described in Sect. 3.2.

3.1 Agent Model

The agent model is shown in Fig. 1b. Each agent i receives an input reference, r_i, which influences the output, y_i, of the agent through the system mapping, H_i. The mapping H_i describes a compensated plant, and is required to be strictly output passive. The variables x_i and y_i are transformed into the wave domain through the scattering transformation. The node's wave variables u_{ii} and v_{ii} are coupled to other nodes through a power junction, PJ_i, which allows two or more systems to be connected in a passivity-preserving manner [11]. The scattering transformation and power junction are crucial to ensuring passivity of the networked system and will be described in the next section.

[1] We assume the agents use a clock synchronization algorithm prior to deployment to ensure this assumption holds.



(a) A three node network. (b) Node structure.

Fig. 1. Network and node structure.

For simplicity, we model the quadrotor UAVs as a point mass in two dimensions. We denote the point mass system, $H_p : f_I \rightarrow y_I$, in which $f_I \in \mathbb{R}^2$ is the control force and $y_I \in \mathbb{R}^2$ is the position as depicted in Fig. 2a. The equations of motion are

$$\dot{y}_I(t) = v_I(t), \quad M\dot{v}_I(t) = f_I(t) \; .$$

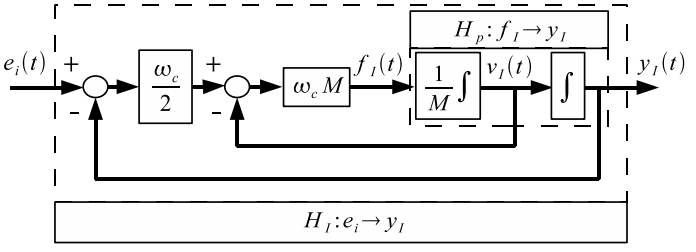
Using the point mass model for each agent i, we design a position control system, which we denote $H_I : e_i \rightarrow y_I$, shown in Fig. 2a. The inner loop gain of the compensator is $\omega_c M$ ($\omega_c > 0$) and the outer loop gain $\frac{\omega_c}{2}$. The overall equation of motion

$$\ddot{y}_I = -\omega_c \dot{y}_I - \frac{\omega_c^2}{2}(y_I - e_i) = -2\zeta\omega_n \dot{y}_I - \omega_n^2(y_I - e_i) \; ,$$

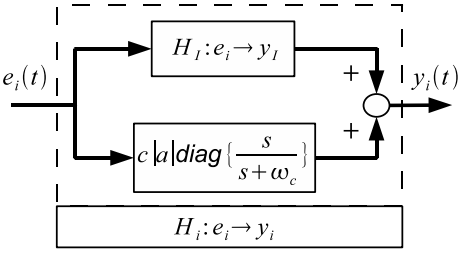
clearly indicates a stable second order system with natural frequency $\omega_n = \frac{\omega_c}{\sqrt{2}}$ and damping coefficient $\zeta = \frac{1}{\sqrt{2}}$, where $y_I = e_i$ at steady-state. It can be shown that the position control system is inside the sector $[a, 1]$, where $a = -\frac{1}{2(1+\sqrt{2})}$ [21], [10]. Therefore, the system $H_I : e_i \rightarrow y_I$ is not strictly output passive; however, by adding a high-pass filter in parallel, the system may be rendered strictly output passive, as depicted in Fig. 2b (with $c = 2$). Since $e_i = y_i = y_I$ at steady-state, the position of the system may be directly controlled. This model is discretized using a bilinear-like transform, called the inner-product equivalent sample and hold (IPESH) transform, which preserves the passive properties of the system [11].

3.2 Network Model

In distributed control applications the information transmitted across the network has inherent physical meaning. It is well known that transforming these physical variables into the wave domain can preserve passivity and stability for a single bidirectional connection [3] and for star networks [11]. In this paper, we extend these discrete-time approaches to distributed networks with arbitrary overlay topology. The network model is distributed in the sense that all nodes in the network communicate only locally.



(a) Non-passive UAV model.



(b) Strictly output passive UAV model.

Fig. 2. Simplified quadrotor UAV model.

We formally define the scattering transformation as follows. For each $i \in V$, the *scattering transformation* produces power waves $u_{ii}(k)$ and $v_{ii}(k)$ defined by

$$u_{ii}(k) = \tfrac{1}{\sqrt{2b_i}}(b_i y_i(k) + x_i(k)) \,, \tag{1a}$$

$$v_{ii}(k) = \tfrac{1}{\sqrt{2b_i}}(b_i y_i(k) - x_i(k)) \,. \tag{1b}$$

This definition is similar to the one in [14], with the force and velocity variables replaced with x_i and y_i. Traditionally, the scattering formalism has been applied to power variables (effort and flow) while closing the loop on velocity. In this work, the scattering formalism is used abstractly (without the physical interpretation) to close the loop on position. The scattering transformation is treated as a mathematical definition, where the characteristic impedance, b_i, has the appropriate units for physical consistency (in this case b_i is unitless).

Next, we define the power junction, which allows two or more systems to be connected in the wave domain in a passivity-preserving manner.

Definition 3. *Fix $m, p \in \mathbb{N}, p \geq 2$. Then, a power junction is a function $f\colon l_{2e}^{mp} \to l_{2e}^{mp}$, which satisfies for all $\xi \in l_{2e}^{mp}$ and all $k \in \mathbb{Z}^+$ the inequality $\xi^{\mathsf{T}}(k)\xi(k) \geq f(\xi(k))^{\mathsf{T}} f(\xi(k))$.*

The vector $\xi(k)$ in the definition of the power junction is formed by concatenating the p inputs in l_{2e}^{m} into a single mp-dimensional column vector. For analyzing our network model, it is useful to pair the p inputs to their corresponding outputs in the output column vector, $f(\xi(k))$, and partition the set of pairs into two disjoint sets S_{in} and S_{out}. These sets denote the net flow of power into and out of the power junction, respectively.

Formally, for $i \in S_{in}$ and $o \in S_{out}$, let $u_i, v_o \in l_{2e}^m$ denote the inputs and $v_i, u_o \in l_{2e}^m$ denote the outputs of the power junction. Then the inequality in Def. 3 may be rewritten as

$$\sum_{i \in S_{in}} u_i^{\mathsf{T}}(k) u_i(k) - v_i^{\mathsf{T}}(k) v_i(k) \geq \sum_{o \in S_{out}} u_o^{\mathsf{T}}(k) u_o(k) - v_o^{\mathsf{T}}(k) v_o(k) \; . \tag{2}$$

We implement each node's power junction as a linear set of equations. Specifically, we use the following equations. For each $i \in V$, $j \in N_i$, and $k \in \mathbb{Z}^+$, the outgoing waves are computed as

$$u_{ij}(k) = \tfrac{1}{\sqrt{n_i}} u_{ii}(k) \; , \tag{3a}$$

$$v_{ii}(k) = \tfrac{1}{\sqrt{n_i}} \sum_{j \in N_i} v_{ji}(k) \; . \tag{3b}$$

Although the functional form of the power junction is not constrained to be linear, these equations simplify the steady-state analysis and exhibit a local averaging behavior in regular networks. This can be seen as follows. Consider the wave variables that influence the power junction at a given node i and suppose $n_i = n_j \equiv \eta, \forall i, j \in V$ (i.e., a regular network). Then, for each $j \in N_i$, $v_{ji}(k) = u_{ji}(k - d_{ji}(k))$. Thus, an expression for $v_{ii}(k)$ is given by

$$v_{ii}(k) = \tfrac{1}{\sqrt{\eta}} \sum_{j \in N_i} v_{ji}(k) = \tfrac{1}{\sqrt{\eta}} \sum_{j \in N_i} u_{ji}(k - d_{ji}(k)) = \tfrac{1}{\eta} \sum_{j \in N_i} u_{jj}(k - d_{ji}(k)) \; .$$

Therefore, in regular networks, the input wave variable, $v_{ii}(k)$, is the average of its neighbors' delayed output wave variables, $u_{jj}(k - d_{ji}(k))$, $j \in N_i$.

Due to the presence of delays and data loss, some (or all) of the $v_{ji}(k)$ may not be received at time k, in which case $v_{ji}(k) \triangleq 0$. Handling delayed and dropped packets as null packets satisfies the synchronous assumption and preserves passivity [3]. Before proceeding to describe the constraints on delayed and lost data, we prove our claim that the implementation given by (3) satisfies the definition of a power junction.

Lemma 1. *The implementation defined by (3) satisfies the power junction constraint.*

Proof. From the remarks following the power junction definition, we must show

$$u_{ii}^{\mathsf{T}}(k) u_{ii}(k) - v_{ii}^{\mathsf{T}}(k) v_{ii}(k) \geq \sum_{j \in N_i} u_{ij}^{\mathsf{T}}(k) u_{ij}(k) - v_{ji}^{\mathsf{T}}(k) v_{ji}(k) \; . \tag{4}$$

Clearly, a sufficient condition for satisfying (4) is to enforce the following constraints for each component $l = 1, 2, \ldots, m$,

$$\sum_{j \in N_i} u_{ij_l}^2(k) \leq u_{ii_l}^2(k) \text{ and } v_{ii_l}^2(k) \leq \sum_{j \in N_i} v_{ji_l}^2(k) \; .$$

The first inequality holds with equality. Indeed, by (3a),

$$\sum_{j \in N_i} u_{ij_l}^2(k) = \sum_{j \in N_i} \tfrac{1}{n_i} u_{ii_l}^2(k) = u_{ii_l}^2(k) \; .$$

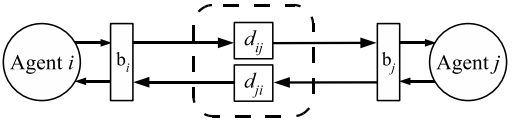


Fig. 3. A point-to-point connection using the scattering formalism to ensure passivity of the bidirectional connection subject to asymmetric time-varying delays, shown inside the dashed box.

For the second inequality, combine (3b) with the Cauchy-Schwarz inequality to get

$$v_{ii_l}^2(k) = \tfrac{1}{n_i}\left(\sum_{j\in N_i} v_{ji_l}(k)\right)^2 \leq \sum_{j\in N_i} v_{ji_l}^2(k)\ .$$

□

Finally, we constrain the network model by preventing retransmission of data for each agent. Also, as mentioned above, whenever receiver's buffers are empty, we process null packets. Based on these assumptions, each channel $(i,j) \in E$ satisfies the following inequality regardless of time-varying delays and data loss [3],

$$\|(v_{ij})_N\|_2^2 \leq \|(u_{ij})_N\|_2^2, \text{ holds } \forall N \in \mathbb{N}\ . \tag{5}$$

This inequality states that each channel, viewed as the input-output mapping shown in Fig. 3, is passive.

4 Analysis

4.1 Passivity of the Networked System

In this section we first prove that the network model is passive and then show that the input-output mapping describing the networked system is strictly output passive. Figure 4 shows the passive network. The following lemma proves that the portion inside the dashed box of Fig. 4 is passive.

Lemma 2. *Consider a network of n interacting dynamic systems constrained to the design framework described in Sect. 3. Then, the global energy constraint*

$$\sum_{i=1}^{n}\left(\|(u_{ii})_N\|_2^2 - \|(v_{ii})_N\|_2^2\right) \geq 0\ , \tag{6}$$

is satisfied for all $N \in \mathbb{N}$, regardless of time-varying delays and data loss.

Proof. Sum the power constraints (4) of each node i from time $k = 0$ to $k = N-1$ and then sum the resulting inequalities over all nodes (swap the position of v_{ji} and v_{ij} in the sum of sums). Then, invoke (5) to obtain

$$\sum_{i=1}^{n}(\|(u_{ii})_N\|_2^2 - \|(v_{ii})_N\|_2^2) \geq \sum_{i=1}^{n}\sum_{j\in N_i}\left(\|(u_{ij})_N\|_2^2 - \|(v_{ij})_N\|_2^2\right) \geq 0\ .$$

□

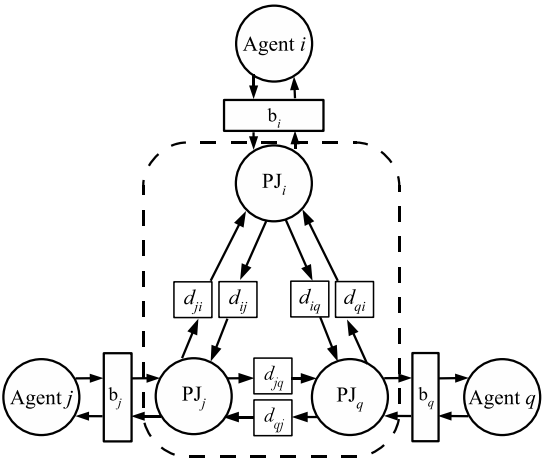


Fig. 4. A three node network illustrating the passive network, shown inside the dashed box.

The energy constraint of Lemma 2 also implies that collectively, the mapping from the y_i to the x_i, $i = 1, \ldots, n$, is passive (see Fig. 1b). To show this, consider the following power constraint, which may easily be derived from (1a) and (1b)

$$\tfrac{1}{2}(u_{ii}^{\mathsf{T}}(k)u_{ii}(k) - v_{ii}^{\mathsf{T}}(k)v_{ii}(k)) = y_i^{\mathsf{T}}(k)x_i(k)\ . \tag{7}$$

Substitute (7) into the global energy constraint of Lemma 2 to obtain

$$\sum_{i=1}^{n}\langle y_i, x_i\rangle_N \geq 0\ . \tag{8}$$

Define $x(k)$ and $y(k)$ as the $nm \times 1$ column vectors formed by concatenating the $x_i(k)$ and $y_i(k)$, respectively, of each node. Then, it follows that $\langle y, x\rangle_N \geq 0$, which satisfies the definition of passivity in Def. 2, with $\beta = 0$.

We conclude the section by proving that the entire networked system (e.g., the three node system in Fig. 4) is strictly output passive for arbitrary network topologies.

Theorem 1. *Consider a network of n interacting dynamic systems constrained to the design framework described in Sect. 3. Define $r(k)$ and $y(k)$ as the $nm \times 1$ column vectors formed by concatenating the $r_i(k)$ and $y_i(k)$, respectively, of each node. Finally, define the input-output mapping $H : l_{2e}^{nm} \to l_{2e}^{nm}$ such that $H(r(k)) = y(k)$. Then, H is strictly output passive.*

Proof. Since each H_i is strictly output passive, there exists $\epsilon_i > 0$ and β_i, for all $i \in V$, such that

$$\langle y_i, e_i\rangle_N \geq \epsilon_i \|(y_i)_N\|_2^2 - \beta_i. \tag{9}$$

Making the substitution, $x_i(k) = r_i(k) - e_i(k)$ into (8) and using the linearity of the inner-product, gives

$$\sum_{i=1}^{n}\langle y_i, r_i\rangle_N \geq \sum_{i=1}^{n}\langle y_i, e_i\rangle_N. \tag{10}$$

Substituting (9) into (10) yields $\sum_{i=1}^{n} \langle y_i, r_i \rangle_N \geq \epsilon \sum_{i=1}^{n} \|(y_i)_N\|_2^2 - \beta$, where $\epsilon = \min_i\{\epsilon_i\}$ and $\beta = \sum_{i=1}^{n} \beta_i$. This can be rewritten as $\langle y, r \rangle_N \geq \epsilon \|(y)_N\|_2^2 - \beta$. □

4.2 Stability

The previous result shows that the networked system defined by the mapping H is strictly output passive. It then follows that H is l_2^m-stable.

Theorem 2. *The mapping $H(r(k)) = y(k)$ defined in Theorem 1 is l_2^m-stable.*

Proof. We begin with the notion of finite l_2^m-gain. The map G has finite l_2^m-gain if there exists finite constants γ, β such that for all $N \in \mathbb{N}$, $\|(G(u))_N\|_2 \leq \gamma \|(u)_N\|_2 + \beta$, $\forall u \in l_{2e}^m$.

It is well known in continuous-time [18] and has been shown for discrete-time [9] that a sufficient condition for a system to have finite l_2^m-gain is for the system to be strictly output passive. Therefore, by Theorem 1, H has finite l_2^m-gain.

Now suppose $u \in l_2^m$ (i.e., $\|u\|_2 < \infty$). Then take $N \to \infty$ in the definition of finite l_2^m-gain . This leads to $\|G(u)\|_2 \leq \gamma \|u\|_2 + \beta < \infty$, $\forall u \in l_2^m$. Therefore, $H(u) \in l_2^m$. By Definition 1, H is l_2^m-stable. □

From the proof of Theorem 2, we see that any system that is strictly output passive is necessarily l_2^m-stable. Therefore, each agent described by H_i is inherently stable. The benefit of the passivity-based network framework is that it ensures that interactions caused by the network do not destabilize the networked multi-agent system. This result holds even in the presence of time-varying delays and data loss (under the assumptions outlined in Sect. 3.2) because the passivity results hold. Moreover, the networked multi-agent system will remain stable regardless of network topology.

4.3 Steady-State Analysis, n-gon Establishment, and Output Synchronization

To analyze the behavior of the coupled multi-agent system, we consider the system at steady-state. Of course, the system will not reach steady-state in the presence of time-varying delays and packet loss. However, the system will approach the ideal steady-state case for constant reference inputs even with time-varying delays and data loss because by Theorem 1, the system is strictly output passive. For simplicity, we assume the system is SISO. If the degrees of freedom of the system are decoupled, this result may be applied to MIMO systems.

Theorem 3. *Consider a network of n SISO agents designed using the framework described in Sect. 3 and assume no time delays or data loss. Assume that constant inputs, r_i, result in steady-state gains g_i, for $i = 1, 2, \ldots, n$. Then the steady-state output of node i is given by*

$$y_i = \frac{g_i}{b_i g_i + 1} \left(r_i + \frac{\sqrt{2b_i}}{\sqrt{n_i}} \sum_{j \in N_i} \frac{1}{\sqrt{2b_j n_j}} \left(\frac{b_j g_j - 1}{g_j} y_j + r_j \right) \right) . \tag{11}$$

Proof. Since time delays and data loss are ignored, we drop the time index. Using the relation $e_i = r_i - x_i$ and replacing H_i with g_i, the input-output relation $y_i = H_i(e_i)$ may be written as $y_i = g_i(r_i - x_i)$. First, substitute this into (1b) and solve for x_i. Then substitute this value of x_i back into $y_i = g_i(r_i - x_i)$ to get

$$y_i = \tfrac{g_i}{b_i g_i + 1} r_i + \tfrac{\sqrt{2b_i} g_i}{b_i g_i + 1} v_{ii} \ . \tag{12}$$

Combining $v_{ji} = u_{ji}$ with (3a) at node j (roles of j and i are reversed), produces $v_{ji} = \frac{1}{\sqrt{n_j}} u_{jj}$. Substituting this into (3b) for node i yields

$$v_{ii} = \tfrac{1}{\sqrt{n_i}} \sum_{j \in N_i} \tfrac{1}{\sqrt{n_j}} u_{jj} \ . \tag{13}$$

Now, solving $y_j = g_j(r_j - x_j)$ for x_j and substituting into (1a) at node j produces

$$u_{jj} = \tfrac{1}{\sqrt{2b_j}} \left(\tfrac{b_j g_j - 1}{g_j} y_j + r_j \right) \ . \tag{14}$$

Substitute (14) into (13) to get $v_{ii} = \frac{1}{\sqrt{n_i}} \sum_{j \in N_i} \frac{1}{\sqrt{2b_j n_j}} \left(\frac{b_j g_j - 1}{g_j} y_j + r_j \right)$. Finally, substitute this into (12) to obtain (11). □

Theorem 3 provides a system of n equations describing the system asymptotically (as $k \to \infty$). The system of equations described by (11) are clearly coupled and depend on the overlay network structure. For the case of a regular network, the following corollary characterizes the system of equations and provides the means to precalculate the reference inputs to asymptotically achieve a desired setpoint. For the two-dimensional agent model described in Sect. 3.1 the two degrees of freedom are decoupled, so we use this corollary to establish an n-gon around the target, as described in Sect. 2.

Corollary 1. *Consider a network of n SISO agents with a regular overlay network topology (i.e., $n_i = n_j \equiv \eta \ \forall i, j \in V$). If all of the systems H_i have identical steady-state gain g and each scattering transformation has the same impedance b, the system of steady-state equations may be written as*

$$y = \tfrac{g}{bg+1} \left(r + \tfrac{1}{\eta} A \left(\tfrac{bg-1}{g} y + r \right) \right) \ , \tag{15}$$

where y and r are defined in Theorem 1 and A is the adjacency matrix of the regular overlay network topology. Assuming the inverse of $(\eta I + A)$ exists, we may solve this equation for r to obtain

$$r = \tfrac{1}{g} (\eta I + A)^{-1} \left((bg+1)\eta I - (bg-1) A \right) y \ . \tag{16}$$

Next, we show that the output synchronization problem is achieved under the conditions in which the steady-state values are achieved in regular networks. In the presence of time-varying delays and packet loss the outputs will approach synchronization, but will not synchronize unless packet loss ceases and the delays become fixed and remain fixed.

Theorem 4. *Assume the hypotheses of Theorem 3, but with $r_i \equiv \rho$, $g_i \equiv g$, and $b_i \equiv b$. Provided $bg > 0$ and the network has a regular network topology with $n_i \equiv \eta > 0$, the output synchronization problem is achieved.*

Proof. For the case of common reference input, ρ, the system of equations given by (15) may be solved for y as follows

$$y = g\left((bg+1)I - \tfrac{bg-1}{\eta}A\right)^{-1}(I + \tfrac{1}{\eta}A)\rho\mathbf{1} \triangleq gB_1^{-1}B_2\rho\mathbf{1} \triangleq B\rho\mathbf{1}\,, \qquad (17)$$

where $\mathbf{1}$ is the column vector of ones, $B_1 \triangleq (bg+1)I - \frac{bg-1}{\eta}A$, $B_2 \triangleq I + \frac{1}{\eta}A$, and $B = gB_1^{-1}B_2$. Therefore, we have to show that the matrix B in (17) exists and has a right eigenvector of $\mathbf{1}$ corresponding to eigenvalue g. Then the asymptotic outputs $\mathbf{y}$ will have an aligned equilibrium point of $g\rho\mathbf{1}$, and thus output synchronization is achieved.

For existence, we must show that the matrix inverse, B_1^{-1}, exists. Using Geršgorin's Disc Theorem, we show that all eigenvalues of B_1 are positive, so that B_1 is positive definite and therefore invertible [6]. Since the overlay network graph is η-regular, the sum of the absolute values of terms in each row i excluding the diagonal term ($B_{1_{ii}}$) is $|bg-1|$. The diagonal term of each row is $bg+1$. Therefore, the Geršgorin discs are all equal, and centered at $bg+1$ with radius $|bg-1|$. Since $bg > 0$, the entirety of all of the discs are positive, so by Geršgorin's Disc Theorem, all eigenvalues of B_1 are positive.

Next, we show that B has right eigenvector $\mathbf{1}$ corresponding to eigenvalue g (for brevity we drop the reference to $\mathbf{1}$). First, since each node has degree η, A has eigenvalue η. Hence, $\lambda_{B_1} = \lambda_{B_2} = 2$. Therefore, since $\lambda_{B_1^{-1}} = \frac{1}{\lambda_{B_1}}$, B has eigenvalue g. □

5 Simulations

The experimental setup involves a network of eight velocity-limited quadrotor UAVs that communicate in a regular overlay network topology, each with degree $\eta = 4$, and a synchronous sampling period of $T = 0.01$ seconds. The UAVs move in the plane, and the goal is to form an octagon with eah UAV 100m from a target centered at the origin. The initial points of the UAVs are randomly selected within the region between a 1000m square and the circle with 100m radius, both centered at the origin. We model the UAVs as described in Section 3.1, but with saturation in the actuators to limit the velocity. The parameters in Corollary 1 used to precalculate the reference values are: $g = 1$ and $b = 1$. The dynamics of the agents are implemented in Simulink, while TrueTime is used to simulate the network dynamics and communication between UAVs. The network protocol used is IEEE 802.11b, with a speed of 11 Mbps.

5.1 Evaluation

In [13], we evalutated the performance of the networked UAVs in establishing an octagon of radius 100m, under various network conditions. In particular, we considered: no delays nor data loss (nominal case), nonuniform constant delays, time-varying

delays, data loss, and combined effects (time-varying delays and data loss). In this paper, we demonstrate the performance in establishing the octagon in the case where there are time-varying delays and varying amounts of data loss.

Scenario 1: No data loss. In [13], we only studied the effects of time-varying delays with no additional delay bias, which implies that there could be time steps with zero delay. Here, we consider the more realistic scenario, where the time-varying delays have additional nonuniform constant delay bias in all the communication channels of the network. The delay biases are between 1 and 2 seconds. As was done in [13], to simulate the time-varying delays, we introduce a disturbance node in the network. The sampling period of the disturbance node is 0.05 seconds, and the disturbance node floods the network with disturbance packets based on a Bernoulli process with parameter d (in this case, $d = 0.5$). The disturbance node samples a uniformly distributed random variable $X[k] \in [0, 1]$ every 0.05 seconds. If $X[k] > d$, a disturbance packet is forced on the network. Figure 5(a) shows the average and maximum errors for the nominal case and the combined delays with no data loss. The results show that in the presence of combined delays, the UAVs remain stable and settle to the desired configuration.

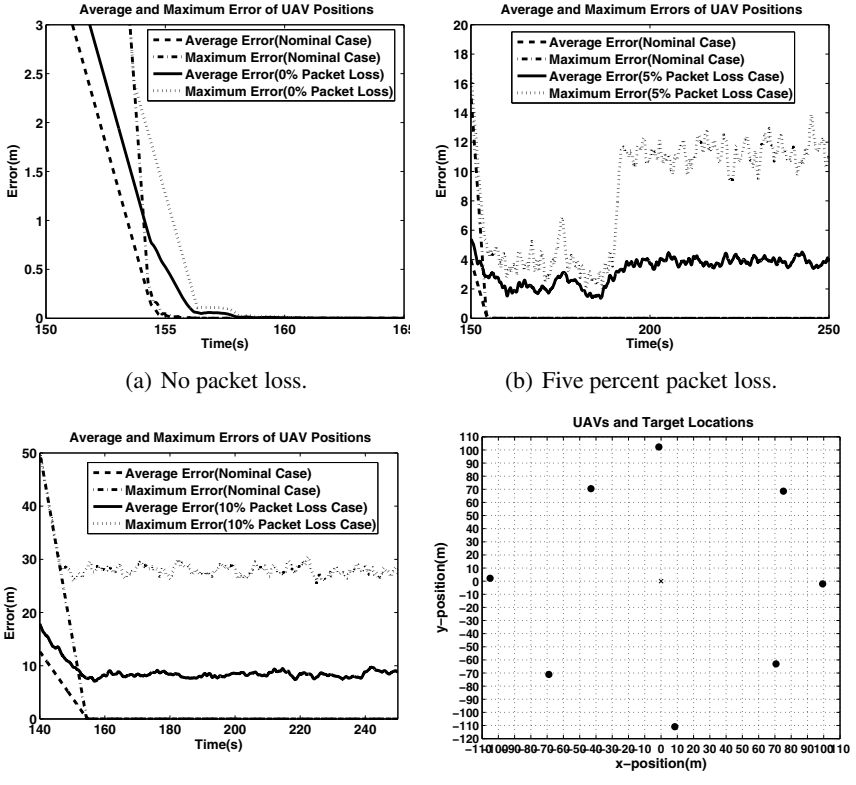
Scenario 2: 5% data loss. In this scenario, 5% packet loss is introduced with the combined delays described above. A probabilistic model is used to implement the loss of packets in the channels. The results for this case are shown in Fig. 5(b). The results illustrate that with packet loss the system does not reach steady-state, and therefore, does not achieve zero error. However, the UAVs still manage to come very close to the desired configuration, demonstrating the resilience of the network. The UAV's end up within a maximum error of 14 meters and an average error of 4 meters of the desired configuration.

Scenario 3: 10% data loss. In this scenario, the probability of packet loss is increased to 10% while still imposing the same combined delays as in Scenarios 1 and 2. The UAVs will never reach steady-state; however, the system is still stable (in an input-output sense). The results for this case are shown in Fig. 5(c) and Fig. 5(d). The maximum error in this case is considerably larger than in Scenario 2. This is caused by the UAV located above and to the left of the target, shown in Fig. 5(d).

Table 1 shows a summary of the results, where average and maximum errors are averaged over time. The averaging is done over time samples after the errors have settled within 2 meters of the final value of the errors. For each of the three cases, the same nonuniform constant delays are used. From the table it can be seen that increasing packet loss significantly affects performance.

Table 1. Experimental Results Summary.

Scenarios	0%	5%	10%
Average Error over time (m)	0.0	3.9	8.5
Maximum Error over time (m)	0.0	11.5	28.0



(a) No packet loss.

(b) Five percent packet loss.

(c) Ten percent packet loss.

(d) UAV positions with ten percent loss.

Fig. 5. Results with combined delays and varying amounts of data loss.

6 Conclusions

Discrete-time implementation of networked multi-agent systems introduces significant challenges caused by effects such as network delays and data loss. This paper proposes a passive-by-construction distributed network control design framework that ensures l_2^m-stability in the presence of these network effects. Using steady-state analysis, we show how to control the agents in the multi-agent network in order to establish an n-gon upon deployment, and we show that output synchronization can be achieved. Simulations of velocity-limited quadrotor unmanned air vehicles (UAVs) are presented to show the performance in the presence of network delays and varying amounts of data loss.

Acknowledgements. This work is supported in part by the National Science Foundation (CNS-1035655, CCF-0820088), the U.S. Army Research Office (ARO W911NF-10-1-0005), the U.S. Air Force Office of Scientific Research (MURI FA9550-06-0312), the U.S Army Research Laboratory (ARL W911NF-087-2-0004), and Lockheed-Martin. The views and conclusions contained herein are those of the authors and should not

be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

1. Arcak, M.: Passivity as a Design Tool for Group Coordination. IEEE Transactions on Automatic Control 52(8), 1380–1390 (2007)
2. Bai, H., Arcak, M., Wen, J.T.: Rigid Body Attitude Coordination without Inertial Frame Information. Automatica 44(12), 3170–3175 (2008)
3. Chopra, N., Berestesky, P., Spong, M.W.: Bilateral Teleoperation over Unreliable Communication Networks. IEEE Transactions on Control Systems Technology 16(2), 304–313 (2008)
4. Fax, J.A., Murray, R.M.: Information Flow and Cooperative Control of Vehicle Formations. IEEE Transactions on Automatic Control 49(9), 1465–1476 (2004)
5. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer-Verlag, New York, Inc. (2001)
6. Horn, R.A., Johnson, C.R.: Matrix Analysis. University Press, Cambridge (1990)
7. Igarashi, Y., Hatanaka, T., Fujita, M., Spong, M.W.: Passivity-Based Output Synchronization in SE(3). In: American Control Conference, pp. 723–728 (2008)
8. Ihle, I.A.F., Arcak, M., Fossen, T.I.: Passivity-Based Designs for Synchronized Path-Following. Automatica 43(9), 1508–1518 (2007)
9. Kottenstette, N., Antsaklis, P.: Stable Digital Control Networks for Continuous Passive Plants Subject to Delays and Data Dropouts. In: 46th IEEE Conference on Decision and Control, New Orleans, Louisiana, USA, pp. 4433–4440 (2007)
10. Kottenstette, N., Porter, J.: Digital Passive Attitude and Altitude Control Schemes for Quadrotor Aircraft. In: 7th International Conference on Control and Automation, pp. 1761–1768 (2009)
11. Kottenstette, N., Hall, J., Koutsoukos, X., Antsaklis, P., Sztipanovits, J.: Digital Control of Multiple Discrete Passive Plants over Networks. International Journal of Systems, Control and Communications (IJSCC): Special Issue on Progress in Networked Control Systems (2009)
12. Lawrence, D.A., Frew, E.W., Pisano, W.J.: Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control. Journal of Guidance, Control, and Dynamics 31(5), 1220–1229 (2008)
13. LeBlanc, H., Eyisi, E., Kottenstette, N., Koutsoukos, X., Sztipanovits, J.: A Passivity-Based Approach to Deployment in Multi-Agent Networks. In: 7th International Conference on Informatics in Control, Automation and Robotics, vol. 1, pp. 53–62 (2010)
14. Niemeyer, G., Slotine, J.J.E.: Telemanipulation with Time Delays. International Journal of Robotics Research 23(9), 873–890 (2004)
15. Olfati-Saber, R.: Flocking for Multi-agent Dynamic Systems: Algorithms and Theory. IEEE Transactions on Automatic Control 51(3), 401–420 (2006)
16. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and Cooperation in Networked Multi-agent Systems. Proceedings of the IEEE 95(1), 215–233 (2007)
17. Ren, W., Beard, R., Atkins, E.: A Survey of Consensus Problems in Multi-agent Coordination. In: Proceedings of the American Control Conference, vol. 3, pp. 1859–1864 (2005)
18. van der Schaft, A.J.: L_2-Gain and Passivity in Nonlinear Control. Springer-Verlag New York, Inc., Secaucus (1999)
19. Stramigioli, S., Secchi, C., van der Schaft, A.J., Fantuzzi, C.: Sampled Data Systems Passivity and Discrete Port-Hamiltonian Systems. IEEE Transactions on Robotics 21(4), 574–587 (2005)

20. Wang, W., Slotine, J.J.: Contraction Analysis of Time-Delayed Communications and Group Cooperation. IEEE Transactions on Automatic Control 51(4), 712–717 (2006)
21. Zames, G.: On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems Part One: Conditions Derived Using Concepts of Loop Gain, Conicity, and Positivity. IEEE Transactions on Automatic Control 11(2), 228–238 (1966)

Part II
Robotics and Automation

Adhering to Terrain Characteristics for Position Estimation of Mobile Robots

Michael Brunner[1], Dirk Schulz[1], and Armin B. Cremers[2]

[1] Department of Unmanned Systems, Fraunhofer-Institute FKIE
Neuenahrer Straße 20, 53343 Wachtberg, Germany
[2] Department of Computer Science, University of Bonn
Roemerstr. 164, 53117 Bonn, Germany
{michael.brunner,dirk.schulz}@fkie.fraunhofer.de,
abc@cs.uni-bonn.de

Abstract. Outdoor environments bear the problem of different terrains along with changing driving properties. Therefore, compared to indoor environments, the kinematics of mobile robots is much more complex. In this paper we present a comprehensive approach to learn the function of outdoor kinematics for mobile robots. Future robot positions are estimated by employing Gaussian process regression (GPR) in combination with an Unscented Kalman filter (UKF). Our approach uses optimized terrain models according to the classification of the current terrain – accomplished through Gaussian process classification (GPC) and a second order Bayesian filter (BF). Experiments showed our approach to provide more accurate estimates compared to single terrain model methods, as well as to be competitive to other dynamic approaches.

Keywords: Mobile robots, Position estimation, Terrain classification, Gaussian process, Machine learning.

1 Introduction

The kinematics of mobile robots in outdoor scenarios is much more complex than in indoor environments due to the varying terrain conditions. Therefore, truly reliable velocity controls for robots which are able to drive up to *4 m/s* or even faster (e.g. Figure 1) are hard to design. The drivability is primarily determined by the terrain conditions. Thus, we developed a machine learning system which predicts the future positions of mobile robots using optimized terrain models.

In this paper we present a Gaussian processes (GPs) based method for estimating the positions of a mobile robot. Our approach considers the different terrain conditions to improve prediction quality. Gaussian process regression (GPR) models are utilized to estimate the translational and rotational velocities of the robot. These estimated velocities are transferred into the position space using an unscented Kalman filter (UKF). By projecting the uncertainty values of the GPR estimates onto the positions, the UKF enables us to also capitalize the GPR uncertainties. To distinguish the different terrains the robot is traversing, we classify the spectrums of vertical accelerations using Gaussian process classification (GPC). The transitions between terrains are modeled by a 2nd

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 153–169.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

Fig. 1. The experimental Longcross platform *Suworow*.

order Bayesian filter (BF). This allows us to assign different probabilities to distinct terrain sequences, as we incorporate the properties of the classifier.

The remainder of this paper is organized as follows: Related work is presented in Section 2. In Section 3 we explain GPs. In Section 4 we describe our dynamic approach. Experiments are shown in Section 5. We conclude in Section 6.

2 Related Work

Several works study Gaussian processes (GPs) and their application to machine learning problems. A detailed view is provided by Rasmussen and Williams [15]. Other works are Williams [21], and MacKay [13].

There is a lot of research on prediction of positions or trajectories of mobile robots. Many different systems are proposed, like a stereo-vision approach [1], probability networks [3] or an approach using a particle filter combined with a Monte-Carlo method [17]. In [16] artificial neural networks (ANN) are employed to build a model predictive controller (MPC). The current prediction error is considered to improve the quality of the velocity estimations. Another least-square support vector machine (LS-SVM) based controller adjusts it's data model iteratively by removing the least important data point from the model when adding a new point [12].

Similar to the work presented here, GPs are used by Girard et al. [7] to make predictions several time steps ahead. The uncertainty of the previous step is integrated in the regression to track the increasing uncertainty of the estimation. Hence, the uncertainty value of the GP is the accumulated uncertainty of the previous time series. In contrast, we are able to relate the Gaussian uncertainties by using an UKF. A similar approach has been suggested by [11], and [10]. They estimated the trajectory of an autonomous blimp by combining GPR with an UKF or ordinary differential equations (ODE), respectively. Localization of wireless devices, like mobile telecommunication devices or mobile robots is solved by modeling the probability distribution of the signal strength with GPs integrated in a BF [6].

Several works employ information about the current terrain conditions to improve the navigation systems of mobile robots. One intuitive approach is to distinguish

between traversable and non-traversable terrain [5], [14]. In contrast to the binary separation, Weiss et al. [20] are using a SVM to classify the vertical accelerations and hence the terrain type. Using spectral density analysis (SDA) and principal component analysis (PCA) Brooks et al. [2] concentrate on the vibrations a mobile robot experiences while driving. The preprocessed data records are categorized through a voting scheme of binary classifiers. Another way to analyse the driving conditions is to measure the slippage [8], [19].

Kapoor et al. [9] are using GPs with pyramid-match kernels to classify objects from visual data. Urtasun and Darrell [18] chose a GP latent variable model to achieve a better classification by reducing the input dimension.

Although many works concentrate on position estimation as well as terrain classification, none combined GPR and GPC to solve both problems at once for a velocity controller of high speed outdoor robots.

3 Gaussian Processes

GPs are applicable to regression as well as classification tasks. In contrast to other methods GPs do not have any parameters that have to be determined manually. However, the kernel function affects the properties of a GP essentially and must be chosen by hand. The parameters of of GPs can be automatically optimized using the training data. Furthermore, GPs provide uncertainty values additionally to the estimates. These properties makes GPs attractive for regression and classification task like position estimation and terrain classification. However, a drawback of GPs is their running time which is quadratic in the number of training cases due to the inversion of the kernel matrix.

3.1 Regression

Computing the predictive distribution of GPs consists of three major steps. First, we determine the Gaussian distribution of the training data and the test data. To integrate the information of the training data into the later distribution, we compute the joint distribution. Finally, this joint distribution is transformed to the predictive equations of GPs by conditioning it completely on the training data.

Let $X = [x_1, \ldots, x_n]^\top$ be the matrix of the n training cases x_i. The measured process outputs are collected in $y = [y_1, \ldots, y_n]^\top$. The noise of the random process is modeled with zero mean and variance σ_n^2. The kernel function is used to computed the similarities between two cases. Our choice is the squared exponential kernel given by

$$k(x_i, x_j) = \exp(-\tfrac{1}{2}|x_i - x_j|^2), \tag{1}$$

where k is the kernel and x_i, x_j are two inputs. A further quantity we need to provide the prior distribution of the training data is the kernel matrix of the training data $K = K(X, X)$. It is given by $K_{ij} = k(x_i, x_j)$ using the kernel function. Now, we are able to specify the distribution of y:

$$y \sim \mathcal{N}(0, K + \sigma_n^2 I). \tag{2}$$

We are trying to learn the underlying function of the process. To get the distribution of the values of the underlying function $f = [f_1, \ldots, f_n]^\top$ we simply need to neglect the noise term from the distribution of the training data.

$$f \sim \mathcal{N}(0, K). \tag{3}$$

Secondly, we determine the distribution of the test data. Let the set of n_* test cases be assembled in a matrix as the training data, $X_* = [x_{*_1}, \ldots, x_{*_{n_*}}]^\top$. $f_* = [f_{*_1}, \ldots, f_{*_{n_*}}]^\top$ are the function values of the test cases. The normal distribution of the test data is therefore given by

$$f_* \sim \mathcal{N}(0, K_{**}), \tag{4}$$

where $K_{**} = K(X_*, X_*)$ is the kernel matrix of the test cases, representing the similarities of this data points.

To combine training and test data distributions in a joint Gaussian distribution we further require the kernel matrix of both sets, denoted by $K_* = K(X, X_*)$. Consequently, the joint distribution of the training and test data is:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma_n^2 I & K_* \\ K_*^\top & K_{**} \end{bmatrix}\right). \tag{5}$$

This combination allows us to incorporate the knowledge contained in the training data into the distribution of the function values of the test cases f_*, the values of interest.

Since the process outputs y are known, we can compute the distribution of f_* by conditioning it on y, resulting in the defining predictive distribution of GP models.

$$f_* | X, y, X_* \sim \mathcal{N}(\mathbb{E}[f_*],\ \mathbb{V}[f_*]) \tag{6}$$

with the mean and the variance given as

$$\mathbb{E}[f_*] = K_*^\top [K + \sigma_n^2 I]^{-1} y \tag{7}$$

$$\mathbb{V}[f_*] = K_{**} - K_*^\top [K + \sigma_n^2 I]^{-1} K_*. \tag{8}$$

The final distribution is defined only in terms of the three different kernel matrices and the training targets y.

3.2 Classification

The basic principle of GPC for multi-class problems is similar to GPR. Yet GPC suffers from the problem that the class labels $p(y|f)$ are not Gaussian distributed. Hence, the distribution of the function values given the training data

$$p(f|X, y) = \frac{p(y|f)p(f|X)}{p(y|X)} \tag{9}$$

is also not Gaussian. X and f still denote the training data and its function values, whereas y now represents the class labels of the training data. Yet to use GPs we have to have Gaussian distributed variables and must approximate $p(f|X, y)$, by the Laplace

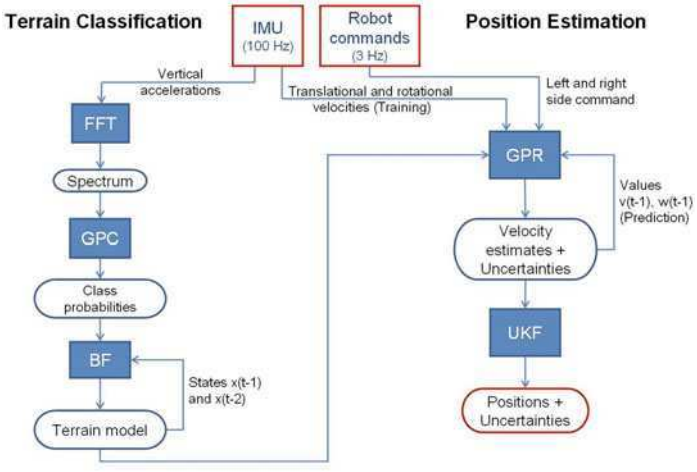


Fig. 2. Graphical model of our dynamic approach showing the processing of the data and the dependencies between the terrain classification and position estimation.

method for instance. Once the approximation is done, the further procedure is similar to GPR models, resulting in the predictive distribution of the GPC models $f_*|X, y, x_*$. Class probabilities are computed by drawing samples from this final distribution and applying the softmax function to squeeze the values into $[0, 1]$.

GPC have the same properties as GPR. The class probabilities can be seen as a confidence value and are perfectly suited to be combined with probabilistic filters.

4 Dynamic Approach

Simple controller consider only one type of terrain or use one averaged terrain model. In contrast, our approach considers the different effects on the kinematics of mobile robots caused by varying terrain conditions. Our algorithm splits naturally in two parts, the estimation of the robot's position and the classification of the terrain. Figure 2 shows a graphical model of our dynamic approach.

4.1 Position Estimation

The position estimation consists of two methods. We use GPs to do regression on the robot's velocities and an UKF to transfer the results into the position space.

The values used were provided by an IMU installed on the robot. A series of preceding experiments were done to determine the composition of data types which in combination with the projection into the position space allowed the best position estimation. The data compositions compared in these experiments varied in two aspects, first, the type of the data, and second, the amount of past information.

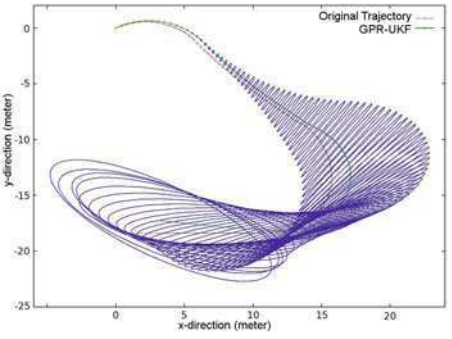


Fig. 3. Original (red) and predicted (green) trajectory of a 30 second drive. The UKF error ellipses are displayed in blue. Starting out with a small uncertainty the error ellipses (blue) increase over time, due to the uncertainty of previous positions and the velocity estimates.

The data combinations considered were: (1) the change in x-direction[1] and y-direction, (2) the change in x-direction, y-direction and in the orientation, (3) the x-velocity and y-velocity, (4) the x-velocity, y-velocity and the change in the orientation, (5) the translational and rotational velocity, and (6) the translational and rotational velocity and the orientation change. The number of past values was altered from 1 to 4 values to determine the minimal amount of previous information necessary for the system to be robust, while at the same time does not cloud the data records and does not complicate the learning task. Leading to a total number of 24 different data compositions. While differing in the motion model and complexity, all approaches can be easily transformed into positions. The approaches (5) and (6) are utilizing lines and circular paths to approximate the robot's trajectory, whereas (1)-(4) are just using lines. Additionally, the approaches (2), (4), and (6) are employing an independent estimation of the orientation, which may cover possible rotational slip. However, some approaches suffer from a poor transformation function, making them rather useless.

The results showed that a single past value of the translational and rotational velocity works best for position estimation. Thus, the data records of the GPR reads as follows:

$$[l_{t-1},\, l_t,\, r_{t-1},\, r_t,\, v_{t-1},\, \omega_{t-1}] \tag{10}$$

Since our Longcross robot has a differential drive, l_i and r_i denote the values of the speed commands to the robot for the left and right side of it's drive. v_i and ω_i represent the translational and rotational velocity, respectively. To learn a controller we need both the commands given to the robot and the implementation of these commands. The values of interest are the current velocities, v_t and ω_t.

We use GPR to solve this regression task which not only provides the estimates of translational and rotational velocities but also gives us uncertainty values. We transform the estimates into positions using an UKF. Additionally, the uncertainties are projected onto these positions and are propagated over time (Figure 3). Starting with a quite certain position the sizes of the error ellipses are increasing, due to the uncertainties of

[1] All directions are relative to the robot. The x-directions points always in the front direction of the robot, and the y-direction orthogonal laterally.

the velocity estimates and the increasing uncertainties of the previous positions they rely on.

Given only the first translational and rotational velocity, the regression does not rely on any further IMU values. It takes previous estimates as inputs for the next predictions. Assuming the first velocities to be zero (i.e. the robot is standing) makes our approach completely independent of the IMU device. However, IMU data is required for training.

For each type of terrain we trained two[2] GPR models with data recorded only on that terrain. This has several advantages: Firstly, the effects of the robot commands on a specific terrain can be learned more accurately, since they are not contaminated by effects on other terrain types. Secondly, in combination more effects can be learned. And thirdly, due to the allocation of the training data to several GPR models the size and complexity of each model is smaller compared to what a single model covering all terrain types would require. The classification described in the next paragraph enables us to select the appropriate terrain models for regression.

4.2 Terrain Classification

The terrain classification is implemented with a GPC model, and a 2nd order BF is used to model the transitions between terrains and to smooth the classification results.

As for the position estimation task preceding experiments were conducted to determine the best suited data records for the classification task. In contrast, here are mainly two possible data types, the vertical velocities and accelerations.

By taking a look at the raw IMU data it is evident that the different terrains are hardly distingushable and that it may need to be preprocessed. To be sure, we classified vertical velocity and acceleration records of varying sizes. Since the single values carry no information about the terrain, we took vectors of 25, 30, 35, and 40 values. The results showed the raw data to be improper for the classification task at hand.

The information about the terrain types is enclosed in the vibrations a mobile robot experiences while driving. We used the fast Fourier transform (FFT) to extract the frequency information. Given that the FFT works best with window sizes being a power of 2, we chose 16, 32, 64, and 128 as input lengths. Due to the symmetric structure of the FFT output, it is sufficient to use only the first half of the result. This reduces the complexity of the problem and allows us to consider larger window sizes as for the raw data. Using Fourier transformed vertical velocities and accelerations improved the classification results, as was expected. We found the vector of 128 vertical acceleration values to work best for our task. Hence, the input to the GPC model is given by

$$[fftaz_0, \ldots, fftaz_{63}], \tag{11}$$

where $fftaz_i$ denotes the ith value of the FFT result. However, we did not consider input lengths of 256 or longer, since it would take more than 2 seconds to collect the data for a single classification.

If the robot's speed is too slow the terrain characteristics are not longer present in the vibrations, and therefore a good classification is impossible. Thus, we omit data for which the robot's translational velocity is below some threshold τ.

[2] GPs are not able to handle multi-dimensional outputs. Hence, we need one GPR model for each velocity, translational and rotational.

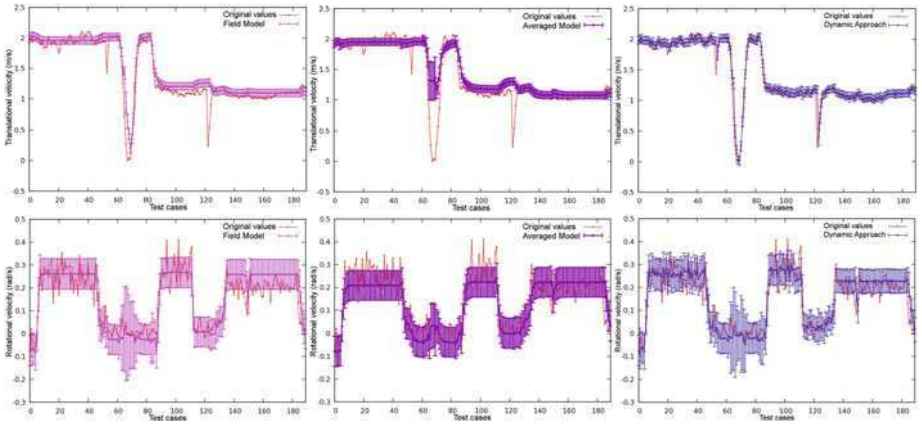


Fig. 4. Estimation of the translational velocities (upper row), and of the rotational velocities (lower row) with error bars. Even if the differences between the models are not significant, they result in rather distinct trajectories.

Due to the native charaters of the terrains, the driving conditions vary within a terrain type. This makes it hard to achieve a perfect classification. To deal with false predictions and to account for the properties of the classifier, we used a 2nd order BF to model the transitions between terrains. The quality values GPC models provide, come in terms of class probabilities which are well suited to be included into a probabilistic filter like the BF. The defining equation of our 2nd order BF is slightly modified to accommodate the classification and the dependency on two previous beliefs.

$$b(x_{t|t-1}) \propto \sum_{x_{t-1|t-1}} \sum_{x_{t-2|t-2}} p(x_{t|t-1}|x_{t-1|t-1}, x_{t-2|t-2}) \cdot b(x_{t-1|t-1}) \cdot b(x_{t-2|t-2}) \tag{12}$$

The notation $x_{t|t-1}$ denotes the value of the state x at time t given all observations up to time $t-1$. $b(\cdot)$ is the belief, and $p(x_{t|t-1}|x_{t-1|t-1}, x_{t-2|t-2})$ represents the transition probabilities, i.e. the dependency on the previous states. At this point the belief of the current state x_t depends only on the observations up to z_{t-1}. We include the current observation z_t via the result of the classifier c_t.

$$b(x_{t|t}) \propto \frac{p(x_{t|t-1}|c_t, z_t)p(c_t|z_t)}{p(x_{t|t-1}|z_t)} \cdot b(x_{t|t-1}), \tag{13}$$

where $p(c_t|z_t)$ is the classification result. The output of our BF is the belief of state x_t given all observations including z_t. It considers the classification and two previous states.

As already mentioned, the terrain classification is used to determine on which terrain the robot is currently driving and hence which terrain model should be used to estimate the next position. Since the robot commands required for the position estimation are given at a rate of approximately 3 Hz, whereas the terrain classification relies only on

Algorithm 1. Dynamic Approach.

Input: initialized and trained GPR and GPC models
while sensor data q_* available **do**
  get p_* and q_* from sensors
  if p_* available and $speed > \tau$ **then**
    s_* = FFT(p_*)
    t_p = GPC.classify(s_*)
    b_p = BF.process(t_p)
    $model$ = selectModel(b_p)
  end if
  $[v, \omega]$ = GPR.predict($model$,q_*)
  pos = UKF.process(v, ω)
  publish(pos)
end while

The algorithm requires trained GP models. If sensor data q_* is available, the algorithm predicts the velocity values using the current terrain model. If sufficient vibration data p_* is available for classification (omitting low speed data), a new classification of the terrain will be performed and the current model will be updated.

IMU data provided at $100\,Hz$, we implemented both parts to work independently of each other. However, it takes about a second to acquire enough data for classification, thus we reuse the last terrain model until a new classification is available. Below we will refer to the terrain models used by the GPR as *applied models* to distinguish them from the fewer classifications which were actually performed. Algorithm 1 summarizes our approach.

5 Experimental Results

The Longcross (Figure 1) is an experimental platform weighing about *340 kg* with a payload capacity of at least *150 kg*. The compartment consists of carbon-fibre and is environmentally shielded. Our version is equipped with a SICK LMS 200 2D laser scanner, a Velodyne Lidar HDL-64E S2 3D laser scanner, and an Oxford Technical Solutions Ltd RT3000 combined GPS receiver and inertial unit. The software runs on a dedicated notebook with a Intel Core 2 Extreme Quad processor and *8 GB* memory.

5.1 Naive Approaches

We evaluated our approach on a test drive covering three different terrains, starting on field, continuing with grass, and ending on asphalt. The test sequence consists of 190 positions (about a 1 minute) almost evenly partitioned on the three terrains. We compared our dynamic approach to a *field model*, an *asphalt model*, a *grass model* and an *averaged model*. The first three models were trained only on the specific terrain. The *averaged model* was trained on all three terrains equally. All models used a total of 1500 training cases. The dynamic approach utilizes the three terrain specific models.

We used the mean squared error (MSE) and the standardized MSE to evaluate the quality of the regression. The SMSE is the MSE divided by the variance of the test points, thus it does not depend on the overall scaling of the values. The MSE is not applicable to measure the quality of a trajectory because errors at the beginning are propagated through the entire trajectory influencing all proceeding positions. So we introduced the mean segment distance error (MSDE). First, we split the original trajectory and the prediction in segments of fixed size. Each pair of segments is normalized to the same starting position and orientation. The distance of the resulting end positions is computed subsequently. The mean of all these distances constitutes the MSDE. We found a segment size of 10 to be a reasonable size.

The estimation results of the translational (upper row) and the rotational velocities (lower row) of the *field model*, the *averaged model* and of our dynamic approach are shown in Figure 4. The *field model* is the best of the simple models. The *averaged model* would be the model of choice if one wants to consider different terrain conditions but does not want to employ a dynamic approach.

The estimation of the translational velocities by the *averaged model* shows little or no reaction to any of the outliers. Averaging the effects of commands over several terrains involves diminishing effects of certain command-terrain combinations, hence causing inaccurate estimates. The *field model* performs best of the simple approaches, mainly because it recognizes the slowdown around the 70th test case. However, the other two outliers are not identified. The estimates of the remaining test cases are similar to the *averaged model*, yet the *field model* is slightly superior. Our dynamic approach outperforms both previous models. In contrast, it recognizes all outliers, as the right terrain model is applied most of the times. Additionally, the estimates of each test point are pretty accurate.

The estimates of the rotational velocities give a similar picture. The *averaged model* is somewhat off at the beginning, underestimating the true values. The *field model* is much more accurate. The error bars seem to be larger than in the upper row but it is due to the scale of the data. However, our approach again provides the best estimation, reacting even to minor changes in the velocity values.

The velocities are translated into positions using an UKF. The resulting trajectories of all models tested are shown in Figure 5. The previous tendencies are reflected in the quality of the trajectories. The *averaged model's* trajectory is rather inaccurate, followed by the trajectory of the *field model* which is outperformed by our dynamic approach. At the beginning the predicted tracks are close, the differences start during the first turn and increase by the time. Even though the distinction between the velocity estimates are relatively small, the impact on the trajectories are formidable. Especially the rotation values are crucial to the trajectory quality.

We analyzed the quality of the GPR estimations of the velocities with the MSE and the SMSE. Due to the reasons mentioned above, the trajectories are evaluated by the MSDE. The results are presented in Table 1.

We also examined the classification performance. Figure 6 displays the results of the classification after applying the BF, and the terrain models used for the regression. The figures show that the classifier tends to categorize field as grass vibrations and grass records as asphalt. One problem is that the field and grass terrains are alike. At the low

Table 1. Prediction quality. The translational velocity unit is m/s, the rotational velocity unit is rad/s, and the MSDE unit is m.

Model		MSE	SMSE	$MSDE_{10}$
Field	v	0.02522	0.10588	0.42923
	ω	0.00331	0.23605	
Asphalt	v	0.07032	0.29517	0.56363
	ω	0.00533	0.38034	
Grass	v	0.05161	0.21662	0.51907
	ω	0.00365	0.26041	
Averaged	v	0.06203	0.26038	0.45478
	ω	0.00377	0.26929	
Dynamic	v	0.01181	0.04956	0.31407
	ω	0.00279	0.19917	

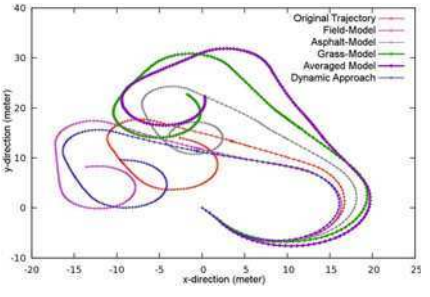


Fig. 5. Predicted trajectories. Starting the models' predictions are exact but differ increasingly along the time line, leading to unlike trajectories.

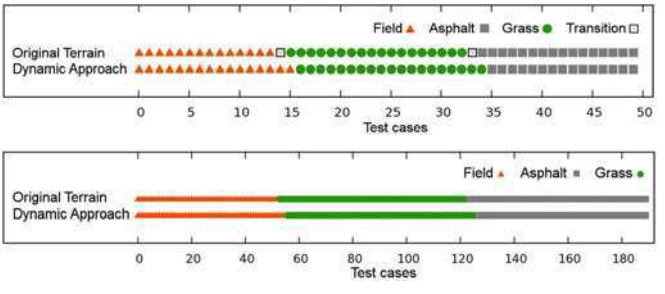


Fig. 6. Classification results (upper image) and applied models (lower image). The classification matches the sequence of field (red triangle), grass (green circles), and asphalt (grey squares) with high accuracy. Transitions are not classifiable because the records contain data of two terrains.

Table 2. Distribution of the classification results of the GP-Approach.

Classification		Original Terrain		
		Field	**Asphalt**	**Grass**
Guess	**Field**	*100.00%*	0.00%	5.56%
	Asphalt	0.00%	*93.75%*	0.00%
	Grass	0.00%	6.25%	*94.44%*

Table 3. Distribution of the applied models of the GP-Approach.

Applied Models		Original Terrain		
		Field	**Asphalt**	**Grass**
Guess	**Field**	*100.00%*	0.00%	5.71%
	Asphalt	0.00%	*95,52%*	0.00%
	Grass	0.00%	4.48%	*94,29%*

speed of approximately *1 m/s* the vibrations experienced on grass are close to the ones recorded on asphalt. Nonetheless, almost all false classifications are compensated by the BF.

Table 2 and 3 list the classification quality in terms of correct classifications and applied models, respectively. The GP-Approach determined the correct class with 96.06% accuracy, and hence, applied the right model in 96.84% of all cases. The results are broken down into the single terrain classes, giving more insight into the classifier's bias.

5.2 Dynamic Approaches

In the previous section we compared our dynamic approach to single terrain models, which by definition cannot be optimal when the robot drives on several different terrains. Therefore, we constructed another dynamic approach similar to the one presented. By doing so, we combined a support vector machine (SVM) to perform the regression of the velocities, and a k-nearest neighbor (k-NN) method for classification. We chose to combine a SVM and k-NN because of their low runtimes. The conducted experiments showed the overall runtime of the GP approach to be a serious issue.

We omitted the UKF from the system because the SVM does not provide any uncertainty values. These values are essential to the UKF in order to work properly. However, we kept the 2nd order BF, even though the k-NN method usually returns simple class labels rather than class probabilities. We counted the votes for each class and divided them by the total number of neighbors k to provide class probabilities anyway. We used the *LibSVM* library [4] with a radial basis function (RBF) as SVM kernel and found the *OpenCV* k-NN implementation with $k = 8$ neighbors to work best for vibrations.

To train the terrain models of the SVM we used the same training data as we did for the GP models. The training set of the classifiers are also unchanged.

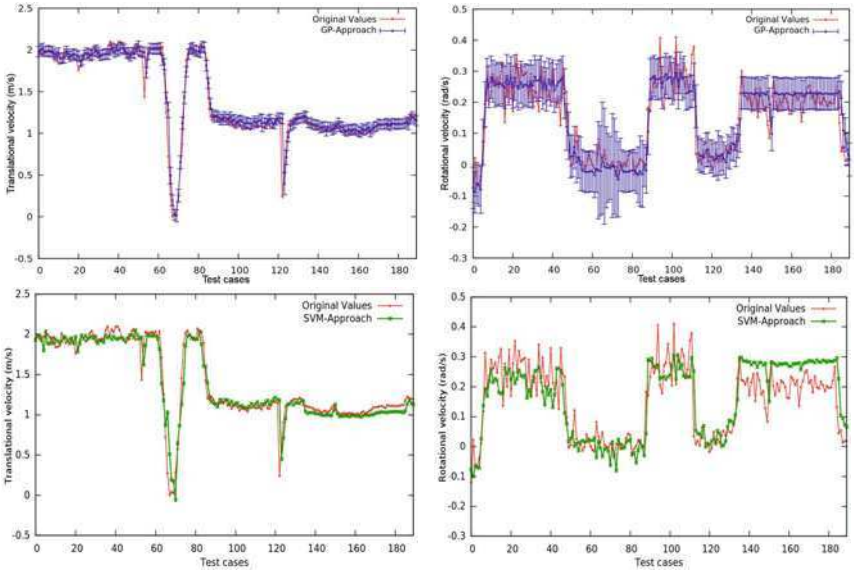


Fig. 7. Comparison of the two dynamic approaches. Estimation of the translational velocities (left column), and of the rotational velocities (right column) with error bars.

Since this approach is able utilize several terrain models for prediction, it should be more competitive than the single terrain model approaches. To distinguish the two dynamic approaches, we will refer to the GP based system as GP-Approach and to the SVM and k-NN system as SVM-Approach.

The top row of Figure 7 shows the estimations of the translational and rotational velocities by the GP-Approach from the previous section. The prediction results of the SVM-Approach are illustrated below. Both dynamic approaches fit the original translational velocities fairly accurately and recognize all outliers. The estimation differences between the two systems are somewhat more apparent when we concentrate on the rotational velocities. The SVM-Approach underestimates the first part and overestimates the velocities towards the end. Its predictions are more agitated in contrast to the averaging characteristic of the GP estimations.

The observations are reflected in the error values of Table 4. Even though the errors of the dynamic approaches are very small - they are the smallest of all tested models - the values of the SVM-Approach are more than $\frac{1}{3}$ higher. As we will see later, this is partly due to the lower k-NN classification quality.

The SVM's underestimation of the rotational velocity corresponds to the assumption of a less sharper turn. The green trajectory in Figure 8 shows exactly this behavior. Furthermore, the higher estimated rotational velocities at the end of the trajectory result in tighter turns. Regardless of the rather different predictive trajectories, the trajectory qualities in Table 4 are almost the same.

With respect to the classification task the GP classifier performs much better than the k-NN classifier. The sequence of classifications and applied models are compared in Figure 9. While the GP-Approach keeps the current terrain until enough evidence

Table 4. Prediction quality of the two dynamic approaches. The translational velocity unit is m/s, the rotational velocity unit is rad/s, and the MSDE unit is m.

Model		MSE	SMSE	$MSDE_{10}$
GP	v	0.01181	0.04956	0.31407
Approach	ω	0.00279	0.19917	
SVM	v	0.01923	0.08073	0.30336
Approach	ω	0.00464	0.33089	

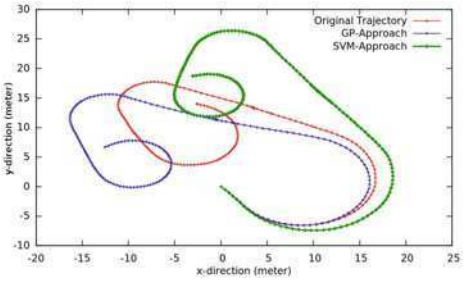


Fig. 8. Predicted trajectories of the two dynamic approaches.

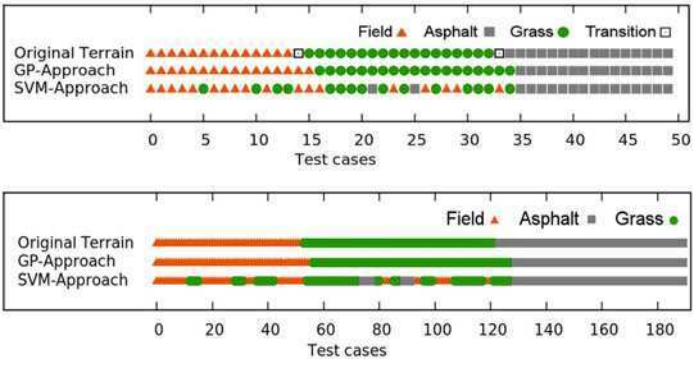


Fig. 9. Comparison of the two dynamic approaches. Classification results (upper image) and applied models (lower image). Transitions are not classifiable because the records contain data of two terrains.

is present that the terrain may have changed, the SVM-Approach tends to change the terrain class more quickly. This makes the system to apply the wrong terrain model more often.

Table 5 and 6 point out that the k-NN classifier struggles to distinguish grass and field records. The overall performance of our GP-Approach is 96.06% correct classification in contrast to the SVM-Approach's 70.00%. This is a gap in performance of more than 25%. Evaluating the quality of the applied models, we can see a slight improvement

Table 5. Distribution of the classification results of the dynamic approaches.

Classification		Original Terrain: Field		Asphalt		Grass	
		GP	SVM	GP	SVM	GP	SVM
Guess	Field	*100.00%*	*71.43%*	0.00%	0.00%	5.56%	33.33%
	Asphalt	0.00%	0.00%	*93.75%*	*93.75%*	0.00%	11.11%
	Grass	0.00%	28.57%	6.25%	6.25%	*94.44%*	*55.56%*

Table 6. Distribution of the applied models of the dynamic approaches.

Applied Model		Original Terrain: Field		Asphalt		Grass	
		GP	SVM	GP	SVM	GP	SVM
Guess	Field	*100.00%*	*71.70%*	0.00%	0.00%	4.29%	27.14%
	Asphalt	0.00%	0.00%	*95.52%*	*95.52%*	0.00%	10.00%
	Grass	0.00%	28.30%	4.48%	4.48%	*95.71%*	*62.86%*

by the SVM-Approach of about 9%, resulting in 78.86% correctly applied models. In comparision the GP-Approach applied the right model in 96.84% of the cases, still staying way ahead.

As stated earlier the classification quality is crucial to the performance of the overall system, since false classifications lead to the application of wrong terrain models, resulting in worse velocity estimations. However, the SVM-Approach, incorporating SVM regression and k-NN classification, is significantly faster than our GP-Approach.

6 Conclusions and Future Work

In this paper we introduced a dynamic approach to estimate positions of a mobile robot. Since the terrain conditions are crucial to the robot's implementation of velocity commands, we consider the terrain for the prediction of future positions. We used GPs for the regression of translational and rotational velocities. An UKF transfers the velocity estimates into positions and propagates the uncertainties of the positions over time. The vibration affecting the robot are classified by a GPC model in order to separate different terrains. A 2nd order BF is used to compensate for classification errors and to model the terrain transitions.

The prediction problem considered in this paper has several difficult properties: first, with respect to the classification task, the visual ground truth is not always accurate. Due to natural terrain types, which in themselves can be vastly different, vibrations induced by separate terrains may sometimes look alike or even show characteristics of another terrain. In a consequence, making it tricky to evaluate the classifier's performance. Second, finding a reasonable diversity of terrains is quite challenging, which aggravates the task of determining the transition probabilities from real data.

Nevertheless, our approach proved to be more effective than simple approaches utilizing a single model for all terrains. Furthermore, our GP-Approach is competitive with other dynamic approaches using different machine learning techniques. The dynamic systems, our GP-Approach and the slightly structurally altered SVM-Approach, are both better than all simple models, proving the basic idea of our approach to be well suited to solve such prediction problems.

Future work will be to reduce the run time. Due to the use of GPs the entire systems suffers of a long run time. Several sparse algorithms for GPs exist which would help to improve this issue. We are working on the implementation of such a sparse method at the moment.

Also GPRs and GPCs were compared to other well known methods for regression and classification, respectively. The run time of SVMs is smaller while performing equally. However, this would mean losing the uncertainty estimates, i.e. the quality indicators. This trade-off might be worth further investigation. Therefore and to determine the benefit of such a dynamic method in practice, we are currently integrating the SVM-based version in our local navigation software. Once the run time of the GP-based version allows the use in online applications, the integration in our local navigation framework will follow.

Moreover, expanding the current system by additional terrains as sand or gravel would be desirable. Extending the classification records by the translational velocity could improve classification results, since it allows to learn the effects of different speeds on the vibrations.

References

1. Agrawal, M., Konolige, K.: Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In: International Conference on Pattern Recognition (ICPR), vol. 3, pp. 1063–1068 (2006)
2. Brooks, C.A., Iagnemma, K., Dubowsky, S.: Vibration-based terrain analysis for mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3415–3420 (2005)
3. Burgard, W., Fox, D., Hennig, D., Schmidt, T.: Estimating the absolute position of a mobile robot using position probability grids. In: AAAI National Conference on Artificial Intelligence, vol. 2 (1996)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Software `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G.R.: Self-supervised monocular road detection in desert terrain. In: Robotics: Science and Systems (2006)
6. Ferris, B., Hähnel, D., Fox, D.: Gaussian processes for signal strength-based location estimation. In: Robotics: Science and Systems (2006)
7. Girard, A., Rasmussen, C.E., Quinonero-Candela, J., Murray-Smith, R.: Gaussian process priors with uncertain inputs - Application to multiple-step ahead time series forecasting. In: Advances in Neural Information Processing Systems (NIPS), pp. 529–536 (2003)
8. Iagnemma, K., Ward, C.C.: Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain. Autonomous Robots 26(1), 33–46 (2009)
9. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: IEEE International Conference on Computer Vision (ICCV), vol. 11, pp. 1–8 (2007)

10. Ko, J., Klein, D.J., Fox, D., Hähnel, D.: Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 742–747 (2007a)
11. Ko, J., Klein, D.J., Fox, D., Hähnel, D.: GP-UKF: Unscented Kalman filters with gaussian process prediction and observationmodels. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1901–1907 (2007b)
12. Li-Juan, L., Hong-Ye, S., Jian, C.: Generalized predictive control with online least squares support vector machines. Acta Automatica Sinica (AAS) 33, 1182–1188 (2007)
13. MacKay, D.J.C.: Introduction to gaussian processes. In: Bishop, C.M. (ed.) Neural Networks and Machine Learning. NATO ASI Series, pp. 133–166. Springer, Heidelberg (1998)
14. Rasmussen, C.E.: Combining laser range, color, and texture cues for autonomous road following. In: IEEE International Conference on Robotics and Automation, ICRA (2002)
15. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2005)
16. Seyr, M., Jakubek, S., Novak, G.: Neural network predictive trajectory tracking of an autonomous two-wheeled mobile robot. In: International Federation of Automatic Control (IFAC) World Congress (2005)
17. Thrun, S., Fox, D., Burgard, W., Dellaert, F.: Robust monte carlo localization for mobile robot. Artificial Intelligence 128, 99–141 (2000)
18. Urtasun, R., Darrell, T.: Discriminative gaussian process latent variable models for classification. In: International Conference on Machine Learning, ICML (2007)
19. Ward, C.C., Iagnemma, K.: Model-based wheel slip detection for outdoor mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2724–2729 (2007)
20. Weiss, C., Fröhlich, H., Zell, A.: Vibration-based terrain classification using support vector machines. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4429–4434 (2006)
21. Williams, C.K.I.: Gaussian processes. In: The Handbook of Brain Theory and Neural Networks, 2nd edn. MIT Press, Cambridge (2002)

Orientation of Doubly Nonholonomic Mobile Manipulator in the Path Tracking Problem

Alicja Mazur

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology
ul. Janiszewskiego 11/17, 50-372 Wrocław, Poland
alicja.mazur@pwr.wroc.pl

Abstract. In the paper we want to present a problem of path following for nonholonomic mobile manipulators. In our consideration we restrict ourself to doubly nonholonomic mobile manipulators. Nonholonomic constraints appear due to motion of the robotic system without slipping effect between wheels and the surface (for nonholonomic platform) or between special supporting wheels and ball in an onboard manipulator equipped with nonholonomic gears. As a point of departure for mathematical formulation of this problem we will use so-called Frenet parametrization. It establishes some tool to find some control laws solving the formulated problem. A role of orientation in the path tracking will depend on the user-choice and number of control inputs.

Keywords: Mobile manipulator, Nonholonomic constraints, Path following, Frenet parametrization.

1 Introduction

An object of our considerations will be a special robotic system, i.e. a nonholonomic mobile manipulator. The mobile manipulator consists of a mobile platform, equipped with non-deformable wheels, and an onboard rigid manipulator mounted on the platform.

The problem with a path definition for the end-effector of the mobile manipulator is that behavior of the subsystems is unpredictable, because the same path defined in global coordinates can be executed by separate subsystems or by both of them. Sometimes, it is important to move the platform and simultaneously unload a payload: such a task is defined relative to the base of the manipulator mounted on the platform; definition relative to the end-effector is ill-conditioned. In such situation the decomposition of the task into sub-tasks defined separately for both subsystems is more natural and convenient [5].

In the paper we will make an assumption that a desired task can be decomposed into two separate sub-tasks defined for each subsystem independently: the end-effector has to follow a desired geometric path $\Pi(s)$ described relative to the base of manipulator (i.e. relative to the platform) and the task of the platform is to follow a desired curve $P(s)$ lying on a plane. Such a formulation of the task makes possible successive unloading of payload transported by the mobile manipulator during the control process.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 171–184.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

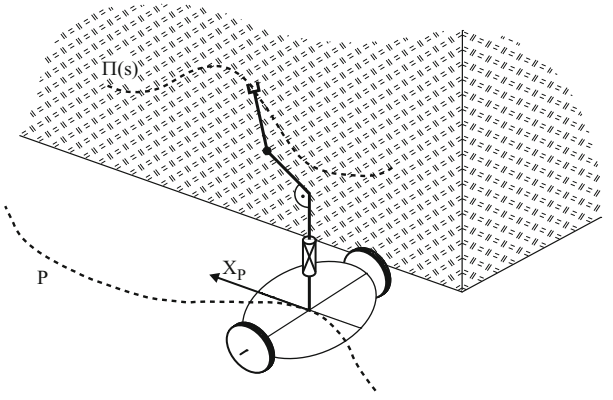


Fig. 1. Decomposition of a task for doubly nonholonomic mobile manipulator: $P(s)$ – desired path for a nonholonomic platform, $\Pi(s)$ – desired path for a nonholonomic manipulator.

If we take into account the type of components mobility of mobile manipulators, we obtain 4 possible configurations: type (h, h) – both the platform and the manipulator holonomic, type (h, nh) – a holonomic platform with a nonholonomic manipulator, type (nh, h) – a nonholonomic platform with a holonomic manipulator, and finally type (nh, nh) – both the platform and the manipulator nonholonomic. The notion *doubly nonholonomic* manipulator was introduced in [10] for the type (nh, nh).

2 Mathematical Models of Doubly Nonholonomic Mobile Manipulator

In this paper we restrict our considerations to mobile manipulators of (nh, nh) type, i.e. to doubly nonholonomic objects. We will discuss the constraints occurring in the motion of both subsystems. Nonholonomic constraints appearing in the motion of mechanical systems come from different sources. Very often they come from an assumption that a motion of the system can be treated as pure rolling of components, without slippage effect. We have taken such assumption in description of constrained motion of the considered mobile manipulator.

It is worth to mention that there are other sources of nonholonomic constraints, i.e. principle of conservation of momentum, non-integrable relationship between orientation of a rigid body expressed in local and global coordinates etc.

In the next step we want to calculate dynamics of the whole complex system i.e. mobile manipulator with nonholonomic restriction in the motion.

2.1 Nonholonomic Constraints for Wheeled Mobile Platform

Motion of the mobile platform can be described by generalized coordinates $q_m \in R^n$ and generalized velocities $\dot{q}_m \in R^n$. The wheeled mobile platform should move without slippage of its wheels. It is equivalent to an assumption that the momentary velocity at the contact point between each wheel and the motion plane is equal to zero. This

assumption implies the existence of l $(l < n)$ independent nonholonomic constraints expressed in Pfaffian form

$$A_1(q_m)\dot{q}_m = 0, \tag{1}$$

where $A_1(q_m)$ is a full rank matrix of $(l \times n)$ size. Since due to (1) the platform velocity is in a null space of $A(q_m)$, it is always possible to find a vector of special auxiliary velocities $\eta \in R^m$, $m = n - l$, such that

$$\dot{q}_m = G_1(q_m)\eta, \tag{2}$$

where G_1 is an $n \times m$ full rank matrix satisfying the relationship $A_1 G_1 = 0$. We will call the equation (2) the kinematics of the nonholonomic mobile platform.

2.2 Nonholonomic Constraints for Manipulator

A rigid manipulator can be a holonomic or a nonholonomic system – it depends on construction of its drives. In [7,1] the authors have presented a new nonholonomic mechanical gear, which could transmit velocities from the inputs to many passive joints, see Fig. 2. The prototype of 4-link manipulator with such gears has been developed in last 1990th years. The nonholonomic constraints in the gear appear by assumption on rolling contact without slippage between balls of gear and wheels in the robot joints.

The basic components of the gear presented in Fig. 2 are a ball and three wheels – an input wheel IW and two output wheels OW_1 and OW_2. The velocity constraints of the ball are only due to point contact with the wheels. The input wheel IW is mounted in the first joint, the output wheels are mounted in the next joint. The wheel IW rotates around the fixed axis α_I with an angular velocity u_2, which plays the role of a control input. The rotating input wheel IW makes the ball rotate. The wheel OW_1 rotates around an axis α_O, which makes with the axis of the input wheel a joint angle θ_1. The angular velocity $\dot{\theta}_1 = u_1$ is the second control input for the manipulator with nonholonomic gears.

Nonholonomic constraints (the kinematics) of n-pendulum have the form

$$\dot{\theta}_1 = u_1, \tag{3}$$

$$\dot{\theta}_i = a_i \sin\theta_{i-1} \prod_{j=1}^{i-2} \cos\theta_j u_2, \quad i \in \{2, \ldots, n\}, \tag{4}$$

with positive coefficients a_i depending on gear ratios or the same in matrix form

$$\dot{q}_r = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} = G_2(q_r)u. \tag{5}$$

It is worth to emphasize that only two inputs u_1 and u_2 are able to control many joints of manipulator equipped with gears designed by Nakamura, Chung and Sørdalen.

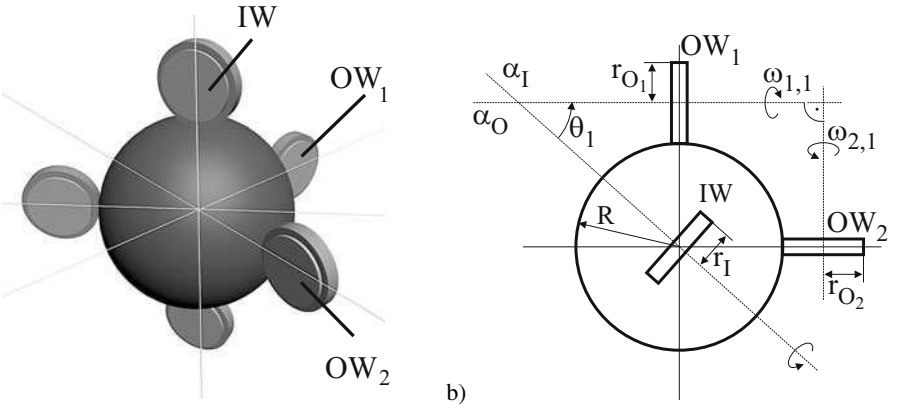


Fig. 2. Nonholonomic gear: a) schematic of the the gear, b) the gear seen from above.

2.3 Dynamics of Doubly Nonholonomic Mobile Manipulator

Because of the nonholonomy of constraints, to obtain the dynamic model of mobile manipulator the d'Alembert Principle has to be used

$$Q(q)\ddot{q} + C(q,\dot{q})\dot{q} + D(q) = A_{11}(q_m)\lambda_1 + A_{21}(q_r)\lambda_2 + B\tau, \tag{6}$$

where:
$Q(q)$ – inertia matrix of the mobile manipulator,
$C(q,\dot{q})$ – matrix of Coriolis and centrifugal forces,
$D(q)$ – vector of gravitational terms
A_{i1} – matrix of nonholonomic constraints for ith subsystem,
λ_i – vector of Lagrange multipliers for ith subsystem,
B – input matrix,
τ – vector of controls,
$q^T = (q_m^T, q_r^T)$ – generalized coordinates of the mobile manipulator.

Matrices $A_{11}(q_m)$ and $A_{21}(q_r)$ are Pfaff's matrices for the platform and the manipulator respectively, whereas B is so-called input matrix

$$A_{11} = \begin{bmatrix} A_1^T(q_m) \\ 0 \end{bmatrix}, \qquad A_{21} = \begin{bmatrix} 0 \\ A_2^T(q_r) \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}.$$

Submatrices B_1 and B_2 describe which coordinates of both subsystems are directly driven by actuators. Equations of constraints (2) and (5) can be rewritten as the kinematics in one block as follows

$$\dot{q} = \begin{pmatrix} \dot{q}_m \\ \dot{q}_r \end{pmatrix} = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \begin{pmatrix} \eta \\ u \end{pmatrix} = G\zeta, \qquad \zeta = \begin{pmatrix} \eta \\ u \end{pmatrix}, \tag{7}$$

where ζ is the vector of auxiliary velocities for both subsystems. After substituting the equation (7) into the dynamics (6) we get

$$Q^*\dot{\zeta} + C^*\zeta + D^* = B^*\tau \tag{8}$$

with elements defined in the following way

$$Q^* = G^T Q G, \qquad C^* = G^T(Q\dot{G} + CG), \qquad D^* = G^T D, \qquad B^* = G^T B.$$

The equation (8) describes the dynamics of doubly nonholonomic mobile manipulator expressed in the auxiliary coordinates.

3 Control Problem Statement

As it has been mentioned in the introduction, the desired task for the mobile manipulator can be decomposed into two independent parts: the end-effector of robotic arm has to follow a geometric path $\Pi(s)$ described relative to the platform *without stopping at the end* and the task of the platform is to move continuously along desired path $P(s)$ lying on the plane, see Fig. 1.

A goal of this chapter will be to address the following control problem for mobile manipulators:

> Design a control law τ such that a mobile manipulator with fully known dynamics follows the desired paths defined separately for each subsystem, and tracking errors converge asymptotically to zero.

Note, that a complete model of the nonholonomic system has a structure of two cascaded equations: kinematics (nonholonomic constraints) and dynamics. For this reason the backstepping-like procedure for the designing of control law should be used, [3]:

- kinematic controller $\zeta_r(t)$ – represents an embedded control input, which ensures the realizability of the path following for the nonholonomic constraints. Kinematic controller can be treated as a solution to the kinematics (7), if the dynamics were not present. Such the controller generates a 'velocity profile' which can be executed in practice. The convergence of the kinematic control algorithm must be proven.
- dynamic controller τ – as a consequence of cascaded structure of the model, the system's auxiliary velocities ζ cannot be commanded directly, as it is assumed in the designing of kinematic control, and instead they must be realized as the output of the dynamics (8) driven by τ. The dynamic input τ intends to regulate the real velocities ζ toward the reference control ζ_r and, therefore, attempts to provide control input necessary to achieve the desired task.

4 Description of Nonholonomic System Relative to a Given Path

For nonholonomic systems whose workspace is planar, it is possible to describe state variables relative to global inertial frame as well as to a given path [4], see Fig. 3.

The path P is characterized by a curvature $c(s)$, which is the inversion of the radius of the circle tangent to the path at a point characterized by the parameter s. Consider a moving point M and the associated Frenet frame defined on the curve P by the normal and tangent unit vectors x_n and $\frac{dr}{ds}$. The point M is the mass center of a mobile platform and M' is the orthogonal projection of the point M on the path P. The point M' exists and is uniquely defined if some conditions are satisfied, see [2] for details.

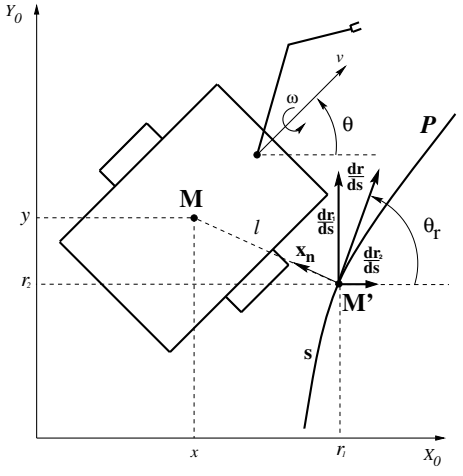


Fig. 3. Illustration of path following problem for the nonholonomic platform.

The coordinates of the point M relative to the Frenet frame are $(0, l)$ and relative to the basic frame X_0Y_0 are equal to (x, y), where l is the distance between M and M'. A curvilinear abscissa of M' is equal to s, where s is a distance along the path from some arbitrarily chosen point.

If we want to express the position of the point M not in coordinates (x, y) relative to inertial frame, but relative to the given path P, we should use some geometric relationships, [4],

$$\dot{l} = (-\sin\theta_r \quad \cos\theta_r) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \qquad \dot{s} = \frac{(\cos\theta_r \quad \sin\theta_r)}{1 - c(s)l} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \tag{9}$$

where $\dot{x}$ and $\dot{y}$ are defined by nonholonomic constraints for the system (wheeled mobile platform or nonholonomic manipulator) and θ_r is a desired orientation at the point M' on the path.

4.1 Path Following with Desired Orientation

Sometimes it is very important, especially for nonholonomic platforms, to keep the orientation of the subsystems under control. In such a situation we have only two control inputs and more than two coordinates to control. Therefore we make take an assumption that we will not to regulate the variable s. Such the assumption is good conditioned because for asymptotic path tracking (we have formulated in such a way our control problem)

Nonholonomic Mobile Platform. Posture of the mobile platform is defined not only by the position of the mass center, but by the orientation, too. For this reason, it is necessary to define the orientation tracking error equal to $\tilde{\theta} = \theta - \theta_r$. Moreover, at the point M' the desired orientation of the platform fulfills a condition, [8],

$$\dot{\theta}_r = c(s)\dot{s}. \tag{10}$$

Then the coordinates

$$\xi = (l, \tilde{\theta}, s)^T \tag{11}$$

i.e. the Frenet coordinates (l, s) and orientation tracking error $\tilde{\theta}$, constitute path following errors for nonholonomic mobile platform. It is worth to mention that Frenet parametrization is valid only locally, near the desired path.

If we want to solve the path following problem, it is necessary to express the kinematics of nonholonomic mobile platform in Frenet coordinates (11) instead of generalized coordinates q_m. After using equations (9) and (10), the nonholonomic constraints can be represented by the dynamic driftless system.

In our solution to the path following problem for nonholonomic wheeled mobile platform we have omitted the differential equation for $\dot{s}$, as we mentioned earlier. It means that it is enough to consider only l and $\tilde{\theta}$.

For example, the nonholonomic constraints for the platform of the class (2,0) are given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = G_1(q_m)\eta. \tag{12}$$

After substituting (12) into (9) and (10), we get Frenet variables for the mobile platform of $(2, 0)$ class in the form

$$\begin{aligned} \dot{l} &= v \sin\tilde{\theta}, \\ \dot{\tilde{\theta}} &= \dot{\theta} - \dot{\theta}_r = \omega - v\cos\tilde{\theta}\frac{c(s)}{1 - c(s)l} = w, \end{aligned} \tag{13}$$

where w is a new control input for the second equation.

For the system (13) we can use many kinematic control laws, e.g. Samson algorithm introduced in [9],

$$\eta_r = \begin{pmatrix} v_r \\ w_r \end{pmatrix} = \begin{pmatrix} const \\ v_r \frac{\sin\tilde{\theta}}{\tilde{\theta}} - k_3\tilde{\theta}, \end{pmatrix} \qquad k_2, k_3 > 0, \tag{14}$$

which is asymptotically stable. It can be shown using the following Lyapunov-like function

$$V(l, \tilde{\theta}) = k_2\frac{l^2}{2} + \frac{\tilde{\theta}^2}{2} \tag{15}$$

and Barbalat lemma.

Path following with the desired orientation is very important for mobile systems, especially for mobile manipulators. It comes e.g. from the fact that it would be impossible to unload a payload if the platform had wrong orientation, i.e. it would be in a right place but it would be back-oriented.

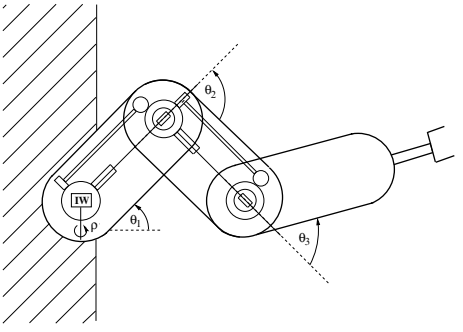


Fig. 4. Schematic of 3-link nonholonomic manipulator.

Nonholonomic Manipulator. For a nonholonomic manipulator it is possible to follow the desired path with prescribed orientation. However, this issue has a drawback: the nonholonomic manipulator has only two control inputs, therefore it is impossible to have all parameters $(l, s, \tilde{\theta})$ under control. In such a case many authors decide to regulate only two tracking errors $(l, \tilde{\theta})$ to zero. Such a case of the path following problem is so-called the asymptotic path following.

The Frenet parametrization can be evoked once again in the problem of path following for the planar manipulator, e.g. 3-pendulum with nonholonomic gears moving on the XZ surface, see Fig. 4. The role of the point M in Fig. 3 plays a point at the end of a gripper. The orientation of the end-effector θ_m is an rotation angle of the frame associated with the gripper around $-Y_b$ axis, which is located in the base of the manipulator. It means that the orientation of the end-effector in the planar nonholonomic n-pendulum is then equal to

$$\theta_m = \sum_{i=1}^{n} \theta_i.$$

In the considered planar nonholonomic manipulator lying in XZ-plane, relationships between velocity of the working point M expressed in Cartesian and curvilinear coordinates have the form

$$\dot{l}_m = (-\sin\theta_{rm} \quad \cos\theta_{rm}) \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix}, \quad \dot{s} = \frac{(\cos\theta_{rm} \quad \sin\theta_{rm})}{1 - c(s) l_m} \begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix}, \tag{16}$$

where l_m denotes distance between the point M and the path $\Pi(s)$, and θ_{rm} is the orientation of the Frenet frame in the point M'. Subscripts were introduced to distinguish Frenet variables for both subsystems of the (nh, nh) mobile manipulator.

Coordinates of the end-effector in the n-pendulum relative to its base are equal to

$$x = \sum_{i=1}^{n} l_i \cos(\sum_{j=1}^{i} \theta_j) \qquad z = \sum_{i=1}^{n} l_i \sin(\sum_{j=1}^{i} \theta_j). \tag{17}$$

Substituting time derivatives of variables (17), we obtain the following equations

$$\dot{l}_m = \sum_{i=1}^{n} \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_j) \cdot l_i \sum_{k=1}^{i} \dot{\theta}_k, \tag{18}$$

$$\dot{\tilde{\theta}}_m = \sum_{i=1}^{n} \dot{\theta}_i - \frac{c(s)}{1 - c(s) l_m} \cdot \sum_{i=1}^{n} \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_j) l_i \sum_{k=1}^{i} \dot{\theta}_k. \tag{19}$$

Using the kinematics of the nonholonomic manipulator given by (3)-(4), the equations (18) and (19) can be expressed in the matrix form as follows

$$\dot{\xi}_m = \begin{pmatrix} \dot{l}_m \\ \dot{\tilde{\theta}}_m \end{pmatrix} = H(q_r, \xi_m) \begin{pmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{pmatrix} = H(q_r, \xi_m) G_2(q_r) u = K_l(q_r, \xi_m) u. \tag{20}$$

Matrix $K_l(q_r, \xi_m)$ fulfills the regularity condition (i.e. it is invertible) if some configurations, which imply the matrix singularity, are excluded from a set of possibly achieved poses of the nonholonomic manipulator.

For nonholonomic 3-pendulum matrix $K_l(q_r, \xi_m)$ has the form

$$K_l(q_r, \xi_m) = \begin{bmatrix} K_{l11} & K_{l12} \\ K_{l21} & K_{l22} \end{bmatrix},$$

with elements defined below

$$K_{l11} = \sum_{i=1}^{3} l_i \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_i),$$

$$K_{l12} = a_2 s_1 \sum_{i=2}^{3} l_i \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_i) + a_3 s_2 c_1 l_3 \cos(\theta_{rm} - \sum_{j=1}^{3} \theta_i),$$

$$K_{l21} = 1 - \frac{c(s)}{1 - c(s) l_m} \sum_{i=1}^{3} l_i \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_i),$$

$$K_{l22} = a_2 s_1 [1 - \frac{c(s)}{1 - c(s) l_m} \sum_{i=2}^{3} l_i \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_i)]$$

$$+ a_3 s_2 c_1 [1 - \frac{c(s)}{1 - c(s) l_m} l_3 \sin(\theta_{rm} - \sum_{j=1}^{3} \theta_i)].$$

Note that Frenet transformation is valid only locally, i.e. $l_m(0) < r_{min}$, therefore nominators of all fractions are well defined. The nonholonomic planar 3-pendulum cannot achieve angles equal to $\theta_1, \theta_2 = 0, \pm\pi$. Moreover, singularities in K_l matrix appear for $\sin(\theta_{rm} - \theta_1) = \sin(\theta_{rm} - \theta_1 - \theta_2) = \sin(\theta_{rm} - \theta_1 - \theta_2 - \theta_3) = 0$.

For the regular matrix K_l, the following control signals guaranteeing a convergence of tracking errors to zero for pure kinematic constraints can be proposed

$$u_r = -K_l^{-1}(q_r, \xi_m) \Lambda \xi_m, \qquad \Lambda = \Lambda^T > 0. \tag{21}$$

It is easy to observe that the system (20) with closed-loop of the feedback signal (21) is asymptotically stable and has a form

$$\dot{\xi}_m + \Lambda \xi_m = 0.$$

4.2 Path Following without Desired Orientation

The manipulator with gears designed by Nakamura, Chung and Sørdalen has two control inputs. It means that only two parameters can be regulated during the path following process. If we mean that the orientation of the end-effector of such manipulator is not very important, it is possible to control other Frenet parameters, e.g. l_m – distance error from the desired path and curvilinear length s of the path.

In such a case the following differential equations

$$\dot{l}_m = \sum_{i=1}^{n} \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_j) l_i \sum_{k=1}^{i} \dot{\theta}_k, \quad \dot{s} = \frac{1}{1 - c(s) l_m} \cdot \sum_{i=1}^{n} \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_j) l_i \sum_{k=1}^{i} \dot{\theta}_k$$

have to be considered. Similarly to (18) and (19), using the kinematics of the nonholonomic manipulator (3)–(4), these equations can be expressed in the matrix form as follows

$$\begin{pmatrix} \dot{l}_m \\ \dot{s} \end{pmatrix} = H(q_r, \xi_m) \begin{pmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{pmatrix} = H(q_r, \xi_m) G_2(q_r) u = K_s(q_r, \xi_m) u. \tag{22}$$

Matrix $K_s(q_r, \xi_m)$ fulfills the regularity condition (i.e. it is invertible) if some configurations, which imply the matrix singularity, are excluded from a set of possibly achieved poses of the nonholonomic manipulator.

For nonholonomic 3-pendulum matrix $K_s(q_r, \xi_m)$ has the form

$$K_s(q_r, \xi_m) = \begin{bmatrix} K_{s11} & K_{s12} \\ K_{s21} & K_{s22} \end{bmatrix},$$

with elements defined below

$$K_{s11} = \sum_{i=1}^{3} l_i \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_i),$$

$$K_{s12} = a_2 s_1 \sum_{i=2}^{3} l_i \cos(\theta_{rm} - \sum_{j=1}^{i} \theta_i) + a_3 s_2 c_1 l_3 \cos(\theta_{rm} - \sum_{j=1}^{3} \theta_i),$$

$$K_{s21} = \frac{c(s)}{1 - c(s) l_m} \sum_{i=1}^{3} l_i \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_i),$$

$$K_{s22} = \frac{a_2 s_1 c(s)}{1 - c(s) l_m} \sum_{i=2}^{3} l_i \sin(\theta_{rm} - \sum_{j=1}^{i} \theta_i) + \frac{a_3 s_2 c_1 c(s)}{1 - c(s) l_m} l_3 \sin(\theta_{rm} - \sum_{j=1}^{3} \theta_i).$$

Singular configurations of nonholonomic 3-pendulum for the matrix K_s are equal to configurations $\sin(\theta_1) = \sin(\theta_1 - \theta_2) = \sin(\theta_1 - \theta_2 - \theta_3) = 0$.

If the following control law is applied

$$u_r = -K_s^{-1}(q_r, \xi_m) v, \tag{23}$$

where $v \in R^2$ is a new input to the system (22), then we obtain the decoupled and linearized control system of the form

$$\begin{pmatrix} \dot{l}_m \\ \dot{s} \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{24}$$

Now it is possible to control each variable separately. For instance, if we want to move along the desired path, not only to converge to this path, it seems to be a good idea to get

$$v_1 = -\Lambda l_m, \qquad v_2 = const, \qquad \Lambda > 0.$$

Such control algorithm guarantees the motion along the geometrical curve with constant velocity and, simultaneously, the convergence of the distance tracking error l_m to 0.

5 Dynamic Control Algorithm

Lets consider the model of a mobile manipulator (8) with nonholonomic constraints (7). We assume that reference kinematic controls $\zeta_r(t) = (\eta_r^T(t), u_r^T(t))$ solve the path following problem for both nonholonomic subsystems, where $\eta_r(t)$ was computed due to Samson algorithm (14) for the mobile platform and $u_r(t)$ for nonholonomic manipulator is given by (21) or (23).

Then we propose a dynamic control law

$$\tau = (B^*)^{-1} \left\{ Q^* \left(\dot{\zeta}_r - K e_\zeta \right) + C^* \zeta + D^* \right\} \tag{25}$$

with symbols defined as follows

$$e_\zeta = \begin{pmatrix} e_\eta \\ e_u \end{pmatrix} = \begin{pmatrix} \eta - \eta_r \\ u - u_r \end{pmatrix}, \qquad K = \begin{bmatrix} K_1 I_2 & 0 \\ 0 & K_2 I_2 \end{bmatrix},$$

and $K_1, K_2 > 0$, which preserves asymptotic convergence for full kinematic and dynamic coordinates of the (nh, nh) mobile manipulator to their desired values.

6 Simulations

As an object of a simulation study we have chosen a planar vertical 3-pendulum with nonholonomic gears mounted on a unicycle. Parameter of the dynamic control algorithm (25) was equal to $K = 200 \cdot I_4$. The desired path for the manipulator (a circle) was selected as

$$\Pi_1(s) = 0.25 \cos 4s + 1 \; [m],$$
$$\Pi_2(s) = -0.25 \sin 4s + 0.6 \; [m],$$

and the desired path for the mobile platform was a straight line

$$P(s): \qquad x(s) = \frac{\sqrt{2}}{2} s \; [m], \qquad y(s) = \frac{\sqrt{2}}{2} \; [m].$$

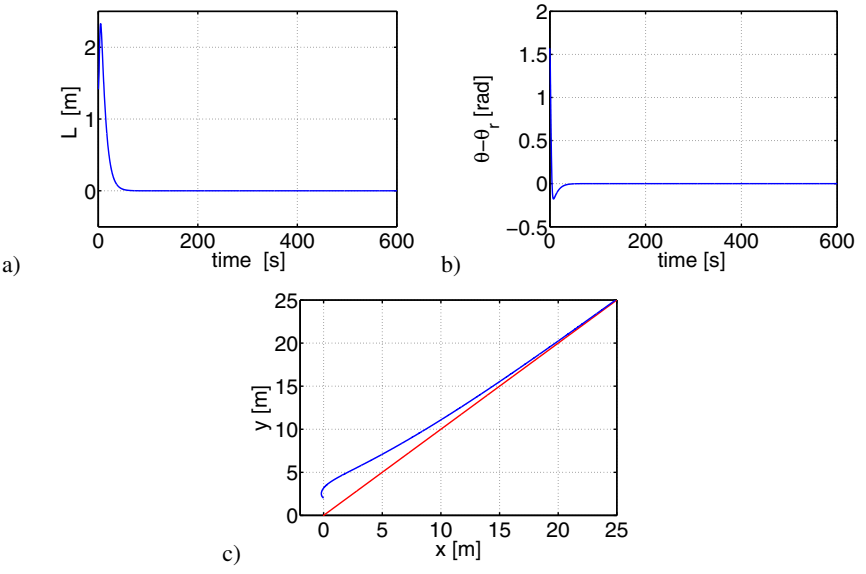


Fig. 5. Path tracking for the mobile platform: a) distance error L, b) orientation error $\tilde{\theta}$, c) XY plot.

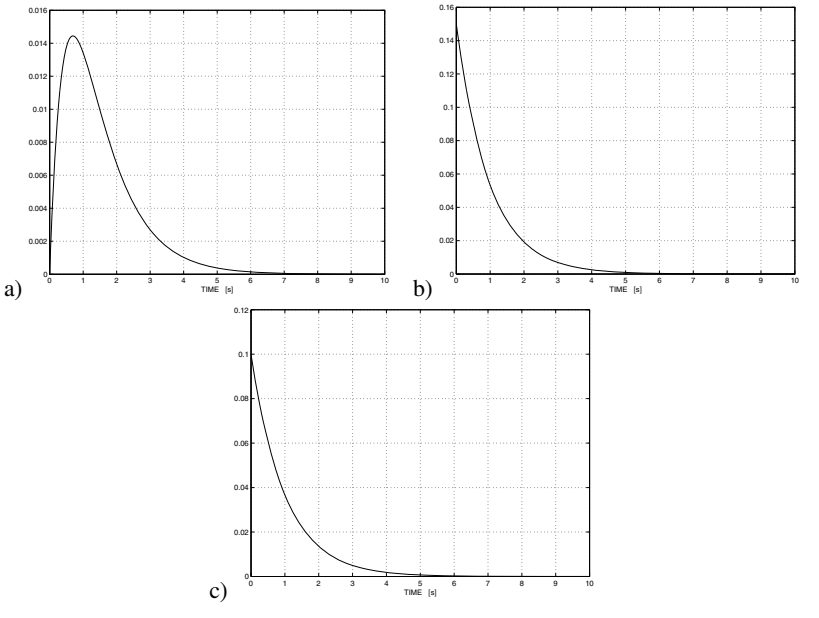


Fig. 6. Path tracking for the 3-pendulum: a) an error of x coordinate, b) an error of z coordinate, c) an error of orientation $\tilde{\theta}$.

The initial posture of the platform was selected as $(x, y, \theta)(0) = (0, 2, 3\pi/4)$ and configuration of the manipulator was equal to $(\theta_1, \theta_2, \theta_3)(0) = (0, 0.6732, -\pi/3)$.

Tracking of the desired path for the mobile platform has been presented in Fig. 5. Parameters of the Samson algorithm were selected as $v = 1, k_2 = 0.1, k_3 = 1$. Tracking of the desired path with prescribed orientation for the nonholonomic 3-pendulum has been presented in Fig. 6. Tracking of the desired path without preserved orientation for the same manipulator has been presented in Fig. 7.

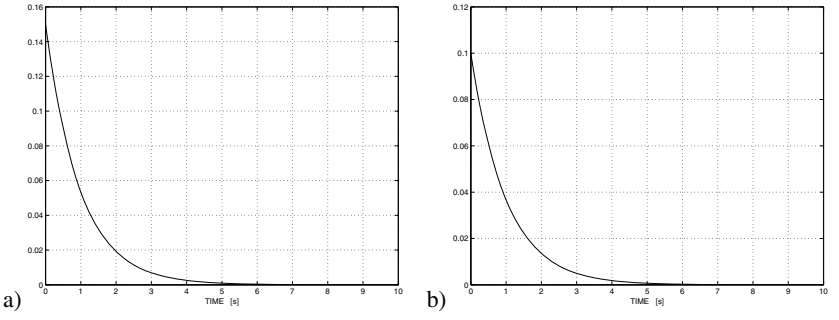


Fig. 7. Path tracking for the 3-pendulum: a) XZ plot, b) a distance error l_m.

7 Conclusions

In the paper the problem of defining the path for doubly nonholonomic mobile manipulators has been considered. We have proposed a new approach to the path as a geometric curve defined with the orientation or not. Path following problem with prescibed orientation is very important for mobile systems, especially for mobile manipulators – it results from the fact that it is impossible to realize a task, namely unload a payload if the platform has wrong orientation during the regulation process.

In turn, for nonholonomic manipulator the desired path need not be defined with orientation. In such a case a new approach to the path following problem has been presented in the paper. A new control algorithm, guaranteeing not only asymptotic convergence to the desired path but simultaneously the motion along the path with nonzero velocity, has been introduced. It is possible to define another kinematic control algorithms with specific properties using Frenet parametrization.

References

1. Chung, W.: Design of the nonholonomic manipulator. Springer Tracts in Advanced Robotics 13, 17–29 (2004)
2. Fradkov, A., Miroshnik, I., Nikiforov, V.: Nonlinear and Adaptive Control of Complex Systems. Kluwer Academic Publishers, Dordrecht (1999)
3. Krstić, M., Kanellakopoulos, I., Kokotović, P.: Nonlinear and Adaptive Control Design. J. Wiley, New York (1995)

4. Mazur, A.: Hybrid adaptive control laws solving a path following problem for nonholonomic mobile manipulators. International Journal of Control 77(15), 1297–1306 (2004)
5. Mazur, A., Szakiel, D.: On path following control of nonholonomic mobile manipulators. International Journal of Applied Mathematics and Computer Science 19(4), 561–574 (2009)
6. Michałek, M., Kozłowski, K.: Tracking controller with vector field orientation for 3-D nonholonomic manipulator. In: Proceedings of 4th Int. Workshop on Robot Motion and Control, pp. 181–191. Puszczykowo, Poland (2004)
7. Nakamura, Y., Chung, W., Sørdalen, O.J.: Design and control of the nonholonomic manipulator. IEEE Transactions on Robotics and Automation 17(1), 48–59 (2001)
8. Samson, C.: Control of chained systems - application to path following and time-varying point-stabilization of mobile robots. IEEE Transactions on Automatic Control 40(1), 147–158 (1995)
9. Samson, C., Ait-Abderrahim, K.: Feedback control of a nonholonomic wheeled cart in cartesian space. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Sacramento, pp. 1136–1141 (1991)
10. Tchoń, K., Jakubiak, J., Zadarnowska, K.: Doubly nonholonomic mobile manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, pp. 4590–4595 (2004)

Extracting the Frequency of Robotic Tasks with an Adaptive Fourier Series: Application to Yo-Yo

Tadej Petrič, Andrej Gams, and Leon Žlajpah

Department of Automation, Biocybernetics and Robotics, Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
{tadej.petric,andrej.gams,leon.zlajpah}@ijs.si
http://abr.ijs.si/

Abstract. On-line determination of the basic frequency of an unknown periodic signal with an arbitrary waveform is crucial in imitating and performing rhythmic tasks with robots. We present a novel method to determine the basic frequency of a periodic signal on-line. The method is based on adaptive frequency oscillators in a feedback loop. While so far several adaptive frequency oscillators in a loop had to be used and the basic frequency determined using logical algorithms that choose from the determined frequency components, our method extracts the basic frequency of the input signal without any additional logical operations. The proposed novel method uses a single oscillator combined with a whole Fourier series representation in a feedback loop. Such formulation allows extracting the frequency and the phase of an unknown periodic signal in real-time and without any additional signal processing or preprocessing. The method also determines the Fourier series coefficients and can be used for dynamic Fourier series implementation. The method can be used for the control of rhythmic robotic tasks, where successful performing of a task crucially depends on the extraction of the fundamental frequency. We demonstrate the properties and usefulness of the method in simulation and on a highly nonlinear and dynamic task of playing the robotic yo-yo.

Keywords: Adaptive frequency oscillators, Nonlinear dynamical systems, Fourier series, Imitation, Yo-yo.

1 Introduction

Rhythmic tasks, such as locomotion [7], drumming [3], handshaking [9] or playing with different toys, like the yo-yo[17] or the gyroscopic device called Powerball [6], require both accurate trajectory generation and frequency tuning. Controlling such tasks with robots is a difficult problem that requires complex sensory system and advanced knowledge of the task and the the actuated device [11].

The task itself can often be separated in determining the frequency of movement on one hand, and determining the waveform of the movement on the other. Determining the frequency of a task is a complex problem, even more so when the measured signal has several frequency components as is the case in imitating human movement. Extracting the fundamental frequency of a task can be achieved in different ways, e.g. with signal processing methods, such as FFT, or for example with the use of nonlinear oscillators

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 185–196.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

[10]. Furthermore, to perform the task with a robot, one has to constantly generate appropriate trajectories and modulate them on-line. A possible approach to trajectory generation and modulation is movement imitation [14], which can be preformed in several ways and with different trajectory encoding methods, e.g. splines [16] or dynamic movement primitives (DMP) [15].

One of the approaches that successfully combines frequency extraction and trajectory generation is the use of a two-layered imitation system based on nonlinear dynamical systems [5]. In their work the authors explain that the imitation system can be used for extracting the frequency of the input signal, learning its waveform, and imitating the waveform at the extracted or any other frequency. Similar, but with less favorable properties of robustness against perturbations, trajectory generation and modulation can be achieved by using only the first layer of such system for both frequency extraction and waveform learning [13]. Both systems, the one-layered and the two-layered are based on adaptive frequency oscillators in a feedback loop. Using adaptive frequency oscillators in a feedback loop can determine several frequency components of an input signal. Despite favorable properties of such systems there is a considerable drawback in determining the basic or fundamental frequency of the input signal.

When dealing with complex periodic and pseudo-periodic signals with several frequency components, such as measurements of human movement, the first layer of the imitation system, referred to as the canonical dynamical system, has to include a high number of oscillators in the feedback loop. Using this system for movement imitation requires determining the basic of fundamental frequency. This is accomplished by a logical algorithm that follows the feedback loop [5]. With a high number of oscillators, and when several of the oscillators tune to the same frequency, the decision making is not straight-forward anymore and becomes complex. As a high number of oscillators is practically necessary, determining the fundamental frequency from a list of determined frequencies cannot be avoided.

In this paper we propose a novel design of the first layer (the canonical dynamical system) for the two-layered imitation system. Contrary to the original approach [5] the proposed approach does not require a logic algorithm to determine the fundamental frequency of the input signal. We use a single adaptive phase oscillator in a feedback loop, which is followed by a complete Fourier series approximation. An algorithm to determine the Fourier coefficients is built into the series approximation. The combination of an adaptive phase oscillator and the adaptive Fourier series allows us to easily extract the fundamental frequency of the input signal and use it to control rhythmic robotic tasks. Our novel approach essentially implements a real-time adaptive Fourier series analysis. It is able to calculate the Fourier coefficients of an unknown periodic signal in real-time and is computationally inexpensive.

The usefulness of this system is presented on the case of playing the yo-yo. Controlling the yo-yo with a robot is a difficult task and has already been a subject of several studies [17,8,4] which mostly rely on complex, specially designed controllers based on the models of the device. The task of playing yo-yo is highly non-linear and requires on-line frequency adaptation. The proposed approach simplifies the synchronization between the necessary upward jerk of the robot and the movement of the yo-yo by determining the frequency of the up-down motion from a measurable periodic quantity,

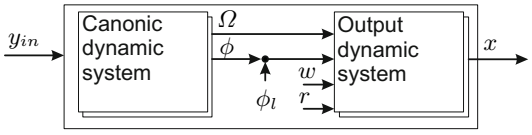


Fig. 1. Two-layered structure of the imitation system. The input y_{in} is a measured quantity and the output is the desired trajectory x of the robot. The input ϕ_l is the additional phase lag and r is the amplitude of the trajectory. The system can work in parallel for an arbitrary number of dimensions.

which is either the measured force of the impact, or the length of the unwound yo-yo string.

In the rest of the paper we give in section 2 a brief description of the original two-layered imitation system with an emphasis on the first layer - the canonical dynamical system. In section 3 we describe the novel approach using the Fourier series in the feedback loop. In section 4 we evaluate the proposed approach in both simulation and on a real-world experiment of playing the yo-yo. Conclusions and summary are given in section 5.

2 Two-Layered Imitation System

The two-layered imitation system was presented in detail in [5]. In their work the authors explained that the system can be used for extracting the frequency spectrum of the input signal, learning the waveform of one period, and imitating the desired waveform at an arbitrary frequency. The system structure is presented in Fig. 1. The first layer, i.e. the canonical dynamical system, is used for frequency extraction. It is based on a set of adaptive frequency oscillators in a feedback loop. The second layer is called the output dynamical system and is used for learning and repeating the desired waveform. The latter is based on dynamic movement primitives - DMPs, e.g. [15].

The first layer of the system has two major tasks. It has to extract the fundamental frequency Ω of the input signal and it has to exhibit stable limit cycle behavior in order to provide the phase signal Φ. The basis of the canonical dynamical system is a set of adaptive phase oscillator with applied learning rule as introduced in [1]. In order to accurately determine the frequency it is combined with a feedback structure [2], as is presented in Fig. 2. The feedback structure of M adaptive frequency oscillators is governed by

$$\dot{\phi}_i = \omega_i - K \cdot e \cdot \sin\phi_i, \tag{1}$$

$$\dot{\omega}_i = -K \cdot e \cdot \sin\phi_i, \tag{2}$$

$$e = y_{in} - \hat{y}, \tag{3}$$

$$\hat{y} = \sum_{i=1}^{M} \alpha_i \cos\phi_i, \tag{4}$$

$$\dot{\alpha}_i = \eta \cdot e \cdot \cos\phi_i, \tag{5}$$

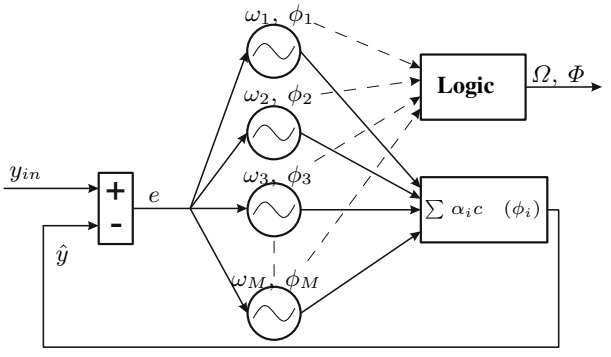


Fig. 2. Feedback structure of M nonlinear adaptive frequency oscillators. Note the logic algorithm that follows the feedback loop.

where K is the coupling strength, ϕ_i, $i = 1...M$ is the phase of separate oscillators, y_{in} is the input signal, M is the number of oscillators, α_i is the amplitude associated with the i-th oscillator, and η is the learning constant.

As shown in Fig. 2, each of the oscillators in the feedback structure receives the same input, i.e. the difference between the input signal and the weighted sum of separate frequency components. Such a feedback structure preforms a kind of Fourier analysis. The number of extracted frequencies depends on how many oscillators are used. As only the fundamental frequency is of interest, the feedback structure is followed by a logic algorithm. Determining the correct frequency and the phase is crucial, because they are the basis for the output dynamic system and the desired behavior of the actuated device.

When choosing the fundamental frequency one possible approach is to choose the first non-zero frequency as was presented in [5]. However, it has a drawback that when more than one oscillators converge to, or oscillate, around the same frequency, the logic algorithm switches between the oscillators, and consequently the phase will not be smooth, leading to oscillations in the output trajectory.

3 Canonical Dynamical System Based on Fourier Series

In this section a novel architecture for canonical dynamical system is presented. As the basis of the canonical dynamical system we use a single phase oscillator with applied learning rule [1]. This is combined with a feed-back structure based on an adaptive Fourier series in order to accurately determine the frequency. A feedback structure with an adaptive frequency oscillator combined with an adaptive Fourier series is shown in Fig. 3. The feedback structure of an adaptive frequency phase oscillator is governed by

$$\dot{\phi} = \Omega - K \cdot e \cdot \sin\phi, \tag{6}$$

$$\dot{\Omega} = -K \cdot e \cdot \sin\phi, \tag{7}$$

$$e = y_{in} - \hat{y}, \tag{8}$$

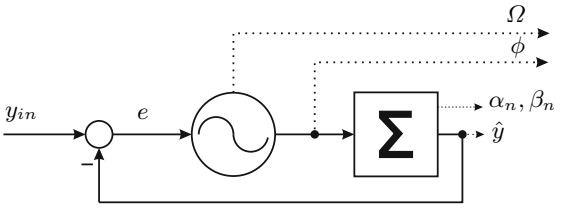


Fig. 3. Feedback structure of nonlinear adaptive frequency oscillator combined with dynamic Fourier series. Note that no logic algorithm is needed.

where K is the coupling strength, ϕ is the phase of the oscillator, e is the input into the oscillator and y_{in} is the input signal. If we compare Eqs. (1, 2) and Eqs. (6, 7), we can see that the frequency Ω and the phase ϕ are now clearly defined. The feedback loop $\hat{y}$ is now represented by the Fourier series

$$\hat{y} = \alpha_0 + \sum_{i=1}^{M} (\alpha_i \cos(i\phi) + \beta_i \sin(i\phi)), \tag{9}$$

and not by the sum of separate frequency components as in Eq. 4. M is the size of the Fourier series and α_0 is the amplitude associated with the first segment of the series. It is updated by

$$\dot{\alpha}_0 = \eta \cdot e, \tag{10}$$

where η is a learning constant. The amplitudes associated with the other terms of the Fourier series are determined by

$$\dot{\alpha}_i = \eta \cos(i\phi) \cdot e, \tag{11}$$

$$\dot{\beta}_i = \eta \sin(i\phi) \cdot e, \tag{12}$$

where $i = 1...M$. As shown in Fig. 3, the oscillator of the feedback structure receives the difference between the input signal and the Fourier series. Since a negative feedback loop is used, the difference approaches zero when the Fourier series representation approaches the input signal. Such a feedback structure preforms an adaptive Fourier analysis, where the phase difference between the harmonics can only be 0, $\pi/2$, π or $3\pi/2$. This is not the case in the original approach [12], where the phase difference can be completely arbitrary.

The proposed approach has the ability to adapt to the basic frequency of the input signal. The number of harmonic frequency components it can accurately extract depends on how many terms of the Fourier series are used. Since in this structure only one oscillator is used and the harmonics are encoded in the Fourier series, the basic frequency and phase are clearly defined. This is an important improvement, especially for the usefulness of the imitation system when performing rhythmic tasks.

The new architecture of the canonical dynamic system can be used as an imitation system by itself, as it is able to learn arbitrary periodic signals. After convergence, e reaches zero (with an accuracy that depends on the number of elements of the Fourier

series). Once e is zero, the periodic signal stays encoded in the Fourier series. The learning process is embedded and is done in real-time. There is no need for any external optimization process or learning algorithm.

Adding the output dynamical system enables us to synchronise the motion of the robot to a measurable periodic quantity of the task we would like to preform. The measured signal is now encoded into the Fourier series and the desired robot trajectory is encoded in the output dynamic system. Sice adaptation of the frequency and the learning of the desired trajectory can be done simultaneously, all of the system time-delays can be automatically included. Furthermore, when a predefined motion pattern for the trajectory is used, the system time-delays can be adjusted with a phase lag parameter ϕ_l. This enables us to either predefine the desired motion or to teach the robot how to preform the desired rhythmic task online.

The output dynamical system also ensures greater robustness against perturbations and smooth modulation. Specially greater robustness to perturbation is crucial when performing fast, dynamic tasks.

4 Evaluation

In the following section we evaluate the proposed imitation system. In Section 4.1 the numerical results from the original and the novel architectures are presented. In Section 4.2 a real-world experiment of playing the yo-yo with the use of the proposed imitation system is shown.

4.1 Simulation

In this numerical experiment the proposed architecture for the canonical dynamical system learns an arbitrary signal. The populating of the frequency spectrum is done without any signal processing, as the whole process of frequency extraction and adaptation of the waveform is completely embedded in the dynamics of the adaptive frequency oscillator combined with the adaptive Fourier series. Unless stated otherwise, we use the following parameters: $\mu = 2$, $K = 20$, $M = 10$.

Frequency adaptation results from time- and shape-varying signals are illustrated in Fig. 4. The input signal itself is of three parts: a periodic pulse signal, a sinusoid, and a sawtooth wave signal. Transitions between the signal parts are instant for both frequency and waveform. We can see that after the change of the input signal, the output frequency stabilises very quickly.

A single adaptive frequency oscillator in a feedback loop is enough, because the harmonics of the input signal are encoded with the Fourier series in the feedback loop. As can be seen from the bottom plots in Fig. 4, the input signal and the feedback signal are very well matched. The approximation error depends only on M.

A comparison with the original approach as proposed in [2] is given in Fig. 5. In their approach, if there are not enough oscillators to encode the input signal, the system will only learn the frequency components with more power. Thus, the output signal will only be an approximation. However, if there are more oscillators than the frequency components to learn, either some of them will not converge to any frequency or the same

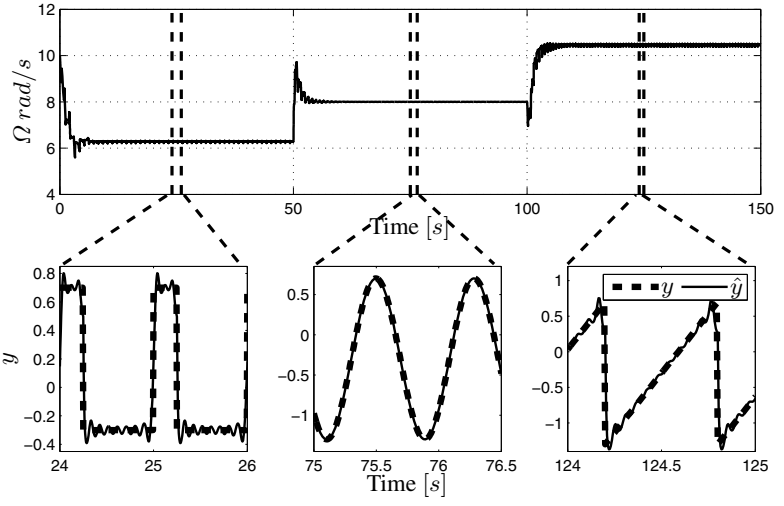


Fig. 4. Typical convergence of an adaptive frequency oscillator combined with an adaptive Fourier series, driven by a periodic signal with different waveforms and frequencies. Frequency adaptation is presented in the top plot and the comparison between the input signal y and the approximation $\hat{y}$ in the bottom plot.

frequency components will be coded by several oscillators, as shown in the top plot in Fig. 5, where a pool of ten oscillators was used. In this particular experiment, five of the oscillators converge to the basic frequency of the signal. Choosing the right oscillator from that pool is a very difficult task and requires a complex logic algorithm. On the other hand, using our new approach, where the feedback is encoded with a Fourier series, the oscillator converges to the basic frequency of the input signal. Therefore, the basic frequency and the phase are clearly defined. Furthermore, the approximation and the convergence of the feedback signal is quicker, as it is shown in the bottom plots in Fig. 5. Even after 350 s, the original architecture from [12] did not produce as good an approximation as it was after 20 s when using our new proposed canonical dynamical system.

4.2 Application to Robotic Yo-Yo

To illustrate the usefulness of the proposed approach we implemented it on a real robot playing a yo-yo. Playing yo-yo with a robot can be achieved in different ways, depending on what one can measure. It can be the length of the unwound string, which can be effectively measured by a vision system. As described in [17], using vision is also one of the ways humans do it, even though approaches using only the measured force were also described [8]. With our proposed system, playing yo-yo can be accomplished either with force feedback or with visual feedback. Furthermore, the proposed system is able to synchronize even if the input signal is changed from one measurable quantity to another during the experiment.

We preformed the experiment on a Mitsubishi PA-10 robot as presented in Fig. 6.

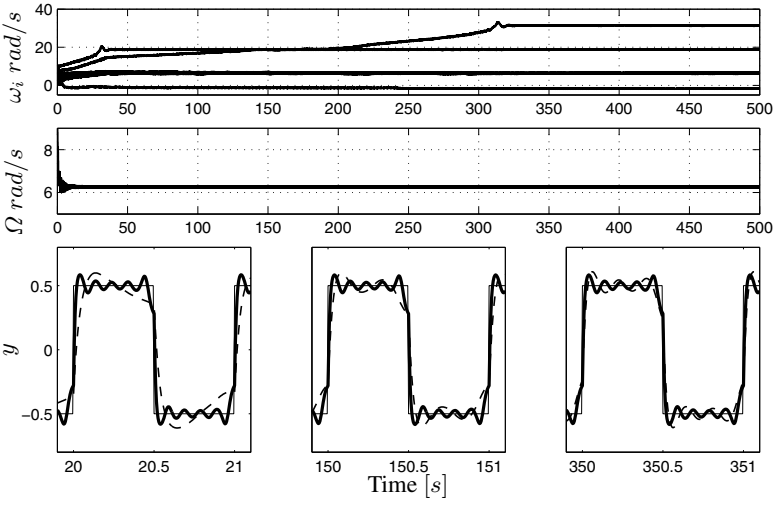


Fig. 5. Comparison between the pool of the adaptive oscillators and our proposed approach. The first plot shows evolution of frequency distribution using a pool of 10 oscillators. The middle plot shows the extracted frequency using an adaptive frequency oscillator combined with Fourier series. The comparison of approximated signals is presented in the bottom plot. The thin solid line presents the input signal, the solid line presents our new proposed approach and the dotted line presents the pool of adaptive oscillators.

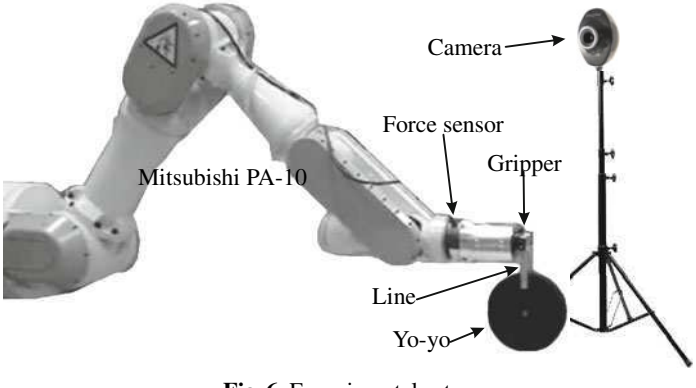


Fig. 6. Experimental setup.

A JR3 force sensor was attached to the end effector to measure the impact force of the yo-yo hitting the end of the string, and a USB camera was used to measure the length of the unwound string.

The two layered imitation system with the novel canonical dynamical system was implemented in Matlab/Simulik. The control scheme is presented in Fig. 7. As we can see, the imitation system, based on a nonlinear oscillator combined with dynamic Fourier series, provides the desired trajectory for the robot with a yo-yo attached at the top. The motion of the robot was constrained to up-down motion using inverse kinematics. The

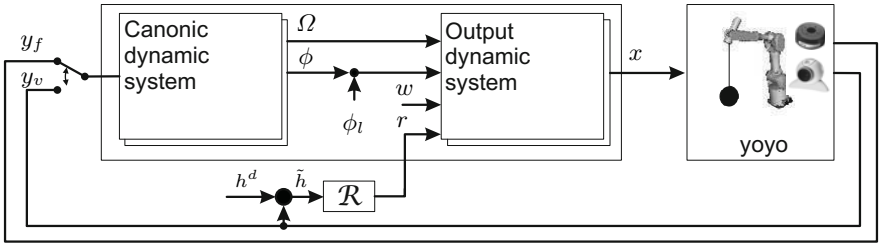


Fig. 7. Proposed two-layered structure of the control system for controlling the peak height of the yo-yo. The input is either the force y_f or the visual feedback y_v.

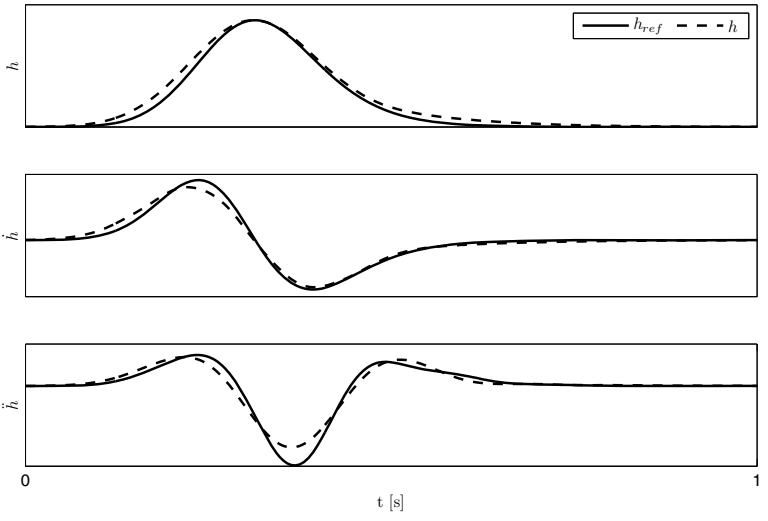


Fig. 8. Pre-defined hand motion pattern for playing yo-yo (solid) and encoded into the output dynamic system (dashed).

length of the string or the force from the top of the robot can be used as the input into the system. Since a measurable force difference appears only as a spike when the yo-yo hits the end of the string, we modify the signal in a way that it carries more energy. In our particular case we use the measured spike to create a short pulse.

To preform the task, we first determined the waveform of the required motion pattern. We chose the motion pattern described in [17], which satisfies the required criteria for playing the yo-yo. The robot motion pattern encoded into the output dynamic system (dashed line), and the pre-defined motion pattern (solid line) are presented in Fig.8.

The frequency of the task depends on the parameters of the yo-yo itself, and on how high the yo-yo rolls up along the string. The height can be influenced by the amplitude of the robot (or hand) motion, which can be easily modified using the amplitude parameter r of the motion, see Fig. 7. A PI controller was used to control the peak height of the yo-yo. The controller is given by

$$r(t) = k_p \tilde{h}(t) + k_i \int \tilde{h}(t)dt, \tag{13}$$

where $k_p = 2$, and $k_i = 0.4$ were determined empirically. Fig. 9 shows the results of frequency adaptation and yo-yo height during the experiment.

As we can see, the frequency of the imitated motion quickly adapted to the motion of the yo-yo and stable motion was achieved. At approximately 52 s the input into the imitation system was switched from force feedback to visual feedback. At that point some oscillation in the frequency and the approximation of the input signal can be observed because they have to adapt to the new waveform of the input signal. Furthermore, from the middle sequence in Fig. 10 we can see that the amplitude of the robot motion is higher after switching from the force feedback to the vision feedback. Despite the change, the imitation system still manages to extract the correct frequency and the robot motion returns to steady-state oscillations. Note that in the bottom sequence in Fig. 10 the amplitude of robot motion is smaller than immediately after the switch.

As far as we know, this is the first system which has the capability of playing the yo-yo by force feedback or by vision feedback, without changing the system parameters. Furthermore, switching from one to another measured quantity can even be done during the experiment. This shows that the proposed system is adaptable and robust.

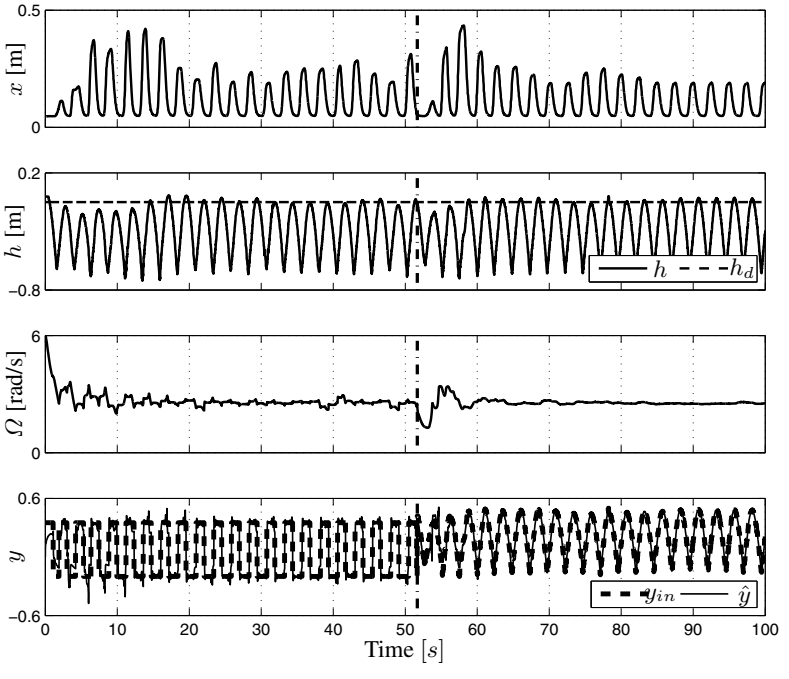


Fig. 9. Robot trajectory x in the top plot, height h of the yo-yo in the second plot, extracted frequency in the third plot and signal adaptation $\hat{y}$ in the bottom plot. At 52 s the input signal is switched from force feedback to visual feedback. Yo-yo parameters in this case are: axle radius $r_a = 0.01$ m and mass $m = 0.2564$ kg.

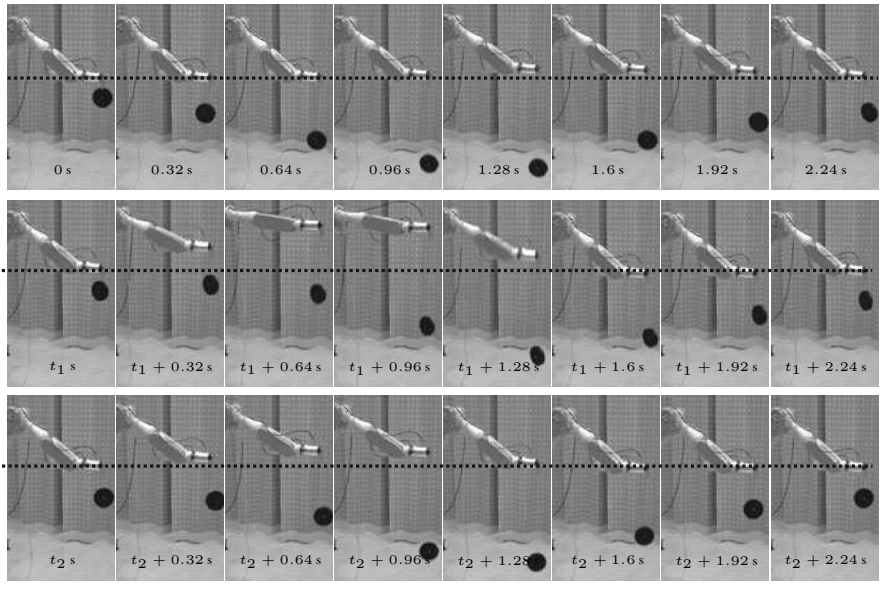


Fig. 10. Image sequence of a robotic yo-yo. The top sequence presents the descent of the yo-yo from the gripper. In the middle sequence the behaviour of the system after switching from force feedback to vision feedback is shown ($t_1 = 54$ s) and in the bottom sequence the behaviour in steady state is presented ($t_2 = 81$ s).

5 Conclusions

We presented a new architecture for the canonical dynamical system which is a part of a two layered imitation system, but can be used as an imitation system by itself. The dynamical system used to extract the frequency is composed of an adaptive phase oscillator combined with a Fourier series. This system essentially implements an adaptive Fourier series of the input signal. It can extract the frequency, the phase and the Fourier series coefficients of an unknown periodic signal. This is done in real-time and without any additional processing of the input signal in the sense of determining the frequency or setting processing parameters. Integrating this system into the imitation system based on dynamic motion primitives enables simple and computationally inexpensive control of rhythmic tasks with at least one measurable periodic quantity.

We presented the use of the proposed imitation system to preform a rhythmic robotic task that requires synchronization with the controlled device. For playing the yo-yo we have shown that the information on how high the yo-yo rolls up along the string or simply the force feedback is enough to achieve stable performance. The proposed approach enables playing yo-yo by measuring either the force or the yo-yo position. Furthermore we also showed that the system has the capability of changing the measured quantity in a single experiment without loosing the synchronization between the robot and the yo-yo.

References

1. Buchli, J., Ijspeert, A.J.: A simple, adaptive locomotion toy-system. In: Schaal, S., Ijspeert, A.J., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J.A. (eds.) From Animals to Animats 8. Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior (SAB 2004), pp. 153–162. MIT Press, Cambridge (2004)
2. Buchli, J., Righetti, L., Ijspeert, A.J.: Frequency Analysis with coupled nonlinear Oscillators. Physica D: Nonlinear Phenomena 237, 1705–1718 (2008)
3. Degallier, S., Righetti, L., Natale, L., Nori, F., Metta, G., Ijspeert, A.: A modular bio-inspired architecture for movement generation for the infant-like robot iCub. In: Proceedings of the 2nd IEEE RAS / EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob) (2008)
4. Gams, A., Petric, T., Zlajpah, L.: Controlling yo-yo and gyroscopic device with nonlinear dynamic systems. In: 18th International Workshop on Robotics in Alpe-Adria-Danube Region, Brasov, Romania, RAAD 2009, May 25-27, p. 6 (2009)
5. Gams, A., Ijspeert, A.J., Schaal, S., Lenarcic, J.: On-line learning and modulation of periodic movements with nonlinear dynamical systems. Auton. Robots 27(1), 3–23 (2009)
6. Gams, A., Žlajpah, L., Lenarčič, J.: Imitating human acceleration of a gyroscopic device. Robotica 25(4), 501–509 (2007)
7. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: A review. Neural Networks 21(4), 642–653 (2008)
8. Jin, H.-L., Ye, Q., Zacksenhouse, M.: Return maps, parameterization, and cycle-wise planning of yo-yo playing. Trans. Rob. 25(2), 438–445 (2009)
9. Kasuga, T., Hashimoto, M.: Human-robot handshaking using neural oscillators. In: ICRA, pp. 3802–3807 (2005)
10. Matsuoka, K., Ohyama, N., Watanabe, A., Ooshima, M.: Control of a giant swing robot using a neural oscillator. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 274–282. Springer, Heidelberg (2005)
11. Petrič, T., Gams, A., Žlajpah, L.: Modeling and control strategy for robotic powerball. In: RAAD, p. 7 (2009)
12. Righetti, L., Ijspeert, A.J.: Programmable Central Pattern Generators: an application to biped locomotion control. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation (2006)
13. Righetti, L., Buchli, J., Ijspeert, A.J.: Dynamic hebbian learning in adaptive frequency oscillators. Physica D 216(2), 269–281 (2006)
14. Schaal, S.: Is imitation learning the route to humanoid robots? Trends in cognitive sciences 6, 233–242 (1999)
15. Schaal, S., Mohajerian, P., Ijspeert, A.: Dynamics systems vs. optimal control — a unifying view, vol.(165), pp. 425–445 (2007)
16. Ude, A., Atkeson, C.G., Riley, M.: Planning of joint trajectories for humanoid robots using b-spline wavelets. In: ICRA, pp. 2223–2228 (2000)
17. Žlajpah, L.: Robotic yo-yo: modelling and control strategies. Robotica 24(2), 211–220 (2006)

Bilateral Teleoperation for Linear Force Sensorless 3D Robots

Stefan Lichiardopol, Nathan van de Wouw, and Henk Nijmeijer

Dept. of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
{s.lichiardopol,n.v.d.wouw,h.nijmeijer}@tue.nl

Abstract. It is well known that for bilateral teleoperation, force feedback information is needed. In this paper, we propose a control approach for bilateral teleoperation with uncertainties in the model of the slave robot and which does not use force sensors for haptic feedback. The controller design is based on a cyclic switching algorithm. In the first phase of the cyclic algorithm, we estimate the environmental force acting on the slave robot and in the second phase a tracking controller ensures that the position of the slave robot is tracking the position of the master robot. A stability analysis of the overall closed-loop system is presented and the approach is illustrated by means of an example.

Keywords: Bilateral teleoperation, Force sensor-less robotic setups, Haptic feedback.

1 Introduction

In this paper, we consider the problem of bilateral teleoperation in force-sensor-less robotic setups. It is well-known that haptic robotic devices and teleoperation systems exploit information regarding the external forces (see [1] and [2]), e.g. for haptic feedback. The slave robot interacts with the environment and its dynamics are dependent on external forces induced by this interaction. These forces can be contact forces (interaction forces between environmental objects and the robot) or exogenous forces induced by the environment.

In bilateral teleoperation, knowledge on the unknown environmental force applied on the slave robot is typically needed to achieve coordinated teleoperation. One option for obtaining such disturbance information is to equip the slave robot with force-sensors; for examples of such robotic devices, especially haptic devices, which use force sensors the reader is referred to [1], [3]. However, in many cases, the most important external forces for multi-link robots appear at the end-effector. Note that force sensing at the end effector of the robot is often not feasible since the external forces will typically interact with the load, which the slave robot is positioning, and not directly with the robot end-effector. Besides, in some cases, the position at which the external forces are applied is a priori unknown and may be on a robot link as opposed to on the end-effector. Moreover, the usage of force-sensors can be expensive and increase the production costs of the robot which can be undesirable especially in domestic applications.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 197–210.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

For these reasons, the usage of a disturbance estimation scheme for force-sensorless robots for the purpose of haptic feedback can be interesting. Disturbance observers (DOB) have been widely used in different motion control applications ([4], [5], [6]) for determining disturbance forces, such as friction forces. However, the performance enhancement of these DOB strategies may lead to smaller stability margins for the motion control ([7]); therefore, a robust design with respect to the environmental disturbances and model uncertainties is needed. Previous results on robustly stable DOB ([8], [9], [10], [11]) are based on linear robust control techniques. Some nonlinear DOB have been developed for the estimation of harmonic disturbance signals ([12], [13]).

Various strategies have also been considered for force-sensor-less control schemes estimating the external force. [14] proposes an adaptive disturbance observer scheme, and [15] and [16] propose an H^{∞} estimation algorithm. In [17], a control strategy called "force observer" is introduced. This design uses an observer-type algorithm for the estimation of the exogenous force. The drawback of this approach is that it assumes perfect knowledge of the model of the system.

In parallel with force estimation strategies, based on disturbance observers, another approach using sensor fusion has been developed to diminish the noise levels of the force sensors. In [18], force and acceleration sensors are used, while in [19], data from force sensors and position encoders are fused. Sensor fusion provides better qualitative results than obtained by employing more expensive force sensors.

Here, we present a control approach for bilateral teleoperation with an estimation strategy for external forces acting on the slave robot with a load with unknown mass. The difficulty in this solving this problem is that due to the uncertainties in the mass of the load it is possible for the previously referenced algorithms to estimate the exogenous force acting upon the slave robot. The method introduced in this chapter extends a result presented in [20], which considered the human-robotic co-manipulation problem. The proposed algorithm is robust for large uncertainties in the mass of the load.

The chapter is structured as follows. Section 2 presents the problem formulation and in Section 3 we describe the control strategy we propose. In Section 4, we apply the algorithm to a master-slave robotic setup. In the final section of the paper, the conclusions and some perspectives on future work are discussed.

2 Problem Statement

The problem that is tackled in this paper is that of bilateral teleoperation in force sensorless robotic setups. We assume that the slave robot is generally carrying a load (e.g. tool or product) and that the exogenous forces act on the slave or on the load. We consider the case in which no force sensor is present to measure the exogenous force directly. Moreover, we consider the realistic case in which the mass of the load is not known exactly which further challenges the estimation of the exogenous force. In order to solve this problem, we propose the design of a force estimator which is robust to the uncertainties in the mass of the load. In order to achieve teleoperation, the position of the slave robot must track the position of the master robot.

In Figure 1, the block diagram of the teleoperation setup is presented with the blocks $Master$ and $Slave$ representing the dynamics of the master and the slave robot, respectively, and the block C representing the control algorithm for bilateral teleoperation.

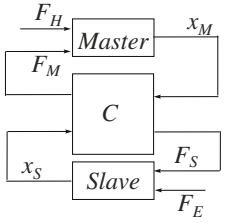


Fig. 1. Problem Setup.

The signals F_H and F_E represent the human and the environmental force, respectively; x_M and x_S are the positions of the master and the slave robots and F_M and F_S are the control signals for the master and slave robots, respectively. We adopt the assumption that the only measurements available are the position of the end-effectors and hence we aim to construct an output-feedback control strategy.

Consider the dynamics of the master and slaves robots dynamics:

$$\begin{cases} m_M \ddot{x}_M = F_H(t) + F_M(t) \\ m_S \ddot{x}_S = F_E(t) + F_S(t) \end{cases}, \tag{1}$$

where $x_M, x_S \in \mathbb{R}^3$ are the position of the master and the slave robots, respectively, $F_H, F_E \in \mathbb{R}^3$ are the human and the environmental force, respectively, m_M is the unknown inertia of the master robot (including some unknown inertia of the human hand) and m_S is the unknown inertia of the slave with the load (the masses are assumed to be bounded by $m_M, m_S \in [M_{min}, M_{ma}\ ]$).

The objective of this work is to design the controller C such that the following goals are met:

- the position of the slave robot is tracking the position of the master robot;
- an accurate estimate of the environmental force is transmitted to the master robot;
- the overall system is stable.

3 Control Design

Due to the uncertainties in the model of the slave robot we can not estimate the unknown environmental force and track the master robot position at the same time (the unknown inertia of the load and the fact that only position measurement data is available do not allow for simultaneous force estimation and position tracking). Therefore, we are proposing a switching controller based on a cyclic algorithm. During one cycle of duration T, we will have two phases as in Figure 2:

1. estimation of the environmental force;
2. position tracking.

During the first phase, which last for a period of T_0 ($T_0 < T$), the controller will behave as a force estimator. Here we are using the force observer introduced in [20] to estimate the external force which will be used for the purpose of haptic feedback

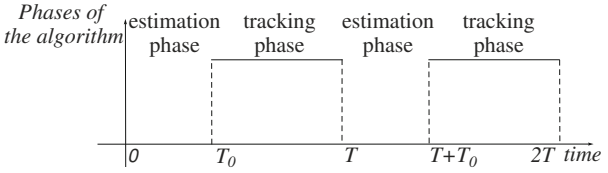


Fig. 2. Temporal division of the control strategy.

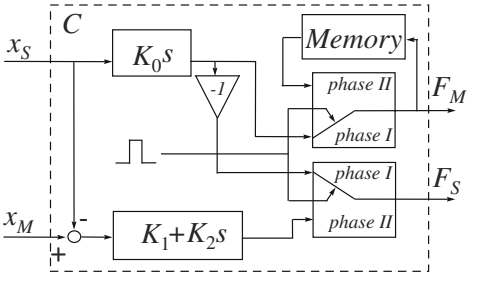


Fig. 3. Controller Design.

and during the second phase we are keeping the estimated force constant, $F_M(t) = \hat{F}_E(t) = K_0 \dot{x}_S$ and $F_S(t) = -K_0 \dot{x}_S$, for $kT \leq t < kT + T_0$, $k \in \mathbb{N}$, where parameter K_0 is a scalar parameter in the force estimation algorithm and is chosen such that the settling time of the estimation of the force is smaller than T_0. In the second phase, we are using a PD controller for the slave robot to track the position of the master robot, $F_M(t) = \hat{F}_E(kT + T_0)$ and $F_S(t) = K_1(x_M - x_S) + K_2(\dot{x}_M - \dot{x}_S)$, for $kT + T_0 \leq t < (k+1)T$, $k \in \mathbb{N}$, where scalars K_1 and K_2 represent parameters of the PD controller that ensures the tracking of the master robot position by the slave robot (these parameters are chosen such that the polynomial $m_S s^2 + K_2 s + K_1$ is Hurwitz $\forall m_S \in [M_{min}, M_{max}]$) and $\hat{F}_E(KT+T_0)$ is the estimation of the environmental force at the end of the first phase. In Figure 3, we present the block diagram representation of the controller where the controller blocks are represented by their transfer functions in the Laplace domain ($s \in \mathbb{C}$) and the block called $Memory$ saves the last estimate of the environmental force at the end of the first phase and provides the same constant output during the entire second phase.This means that (estimation phase) and (tracking phase). The switches in Figure 3 are set on positions corresponding to the first phase of the algorithm.

In the following section, we study the stability for the closed-loop system (including force estimation error dynamics and tracking error dynamics).

3.1 Description of Model Dynamics

For the purpose of stability analysis, we first formulate the model of the error dynamics. In order to obtain the error dynamics, the dynamics of the master and slaves robots are needed in both phases. During the first phase ($kT \leq t < kT + T_0$, $k \in \mathbb{N}$), the model dynamics are:

$$\begin{cases} m_M \ddot{x}_M = F_H(t) + K_0 \dot{x}_S \\ m_S \ddot{x}_S = F_E(t) - K_0 \dot{x}_S \end{cases}. \tag{2}$$

In the second phase of the algorithm ($kT + T_0 \leq t < (k+1)T$, $k \in \mathbb{N}$), the system behavior is described by:

$$\begin{cases} m_M \ddot{x}_M = F_H(t) + \hat{F}_E(kT + T_0) \\ m_S \ddot{x}_S = F_E(t) + K_1(x_M - x_S) + K_2(\dot{x}_M - \dot{x}_S) \end{cases}. \tag{3}$$

In the sequel, we assume that the exogenous forces acting on the system (human force $F_H(t)$ and environmental force $F_E(t)$) and their derivatives are bounded.

Given the fact that the same world frame is considered for both master and slave robots, the 3D system described by relations (2) and (3) can be seen as a set of three decoupled 1-DOF systems along each space dimension. Since the behavior for the three 1-dof system is similar, for simplicity, we are going to consider in the sequel only 1-DOF system, knowing that the extension for 3D systems is trivial.

3.2 Stability Analysis

Let us define the vector of tracking error coordinates $\epsilon = [e_x, \dot{e}_x]^T = [x_M(t) - x_S(t), \dot{x}_M(t) - \dot{x}_S(t)]^T$, which contains both the position and the velocity tracking errors, and the force estimation error $e_F = \hat{F}_E - F_E$. Then the force estimation error dynamics are described by:

$$\dot{e}_F = -\frac{K_0}{m_S} e_F - \dot{F}_E, \tag{4}$$

during the first step of the algorithm ($kT \leq t < kT + T_0$, $k \in \mathbb{N}$) and

$$\dot{e}_F = -\dot{F}_E, \tag{5}$$

during the second phase ($kT + T_0 \leq t < (k+1)T$, $k \in \mathbb{N}$).

The tracking error dynamics is represented by:

$$\dot{\epsilon} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \epsilon + \begin{pmatrix} 0 & 0 \\ \frac{1}{m_M} & \frac{1}{m_M} \end{pmatrix} \begin{pmatrix} F_H \\ F_E \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m_M} + \frac{1}{m_S} \end{pmatrix} e_F, \tag{6}$$

for $t \in [kT, kT + T_0)$, with $k \in \mathbb{N}$ and

$$\dot{\epsilon} = \begin{pmatrix} 0 & 1 \\ -\frac{K_1}{m_S} & -\frac{K_2}{m_S} \end{pmatrix} \epsilon + \begin{pmatrix} 0 & 0 \\ \frac{1}{m_M} & \frac{1}{m_M} - \frac{1}{m_S} \end{pmatrix} \begin{pmatrix} F_H \\ F_E \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m_M} \end{pmatrix} e_F, \tag{7}$$

for $t \in [kT + T_0, (k+1)T)$, with $k \in \mathbb{N}$.

The goal of this section is to prove that the composed estimation and tracking error dynamics (4)-(7) are input-to-state stable with respect to the inputs F_H, F_E and $\dot{F}_E$. For this we are going to use a result introduced in [21] that states that the series connection of two input-to-state stable systems is also an input-to-state stable system. In the sequel, this proof will be split into two parts:

- first, we prove that the force error dynamics (4), (5) are input-to-state stable with respect to the input $\dot{F}_E$;
- secondly, we prove that the tracking error dynamics (6), (7) are input-to-state stable with respect to the inputs F_H, F_E and e_F.

Input-to-State Stability of the Force Estimation Error Dynamics. The stability analysis of the force estimation error dynamics is done by studying an exact discretisation of the system (4), (5) at the sampling instances kT. Let us now construct such a discrete-time model.

The solution of system (4) at time $t = kT + T_0$, with $k \in \mathbb{N}$, is:

$$e_F(kT + T_0) = e^{-\frac{K_0}{m_S}T_0} e_F(kT) + + \int_0^{T_0} e^{-\frac{K_0}{m_S}(T_0-\tau)} \dot{F}_E(kT + \tau)d\tau. \tag{8}$$

The solution of system (5) at time $t = (k + 1)T$, with $k \in \mathbb{N}$, is:

$$e_F((k + 1)T) = e_F(kT + T_0) - \int_0^{T-T_0} \dot{F}_E(kT + T_0 + \tau)d\tau. \tag{9}$$

Define the sampled force estimation error by $e_k := e_F(kT)$, with $k \in \mathbb{N}$. Combining relations (8) and (9), one can obtain the discrete-time force estimation error dynamics:

$$e_{k+1} = e^{-\frac{K_0}{m_S}T_0} e_k + w_k, \tag{10}$$

with $w_k = \int_0^{T_0} e^{-\frac{K_0}{m_S}(T_0-\tau)} \dot{F}_E(kT+\tau)d\tau - \int_0^{T-T_0} \dot{F}_E(kT+T_0+\tau)d\tau$. The system (10) is input-to-state stable with respect to the input w_k because $\left|e^{-\frac{K_0}{m_S}T_0}\right| < 1$, since the parameters K_0, T_0 and the inertia m_S are positive. Note that w_k is bounded for any bounded $\dot{F}_E(t)$ and bounded T_0.

Now we exploit a result in [22] that says that if the discrete-time dynamics is ISS and the intersample behavior is uniformly globally bounded over T, then the corresponding continuous-time dynamics is ISS. The fact that the intersample behavior is uniformly globally bounded over T directly follows from (4), (5) with $\dot{F}_E$ bounded, since

$$e(t) = \begin{cases} e^{-\frac{K_0}{m_S}(t-kT)} e_F(kT) + \int_{kT}^{t} e^{-\frac{K_0}{m_S}(t-\tau)} \dot{F}_E(\tau)d\tau\,, & kT \le t < kT + T_0 \\ e_F(kT + T_0) - \int_{kT+T_0}^{t} \dot{F}_E(\tau)d\tau\,, & kT + T_0 \le t < (k + 1)T \end{cases}. \tag{11}$$

Input-to-State Stability of the Tracking Error Dynamics. Similarly to the study of the force estimation error dynamics, we evaluate the input-to-state stability property of the tracking error dynamics with respect to the inputs F_H, F_E and e_F.

The solution of system (6) at time $t = kT + T_0$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} \epsilon(kT + T_0) = e^{A_1 T_0}\epsilon(kT) + \int_0^{T_0} e^{A_1(T_0-\tau)} B_{11} u(kT + \tau)d\tau \\ + \int_0^{T_0} e^{A_1(T_0-\tau)} B_{12} e_F(kT + \tau)d\tau, \end{aligned} \tag{12}$$

where $A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B_{11} = \begin{pmatrix} 0 & 0 \\ \frac{1}{m_M} & \frac{1}{m_M} \end{pmatrix}$, $B_{12} = \begin{pmatrix} 0 \\ \frac{1}{m_M} + \frac{1}{m_S} \end{pmatrix}$ and $u(t) = \begin{pmatrix} F_H(t) \\ F_E(t) \end{pmatrix}$.

The solution of system (7) at time $t = (k + 1)T$, with $k \in \mathbb{N}$, is:

$$\begin{aligned} \epsilon((k + 1)T) = e^{A_2(T-T_0)}\epsilon(kT + T_0) + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{21} u(kT + T_0 + \tau)d\tau \\ + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)} B_{22} e_F(kT + T_0 + \tau)d\tau, \end{aligned} \tag{13}$$

where $A_2 = \begin{pmatrix} 0 & 1 \\ -\frac{K_1}{m_S} & -\frac{K_2}{m_S} \end{pmatrix}$, $B_{21} = \begin{pmatrix} 0 & 0 \\ \frac{1}{m_M} & \frac{1}{m_M} - \frac{1}{m_S} \end{pmatrix}$ and $B_{22} = \begin{pmatrix} 0 \\ \frac{1}{m_M} \end{pmatrix}$.
Let us define

$$\omega_k := e^{A_2(T-T_0)}\left(\int_0^{T_0} e^{A_1(T_0-\tau)}B_{11}u(kT+\tau)d\tau + \int_0^{T_0} e^{A_1(T_0-\tau)}B_{12}e_F(kT+\tau)d\tau\right) + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)}B_{21}u(kT+T_0+\tau)d\tau + \int_0^{T-T_0} e^{A_2(T-T_0-\tau)}B_{22}e_F(kT+T_0+\tau)d\tau \tag{14}$$

and $\epsilon_k := \epsilon(kT)$, with $k \in \mathbb{N}$. Combining relations (12) and (13), we obtain the following discrete-time error dynamics:

$$\epsilon_{k+1} := e^{A_2(T-T_0)}e^{A_1T_0}\epsilon_k + \omega_k, \tag{15}$$

where ω_k is bounded for all k, since T, T_0 are bounded, F_E, F_H are bounded by assumption and e_F is bounded due to the fact that the force estimation error dynamics is ISS with respect to $\dot{F}_E$.

Next, we study the input-to-state stability property of the system (15) with respect to the input ω_k. But before we carry out this step, we need to evaluate the matrix $Q = e^{A_2(T-T_0)}e^{A_1T_0}$. Namely, input-to-state stability of (15) implies, firstly, the global uniform asymptotic stability of $\epsilon = 0$ when the input ω_k is zero and the boundness of the error ϵ for bounded inputs ω_k.

For the evaluation of the matrix Q, two exponential matrices must be determined; as the matrix A_1T_0 depends only on known parameters, we can easily determine its exponential:

$$E_1 := e^{A_1T_0} = \begin{pmatrix} 1 & T_0 \\ 0 & 1 \end{pmatrix}. \tag{16}$$

In order to compute the exponential of matrix $P = A_2(T-T_0)$, we are using a procedure similar to the one introduced in [23], which employs the Cayley-Hamilton theorem, which says that if $p(\lambda) = \det(\lambda I_n - A)$, with I_n the $n \times n$ identity matrix, is the characteristic polynomial of a matrix $A \in \mathbb{R}^{n\times n}$ then $p(A) = 0$. This means that given the matrix P, for any $i \geq 2$, there exists a set of coefficients $a_i, b_i \in \mathbb{R}$ such that the i^{th} power of P can be expressed in terms of its first two powers:

$$P^i = a_iI_n + b_iP. \tag{17}$$

Let us now exploit (17) to determine the exponential of the matrix P:

$$e^P = \sum_{i=0}^{\infty}\frac{P^i}{i!} = \sum_{i=0}^{\infty}\frac{1}{i!}(a_iI_2 + b_iP), \tag{18}$$

or

$$e^P = \left(\sum_{i=0}^{\infty}\frac{a_i}{i!}\right)I_2 + \left(\sum_{i=0}^{\infty}\frac{b_i}{i!}\right)P. \tag{19}$$

Using the expression of A_2, we can decompose P as follows: $P = U + \frac{1}{m_S}L$, where

$$U = \begin{pmatrix} 0 & T-T_0 \\ 0 & 0 \end{pmatrix} \tag{20}$$

and

$$L = \begin{pmatrix} 0 & 0 \\ -K_1(T-T_0) & -K_2(T-T_0) \end{pmatrix}. \tag{21}$$

Consequently, the expression for the exponential matrix becomes:

$$e^P = \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right) I_2 + \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) U + \frac{1}{m_S}\left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) L. \tag{22}$$

Let us now define the following scalars:

$$\underline{\alpha} = \min_{m_S \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right), \tag{23}$$

$$\overline{\alpha} = \max_{m_S \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right), \tag{24}$$

$$\underline{\beta} = \min_{m_S \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right), \tag{25}$$

and

$$\overline{\beta} = \max_{m \in [M_{min}, M_{max}]} \left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right). \tag{26}$$

Given the fact that $m_S \in [M_{min}, M_{max}]$, we can define the scalars $\underline{\gamma} = \frac{1}{M_{max}}$ and $\overline{\gamma} = \frac{1}{M_{min}}$.

Then there always exist $\zeta_1, \zeta_2, \zeta_3 \in [0, 1]$ such that:

$$\left(\sum_{i=0}^{\infty} \frac{a_i}{i!}\right) = \zeta_1 \underline{\alpha} + (1 - \zeta_1)\overline{\alpha}, \tag{27}$$

$$\left(\sum_{i=0}^{\infty} \frac{b_i}{i!}\right) = \zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta} \tag{28}$$

and

$$\frac{1}{m_S} = \zeta_3 \underline{\gamma} + (1 - \zeta_3)\overline{\gamma}. \tag{29}$$

Introducing relations (27), (28) and (29) into expression (22) leads to:

$$\begin{aligned} e^P = &\left(\zeta_1 \underline{\alpha} + (1 - \zeta_1)\overline{\alpha}\right) I_2 + \left(\zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta}\right) U \\ &+ \left(\zeta_3 \underline{\gamma} + (1 - \zeta_3)\overline{\gamma}\right)\left(\zeta_2 \underline{\beta} + (1 - \zeta_2)\overline{\beta}\right) L, \end{aligned} \tag{30}$$

for some $\zeta_1, \zeta_2, \zeta_3 \in [0, 1]$.

Let us define the matrices $\Gamma_1 = 3\underline{\alpha}E_1$, $\Gamma_2 = 3\overline{\alpha}E_1$, $\Gamma_3 = 3\underline{\beta}UE_1$, $\Gamma_4 = 3\overline{\beta}UE_1$, $\Gamma_5 = 3\underline{\beta}\underline{\gamma}LE_1$, $\Gamma_6 = 3\underline{\beta}\overline{\gamma}LE_1$, $\Gamma_7 = 3\overline{\beta}\underline{\gamma}LE_1$ and $\Gamma_8 = 3\overline{\beta}\overline{\gamma}LE_1$, and the scalars $\varrho_1 = \frac{\zeta_1}{3}$, $\varrho_2 = \frac{1-\zeta_1}{3}$, $\varrho_3 = \frac{\zeta_2}{3}$, $\varrho_4 = \frac{1-\zeta_2}{3}$, $\varrho_5 = \frac{\zeta_2\zeta_3}{3}$, $\varrho_6 = \frac{\zeta_2(1-\zeta_3)}{3}$, $\varrho_7 = \frac{(1-\zeta_2)\zeta_3}{3}$, $\varrho_8 = \frac{(1-\zeta_2)(1-\zeta_3)}{3}$. This means that the expression of matrix Q is equivalent to:

$$Q = \sum_{i=1}^{8} \varrho_i \Gamma_i, \tag{31}$$

with $\sum_{i=1}^{8} \varrho_i = 1$.

Thus we have now found the generators for a convex (polytopic) matrix set that overapproximates the matrix set Q, with the uncertain parameter $m_S \in [M_{min}, M_{max}]$. Notice that $\sum_{i=0}^{\infty} \frac{a_i}{i!}$ and $\sum_{i=0}^{\infty} \frac{b_i}{i!}$ are infinite sums and will in practice be approximated by finite sums of length N. Next, we provide an explicit upper bound on the 2-norm of the approximation error induced by such truncation.

Theorem 1. *Consider an integer $N \in \mathbb{N}$ and a real positive scalar ϑ such that*

- $\mu = \sqrt{\frac{\lambda_{max}}{\vartheta}} < 1$, *where*

$$\lambda_{max} = \max_{m_S \in [M_{min}, M_{max}]} \left\{ eig(P^T P) \right\}, \tag{32}$$

- $\forall i \geq N$, $\sqrt{\vartheta^i} < i!$.

Then:

$$\left\| \sum_{i=N}^{\infty} \frac{P^i}{i!} \right\|_2 \leq \frac{\mu^N}{1-\mu}. \tag{33}$$

Proof.

$$\left\| \sum_{i=N}^{\infty} \frac{P^i}{i!} \right\|_2 \leq \sum_{i=N}^{\infty} \left\| \frac{P^i}{i!} \right\|_2 \leq \sum_{i=N}^{\infty} \frac{\|P^i\|_2}{i!} \leq \sum_{i=N}^{\infty} \frac{\sqrt{(\lambda_{max})^i}}{i!}, \tag{34}$$

where the inequality $\|A^i\|_2^2 \leq \|A\|_2^2 \times \ldots \times \|A\|_2^2 = \max(eig((A^T A))^i$ has been used. Using the property that $\forall a \in \mathbb{R}^+$, $\exists N \in \mathbb{N}$ such that $\forall i \geq N$, $\sqrt{a^i} < i!$, inequality (34) becomes:

$$\left\| \sum_{i=N}^{\infty} \frac{P^i}{i!} \right\|_2 \leq \sum_{i=N}^{\infty} \frac{\sqrt{(\lambda_{max})^i}}{i!} \leq \sum_{i=N}^{\infty} \mu^i. \tag{35}$$

Let us now employ the known result of convergence of geometric series which states that $\forall a \in [0, 1)$, $\lim_{n\to\infty} \sum_{i=0}^{n} a^i = \lim_{n\to\infty} \frac{1-a^{n+1}}{1-a} = \frac{1}{1-a}$.

$$\left\| \sum_{i=N}^{\infty} \frac{P^i}{i!} \right\|_2 \leq \frac{\mu^N}{1-\mu}. \tag{36}$$

□

Using Theorem 1, we can choose N such that the approximation error is small (even as low as the machine accuracy).

In the next theorem, we provide LMI-based stability conditions for the discrete-time tracking error dynamics (15) to be ISS with respect to the input ω_k.

Theorem 2. *Consider the discrete-time system (15). If there exists a matrix $\Omega = \Omega^T > 0$ and scalar $\varsigma \in (0, 1)$, such that the following linear matrix inequalities are satisfied:*

$$\Gamma_i{}^T \Omega \Gamma_i - \Omega \leq -\varsigma \Omega, i \in \{1, \ldots, 8\}, \tag{37}$$

where Γ_i are defined above, then the system (15) is ISS with respect to the input ω_k.

Proof: Using the Schur complement, relations (37) can be written as:

$$\begin{pmatrix} -\Omega & {\Gamma_i}^T\Omega \\ \Omega\Gamma_i & -\Omega \end{pmatrix} \leq -\varsigma\Omega, i \in \{1,\ldots,8\}. \tag{38}$$

Multiplying every inequality (38) with ϱ_i and summing them up, we obtain:

$$\begin{pmatrix} -\Omega\sum_{i=1}^{8}\varrho_i & \sum_{i=1}^{8}\varrho_i{\Gamma_i}^T\Omega \\ \Omega\sum_{i=1}^{8}\varrho_i\Gamma_i & -\Omega\sum_{i=1}^{8}\varrho_i \end{pmatrix} \leq -\varsigma\Omega\sum_{i=1}^{8}\varrho_i, \tag{39}$$

which according to equation (31) is:

$$\begin{pmatrix} -\Omega & Q^T\Omega \\ \Omega Q & -\Omega \end{pmatrix} \leq -\varsigma\Omega, \tag{40}$$

or

$$Q^T\Omega Q - \Omega \leq -\varsigma\Omega. \tag{41}$$

Let the candidate ISS-Lyapunov function be $V_k = (\epsilon_k)^T\Omega\epsilon_k$. We compute $\triangle V_k = V_{k+1} - V_k$:

$$\triangle V_k = (\epsilon_k)^T Q^T\Omega Q\epsilon_k - (\epsilon_k)^T\Omega\epsilon_k + 2(\epsilon_k)^T Q^T\Omega\omega_k + (\omega_k)^T\Omega\omega_k, \tag{42}$$

which according to (41) gives:

$$\triangle V_k \leq -\varsigma(\epsilon_k)^T\Omega\epsilon_k + 2(\epsilon_k)^T Q^T\Omega\omega_k + (\omega_k)^T\Omega\omega_k \tag{43}$$

After some straightforward computations, we can show that:

$$\|\epsilon\|_2 \geq \frac{2}{\varsigma}\sqrt{\frac{\lambda_{max}}{\lambda_{min}}}\sup_{k\in\mathbb{N}}(\omega_k) \Rightarrow \triangle V \leq -\frac{\varsigma}{2}\|\epsilon\|_2^2, \tag{44}$$

where λ_{max} and λ_{min} are the largest and the smallest eigenvalues of matrix Ω, respectively.

(44) implies that system (15) is input-to-state stable with respect to the input ω_k; see [24] for sufficient conditions for the ISS of discrete-time systems.

Remark 1. *For the sake of simplicity, Theorem 2 is based on a common quadratic ISS Lyapunov function $V = \epsilon^T\Omega\epsilon$. Alternatively, a parameter-dependent Lyapunov function approach could straight-forwardly be exploited to formulate less conservative stability conditions.*

The LMIs (37) are defined for the non-truncated Γ_i, but in practice we evaluate the vertex matrices using a truncation after N iterations as provided by Theorem 1. The errors can be ensured to be as low as the machine accuracy by choosing N large enough, just as the errors obtained from the numerical solver of the LMIs.

The last part of the study of the ISS property of the tracking error dynamics is to analyze the intersample behavior. Using Theorem 2, we can prove that the error dynamics are ISS on the sampling instances $t = kT$, with $k \in \mathbb{N}$. Given the choice of the parameters K_1 and K_2 such that the system (7) is Hurwitz for all $m_S \in [M_{min}, M_{max}]$, we

can conclude that during the second phase ($t \in [kT + T_0, (k+1)T)$) the evolution of the tracking error is bounded. In order to prove the stability of the overall continuous-time system, we need to show that the position error dynamics are also bounded for $t \in (kT, kT + T_0)$.

The solution of system (6), for $t \in (kT, kT + T_0)$, is given by:

$$\begin{aligned}\epsilon(kT+t) = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \epsilon(kT) + \int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{11} u(kT+\tau) d\tau \\ + \int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{12} e_F(kT+\tau) d\tau,\end{aligned} \tag{45}$$

with $t \in [0, T_0]$. As the human force and the environmental force are bounded, we can define $F = \max_{t \in (kT, kT+T_0)} (|F_H(t)| + |F_E(t)|)$. In the previous section, we have proven that the force estimation error dynamics are ISS and consequently $e_F(t)$ is bounded; therefore, there exists $E_F = \max_{t \in (kT, kT+T_0)} (|e_F(t)|)$, with E_F bounded. Considering the three terms in the right-hand side of (45), we can conclude that the first one is bounded due to the boundness of the discrete-time error dynamics and the fact that $t \in [0, T_0]$, the second term satisfies:

$$\left| \int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{11} u(kT+\tau) d\tau \right| \leq \left| \frac{F}{m_M} \begin{pmatrix} \frac{T_0^2}{2} \\ T_0 \end{pmatrix} \right|, \tag{46}$$

and the third term satisfies:

$$\left| \int_0^t \begin{pmatrix} 1 & t-\tau \\ 0 & 1 \end{pmatrix} B_{12} e_F(kT+\tau) d\tau \right| \leq \left| \left(\frac{1}{m_M} + \frac{1}{m_S} \right) E_F \begin{pmatrix} \frac{T_0^2}{2} \\ T_0 \end{pmatrix} \right|. \tag{47}$$

Therefore, we can conclude that the inter-sample evolution of tracking error is also bounded for $t \in (kT, kT + T_0)$. Once more, we can employ the result from [22] to prove that the continuous-time tracking error dynamics (6), (7) is ISS with respect to the inputs F_H, F_E and e_F because the discrete-time tracking error dynamics is ISS and the intersample behavior is uniformly globally bounded over T.

Since the force estimation error dynamics (4), (5) is ISS with respect to the input $\dot{F}_E$ and the tracking error dynamics (6), (7) is ISS with respect to the inputs F_H, F_E and e_F, we use the result introduced by [21] concerning the series connection of ISS systems to conclude that the closed-loop system (4)-(7), see also Figure 1 with the controller C with the block diagram representation from Figure 3, is ISS with respect to the inputs F_H, F_E and $\dot{F}_E$.

Remark 2. *By studying the ISS property of the estimation and tracking error dynamics, one can observe that the steady-state force estimation and tracking errors can be influenced by tuning the parameters T, T_0, K_0, K_1 and K_2. The algorithm provides a deeper insight into these relations. If we consider the converging manifold that bounds the error signal we can determine these parameters in accordance with the desired convergence rate.*

Remark 3. *In case the environmental force F_E is constant, i.e. $\dot{F}_E = 0$, the force estimation dynamics are globally exponentially stable and the tracking error dynamics is ISS with respect to the inputs F_H and F_E. This means that "perfect" haptic feedback is provided and that bounded tracking errors remain; therefore the closed loop system is stable.*

Remark 4. *The exact "tracking" regulation of the slave robot with respect to what the human has in mind is up to the human (since the human is in charge of the ultimate positioning).*

4 An Illustrative Example

In this section, we will apply the control design proposed in the previous section to a master-slave teleoperation setup consisting of two 1-DOF robots. The inertia of the robots is considered to be in the range $m_M, m_S \in [0.1, 10]kg$.

The "human" controller has been emulated by a linear transfer function:

$$H(s) = \frac{K_d(T_d s + 1)}{T_{PL} s + 1} = \frac{500(1+s)}{0.1s + 1}, \tag{48}$$

with saturation at $\pm 100N$. Here we use real human parameters, since the human movement is lower than $6Hz$. Moreover, to comply with the human sensing range, which is between $0Hz$ and $40 - 400Hz$ depending on the amplitude of the input signal, we have chosen the parameters of the cycle period of the controller $T = 0.01s$ and the duration of the first stage $T_0 = T/2 = 0.005s$. The force estimator acting in the first phase of the algorithm is defined by parameter $K_0 = 10^5$. The tracking PD controller which is active during the second phase has the parameters $K_1 = 200$ and $K_2 = 1$.

In Figure 4, we present the results of the simulation of the master-slave position tracking when the "human" is performing a movement from $0m$ to $0.25m$ on the master robot (i.e. the "human" force F_H is computed such that this motion is achieved) and a sinusoidal external force $F_E(t) = 0.5sin(2\pi t)$ is disturbing the slave robot. The dotted line is the position of the master and the solid line is the position of the slave.

One can observe that because no disturbance rejection controller is implemented, the external force is preventing the position signal to settle exactly at $0.25m$. In Figure 5, a zoomed in version of the Figure 4 that emphasizes this aspect is presented.

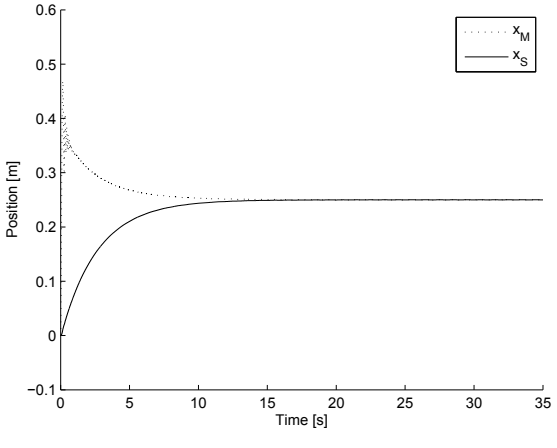


Fig. 4. Position tracking.

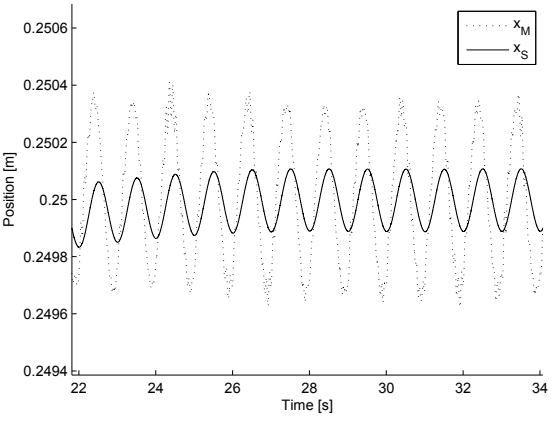


Fig. 5. Position tracking (zoomed version of Figure 4).

5 Conclusions and Perspectives

In this paper, we have introduced a new control algorithm for bilateral teleoperation of 3D robots in force-sensorless setups using a switching strategy between a force estimating controller and a tracking controller. This switching algorithm guarantees both the estimation of the environmental force acting upon the slave robot (to be used in haptic feedback) in the absence of force sensors and the convergence of the tracking errors in the case of external perturbations acting on the slave robot. We note that the ultimate position setting is the responsibility of the human, as he is in charge of the position of the master robot. Finally, we remark that the proposed algorithm is robust for an unknown inertia of the load to be carried by the slave robot and an unknown inertia of the human hand on the master side.

Future perspectives of this work will mainly focus on an extension to multi-degree-of-freedom robots with nonlinear dynamics. The reader should note that the extension of this result for robots with nonlinear dynamics using a pseudo-linearizing controller as in [20] is not straight forward, due to the variable change which depends on the velocities and accelerations.

References

1. Lawrence, D.: Stability and transparency in bilateral teleoperation. IEEE Transactions on Robotics and Automation 9, 624–637 (1993)
2. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: A historical survey. Automatica 42, 2035–2057 (2006)
3. Yokokohji, Y., Yoshikawa, T.: Bilateral control of master-slave manipulators for ideal kinematics coupling formulation and experiment. IEEE Transactions on Robotics and Automation 10, 605–620 (1994)
4. White, M.T., Tomizuka, M., Smith, C.: Improved track following in magnetic disk drives using a disturbance observer. IEEE Transactions on Mechatronics 5, 3–11 (1998)

5. Fujiyama, K., Katayama, R., Hamaguchi, T., Kawakami, K.: Digital controller design for recordable optical disk player using disturbance observer. In: Proceedings of the International Workshop on Advanced Motion Control, pp. 141–146 (2000)
6. Iwasaki, M., Shibata, T., Matsui, N.: Disturbance-observer-based nonlinear friction compensation in table drive system. IEEE Transactions on Mechatronics 4, 3–8 (1999)
7. Komada, S., Machii, N., Hori, T.: Control of redundant manipulators considering order of disturbance observer. IEEE Transactions on Industrial Electronics 47, 413–420 (2000)
8. Kempf, C.J., Kobayashi, S.: Disturbance observer and feedforward design for high-speed direct-drive position table. IEEE Transactions on Control System Technology 7, 513–526 (1999)
9. Eom, K.S., Suh, I.H., Chung, W.K.: Disturbance observer based path tracking of robot manipulator considering torque saturation. Mechatronics 11, 325–343 (2000)
10. Güvenc, B.A., Güvenc, L.: Robustness of disturbance observers in the presence of structured real parametric uncertainty. In: Proceedings of the American Control Conference, pp. 4222–4227 (2001)
11. Ryoo, J.R., Doh, T.Y., Chung, M.J.: Robust disturbance observer for the track-following control system of an optical disk drive. Control Engineering Practice 12, 577–585 (2004)
12. Chen, X., Komada, S., Fukuda, T.: Design of a nonlinear disturbance observer. IEEE Transactions on Industrial Electronics 47, 429–436 (2000)
13. Liu, C.S., Peng, H.: Disturbance observer based tracking control. Journal of Dynamic Systems, Measurement and Control 122, 332–335 (2000)
14. Eom, K.S., Suh, I.H., Chung, W.K., Oh, S.R.: Disturbance observer based force control of robot manipulator without force sensor. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3012–3017 (1998)
15. Ohishi, K., Miyazaki, M., Fujita, M., Ogino, Y.: H^{∞} observer based force control without force sensor. In: Proceedings of International Conference on Industrial Electronics, Control and Instrumentation, pp. 1049–1054 (1991)
16. Ohishi, K., Miyazaki, M., Fujita, M., Ogino, Y.: Force control without force sensor based on mixed sensitivity H^{∞} design method. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 1356–1361 (1992)
17. Alcocer, A., Robertsson, A., Valera, A., Johansson, R.: Force estimation and control in robot manipulators. In: Proceedings of the 7^{th} Symposium on Robot Control (SYROCO 2003), pp. 31–36 (2003)
18. Kröger, T., Kubus, D., Wahl, F.M.: Force and acceleration sensor fusion for compliant manipulation control in 6 degrees of freedom. Advanced Robotics 21, 1603–1616 (2007)
19. Garcia, J.G., Robertsson, A., Ortega, J.G., Johansson, R.: Sensor fusion for compliant robot motion control. IEEE Transations on Robotics 24, 430–441 (2008)
20. Lichiardopol, S., van de Wouw, N., Nijmeijer, H.: Boosting human force: A robotic enhancement of a human operator's force. In: Proceedings of International Conference on Decision and Control, pp. 4576–4581 (2008)
21. Jiang, Z.P., Mareels, I.M.Y., Wang, Y.: A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems. Automatica 32, 1211–1215 (1996)
22. Nešić, D., Teel, A.R., Sontag, E.D.: Formulas relating $\mathcal{KL}$-stability estimates of discrete-time and sampled-data nonlinear systems. Systems & Control Letters 38, 49–60 (1999)
23. Gielen, R., Olaru, S., Lazar, M., Heemels, W.P.M.H., van de Wouw, N., Niculescu, S.I.: On polytopic approximations as a modeling framework for systems with time-varying delays. Automatica (2010) (accepted)
24. Jiang, Z.P., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica 37, 857–869 (2001)

Grasping with Vision Descriptors and Motor Primitives

Oliver Kroemer[1], Renaud Detry[2], Justus Piater[2], and Jan Peters[1]

[1] Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany
oliverkro@tuebingen.mpg.de, jan.peters@tuebingen.mpg.de
[2] University of Liege, Grande Traverse 10, 4000 Liege, Belgium
{renaud.detry,Justus.Piater}@ulg.ac.be

Abstract. Grasping is one of the most important abilities needed for future service robots. Given the task of picking up an object from betweem clutter, traditional robotics approaches would determine a suitable grasping point and then use a movement planner to reach the goal. The planner would require precise and accurate information about the environment and long computation times, both of which may not always be available. Therefore, methods for executing grasps are required, which perform well with information gathered from only standard stereo vision, and make only a few necessary assumptions about the task environment. We propose techniques that reactively modify the robot's learned motor primitives based on information derived from Early Cognitive Vision descriptors. The proposed techniques employ non-parametric potential fields centered on the Early Cognitive Vision descriptors to allow for curving hand trajectories around objects, and finger motions that adapt to the object's local geometry. The methods were tested on a real robot and found to allow for easier imitation learning of human movements and give a considerable improvement to the robot's performance in grasping tasks.

Keywords: Dynamical motor primitives, Early cognitive vision descriptors, Grasping.

1 Introduction

Consider the scenario wherein you want to have a humanoid robot grasp an object in a cluttered space. The first stage of most grasp planners determines a suitable grasp location on the object [21,1,3]. Having selected a final location and orientation for the hand, the robot must then determine how to execute the grasp so as not to collide with the object or any of the surrounding objects.

The traditional solution for this scenario involves supplying the robot with a CAD model of the objects and a laser scanner or other means (ERFID, previous position, etc.) for obtaining their precise positions. These tools give the robot ample knowledge to apply a planning algorithm that determines a suitable path to the goal. This process relies on precise sensor information and can be very time consuming given a complex scene with numerous possible object collisions to test for at each step. In contrast, humans can perform successful grasps of objects in the periphery of their vision, where visual information is limited.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 211–223.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

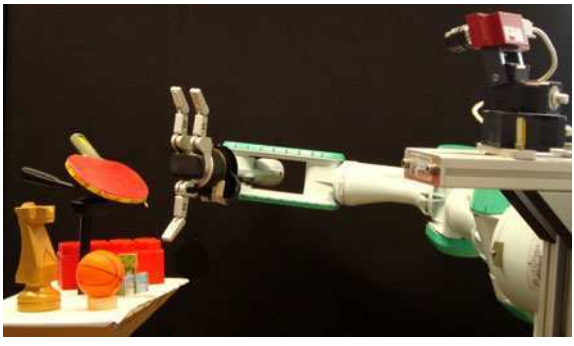


Fig. 1. The robot used in our experiments and an example of a grasping task in a cluttered environment.

Taking inspiration from human movements, we propose a reactive method for robots grasping objects in cluttered environments using potential fields based on only a small amount of visual information. Specifically, we present methods for incorporating information derived from Early Cognitive Vision Descriptors (ECVD) [20] into the dynamical system motor primitives (DMP) [22] framework. The Early Cognitive Vision system (see Appendix and Figure 2) was chosen since it makes only a few assumptions about the object being grasped, while the motor primitives (see Appendix) were chosen because they generalize well to new situations and can be learned through imitation [12]. The two frameworks are also compatible with each other and thus straightforward to combine.

The ECVDs were used to elegantly augment the DMPs for grasping tasks, resulting in the robot being able to avoid obstacles, curve its reaching trajectories around the object to grasp, and adapting the fingers to the local geometry of the object.

2 Methods for Reactive Grasping

The methods proposed in this section were inspired by human movements. Human grasping movements can be modeled as two linked components, transportation and finger posture, synchronized by a shared timer or canonical system [5,18]. Transportation refers to the actions of the arm in moving the hand, while the finger posture aspect relates to the preshaping and closing of the fingers [14].

Humans perform the reaching/transportation component in a task-specific combination of retina and hand coordinates [8], which allows for easier specification of object trajectories in a manipulation task than joint coordinates would and results in a reduction in dimensionality. These movements also have curved trajectories that are needed for avoiding obstacles and reaching around objects, which mainly occurs in a planar subspace [24].

Similar to the transportation component, the main purpose of the finger posture component is to preshape the hand by extending the fingers sufficiently for them to pass around the object upon approach, and then close on the object simultaneously for a good grasp. Over-extending the fingers is undesirable as it makes collisions with the

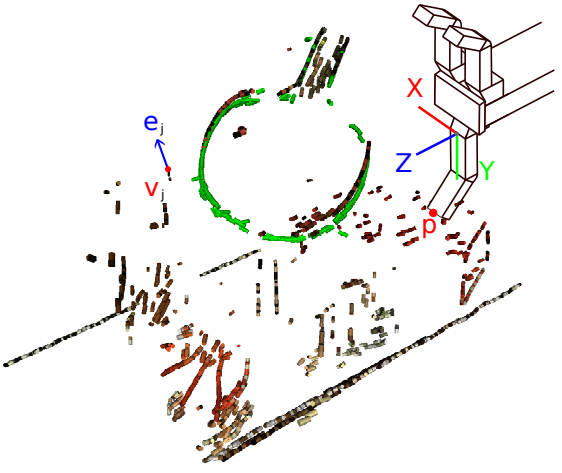


Fig. 2. The green ECVD represent the object to be grasped, while the surrounding ECVDs in the scene are clutter. The coordinate frame of the third finger of the Barrett hand (the lower finger in the image) and variables used in section 2 are shown. The x-y-z coordinate system is located at the base of the finger, with z orthogonal to the palm, and y in the direction of the finger. The marked ECVD on the left signifies the j^{th} descriptor, with its position at $\mathbf{v}_j = (v_{jx}, v_{jy}, v_{jz})^T$, and edge direction $\mathbf{e}_j = (e_{jx}, e_{jy}, e_{jz})^T$ of unit length. The position of the finger tip is given by $\mathbf{p} = (p\_x, p\_y, p\_z)^T$.

environment more likely and is therefore usually restricted to situations that present large uncertainties about the object [17,5].

Curved reaching trajectories and preshaping of the hand were incorporated into the robot via a potential field, as described in Sections 2.1 and 2.2. Subsequently, a higher level controller is proposed in Section 2.3, which allows the grasping movements to be interpolated better to new target grasp locations.

2.1 DMP Based Attractor Field

The first step towards specifying the grasping movements is to define an attractor field as a DMP that encodes the desired movements given no obstacles. The principal features that need to be defined for these DMPs are 1) the goal positions, and 2) the generic shape of the trajectories to reach the goal.

Determining the goal posture of the hand using the ECV descriptors has been previously investigated in. [6]. In this work, possible grasp locations were hypothesized from the geometry and color features of the ECVDs, and used to create a kernel density estimate of suitable grasps, which is then refined by attempting grasps to test them.

However, this grasp synthesizer only gives the desired location and orientation of the hand, but leaves finger placement to a secondary finger controller, e.g., [10,23]. Using the ECVDs, the goal position of each finger is approximated by first estimating a contact plane for the object in the finger coordinate system shown in Figure 2. To make it a local approximation, the influence of the i^{th} ECVD is weighted by $w_i = \exp(-\sigma_x^{-2} v_{ix}^2 -$

$\sigma_y^{-2} v_{iy}^2 - \sigma_z^{-2} v_{iz}^2$), where σ_x, σ_y, and σ_z are length scale constants, and $\mathbf{v}_i$ is the position of the ECVD in the finger reference frame. The hand orientation was chosen such that the Z direction of the finger is parallel to the approximated contact plane, which reduces the problem to describing the plane as a line in the 2D X-Y space. The X-Y gradient of the plane is approximated by $\phi = (\sum_{i=1}^{N} w_i)^{-1} \sum_{i=1}^{N} w_i \arctan(e_{iy}/e_{ix})$, where N is the number of vision descriptors, and $\mathbf{e}_i$ is the direction of the i^{th} edge. The desired Y position of the fingertip is then given by

$$\tilde{p}_y = \frac{\sum_{i=1}^{N} (w_i v_{iy} - \tan(\phi) w_i v_{ix})}{\sum_{i=1}^{N} w_i},$$

which can be easily converted to a joint parameter using the inverse kinematics of the hand.

Having determined the goals of both transportation and finger-posture components, the next step is to define the trajectories used to reach these goals. Many of the beneficial traits of human movements, as described earlier, can be transferred to the robot through imitation learning. Learning by imitation involves a human demonstrating a motion and the robot then mimicking the movement. Details for imitation learning with DMPs can be found in [12].

We can now combine the goals and imitation learned trajectories to specify the DMPs and thus the attractor fields.

2.2 ECVD Based Detractor Fields

Having specified the rudimentary grasping movements, a detractor field is employed to refine the motions in order to include obstacle avoidance for the transportation and ensure that the finger tips do not collide with the object during the hand's approach.

The detractor field will be based on ECVDs, which can be envisioned as small line segments of an object's edges localized in 3D, as shown in Figure 2 for a scene as shown in Figure 1. The detractive potential fields for ECVDs are characterized by two main features; i.e., the detractive forces of multiple ECVDs describing a single line do not superimpose, and the field does not stop DMPs from reaching their ultimate goals. The system therefore uses a Nadaraya-Watson model [4] of the form

$$u = -s(x) \frac{\sum_{i=1}^{N} r_i c_i}{\sum_{j=1}^{N} r_j},$$

to generate a suitable detractor field, where r_i is a weight assigned to the i^{th} ECVD, s is the strength of the overall field, x is the state of the DMPs' canonical system, and c_i is the detracting force for a single descriptor.

The weight of an ECVD for collision avoidance is given by $r_i = \exp(-(\mathbf{v}_i - \mathbf{p})^{\mathrm{T}} \mathbf{h} (\mathbf{v}_i - \mathbf{p}))$, where $\mathbf{v}_i$ is the position of the i^{th} ECVD in the local coordinate system, $\mathbf{h}$ is a vector of positive length scale hyperparameters, and $\mathbf{p}$ is the finger tip position, as shown in Figure 2. The detractor therefore puts more importance on ECVDs in the vicinity of the finger.

The strength factor ensures that the detractor forces always tend to zero at the end of a movement and thus it can not obstruct the attractor from achieving its goal at the end. Therefore, the strength of the detractors is coupled to the canonical system of the DMP; i.e., $s(x) = (\sum_{j=1}^{M} \psi_j)^{-1} \sum_{i=1}^{M} \psi_i w_i x$, where x is the value of the canonical system, ψ are its basis functions, and w specify the varying strength of the field during the trajectory.

The transportation and finger-posture movements react differently to edges and thus employ different types of basis functions c_i for their respective potential fields. For the fingers, the individual potential fields are logistic sigmoid functions about the edge of each ECVD of the form $\rho(1+\exp(d_i\sigma_c^{-2}))^{-1}$, where $d_i = \left\| (\mathbf{p}-\mathbf{v}_i) - \mathbf{e}_i(\mathbf{p}-\mathbf{v}_i)^{\mathrm{T}}\mathbf{e}_i \right\|$ is the distance from the finger to the edge, $\rho \geq 0$ is a scaling parameter, and $\sigma_c \geq 0$ is a length parameter. Differentiating the potential field results in a force term of

$$c_i = \rho \frac{\exp\left(d_i\sigma_c^{-2}\right)}{\left(1+\exp\left(d_i\sigma_c^{-2}\right)\right)^2}.$$

As the logistic sigmoid is monotonically increasing, the detractor always forces the fingers open further to move their tips around the ECVDs and thus ensure that they always approach the object from the outside.

The hand uses instead the Gaussian basis functions of the form $\varrho\exp(-0.5\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i\sigma_d^{-2})$, where $\mathbf{d}_i = (\mathbf{q}-\mathbf{v}_i) - \mathbf{e}_i(\mathbf{q}-\mathbf{v}_i)^{\mathrm{T}}\mathbf{e}_i$ is the distance from the end effector position, $\mathbf{q}$, to the edge, and $\varrho \geq 0$ and $\sigma_d \geq 0$ are scale and length parameters respectively. Differentiating the potential with respect to $\mathbf{d}_i$ gives a force term in the Y direction of

$$c_i = \left[\varrho\mathbf{d}_i\sigma_d^{-2}\exp(-0.5\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_i\sigma_d^{-2})\right]_Y,$$

which can be interpreted as a radial force from the edge with an exponentially decaying magnitude.

The detractor fields, of both the grasping and reaching components, have now been defined, and can be superimposed into the DMP framework as

$$\ddot{y} = \left(\alpha_z(\beta_z\tau^{-2}(g-y) - \tau^{-1}\dot{y}) + a\tau^{-2}f(x)\right) - \tau^{-2}u,$$

which then represents the entire ECVD and MP based potential field.

2.3 High Level DMP Controller for Grasping

Having defined the potential field for a single grasping motion, we interpolate the movements to new target grasps. Having a motion representation that can be interpolated to new targets is crucial for imitation learning. Given such a representation, the number of example trajectories required from the demonstrator can be greatly increased, making learning easier. While DMPs can interpolate to arbitrary goal positions, they have two drawbacks for grasping tasks; i.e., 1) the approach direction to the grasp can not be arbitrarily defined, and 2) the amplitude of the trajectory is unneccessarily sensitive to changes in the start position y_0 and the goal positon g if $y_0 \approx g$ during training, which can cause the robot to reach the limits of its workspace.

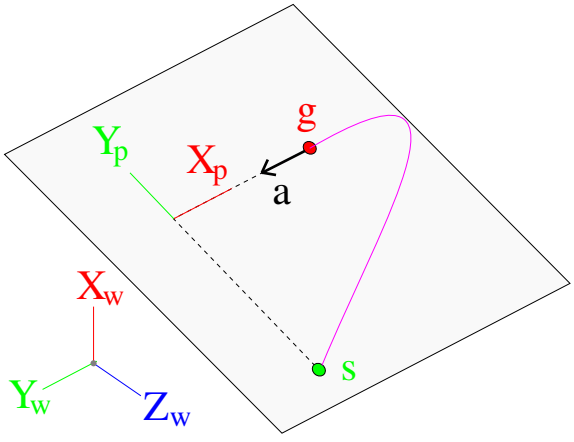


Fig. 3. The above diagram shows the the change in coordinate systems for the transportation DMPs. The axes X_w-Y_w-Z_w are the world coordinate system, while X_p-Y_p-Z_p is the planar right handed coordinate system in which the DMP is specified. The trajectory of the DMP is shown by the **pink** line, starting at the **green point**, and ending at the **red point**. Note that X_p is parallel to the approach direction of the hand, which is shown by the **black arrow** $\mathbf{a}$. The planar axis Y_p is perpendicular to X_p, and pointing from the motor primitive's starting location $\mathbf{s}$ towards the goal $\mathbf{g}$.

These difficulties can be overcome by including a supervisory controller that modifies the hyperparameters of the DMPs appropriately. The supervisor can maintain the correct approach direction by using a task-specific coordinate system. Due to the translation invariance of DMPs, only a rotation, $\mathbf{R} \in \mathbb{SO}(3)$, between the two coordinate systems needs to be determined. The majority of the motions will lie in a plane defined by the start and goal locations, and the final approach direction.

The first new in-plane axis $\mathbf{x}_p$ is set to be along the approach direction of the grasp; i.e., $\mathbf{x}_p = -\mathbf{a}$ as shown in Figure 3. As a result, the approach direction is easily defined and only requires that the Y_p and Z_p primitives reach their goal before the X_p primitive. The second axis, $\mathbf{y}_p$, must be orthogonal to $\mathbf{x}_p$ and also in the plane, as shown in Figure 3. It is set to $\mathbf{y}_p = b^{-1}((\mathbf{g}-\mathbf{s}) - \mathbf{x}_p(\mathbf{g}-\mathbf{s})^T\mathbf{x}_p)$, where b^{-1} is a normalization term, and $\mathbf{s}$ and $\mathbf{g}$ are the motion's 3D start and goal positions respectively. The third vector, $\mathbf{z}_p$, is orthogonal to the plane, and is derived by completing the right-handed coordinate system, i.e., $\mathbf{z}_p = \mathbf{x}_p \times \mathbf{y}_p$. The DMPs can now be specified by the supervisor in the X_p-Y_p-Z_p coordinate system, and mapped to the X_w-Y_w-Z_w world reference frame by multiplying by $\mathbf{R}^T = [\mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p]^T$.

The second problem relates to the scaling of motions with ranges greater than $y_0 - g$, which both components require to move around the outside of objects. In the standard form $a = g - y_0$ [13], which can lead to motions that easily exceed the robot's workspace if $g \approx y_0$ during the training, but not during the motion reproduction. The supervisor can control these trajectories by scaling the shaping force (see Appendix), and thus we propose the amplitude term

$$a = \|\eta(g - y_0) + (1 - \eta)(g_T - y_{0T})\|\,,$$

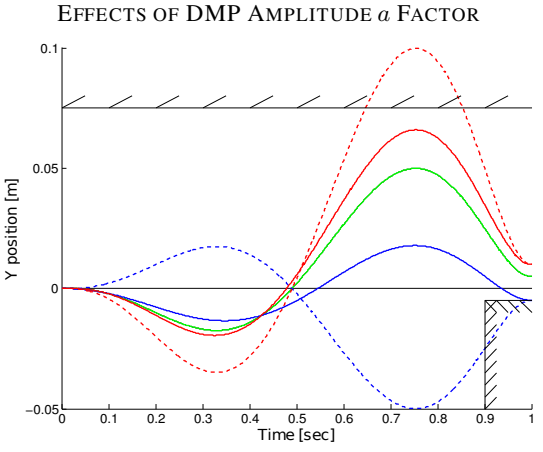


Fig. 4. This is a demonstration of the effects of augmenting the amplitude variable a of DMPs. The black lines represent boundaries. The **green** plot shows the trained trajectory of the DMP going to 0.05, and is the same for all amplitude values. Now consider the scenario wherein the goal is placed at 0.1, but the workspace is limited to ± 0.75 (top boundary). The **dashed red** line is the standard generalization to a larger goal, while the **red** plot uses the new amplitude. Notice how the new amplitude restricts the range of the trajectory to the workspace. In a different scenario, we move the goal to -0.05, but require the goal to be reached from above (lower right boundary), e.g., a finger placed on a surface. The **dashed blue** line is the standard generalization to a negative goal, and the **blue** trajectory uses the new amplitude. Note that the trajectory is not flip in the case of the new amplitude and thus stays within the restricted region. Both of the new trajectories were generated with $\eta = 0.25$, and maintain shapes close to that of the training trajectory.

where g_T and y_{0T} are the goal and start positions of the training data respectively, and $\eta \in [0, 1]$ is a weighting hyperparameter. The resulting trajectory amplitude is in the convex hull of the training amplitude and the standard interpolation value ($a = g - y_0$) [13] and thus only affects how conservative the generalization to new points is, as can be seen in Figure 4. By taking the absolute value of the amplitude, the approach direction is not reversed, giving a result similar to the use of a constant amplitude proposed by Park et al. [19], which corresponds to the special case of $\eta = 0$. Example interpolations of a transportation trajectory can be seen in Figure 5.

3 Grasping Experiments

The methods described in Section 2 were implemented and evaluated on a real robot platform. The robot consists of a Videre stereo camera mounted on a pan-tilt unit, a Barrett hand, and a Mitsubishi PA10 arm. The robot was given the task of grasping an object amongst clutter using only an ECVD model of the object. The results of these trials were then compared to trials of the same grasps using other standard robotics methods for comparison. We hypothesize that our method will result in significantly more successful grasps than the other methods.

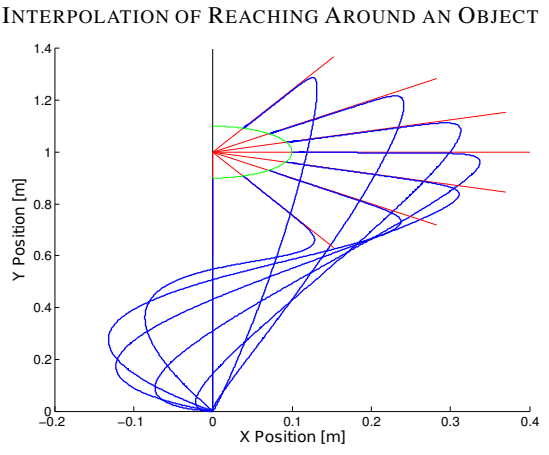


Fig. 5. The plot shows workspace trajectories, wherein the x and y values are governed by two DMPs sharing a canonical system. The **red** lines indicate the desired approach direction while the **green** semicircle indicates the goal positions along them. The **blue** lines show the trajectories for the different goals. They make use of the higher level controller of Subsection 2.3, with $\eta = 0.25$. The approach direction DMP was trained on an amplitude of one.

3.1 Grasping Experiment Procedure

Before the robot can perform a grasping task, its motions must be initialized. Determining the finger goal state and specifying the detractor fields introduces several new hyperparameters that have simple geometrical interpretations. For instance, $\mathbf{h} = 2[w\ l\ l]^{\mathrm{T}}$, where w and l are the width and length of the finger respectively. To reflect the human tendency towards more precise movements during the last 30% of a motion [15], the strength function, $s(x)$, was set to give the highest strengths during the first 70% of the motion for the transportation, and the last 30% for the finger posture.

A VICON$^{\mathrm{TM}}$ motion tracking system was used to record the movements of a human test subject during a grasping task, which used a different object to the one used by the robot. As the reaching trajectories are encoded in task space rather than joint space, the correspondence problem was not an issue for the imitation learning. Similarly, the DMPs of the fingers are homogeneous, which circumvents the correspondence problem of mapping the five human fingers onto the three fingers of the robot. The imitation learning was performed using locally weighted regression in the the X_p-Y_p-Z_p coordinate system, as proposed by Ijspeert et al. [12].

Having defined the basic motions, the robot was then given the task of grasping an object without hitting surrounding obstacles (see Figure 1). Each trial begins with an estimate of the pose of the object relative to the robot [7] and sets its grasp location accordingly. The model's ECVD are then projected into the scene, and the robot attempts to perform the grasp and lift the object 15cm so that it is clear of the stand. The trial is a success if the robot can detect the object in its hand at this point. If the hand collides with an obstacle or knocks the object down, the trial is marked as a failure. Grasps

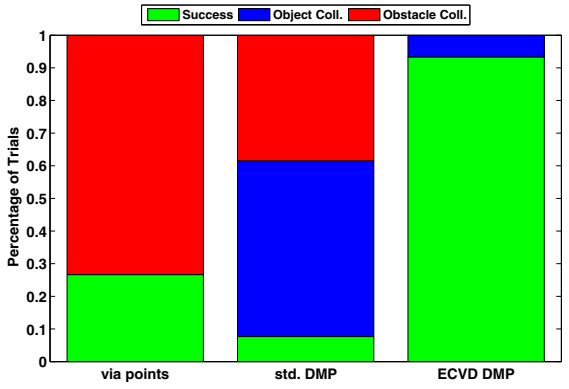


Fig. 6. The occurrences of successes and collision types for the different methods are shown. The first column presents the results for the traditional robotics method of specifying trajectories by via points. The second column corresponds to using standard DMPs, while the final column incorporates the ECVD based potential field and supervisory DMP controller. The occurrences are given as the percentage of trials. Trials that collided multiple times, are classified by their first collision.

were varied to include different approach directions and locations around the object. The experiment consisted of 45 trials.

Two alternative approaches were compared with our proposed method. The first represents a standard robotics approach of specifying a trajectory by straight lines between via points and uses fully extended fingers with no preshaping of the hand. The other approach is to use standard DMPs learned from the same human demonstrated movements as our proposed methods, but without the proposed detractor field and supervisory controller. The same grasp locations were proposed to the different methods, and obstacles were placed in similar positions for the the different trials to allow for a fair comparison between the methods.

3.2 Experimental Results

From the three tested methods, the proposed method acquired the highest success rate, as can be seen in Figure 6. The task was not trivial, and all of the methods encountered both successes and problems during the trials.

The standard DMP method encountered the most problems (success rate of only 7%) a majority of which were caused by collisions with the object. This high failure rate can be attributed to the method not specifically incorporating a desired approach direction. In successful trials, the approach direction was close to that of the initial imitation learning. Therefore the proposed DMP supervisor improved the generalization of the movement to new target grasps, and the system would benefit from it even in uncluttered environments. Similarly, the open-loop preshaping of the hand helped avoid obstacles, but occasionally prevented the hand from being sufficiently open to accept the object. The proposed detractor field successfully overcame this problem for the ECVD DMPs.

The via points method encountered no collisions with the object, and would have worked well in an uncluttered environment. The method still encountered collisions with the obstacles for 73% of the trials, but this is more reflective of the difficulty of the task rather than the via point method. The method can therefore be considered as a good approach if it were combined with a suitable path planning method for obstacle avoidance. However, the path planner would need additional information and assumptions about the scene and possibly even extra hardware to acquire it.

The proposed method had a success rate of 93%, with no occurrences of collisions with obstacles. The trials that did fail were the result of the object falling down while the fingers were closing and thus do not indicate problems with the approach used to reaching the grasp location. The method does have certain restrictions though. The magnitude of the detractor fields needs to be calibrated based on the density of ECVDs for common objects, but some obstacles encountered may present lower densities. As the current set of ECVD relies on object edges, smooth objects can lead to noisy or very sparse descriptors, and therefore not create a suitable basis for obstacle avoidance. As the number of descriptor types increases (e.g., corner and plane descriptors), this will become less of a problem. Occluded obstacles will also need to rely on additional information (e.g., force feedback) to be avoided, although this is a source of error for all vision based planners.

Given a few restrictions, the results still show that our hypothesis was correct and the proposed methods represent a suitable basis for avoiding obstacles without relying on a complicated path planner and using only a small amount of vision information compared to standard robot systems.

4 Conclusions

The proposed methods augment dynamical system motor primitives to incorporate Early Cognitive Vision descriptors by using a potential field. These methods represent important tools that a robot needs to reactively execute grasps of an object in a cluttered environment without relying on a complex planner. The techniques allow for preshaping the fingers to match the shape and size of the object and curving the trajectory of the hand around objects[24]. These modifications were tested on a real robot, and it was discovered that the methods were not only successful at performing the task, but also allowed for easier imitation learning, better interpolation of the learned trajectories, and significantly better chances of a success of a grasp in cluttered environments than standard motor primitives. Although the experiments were performed within a grasping task scenario, the proposed methods can be beneficial for other manipulation tasks, such as pressing buttons and pushing objects.

References

1. Arimoto, S.: Control Theory of Multi-fingered Hands. Springer, London (2008)
2. Bard, C., Troccaz, J., Vercelli, G.: Shape analysis and hand preshaping for grasping. In: Proceedings of IROS 1991(1991)
3. Bicchi, A., Kumar, V.: Robotic grasping and contact: a Review. In: Proceedings of ICRA 2000 (2000)

4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
5. Chieffi, S., Gentilucci, M.: Coordination between the transport and the grasp components during prehension movements (1993)
6. Detry, R., Kroemer, O., Popovic, M., Touati, Y., Baseski, E., Krueger, N., Peters, J., Piater, J.: Object-specific grasp affordance densities. In: ICDL (2009)
7. Detry, R., Pugeault, N., Piater, J.: Probabilistic pose recovery using learned hierarchical object models. In: International Cognitive Vision Workshop (2008)
8. Graziano, M.S.: Progress in understanding spatial coordinate systems in the primate brain. In: Neuron (2006)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
10. Hsiao, K., Nangeroni, P., Huber, M., Saxena, A., Ng, A.: Reactive grasping using optical proximity sensors. In: Proceedings of ICRA 2009 (2009)
11. Iberall, T.: Grasp planning for human prehension. In: Proceedings of ICAI 1987 (1987)
12. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: ICRA (2002)
13. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: NIPS (2003)
14. Jeannerod, M.: Grasping Objects: The Hand as a Pattern Recognition Device. Perspectives of Motor Behaviour and Its Neural Basis (1997)
15. Jeannerod, M.: The study of hand movements during grasping. A historical perspective. In: Sensorimotor Control of Grasping: Physiology and Pathophysiology, Cambridge University Press, Cambridge (2009)
16. Krueger, N., Lappe, M., Woergoetter, F.: Biologically motivated multimodal processing of visual primitives. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour (2004)
17. Oztop, E., Bradley, N.S., Arbib, M.A.: Infant grasp learning: a computational model (2004)
18. Oztop, E., Kawato, M.: Models for the control of grasping. In: Sensorimotor Control of Grasping: Physiology and Pathophysiology, Cambridge University Press, Cambridge (2009)
19. Park, D.-H., Hoffmann, H., Pastor, P., Schaal, S.: Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: IEEE International Conference on Humanoid Robots(HUMANOIDS) (2008)
20. Pugeault, N.: Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation. Vdm Verlag Dr. Mueller (2008)
21. Saxena, A., Dreimeyer, J., Kearns, J., Osondu, C., Ng, A.: Learning to Grasp Novel Objects using Vision. In: Experimental Robotics. Springer, Berlin (2008)
22. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: Proceedings of ISRR 2003 (2003)
23. Steffen, J., Haschke, R., Ritter, H.: Experience-based and tactile-driven dynamic grasp control. In: Proceedings of IRS 2007 (2007)
24. Wank, V., Fischer, A., Bos, K., Boesnach, I., Moldenhauer, J., Beth, T.: Similarities and varieties in human motiontrajectories of predefined grasping and disposing movements. International Journal of Humanoid Robotics (2004)

APPENDIX

Dynamical Systems Motor Primitives

The dynamical systems motor primitives (DMPs) proposed by Ijspeert et al. [13] were inspired by the simple, but highly adaptive, motions that animals employ, and combine to obtain more complex motions. The primitives are implemented as a passive dynamical system with an external force, and represented as

$$\ddot{y} = \alpha_z(\beta_z\tau^{-2}(g - y) - \tau^{-1}\dot{y}) + a\tau^{-2}f(x), \tag{1}$$

where α_z and β_z are constants, τ controls the duration of the primitive, a is an amplitude, $f(x)$ is a nonlinear function, and g is the goal for the state variable y.

By selecting α_z and β_z appropriately, and setting $a = 0$, the system reduces to $\ddot{y} = \alpha_z(\beta_z\tau^2(g - y) - \tau\dot{y})$ and becomes a critically damped global attractor. It can be visualized as a spring and damper system that ensures state y will always end at the goal value g.

The function $f(x)$ is a shaping function based on the state, $x \in [0, 1]$, of the canonical system that synchronizes the DMPs $\dot{x} = -\alpha_x\tau x$, where α_x is a time constant. The function takes the form

$$f(x) = \frac{\sum_{j=1}^{M} \psi_j(x)\, w_j x}{\sum_{i=1}^{M} \psi_i(x)},$$

where M is the number of basis functions, $\psi(x)$ are Gaussian basis functions, and w are weights acquired through locally weighted regression [13]. This function has the effect of introducing a non-linearity that can affect the spring-damper system to output any arbitrary trajectory specified by the user. Due to the dependence of $f(x)$ on x, the shaping term decays to zero with x, so that the spring and damper beneficial properties of the attractor are maintained.

The resulting primitives can encode arbitrary trajectories, and still ensure that the goal state is always achieved. The trajectories can also be scaled in time and space by setting the τ and g variables appropriately and thus generalize to a range of situations.

Early Cognitive Vision System

The entire prehensile process effectively occurs before the hand has even touched the object and thus the vision system plays a very important role [2,11]. Our system uses the Early Cognitive Vision methods of Pugeault et al. [20,9], which makes a minimal number of assumptions about the object, and has been successfully implemented to determine good grasp locations [6]. A principal idea of this vision system is to store additional low level information and perform perceptual grouping on it to later aid the higher level stereo matching and 3D constructions.

The methods extract local features of a scene, which it localizes and orientates in space [16]. Each descriptor is a symbolic representation for an edge in 3D. The resulting features are called early cognitive vision descriptors (ECVD) [20], and can be used in generating models of objects for pose estimation [7], and for symbolically describing 3D scenes. By using a large amount of small ECVDs, any arbitrary object can be represented.

When performing a grasping task, the robot uses a hierarchical Markov model of the object's ECVD geometry [7] to determine its pose, which can then be used to superimpose the ECVDs of the model back into the scene. The grasping techniques can therefore use geometric information of a partially occluded object.

Remote Tracking Control of Unicycle Robots with Network-Induced Delays

Alejandro Alvarez-Aguirre[1], Nathan van de Wouw[1], Toshiki Oguchi[2], Kotaro Kojima[2], and Henk Nijmeijer[1]

[1] Eindhoven University of Technology, Department of Mechanical Engineering
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{a.a.alvarez,n.v.d.wouw,h.nijmeijer}@tue.nl
[2] Tokyo Metropolitan University, Department of Mechanical Engineering
1-1, Minami-osawa, Hachioji-shi, Tokyo 192-0397, Japan
t.oguchi@tmu.ac.jp, kojima-koutarou@ed.tmu.ac.jp

Abstract. In this chapter, the tracking control problem for a unicycle-type mobile robot with network-induced delays is addressed. The time-delay affects the system due to the fact that the controller and the robot are linked via a delay-inducing communication channel, by which the performance and stability of the system are possibly compromised. In order to tackle the problem, a state estimator with a predictor-like structure is proposed. Acting in conjunction with a tracking control law, the resulting control strategy is capable of stabilizing the system and compensates for the negative effects of the time-delay. The local uniform asymptotic stability of the closed-loop system is guaranteed up to a maximum admissible time-delay, for which explicit expressions are provided in terms of the system's control parameters. The applicability of the proposed estimator-control strategy is demonstrated by means of experiments carried out between multi-robot platforms located in Eindhoven, The Netherlands and Tokyo, Japan.

Keywords: Mobile robot, Remote tracking control, Network delay, Nonlinear estimator, Non-holonomic systems.

1 Introduction

In the increasingly fast and diverse technological developments of the last decades the duties and tasks conferred to control systems have become much more complex and decisive. Requirements now encompass flexibility, robustness, ubiquity and transparency, among others.

Specifically, the study of robotic systems controlled over a network has become significatively important as a way to support the design of robotic systems that can perform remote, dangerous or distributed tasks. The remote control, or the control of a system subject to a network-induced delay is important in e.g. teleoperation strategies and is a central topic in Networked Control Systems (NCS).

Several techniques have been proposed so far in order to cope with network-induced delays in these settings; e.g. the use of scattering transformations, wave variables formulation, queuing methodologies, delay compensation techniques and robust control

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 225–238.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

design to name a few. A detailed description of such techniques and many others, together with further references, can be found in e.g. [4], [5], [19].

In this work, a control strategy for the remote tracking control of a unicycle-type mobile robot is proposed. The network-induced delay is compensated by means of a state estimator inspired by the predictor based on synchronization presented in [12], [13]. The main idea behind the state estimator is to reproduce the system's behavior without delay in order to drive an anticipating controller. The problem presents various challenges since the system is nonlinear and subject to a non-holonomic constraint. Additionally, the difficulties faced when implementing the proposed ideas in an experimental setting using the Internet as the communication channel should be taken into account and are also discussed in depth. In [8], a similar state estimator has been applied to a mobile robot subject to a communication delay, and sufficient conditions for the estimator's convergence have been derived. In this work an alternative approach is taken in order to prove the stability of the entire closed-loop system consisting of the mobile robot, the tracking controller and the state estimator.

This chapter is structured in the following way. Section 2 recalls results on the tracking control of a delay-free unicycle-type mobile robot. In Section 3 a control scheme intended to control a mobile robot with a network-induced time-delay is proposed and conditions on the maximum allowable time-delay in terms of the control parameters are posed. Section 4 provides an overview of the experimental platform used to validate the control strategy proposed, explains how the most critical implementation issues have been addressed, and presents the experimental results. Finally, conclusions are provided in Section 5.

2 Tracking Control of a Unicycle Robot

The tracking control design for a unicycle-type mobile robot is discussed in this section. To begin with, consider the posture kinematic model of a unicycle:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\theta(t), \\ \dot{y}(t) &= v(t)\sin\theta(t), \\ \dot{\theta}(t) &= \omega(t), \end{aligned} \tag{1}$$

in which $x(t)$ and $y(t)$ denote the robot's position in the global coordinate frame X-Y (cf. Figure 1), $\theta(t)$ defines its orientation with respect to the X-axis, and $v(t)$ and $\omega(t)$ constitute the robot's translational and rotational velocities, respectively, and are regarded as the system's control inputs. The robot's state is defined by $q(t) = [x(t)\ y(t)\ \theta(t)]^T$ and the non-slip condition on the unicycle's wheels imposes a non-holonomic constraint on the system, as explained in [1].

The control objective is to track a time-varying reference trajectory specified by $q_r(t) = [x_r(t)\ y_r(t)\ \theta_r(t)]^T$. The reference position $(x_r(t), y_r(t))$ satisfies the dynamics,

$$\begin{aligned} \dot{x}_r(t) &= v_r(t)\cos\theta_r(t), \\ \dot{y}_r(t) &= v_r(t)\sin\theta_r(t), \end{aligned} \tag{2}$$

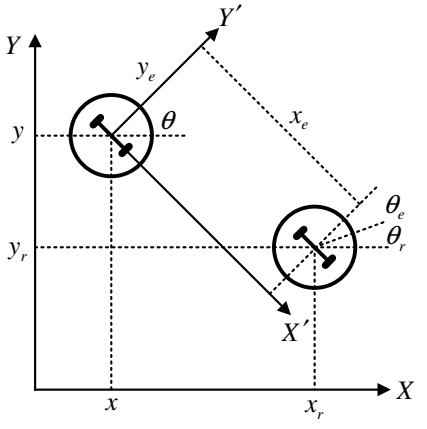


Fig. 1. Mobile robot, reference system, and error coordinates.

while the reference orientation $\theta_r(t)$, translational velocity $v_r(t)$, and rotational velocity $\omega_r(t)$ are defined in terms of the reference Cartesian velocities $\dot{x}_r(t)$, $\dot{y}_r(t)$ and accelerations $\ddot{x}_r(t)$, $\ddot{y}_r(t)$ as follows:

$$\theta_r(t) = \text{arctan2}\left(\dot{y}_r(t), \dot{x}_r(t)\right), \tag{3}$$

$$v_r(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)}, \tag{4}$$

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} = \dot{\theta}_r(t), \tag{5}$$

where atan2 is the arctangent function of two arguments. It is worth noting that computing (3) and (5) requires either $\dot{x}_r(t) \neq 0$ or $\dot{y}_r(t) \neq 0$ at all times.

The difference between the reference trajectory and the state evolution may be expressed with respect to the system's local coordinate frame X'-Y' in order to define the error coordinates $q_e(t) = [x_e(t)\ y_e(t)\ \theta_e(t)]^T$, as proposed by [7] and shown in Figure 1. These tracking error coordinates are given by,

$$\begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) & 0 \\ -\sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r(t) - x(t) \\ y_r(t) - y(t) \\ \theta_r(t) - \theta(t) \end{bmatrix}. \tag{6}$$

Exploiting (1), (2), (5), and (6), the tacking error dynamics result in,

$$\begin{aligned} \dot{x}_e(t) &= \omega(t) y_e(t) + v_r(t)\cos\theta_e(t) - v(t), \\ \dot{y}_e(t) &= -\omega(t) x_e(t) + v_r(t)\sin\theta_e(t), \\ \dot{\theta}_e(t) &= \omega_r(t) - \omega(t). \end{aligned} \tag{7}$$

The following tracking controller has been proposed in [6], [15],

$$\begin{aligned} v(t) &= v_r(t) + c_2 x_e(t) - c_3 \omega_r(t) y_e(t), \\ \omega(t) &= \omega_r(t) + c_1 \sin\theta_e(t), \end{aligned} \tag{8}$$

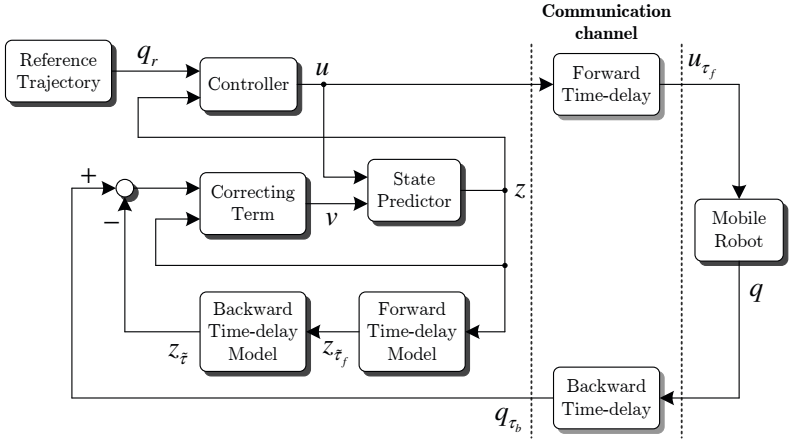


Fig. 2. Block diagram representation of proposed remote tracking control strategy.

which ensures local exponential stability (LES) of the tracking error dynamics (7)-(8) if $c_1, c_2 > 0$ and $c_3 > -1$.

3 Remote Tracking Control

In this section, we consider a mobile robot controlled over a network which induces time-delays, see Figure 2. The robot's controller, consisting of the tracking control law (8) and a state estimator, should ensure that the robot tracks (a delayed version) of the reference trajectory. The state estimator has a predictor-like structure and is similar to the one proposed in [8]. The origin of this type of predictor can be traced back to the appearance of the notion of anticipating synchronization in coupled chaotic systems, which was first noted by [20]. After the same behavior was observed in certain simple physical systems such as specific electronic circuits and lasers, it was studied for more general systems in [14]. As a result of this generalization, a state predictor based on synchronization for nonlinear systems with input time-delay was proposed in [12]. The same concept, which can be seen as a state estimator with a predictor-like structure, is proposed here for a mobile robot subject to a network-induced delay.

3.1 State Estimator and Controller Design

When considering a network-induced delay, the mobile robot is subject not only to a forward τ_f (input) time-delay, but also to a backward τ_b (output) time-delay, as denoted in [5]. Hereinafter the forward and backward time-delays τ_f, τ_b will be assumed to be constant and known, with $\tau := \tau_b + \tau_f$. Given the mobile robot (1) subject to a network-induced input delay τ_f, the robot's posture kinematic model is given by,

$$\begin{aligned} \dot{x}(t) &= v(t-\tau_f)\cos\theta(t), \\ \dot{y}(t) &= v(t-\tau_f)\sin\theta(t), \\ \dot{\theta}(t) &= \omega(t-\tau_f). \end{aligned} \tag{9}$$

Moreover, the system's state measurements are affected by a backward time-delay τ_b: $q(t-\tau_b) = [x(t-\tau_b)\ y(t-\tau_b)\ \theta(t-\tau_b)]^T$.

In order to improve the tracking performance when subject to a communication delay, the following state estimator, with state $z(t) = [z_1(t)\ z_2(t)\ z_3(t)]^T$, is proposed:

$$\begin{aligned}\dot{z}_1(t) &= v(t)\cos z_3(t) + \nu_x(t),\\ \dot{z}_2(t) &= v(t)\sin z_3(t) + \nu_y(t),\\ \dot{z}_3(t) &= \omega(t) + \nu_\theta(t),\end{aligned} \tag{10}$$

with $\nu(t) = [\nu_x(t)\ \nu_y(t)\ \nu_\theta(t)]^T$ defining a correcting term based on the difference between the estimator state and the measured state.

For the purpose of designing the correcting term $\nu(t)$, two new sets of error coordinates are introduced, namely $z_e(t)$ and $p_e(t)$. The first set of error coordinates relates to the difference between the estimator state $z(t)$ and the reference trajectory $q_r(t)$:

$$\begin{bmatrix} z_{1_e}(t)\\ z_{2_e}(t)\\ z_{3_e}(t)\end{bmatrix} = \begin{bmatrix} \cos z_3(t) & \sin z_3(t) & 0\\ -\sin z_3(t) & \cos z_3(t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} x_r(t)-z_1(t)\\ y_r(t)-z_2(t)\\ \theta_r(t)-z_3(t)\end{bmatrix}. \tag{11}$$

The second set of error coordinates relates to the difference between the delayed estimator state $z(t-\tilde{\tau})$ and the delayed system state $q(t-\tau_b)$:

$$\begin{bmatrix} p_{1_e}(t)\\ p_{2_e}(t)\\ p_{3_e}(t)\end{bmatrix} = \begin{bmatrix} \cos z_3(t-\tilde{\tau}) & \sin z_3(t-\tilde{\tau}) & 0\\ -\sin z_3(t-\tilde{\tau}) & \cos z_3(t-\tilde{\tau}) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} x(t-\tau_b)-z_1(t-\tilde{\tau})\\ y(t-\tau_b)-z_2(t-\tilde{\tau})\\ \theta(t-\tau_b)-z_3(t-\tilde{\tau})\end{bmatrix}, \tag{12}$$

where $\tilde{\tau} := \tilde{\tau}_f + \tilde{\tau}_b$ represents the sum of the modeled forward and backward network-induced delays. Recall that the time-delays are assumed to be known or, in other words, modeled perfectly, i.e. $\tilde{\tau}_f = \tau_f$ and $\tilde{\tau}_b = \tau_b$, which yields $\tilde{\tau} = \tau$.

Given the error coordinates (12), the correcting term $\nu(t)$ is proposed as follows:

$$\begin{aligned}\nu_x(t) &= -K_x p_{1_e}(t)\cos z_3(t) + K_y p_{2_e}(t)\sin z_3(t),\\ \nu_y(t) &= -K_x p_{1_e}(t)\sin z_3(t) - K_y p_{2_e}(t)\cos z_3(t),\\ \nu_\theta(t) &= -K_\theta \sin p_{3_e}(t),\end{aligned} \tag{13}$$

where K_x, K_y and K_θ are the correcting term gains.

The block diagram representation of the proposed control scheme is depicted in Figure 2, and shows that the state estimator's output constitutes the controller's input. The tracking control law (8) will now make use of the estimated error coordinates (11) and will be given by,

$$\begin{aligned}v(t) &= v_r(t) + c_2 z_{1_e}(t) - c_3\omega_r(t) z_{2_e}(t),\\ \omega(t) &= \omega_r(t) + c_1 \sin z_{3_e}(t).\end{aligned} \tag{14}$$

Remark 1. Due to the input time-delay τ_f, the control action applied to the robot in (9) is given by:

$$\begin{aligned}v(t-\tau_f) &= v_r(t-\tau_f) + c_2 z_{1_e}(t-\tau_f) - c_3\omega_r(t-\tau_f) z_{2_e}(t-\tau_f),\\ \omega(t-\tau_f) &= \omega_r(t-\tau_f) + c_1 \sin z_{3_e}(t-\tau_f).\end{aligned}$$

The resulting control action already hints at how we would like the system to behave. Intuitively, the robot state $q(t)$ should track the delayed reference trajectory $q_r(t-\tau_f)$. This will be examined in detail during the stability analysis in Section 3.2.

3.2 Stability Analysis

The control objectives may now be defined as follows:

- $q(t) \to q_r(t-\tau_f)$, the system states converge to the reference trajectory delayed by τ_f;
- $z(t) \to q(t+\tau_f)$, the state estimator anticipates the system by τ_f;
- $z(t) \to q_r(t)$, the state estimator converges to the reference trajectory.

Considering these control objectives and taking into account Remark 1, the following control goal can now be formulated:

Given the unicycle-type mobile robot (9) subject to a network induced delay $\tau = \tau_f + \tau_b$, the state estimator (10), (12)-(13), and the control law (11), (14), the robot should track a delayed version $q_r(t-\tau_f)$ of the reference trajectory.

In order to meet this control goal we aim to prove the stability of the equilibrium point $(z_e, p_e) = (z_{1_e}, z_{2_e}, z_{3_e}, p_{1_e}, p_{2_e}, p_{3_e}) = 0$ of the closed-loop system (9)-(14).

Consider the following error coordinate definitions: $\xi_1 = [z_{1_e}\ z_{2_e}\ p_{1_e}\ p_{2_e}]^T$ and $\xi_2 = [z_{3_e}\ p_{3_e}]^T$, with z_{i_e}, p_{i_e}, $i = 1, 2, 3$, defined in (11) and (12), respectively. Using these definitions, the resulting closed-loop error dynamics can be rearranged in the following form:

$$\dot{\xi}_1(t) = A_1(t, t-\tau)\xi_1(t) + A_2\xi_1(t-\tau) + g(t, \xi_{1_t}, \xi_{2_t}), \tag{15}$$

$$\dot{\xi}_2(t) = f_2(t, \xi_{2_t}), \tag{16}$$

where ξ_{i_t}, $i = 1, 2$, is an element of the Banach space $C(n) = C([-\tau, 0], R^n)$ and is defined by the formula $\xi_{i_t}(s) = \xi_{i_t}(t+s)$ for $s \in [-\tau, 0]$. By means of ξ_{i_t} it is possible to represent a state ξ_i of the system throughout the interval $t \in [t-\tau, t]$.

The matrices and functions defining the right-hand side in (15)-(16) are given by

$$A_1(t, t-\tau) = \begin{bmatrix} -c_2 & (1+c_3)\omega_r(t) & K_x & 0 \\ -\omega_r(t) & 0 & 0 & K_y \\ 0 & 0 & 0 & \omega_r(t-\tau) \\ 0 & 0 & -\omega_r(t-\tau) & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0\,0 & 0 & 0 \\ 0\,0 & 0 & 0 \\ 0\,0 & K_x & 0 \\ 0\,0 & 0 & K_y \end{bmatrix},$$

$$g(t, \xi_{1_t}, \xi_{2_t}) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \\ 0 & g_{32} \\ 0 & g_{42} \end{bmatrix} \xi_2(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ h_{31} & h_{32} \\ h_{41} & h_{42} \end{bmatrix} \xi_2(t-\tau),$$

$$f_2(t, \xi_{2_t}) = \begin{bmatrix} -c_1 \sin z_{3_e}(t) + K_\theta \sin p_{3_e}(t) \\ K_\theta \sin p_{3_e}(t-\tau) \end{bmatrix}, \tag{17}$$

with

$$g_{11} = c_1 z_{2_e}(t) \int_0^1 \cos(s z_{3_e}(t)) ds - v_r(t) \int_0^1 \sin(s z_{3_e}(t)) ds,$$

$$g_{12} = -K_\theta z_{2_e}(t) \int_0^1 \cos(s p_{3_e}(t)) ds,$$

$$g_{21} = (v_r(t) - c_1 z_{1_e}(t)) \int_0^1 \cos(s z_{3_e}(t)) ds,$$

$$g_{22} = K_\theta z_{1_e}(t) \int_0^1 \cos(s p_{3_e}(t)) ds,$$

$$g_{32} = -(v_r(t-\tau) + c_2 z_{1_e}(t-\tau) - c_3 \omega_r(t-\tau) z_{2_e}(t-\tau)) \int_0^1 \sin(s p_{3_e}(t)) ds,$$

$$g_{42} = (v_r(t-\tau) + c_2 z_{1_e}(t-\tau) - c_3 \omega_r(t-\tau) z_{2_e}(t-\tau)) \int_0^1 \cos(s p_{3_e}(t)) ds,$$

$$h_{31} = c_1 p_{2_e}(t) \int_0^1 \cos(s z_{3_e}(t-\tau)) ds,$$

$$h_{32} = -K_\theta p_{2_e}(t) \int_0^1 \cos(s p_{3_e}(t-\tau)) ds,$$

$$h_{41} = -c_1 p_{1_e}(t) \int_0^1 \cos(s z_{3_e}(t-\tau)) ds,$$

$$h_{42} = K_\theta p_{1_e}(t) \int_0^1 \cos(s p_{3_e}(t-\tau)) ds.$$

The definition of a persistently exciting (PE) signal will be required in order to formulate a stability result for the system (15)-(17).

Definition 1. *A continuous function $\omega : \mathbb{R}^+ \to \mathbb{R}$ is said to be persistently exciting (PE) if $\omega(t)$ is bounded, Lipschitz, and constants $\delta_c > 0$ and $\epsilon > 0$ exist such that,*

$$\forall t \geq 0,\ \exists s :\ t - \delta_c \leq s \leq t \text{ such that } |\omega(s)| \geq \epsilon.$$

The following theorem formulates sufficient conditions under which $(z_e, p_e) = 0$ is a locally uniformly asymptotically stable equilibrium point of (15)-(17).

Theorem 1. *Consider the posture kinematic model of a unicycle-type mobile robot subject to a constant and known input time-delay τ_f, as given by (9). The robot's reference position is given by $(x_r(t), y_r(t))$, whereas its reference orientation $\theta_r(t)$ is given by (3). Additionally, consider the tracking controller as given in (14), with the feedforward terms $v_r(t)$ and $\omega_r(t)$ defined in (4) and (5), respectively, and the feedback part based on the error between the reference trajectory and an estimate of the state, as given in (11). Moreover, consider the state estimator (10), (12)-(13), which uses state measurements delayed by a constant and known output time-delay τ_b. If the following conditions are satisfied:*

- *$\omega_r(t)$ is bounded and persistently exciting;*
- *the tracking gains satisfy $c_1, c_2 > 0$, $c_3 > -1$;*
- *the correcting term gains satisfy $K_x = K_y = K < 0$, $K_\theta < 0$;*
- *the time-delay $\tau = \tau_b + \tau_f$ belongs to the interval $0 \leq \tau < \tau_{\max}$, with*

$$\tau_{\max} = \min\left\{\frac{-1}{\sqrt{p}K_\theta}, \frac{-1}{\sqrt{p}(K - \bar{\omega}_r)}\right\}, \tag{18}$$

where $p > 1$ and $\bar{\omega}_r = \sup_{t\in\mathbb{R}} |\omega_r(t)|$,

then, $(z_e, p_e) = 0$ is a locally uniformly asymptotically stable equilibrium point of the closed-loop error dynamics (15)-(17). In other words, $z(t) \to q(t + \tau_f)$ as $t \to \infty$ (the state estimator anticipates the state by τ_f) and $q(t) \to q_r(t - \tau_f)$ as $t \to \infty$ (the system tracks the reference trajectory delayed by τ_f).

Proof. For brevity only a sketch of the proof is presented. Recall the closed-loop error dynamics (15)-(17) and note that systems (15)-(16) form a cascade consisting of a nonlinear delayed system $\dot{\xi}_2(t) = f_2(t, \xi_{2_t})$, interconnected to a linear time-varying delayed system $\dot{\xi}_1(t) = A_1(t, t-\tau)\xi_1(t) + A_2\xi_1(t-\tau)$ by means of a nonlinear delayed coupling $g(t, \xi_{1_t}, \xi_{2_t})$.

Based on Theorem 2 in [18], local uniform asymptotic stability of the equilibrium point $(z_e, p_e) = 0$ of the predictor's closed-loop error dynamics may be established if the following conditions are satisfied,

- the coupling term $g(t, \xi_{1_t}, \xi_{2_t})$ vanishes when $\xi_{2_t} \to 0$, i.e. $g(t, \xi_{1_t}, 0) = 0$;
- the unperturbed subsystem $\dot{\xi}_1(t) = A_1(t, t-\tau)\xi_1(t) + A_2\xi_1(t-\tau)$ is uniformly asymptotically stable;
- subsystem $\dot{\xi}_2(t) = f_2(t, \xi_{2_t})$ is locally uniformly asymptotically stable.

Let us now check the validity of these three conditions. Firstly, given $g(t, \xi_{1_t}, \xi_{2_t})$ as defined in (17), it immediately follows that as $\xi_{2_t} \to 0$, the coupling term vanishes and thus the first condition is satisfied.

Regarding the second condition, subsystem $\dot{\xi}_1(t) = A_1(t, t-\tau)\xi_1(t) + A_2\xi_1(t-\tau)$ can be represented by a cascade itself. Using a similar reasoning as for the original cascade (15)-(17), the subsystem's uniform asymptotic stability is concluded if the time-delay satisfies the following condition:

$$\tau < \frac{-1}{\sqrt{p}(K - \bar{\omega}_r)}, \tag{19}$$

and the requirements for c_2, c_3, K_x and K_y stated in Theorem 1 are satisfied.

In order to check the third condition, subsystem $\dot{\xi}_2(t)$ is first linearized around the equilibrium point $z_{3_e} = p_{3_e} = 0$. The uniform asymptotic stability of the linearized subsystem is ensured for

$$\tau < \frac{-1}{\sqrt{p}K_\theta}, \tag{20}$$

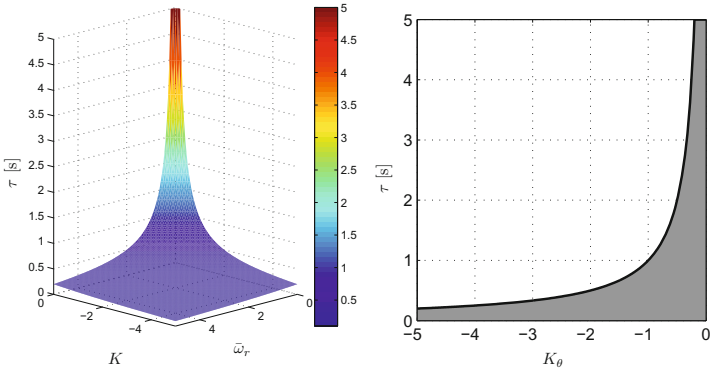


Fig. 3. Maximum allowable time-delay τ for conditions (19) (left) and (20) (right), respectively. To better illustrate the relationship between the gains and the time-delay, the maximum allowable delay in the plot has been cut off at 10 s.

provided c_1 and K_θ satisfy the conditions in Theorem 1. Note that the satisfaction of (19) and (20) is guaranteed by satisfying condition (18) in the theorem.

The local uniform asymptotic stability of the equilibrium point $(z_e, p_e) = 0$ of the closed-loop error dynamics (15)-(17) is then concluded. This means that the state estimator converges to the reference trajectory, since $z_e(t) \to 0$ as $t \to \infty$, or in other words, $z(t) \to q_r(t)$ as $t \to \infty$. It also implies that the state estimator anticipates the system, due to the fact that $p_e(t) \to 0$ as $t \to \infty$, i.e. $z(t) \to q(t + \tau_f)$ as $t \to \infty$. From the previous relations it directly follows that $q(t) \to q_r(t - \tau_f)$ as $t \to \infty$, which means that the unicycle-type mobile robot, subject to a network-induced delay τ, tracks the reference trajectory delayed by τ_f. This completes the sketch of the proof.

The relationship between the allowable time-delay τ and the control parameters for conditions (19) and (20) is shown in Figure 3. The left plot shows the maximum allowable time-delay satisfying (19) considering $p = 1$ and different values for the correcting term gain K and for the maximum reference rotational velocity $\bar{\omega}_r$. Depicted in the right plot is the maximum allowable time-delay satisfying (20) given $p = 1$ and different values for the correcting term gain K_θ. Note that, for both conditions, there exist choices for the correcting term gains such that it becomes possible to accommodate arbitrarily large time-delays ($K \to 0$ and $\bar{\omega}_r \to 0$ for (19) and $K_\theta \to 0$ for (20)). A word of caution is in order, however, since the plots also show that there is a performance tradeoff arising from the relationship between the allowable time-delay, the correcting term gains and the tracking behavior. Namely, decreasing the correcting term gains allows higher robustness for delays at the expense of slower convergence.

4 Experimental Results

Two equivalent multi-robot platforms have been developed at the Eindhoven University of Technology (TU/e), The Netherlands, and at the Tokyo Metropolitan University (TMU), Japan. The proposed remote tracking controller is implemented in such a way that a mobile robot located at TU/e is controlled from TMU and viceversa.

Fig. 4. Experimental setups at TU/e (left) and TMU (right).

4.1 Experimental Platform Description

The experimental platforms' design objectives encompass cost, reliability and flexibility. The corresponding hardware and software choices, together with the implementation of the setup at TU/e are discussed in greater detail in [2], [3] (cf. Figure 4). The setup has already been used to implement cooperation, coordination, collision avoidance and servo vision algorithms, see e.g. [10], [9]. The platform at TMU has similar characteristics, only differing from the one at TU/e in size and in the vision calibration algorithm used. Listed below are some of the experimental platforms' components and characteristics:

- **Mobile Robot.** The unicycle selected is the e-puck mobile robot [11], whose wheels are driven by stepper motors that receive velocity control commands over a BlueTooth connection.
- **Vision.** Each robot is fitted with a fiducial marker of 7 by 7 cm, collected by an industrial FireWire camera, interpreted in the program reacTIVision [17], and calibrated by means of a global transformation (TU/e) or a grid (TMU).
- **Driving Area.** The driving area is of 175×128 cm for TU/e and 100×50 cm for TMU, and is determined based on the required accuracy, the camera lens, and the height at which the camera is positioned.
- **Software.** The e-puck robots and reacTIVision's data stream can be managed in Matlab script, C, or Python. In this work, the controller implementation and signal processing is carried out in Python [16].
- **Bandwidth and Sampling Rate.** Using vision as the localization technique diminishes the system's bandwidth and results in a sampling rate of 25 Hz.

4.2 Data Exchange over the Internet

Due to its widespread availability and low cost, the Internet is chosen as the communication channel to exchange data between TU/e and TMU. Details about the data exchange implementation are given below:

- **Data Exchange.** A Virtual Private Network (VPN) is established between TU/e and TMU in order to implement a reliable and secure data exchange.
- **Socket Configuration.** Data is exchanged between TU/e and TMU as soon as it becomes available using non-blocking Transmission Control Protocol (TCP) sockets running the Internet Protocol (IP). The system's low bandwidth allows for the use of TCP, guaranteeing reliable and orderly data delivery.
- **Data Payload.** The variables exchanged are the following: the current time instant and control signals are sent from the control side to the system side, and the position and orientation measurements are sent from the system side to the controller side.

4.3 Implementation Issues

One of the main implementation issues of the proposed remote tracking controller is the accurate modeling and characterization of the time-delay induced by the communication channel. The use of predictor-like schemes is often discouraged because of their sensitivity to delay model mismatches [5], especially when considering nonlinear systems and a communication channel such as the Internet. To this end, two different methods that ease the implementation of the proposed compensation strategy are suggested. Their objective it to provide an accurate estimate $\tilde{\tau}$ of the real delay τ in practice. The two delay estimation methods studied are explained below:

- **Delay Measurement.** The round trip communication delay between TU/e and TMU (and vice versa) has been measured during different times of the day, for variable amounts of time ranging from 2 min to 10 min, and for a total time of around 60 min. The mean delay value is approximately 265 ms for both cases (267.4917 ms TU/e→TMU, 269.5307 ms TMU→TU/e). Occurrences of delays greater than 300 ms where of 0.27% for TU/e→TMU and 0.34% for TMU→TU/e. Thus, the round trip time-delay is fairly constant and can be modeled with enough accuracy even if the Internet is considered as the communication channel.
- **Signal Bouncing.** The estimator's output may be sent together with the control signals to the mobile robot, and then sent back to the controller without being modified. By using the communication channel itself to delay the estimator's output, modeling the time-delay is no longer necessary (cf. Figure 5).

4.4 Experiments

In the first experiment, a mobile robot at TMU is controlled from TU/e. The reference trajectory is a lemniscate with center at $(0.5\,\mathrm{m}, 0.25\,\mathrm{m})$, a length and width of $0.2\,\mathrm{m}$, and an angular velocity multiplier of $0.2\,\mathrm{m/s}$. The scenario repeats in the second experiment, where a sinusoid with origin at $(0.1\,\mathrm{m}, 0.25\,\mathrm{m})$, an amplitude of $0.15\,\mathrm{m}$, an angular frequency of $0.3\,\mathrm{rad/s}$, and a translational velocity multiplier of $0.01\,\mathrm{m/s}$ constitutes the reference.

The system's initial condition is $q(0) = [0.3235\,\mathrm{m}\ 0.1882\,\mathrm{m}\ 0.2851\,\mathrm{rad}]^T$ for the first experiment and $q(0) = [0.0225\,\mathrm{m}\ 0.1821\,\mathrm{m}\ 0.3916\,\mathrm{rad}]^T$ for the second one. In both cases the estimator's initial condition is set to $z(0) = [0\ 0\ 0]^T$, the controller gains to $c_1 = 1.0$, $c_2 = c_3 = 2.0$ and the correcting term gains to $K_x = K_y = K_\theta = -0.6$. The sampling rate is 25 Hz and the experiments' duration is 60 s and 120 s, respectively.

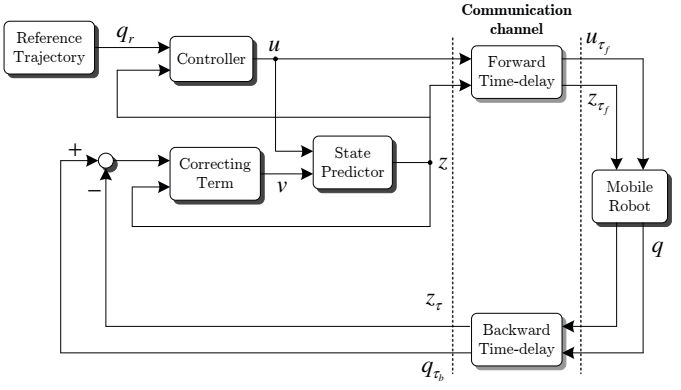


Fig. 5. Remote tracking control strategy block diagram representation using signal bouncing (no time-delay models necessary).

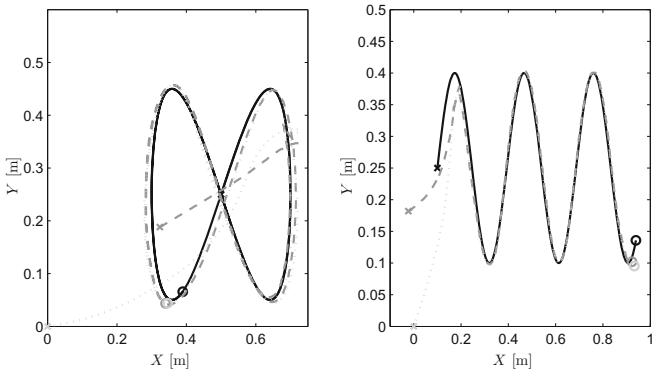


Fig. 6. Reference (black solid line), robot (gray dashed line) and predictor behavior (light gray dotted line) in the X-Y plane for two different trajectories of a robot in Japan controlled from the Netherlands.

The round trip time-delay is modeled as 265 ms based on measurements, although the estimator's output is in fact delayed 280 ms since only delay models which are multiples of 40 ms are allowed due to the setup's sampling time.

The experimental results are shown in Figure 6 and 7 for both experiments. The plots in Figure 6 show the reference (black solid line), robot (gray dashed line) and predictor (light gray dotted line) trajectories in the X-Y plane, with their initial and final position marked with a cross and a circle, respectively. The plots in Figure 7 show the evolution of the error coordinates $z_e(t) = [z_{1_e}(t)\ z_{2_e}(t)\ z_{3_e}(t)]^T$ (black) and $p_e(t) = [p_{1_e}(t)\ p_{2_e}(t)\ p_{3_e}(t)]^T$ (gray) for the first and second experiments (top and bottom, respectively). The error coordinates practically converge to zero even in the presence of a small delay model mismatch and considering a time-varying communication channel. The behavior of the proposed remote tracking controller is consistent with the stability analysis presented and the tracking performance of the robot can be ensured even in the presence of a network-induced delay.

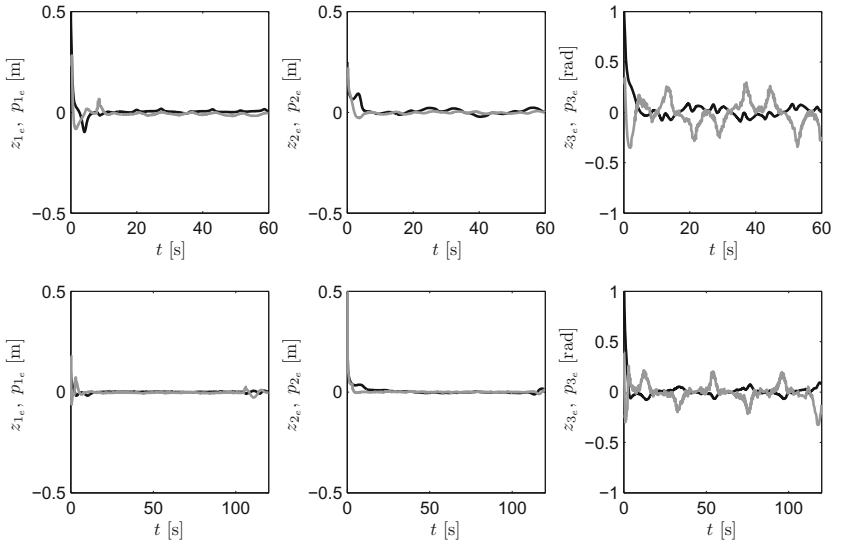


Fig. 7. Practical convergence of the error coordinates $z_e(t)$ (black) and $p_e(t)$ (gray) for the first and second experiments (top and bottom, respectively).

5 Discussion

This paper considers the tracking control problem for a unicycle-type mobile robot controlled over a two-channel communication network which induces time-delays. A tracking control and a state estimator that guarantee tracking a delayed reference trajectory has been proposed. Moreover, a stability analysis showing that the tracking and estimation error dynamics are locally uniformly asymptotically stable has been presented. In addition, experiments validate the effectiveness of the proposed approach and show that the estimator-control strategy can withstand small delay model mismatches and delay variations.

Acknowledgements. The authors are grateful to Prof. Wim Michiels at the Katholieke Universiteit Leuven (KUL) for several useful discussions and communications. This work has been supported by the Mexican Council of Science and Technology (CONACYT), the Mexican Ministry of Education(SEP), the Graduate School of Science and Engineering at TMU, and the Japan Society for the Promotion of Science (JSPS) through Grant-in-Aid for Scientific Research (No. 20560424).

References

1. Brockett, R.: Asymptotic stability and feedback stabilization. In: Differential geometric control theory, pp. 181–191. Birkhäuser, Basel (1983)
2. van den Broek, T.: Formation Control of Unicycle Mobile Robots: Theory and Experiments. Master's thesis, Eindhoven University of Technology (2008)

3. van den Broek, T., van de Wouw, N., Nijmeijer, H.: Formation control of unicycle mobile robots: a virtual structure approach. In: Proceedings of the combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC), pp. 3264–3269 (2009)
4. Heemels, W.P.M.H., van de Wouw, N.: Stability and stabilization of networked control systems. In: Bemporad, A., Heemels, M., Johansson, M. (eds.) Networked Control Systems. Lecture Notes in Control and Information Sciences, vol. 406, pp. 203–253. Springer, Heidelberg (2010)
5. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: An historial survey. Automatica 42(12), 2035–2057 (2006)
6. Jakubiak, J., Lefeber, E., Tchón, K., Nijmeijer, H.: Two observer-based tracking algorithms for a unicycle mobile robot. International Journal of Applied Mathematics and Computer Science 12(4), 513–522 (2002)
7. Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A stable tracking control method for an autonomous mobile robot. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 384–389
8. Kojima, K., Oguchi, T., Alvarez-Aguirre, A., Nijmeijer, H.: Predictor-based tracking control of a mobile robot with time-delays. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS), pp. 167–172 (2010)
9. Kostic, D., Adinandra, S., Caarls, J., Nijmeijer, H.: Collision-free motion coordination of unicycle multi-agent systems. In: Proceedings of the American Control Conference (ACC), pp. 3186–3191 (2010)
10. Kostic, D., Adinandra, S., Caarls, J., van de Wouw, N., Nijmeijer, H.: Collision-free tracking control of unicycle mobile robots. In: Proceedings of the combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC), pp. 5667–5672 (2009)
11. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, pp. 59–65 (2009)
12. Oguchi, T., Nijmeijer, H.: Control of nonlinear systems with time-delay using state prediction based on synchronization. In: Proceedings of the EUROMECH Nonlinear Dynamics Conference (ENOC), pp. 1150–1156 (2005)
13. Oguchi, T., Nijmeijer, H.: Prediction of chaotic behavior. IEEE Trans. on Circuits and Systems I 52(11), 2464–2472 (2005)
14. Oguchi, T., Nijmeijer, H.: Anticipating synchronization of nonlinear systems with uncertainties. In: Proceedings of the 6th IFAC Workschop on Time-Delay Systems (2006)
15. Panteley, E., Lefeber, E., Loría, A., Nijmeijer, H.: Exponential tracking control of a mobile car using a cascaded approach. In: Proceedings of the IFAC Workshop on Motion Control (1998)
16. Python. Python Programming Language (2010), `http://www.python.org`
17. reacTIVision. reacTIVision 1.4: A toolkit for tangible multi-touch surfaces (2010), `http://reactivision.sourceforge.net/`
18. Sedova, N.: Local and semiglobal stabilization in a cascade with delays. Automation and Remote Control 69(6), 968–979 (2008)
19. Tipsuwan, Y., Chow, M.Y.: Control methodologies in networked control systems. Control Engineering Practice 11(10), 1099–1111 (2003)
20. Voss, H.U.: Anticipating chaotic synchronization. Physical Review E 61(15), 5115–5119 (2000)

Development of a Low-Pressure Fluidic Servo-Valve for Wearable Haptic Interfaces and Lightweight Robotic Systems

Michele Folgheraiter, Mathias Jordan, Luis M. Vaca Benitez, Felix Grimminger, Steffen Schmidt, Jan Albiez, and Frank Kirchner

German Research Center for Artificial Intelligence (DFKI)
Robotics Innovation Center, Robert-Hooke-Strasse 5 D-28359 Bremen, Germany
{michele.folgheraiter,mathias.jordan,luis.vaca_benitez,
felix.grimminger,steffen.schmidt,jan.albiez,
frank.kirchner}@dfki.de
http://robotik.dfki-bremen.de/

Abstract. This document presents a low-pressure servo-valve specifically designed for haptic interfaces and lightweight robotic applications. The device is able to work with hydraulic and pneumatic fluidic sources, operating within a pressure range of $(0 - 50 \cdot 10^5 Pa)$. All sensors and electronics were integrated inside the body of the valve, reducing the need for external circuits. Positioning repeatability as well as the capability to fine modulate the hydraulic flow were measured and verified. Furthermore, the static and dynamic behavior of the valve were evaluated for different working conditions, and a non-linear model identified using a recursive Hammerstein-Wiener parameter adaptation algorithm.

Keywords: Proportional valve, Hydraulic valve, Pneumatic valve, Mechatronics, Pressure control, Servo-mechanism.

1 Introduction

Hydraulically actuated robotic systems generally operate at pressures greater than $200 \cdot 10^5 Pa$ [1], [2]. This is mainly due to the fact that it is convenient to increase the force/weight ratio of the actuation system by increasing its operational pressure [3]. If from one side increasing the pressure brings advantages, from the other side it could represent a limitation. First of all, the hydraulic components need to be designed to resist the high forces generated by the fluid pressure; this requires therefore to employ thick and heavy materials for pipes and actuators. Secondly, the usage of high pressure could also cause a dangerous situation for the operators that are in the proximity of the robot. The safety issue is even more critical if the robot, in our case an exoskeleton, is strictly coupled with the human being [4]. Any failure in the hydraulic system could seriously harm the user.

One of the main goals of the VI-Bot project, under development at DFKI Bremen (Robotics Innovation Center), is to design an intrinsically safe, wearable arm exoskeleton for Tele-Robotics applications [5,6,7,8]. As requirements the haptic interface should: enable the operator to control complex robotics systems in an intuitive

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 239–252.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

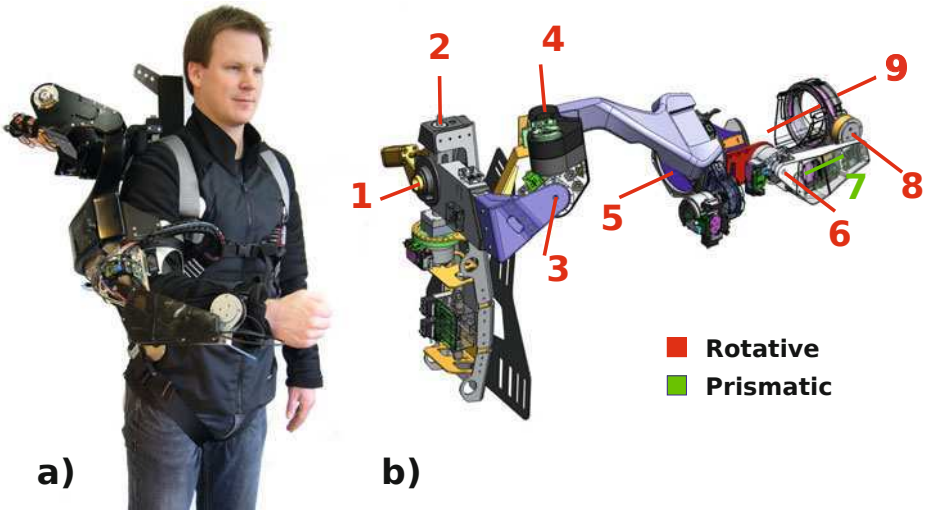


Fig. 1. a) A user wearing the exoskeleton b) The Exoskeleton kinematics equipped with 9 DOF.

way, implement a multi-points haptic feedback to increase the immersion into the work scenario, be lightweight and adaptable to different users, and integrate different levels of safe mechanisms. Furthermore, the kinematics architecture of the system should be designed in order to constrain as less as possible the natural arm movements and workspace.

To achieve these goals and at the same time to reduce the complexity of the system (number of required DOF), it is necessary to keep the exoskeleton's joints close to the human arm, "ideally overlap them with the human articulations" in order to avoid parallel kinematic loops. It turns out that here the necessity is crucial to have a compact, light, highly dynamic actuation. The advantages of using hydraulically actuators to operate the exoskeleton's joints, if directly compared with classical DC motors, are represented by their high force/weight ratio, the possibility to use the axes of the actuator as rotational/prismatic axes of the robotic system, and their back-drivability. Furthermore, with a proper hydraulic supply and a precise fluid regulation, strength and high dynamic range can be achieved [9], [10],[11].

Figure 1 shows the actual version of our haptic interface. In total there are 7 actuated joints: 5 located in the shoulder/upper-arm and 2 in the forearm. Two additional passive joints (8 and 9) allow the wrist supination-pronation. All active joints are hydraulically actuated, valves, sensors and electronics are intended to be mounted directly in proximity of the actuators, this in order to reduce the amount of cables and pipes needed. The hydraulic pump and the primary power supply are located outside the exoskeleton to avoid additional weight to the system.

A central element within the hydraulic system is represented by the proportional servo-valve. On the market there are plenty of proportional (4/3) hydraulic valves, the problem is that most of them are intended to work with high pressure and therefore do not fulfill our needs. According to the authors' knowledge, the smallest, lightweight and dynamically performing valve on the market is currently sold by *MOOG Inc.* company

[12]. The device weights only $92g$, has an hysteresis for the flow characteristics $< 3\%$, and a 90^o phase-lag $>$ 250Hz. Unfortunately, the device is intended to work only with pressure in the range of $160 - 250 \cdot 10^5 Pa$.

Therefore we started to look at the pneumatic components, that generally are light and designed for low pressure (up to $10 \cdot 10^5 Pa$). We adapted them to work with liquids (oil), adding a precise actuation and proper sensory features.

This document is organized as follows: next section describes the experimental setup employed to measure the repeatability and flow-position characteristic of the valve, section 3 presents the static and dynamic models, section 4 introduces a strategy to regulate position and velocity of the valve's spool, section 5 presents a first fully integrated prototype. Finally, section 6 draws the conclusions and future developments of this work.

2 Experimental Setup and Testing

In this section, the testbed developed to evaluate the performance of the servo valve and first experimental results characterizing the valve are presented.

The experimental setup (Fig. 2) consists of the core parts of a commercial pneumatic valve (*Numatics Inc.* series Micro-Air), a stepper motor, a gear pump providing pressure supply between $0 - 30 \cdot 10^5 Pa$, a flow-meter, a pressure sensor, and an electronic board equipped with a STM32 μController (ST-Microelectronics Inc.).

The drive system of the valve is a 3.3 V DC stepper motor from Nanotec working in full-step mode, i.e. 18 degree/step and with an holding torque of $1.6 \cdot 10^{-3} Nm$. The rotor of the stepper motor is a lead screw, driving a cylinder and thus converting the rotational motor movement into a translation. The drive is attached to the valve spool via a permanent magnet, while the actual valve positions are determined using an inductive sensor from *Bahlluff Inc.* by tracking a steel target connected to the extended spool axis. Control of the testbed and its components is performed by a STM32 μController (series F103VE), programmed with a special toolchain consisting of Matlab/Simulink and Rapidstm32 Blockset, Real-Time-Workshop and Keil Microvision μVision. The scheme in Figure 3 sketches the general dependencies of the testbed, were A and B are the connections of the valve to the actuator chambers, P_s and T are the pressure supply line and the tank lines of the gear pump respectively, while the red lines represent the communication between the μController and the experimental setup through sensors and actors.

Via different tests we measured:

1. repeatability of the spool movement with respect to a certain input to the drive system,
2. flow through the valve with respect to spool position and the pressure drop over the valve.

Thus we focus on showing exemplary results of these two features. Position control of the fluidic valve is presented instead in section 4.

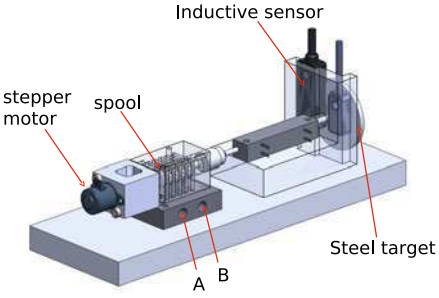


Fig. 2. CAD Model of the valve testbed.

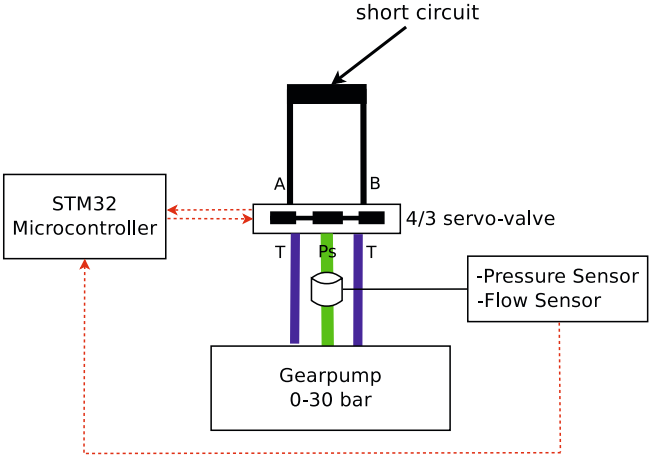


Fig. 3. Scheme of experimental test setup.

2.1 Performance of the Drive System

In the following, the ability of the valve's drive system to exactly position the spool over a long time interval is investigated. Therefore, the repeatability of valve movement is tested by applying a special open-loop control sequence. In the adjustment phase, the spool is driven to the center position, while in the second phase the stepper motor is governed to move the spool 70 steps out of the zero position in both directions, which covers almost the whole working range of the developed prototype. Experiments took place under influence of pressure with $P_s = 15 \cdot 10^5 P$. Note that in this experiment the connectors A and B are connected in short circuit, while the speed of the motor is adjusted to its maximum of 1000 steps per second.

Figure 4 shows the open loop response of the valve prototype to the applied control sequence. The duration of the test was 20 minutes, while data was acquired for a time interval of 20 seconds every five minutes.

As we can determine from Figure 4, the position response of the drive system to the reference signal is matching for the whole experiment. Differences for the travel of the

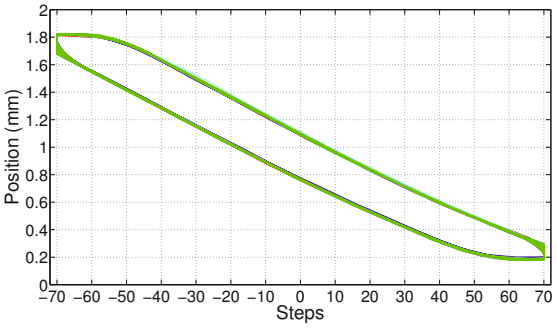


Fig. 4. Position of spool while performing 70 steps out of center in both directions ($P_s = 15 \cdot 10^5 Pa$).

spool, comparing the movement out of the center in both directions, might be caused by friction effects. The hysteresis is due to a backlash of approximately 10 steps (i.e. 0.1mm) between the spindle of the stepper motor and the thread of the cylinder moving the spool, when the motor changes its direction of rotation. This is due to a mechanical behavior and can only be reduced through higher precision in the manufacturing process of these two elements, or via a proper control action.

Alternatively a special ball-screw could be used, which provides nearly zero backlash. Overall the valve shows remarkable repeatability for an open loop control of spool position under influence of pressure for a long time interval.

2.2 Flow Characteristics of Valve

In this section, the resulting static flow characteristics of the valve is presented.

To measure the flow for a fixed pressure drop over the valve, the connectors A and B are again linked in a short circuit, causing the valve to work against the pressure in the tank line T. A flowmeter from *Biotech* (series VZS-007-ALU) is connected to the pressure supply line providing a resolution of 900 pulses/L at a maximum flow of 5 L/min to the digital I/O of the μController. Due to the fact that flow needs a certain time to become constant, the valve is driven at a very low speed of 1 step/2s through the overall working period. Simultaneously, the average flow is calculated and sampled each second.

Fig.5 shows the static flow characteristics of the hydraulic valve for a constant pressure drop of $\Delta P = 29 \cdot 10^5 Pa$.

From Figure 5 we can determine that the valve has a large deadband of approximatively 1.25 mm (between 0.45 mm and 1.7 mm) where it is completely closed. This is caused by the inner structure of the valve which is adapted from a pneumatic solenoid valve and can neither be influenced nor changed at the moment. Therefore, this deadband has to be taken into account in the position control structure presented in section 4.

Having a closer look at the slopes of the flow characteristics, an area of about $0.2mm$ can be attested where flow regulation should be possible. Driving the spool to the

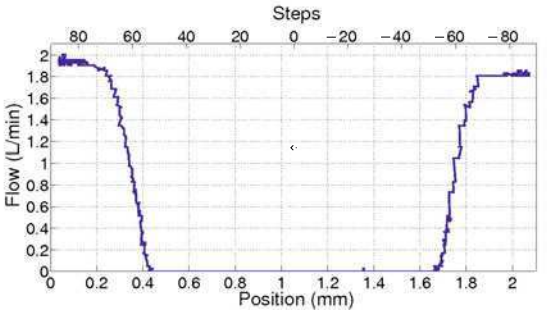


Fig. 5. Flow-characteristics of the valve with respect to the spool position ($\Delta P = 29 \cdot 10^5 Pa$).

extremes causes flow saturation ($U_a \leq 0.2\,mm$ and $1.9\,mm \leq U_a$), thus the overall working range of the valve prototype can be defined to

$$0.15\,mm \leq U_a \leq 1.95\,mm,$$

whereas the amplitude of the flow depends on the pressure drop over the valve.

The exemplary results shown in this section certify a good repeatability to the drive system of the developed prototype. Furthermore, the static flow characteristic promises a possible flow regulation within the defined working range.

3 Valve Model

In order to formalize a complete model of the proportional valve, it was divided in a static and a dynamic part. The static part, describing the position-flow-pressure relationship in the valve, is reported in the next section. The dynamic (electro-mechanical) part is described in section 3.2.

3.1 Static Model

The position-flow-pressure relationship of a hydraulic proportional valve can be described by the following equation:

$$Q = k_v * x_v * \sqrt{|\triangle P|} * sign(\triangle P) \tag{1}$$

where Q is the flow [L/min], x_v the position of the spool in the valve [m], $\triangle P$ the pressure difference in the valves chambers [Pa],and k_v the flow factor.

From our experimental setup, known (measured) values are $Q, x_v, \triangle P$, so it remained to calculate the factor k_v. So from equation 1 we know that

$$k_v = \frac{Q}{x_v * \sqrt{|\triangle P|} * sign(\triangle P)} \tag{2}$$

The flow factor k_v is defined for liquids as the flow of water with temperature ranging $5-30°C$ through a valve in cubic meters per hour (m^3/h) with a pressure drop of 1 bar.

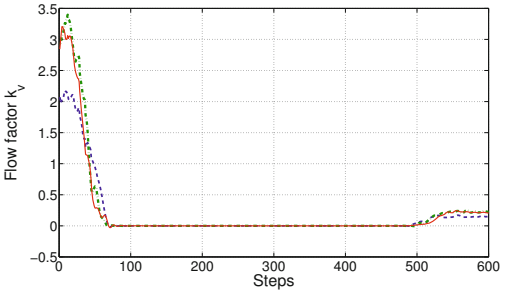


Fig. 6. Values of the flow coefficient k_v for pressure values between 5 and 20 Bar.

The metric equivalent of the flow factor is also in common use and is based on imperial units (mainly used in the United States of America) and is called the flow coefficient c_v. In other words, and explained in a simpler way, the flow factor describes the flow capacity of the valve at fully opened position.

The used pressure signals $\triangle P$ had values between 5 and 20 Bar ($50 \cdot 10^5$ and $200 \cdot 10^5 Pa$), the measured flow signals were filtered with a Butterworth low-pass filter with a band frequency of 3 Hz. Figure 6 shows the values of k_v graphically for all data series, calculated after equation 2.

Now k_v can be estimated as the maximum value of the graph. The factor reached a value of $k_v = 3.398$. With the caculation of the flow factor, the static part of the model is fully defined.

3.2 Dynamic Model

In this section, a non-linear dynamic model that takes into account the electro-mechanical behavior of the stepper motor, the mechanical behavior of the spool and the static/dynamic effects of the frictions present within the system, was identified using a Hammerstein-Wiener structure. The linear part of the model can be considered as an ARX structure, which is well known and described amongst others in [13]. For the input and output blocks, polynomial nonlinearities with order n_l were used:

$$\eta(u) = \sum_{k=2}^{n_l} \beta_k u^k \tag{3}$$

$$\eta(y_{lin}) = \sum_{k=2}^{n_l} \gamma_k y_{lin}^k \tag{4}$$

with y_{lin} as the output of the linear part.

Model parameters were calculated using the *Recursive Least Squares(RLS)* Parameter Adaptation Algorithm (PAA). Using the equation for a Hammerstein-Wiener model's output [14], the structure of the PAA becomes, with $u(t)$ and $y(t)$ being the real input and output signals of the system:

$$\hat{\theta}(t+1) = \theta(t) + F(t+1)\phi(t)\varepsilon^0(t+1) \tag{5}$$

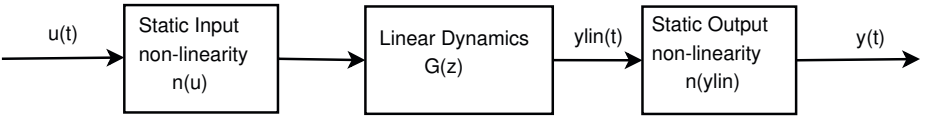


Fig. 7. Block diagram of a non-linear Hammerstein-Wiener model.

with the Adaptation Gain

$$F(t+1) = F(t) - \frac{F(t)\phi(t)\phi(t)^T F(t)}{1 + \phi(t)^T F(t)\phi(t)} \tag{6}$$

and the Prediction Error

$$\varepsilon^0(t+1) = y(t+1) - \hat{\theta}(t)^T \phi(t). \tag{7}$$

$\hat{\theta}$ is the vector of computed parameters, with

$$\hat{\theta}(t)^T = [\hat{a}(t)^T, \hat{b}(t)^T, \hat{\beta}(t)^T, \hat{\gamma}(t)^T], \tag{8}$$

where $\hat{a}(t)^T = [\hat{a}_1(t)...\hat{a}_{n_a}(t)]$ are the parameters of polynomial A with order n_a, $\hat{b}(t)^T = [\hat{b}_1(t)...\hat{b}_{n_b}(t)]$ the parameters of polynomial B with order n_b, $\hat{\beta}(t)^T = [\hat{\beta}_1(t)...\hat{\beta}_{n_l}(t)]$ the parameters of the input non-linearity with order n_l, $\hat{\gamma}(t)^T = [\hat{\gamma}_1(t)... \hat{\gamma}_{n_l}(t)]$ the parameters of the output non-linearity with order n_l.

Furthermore $\phi(t)$ is the Predictor Regressor Vector

$$\phi(t)^T = [-y(t), u(t), u(t)^2, ..., u(t)^{nl}, y_{lin}(t)^2, ..., y_{lin}(t)^{nl}]. \tag{9}$$

The output of the linear part y_{lin} cannot be measured, nevertheless it can be calculated by multiplying the parts of the predictor and the parameters vector corresponding to the linear model:

$$y_{lin}(t+1) = \hat{\theta}\left[1 : n_a + n_b + n_l\right](t)\phi\left[1 : n_a + n_b + n_l\right] \tag{10}$$

Finally, the model output $\hat{y}$ is computed

$$\hat{y}(t+1) = \hat{\theta}(t+1)^T \phi(t) \tag{11}$$

The measured signals used for identification are the current absorbed by the stepper motor i_M as input signal, and the position of the valve's spool x_S in terms of mm as output. In order to characterize the dynamic behavior of the valve we only considered the range of spool positions where the flow can be effectively regulated. As a first step the data was filtered using a bandpass filter allowing frequencies between $0Hz$ and $60Hz$, which does not affect the dynamic range of the model. Different models were identified starting from distinct initial values for the parameter vector, the best data fitting reached an average of 87.49%. The output of the obtained model can be seen in Figure 8, whereas the transfer function of the linear dynamic part is given by:

$$G(z) = \frac{-0.1675z - 0.1039}{z^3 - 0.8512z^2 - 0.1351z - 0.03855} \tag{12}$$

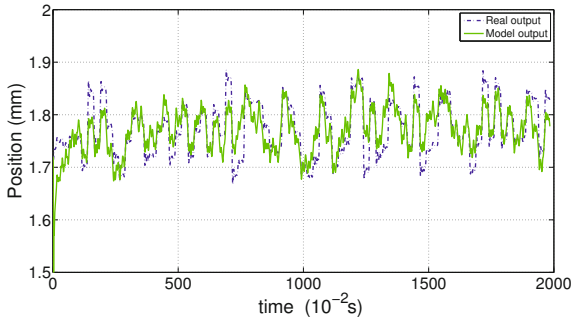


Fig. 8. Comparison of the real and the model output.

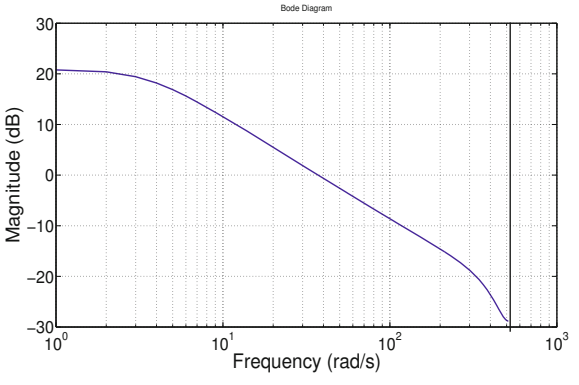


Fig. 9. Magnitude Bode Plot of the linear transfer function.

with the sample time 0.01s. The Bode diagram of G(z) is shown in Figure 9. Finally, the static input non-linearity of the model was identified to:

$$\eta(u) = 0.59831 * u^2 - 0.00047669 * u^3, \tag{13}$$

while the static output non-linearity of the model is given by:

$$\eta(y_{lin}) = 0.030651 * y_{lin}^2 - 0.042948 * y_{lin}^3 \tag{14}$$

4 Valve Control

This section introduces a first approach to control the developed fluidic valve.

To facilitate precise position control of the valve at a high bandwidth, the drive system is to be controlled in closed loop, using the feedback of the inductive sensor to track the desired trajectories given by the μController. Therefore, classical PID-control in combination with a discretization of stepper motor motion is applied.

A basic requirement for position control is the ability of the drive system to run at different speeds.

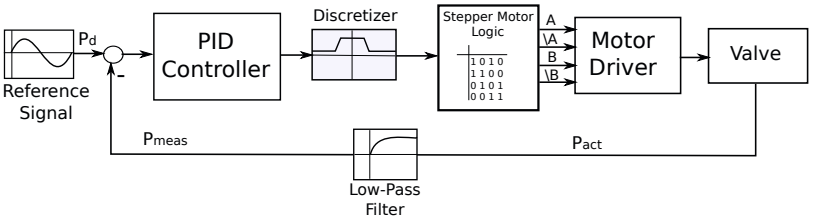


Fig. 10. Scheme of discretized closed loop valve control.

Due to the fact that the stepper motor is somehow a digital drive, which can only have the states *run* or *stop*, a discretizing function is introduced to the control loop. The idea is to vary the number of samples passing between two step commands by defining a variable delay factor τ_d, which causes a step command only every $\tau_d * T_s$, with T_s as basic sampling time of the system.

Thus an increase of τ_d results in a slower motion of the valve spool, because less steps are performed by the drive system in a fixed time interval.

The value of τ_d is set by the control action u of the PID-controller, tracking the desired spool position p_d, as follows

$$\tau_d = \begin{cases} 15 - |u| \cdot 1000 & , \; u \leq 0.013 \\ 1 & , \; u \geq 0.014 \end{cases} \tag{15}$$

Figure 10 shows the resulting scheme of the closed loop control.

Whereas p_{meas} represents the measured and p_d the desired spool position, the discretizing function can be found between the PID-controller and the valve driver.

To reject noise from the position measurement, a 1^{st} order Butterworth low-pass filter is realized via software in the μController. The following Figure 11 shows the modulation of the valve speed through the discretizing function.

To verify the functionality of the discretizing function, the valve is governed to execute a velocity sweep. Starting from $\tau_d = 100$ the delay factor is decreased by an amount of five every 5 ms until the maximum speed at $\tau_d = 1$ is reached. After

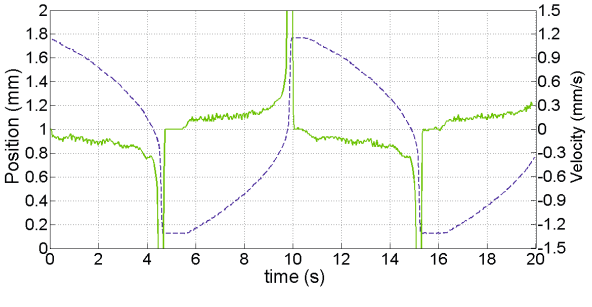


Fig. 11. Speed modulated fluidic valve, blue dashed: valve position, green: velocity.

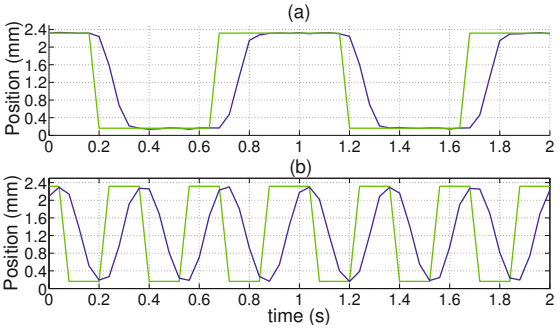


Fig. 12. Step-like reference tracking at (a) f=1Hz, (b) f=3Hz, where the green line represents the reference.

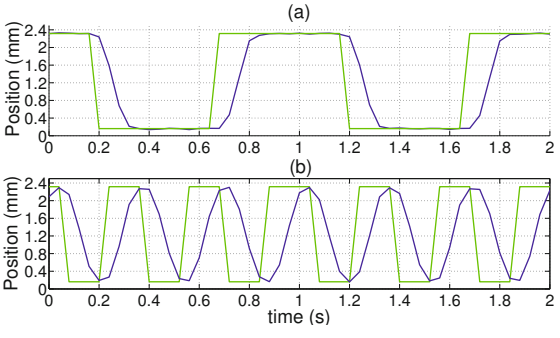


Fig. 13. Step-like reference tracking at (a) f=1Hz, (b) f=3Hz, where the green line represents the reference.

reaching the final position, the direction of movement is changed and τ_d is reset to its starting value. It is obvious that the speed of the valve is increased while the delay factor decreases and vice versa.

To test the overall performances of the developed control scheme, sinusoidals with different frequencies as well as step-like reference trajectories were fed to the control system as reference signals. The parameters of the PID controller have been determined through classical Ziegler-Nichols method, whereas the critical controller gain and the critical oscillation period respectively were identified as $K_{p,crit} = 1.167$ and $T_{crit} = 0.06\,s$. These results in the following controller gains

$$K_p = 0.7\,,\; K_i = 0.03\,,\; K_d = 0.008\,.$$

Figure 13 exemplary presents the tracking results of the position control loop for step-like reference signals at 1 and 3Hz.

As we can extract from the Figure 13, the controller tracks step-like signals without overshoot up to an actual maximum frequency of f=3Hz. Due to the fact that this test is

Table 1. Characteristic values for sinusoidal reference tracking.

f in [Hz]	e_{p_max} in [V]	φ in [deg]
0.5	0.028	0.81
1	0.032	1.8
3	0.11	3.24

performed in the maximum working range, a better dynamic response can be expected for smaller movements.

Finally, Table 1 sums up the characteristic values for a sinusoidal reference tracking, where f is the frequency of the reference, e_{p_max} the maximum error in position, and φ the phase shift between the two signals.

5 The Valve Prototype

After the choice of the proper hardware and electronic components, a new valve was designed and realized (Fig. 14a) using rapid prototyping technique. The final device (Fig. 14b) has a volume of LxWxH=(60mm)x(20mm)x(40mm), and has a weight of $106g$ including one position and two pressure sensors, an amplifier board, the stepper motor, and the electrical/hydraulic connectors.

In this first prototype the PWM motor driver and the control logic are located in a separated device, nevertheless future versions may also include these components on-board. This will reduce the amount of required cables to only a single power and a data line (e.g. a CAN-BUS). Compared with other state of the art hydraulic valves [12], the one presented here is designed to work with relatively low pressures both for hydraulic and pneumatics purposes, it is extremely compact and lightweight, furthermore it integrates two pressure sensors that are directly connected to the two output lines A and B (see schema in figure 2). This allows a fine tuning of the pressure inside the two actuator chambers, and therefore enables a precise control of the generated torque.

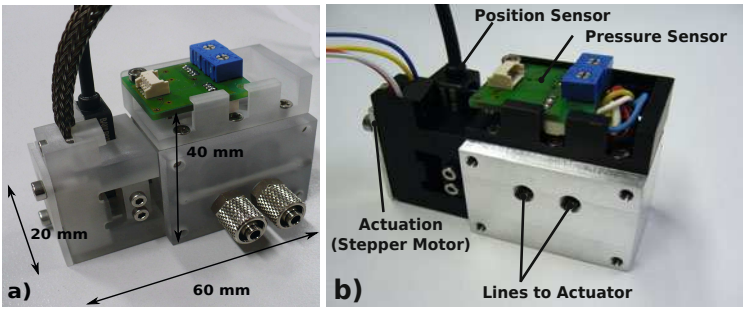


Fig. 14. a) Valve developed using rapid prototyping technique b) Final valve in aluminum with all integrated components.

6 Conclusions and Future Developments

This document presented the development of a new fluid servo-valve. The device is specifically intended for lightweight robotic systems and designed to work in a operational range of $0 - 50 \cdot 10^5 Pa$. The work is motivated by the fact that, according to the authors knowledge, no commercial valve exists for precise low pressure hydraulic actuators control. As general requirements, compactness, lightweight, and high dynamics were considered during the design process. A first series of experiments have been performed to test repeatability, flow-position characteristics and dynamic response. A model of the drive system of the servo-valve was identified using a recursive Hammerstein-Wiener parameter adaptation algorithm. The combination of a linear and dynamic part with a non linear static component let to reach a fit of 87%. Finally to test the overall functionality of the valve and to measure its step response characteristics, a proper control algorithm was implemented that allows to regulate the position and the velocity of the valve's spool.

Future work will be devoted to identify the overall model that will explicitly define the position-flow-pressure relationship. The backlash and dead-band problems need to be properly addressed. In particular the employment of a ball-screw for the roto-translation mechanism of the valve, instead of a normal lead screw , will improve the precision in controlling the position. Furthermore, with a customized design of the spool, will be possible to decrease the switching time between the two opening positions, and therefore to improve the dynamic behavior of the servo-valve.

Acknowledgements. The work presented here was done within the VI-Bot project, funded by the German Ministry of Science (BMBF), [grant number 01IW07003].

References

1. Hayward, V.: Experimental Robotics III. In: Design of a hydraulic robot shoulder based on a combinatorial mechanism, vol. 200. Springer, Heidelberg (1994)
2. Jacobsen, S.C., Smith, F.M., Backman, D.K., Iversen, E.K.: High performance, high dexterity, force reflective teleoperator ii. In: Proc. ANS Topical Meeting on Robotics and Remote Systems (1991)
3. Yoshinada, H., Yamazaki, T., Suwa, T., Naruse, T.: Design and control of a manipulator system driven by seawater hydraulic actuator. In: Second Int. Symposium on Measurement and Control in Robotics (ISMCR), pp. 359–364 (1992)
4. Pons, J.L. (ed.): Wereable Robots: Biomechatronixs Exoskeletons. Wiley, Chichester (2008)
5. Folgheraiter, M., Bongardt, B., Albiez, J., Kirchner, F.: A bio-inspired haptic interface for tele-robotics applications. In: IEEE International Conference on Robotics and Biomemetics (ROBIO 2008), Thailand, Bangkok (2008)
6. Folgheraiter, M., Bongardt, B., Albiez, J., Kirchner, F.: Design of a bio-inspired wearable exoskeleton for applications in robotics. In: International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2009), Portugal, Porto (2009)
7. Folgheraiter, M., de Gea, J., Bongardt, B., Albiez, J., Kirchner, F.: Bio-inspired control of an arm exoskeleton joint with active-compliant actuation system. Applied Bionics and Biomechanics 6, 193–204 (2009)

8. Folgheraiter, M., Schmidt, B.B.S., Fernandéz, d.G., Albiez, Kirchner, F.: Design of an arm exoskeleton using an hybrid motion-capture and model-based technique. In: IEEE International Conference on Robotics and Automation ICRA 2009, Kobe, Japan, May 12–17 (2009)
9. Raibert, M., Blankespoor, K., Nelso, G., Playter, R.: The BigDog Team: Bigdog, The rough-terrain quaduped robot. Technical report, Waltham, MA 02451 USA (2008)
10. Kazerooni, H., Steger, R., Huang, L.: Hybrid control of the berkeley lower extremity exoskeleton (bleex). The International Journal of Robotics Research 25, 561–573 (2006)
11. Kahn, M.: Optimal Control of a Hydraulic Arm. PhD thesis, Stanford University M.E Dept (1969)
12. Moog Inc. Sub miniature servovalve e024. Technical report (2009), `http://www.moog.com/`
13. Landau, I.D., Zito, G.: Digital Control Systems. Springer, Heidelberg (2006)
14. Guo, F.: A New Identification Method for Wiener and Hammerstein System. PhD thesis, Karlsruhe University (2004)

Learning Concepts with Multi-robot Systems

Ana Cristina Palacios-García, Angélica Muñoz-Meléndez, and Eduardo F. Morales

National Institute for Astrophysics, Optics and Electronics
Luis Enrique Erro No. 1, Santa María Tonantzintla, Puebla, México
cpalacios@ccc.inaoep.mx, {munoz,emorales}@inaoep.mx

Abstract. This paper introduces a novel approach to learn representations of objects using a team of robots. Each robot extracts local and global visual features of objects and combines them to represent and recognize objects. Contrary to previous approaches the robots do not know in advance the number or nature of objects to learn. Individual representations of objects are learned on-line while the robots are traversing an environment. Robots share their individual concepts to improve their own concepts, and to acquire a new representation of an object not seen by them. For that, the robots have to detect if they are seeing a new object or an already learned one. We empirically evaluated our approach with a real world robot team with very promising results.

Keywords: Robotics and automation, Mobile robots and autonomous systems, Vision, Recognition and reconstruction, Network robotics.

1 Introduction

Collective robotics is a very active research area in the robotics community. Multi-robot systems or robot teams have effectively emerged as an alternative paradigm for the design and control of robotic systems because of the team's capability to exploit redundancy in sensing and actuation.

The research on collective robotics has focused on developing mechanisms that enable autonomous robots to perform collective tasks, such as strategies for coordination and communication [1,2]; exploration, mapping and deployment [3]; sensing, surveillance and monitoring [4]; and decentralized decision making [5]. In these works, a robot team can reduce time to complete a complex task that is allocated among its members.

Despite constant research on the design of robot teams, very little attention has been paid so far to the development of robot teams capable of learning from their interaction with their environment. In addition to their capability for accelerated learning, learning robot teams can be used to acquire a much richer and varied information compared to the information acquired by single learning robots.

Learning is a key issue to achieve autonomy for both, single robots and robot teams. Learning capabilities can provide robots the flexibility and adaptation needed to cope with complex situations. In the context of robot teams, the most common machine learning approach has been reinforcement learning, where the idea is to learn optimal policies using a set of robots to improve the coordination of individual actions in order to reach common goals [1,2,4,6].

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 253–265.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

In this work we use visual information to learn, with a team of robots, descriptions of objects placed in a particular environment. Learning to recognize particular objects in an environment is important for robotics as it can be used for local and global localization tasks as well as for simple service tasks such as searching for objects in unknown places. Contrary to previous approaches, in our learning setting, the robots are not told the number or nature of the objects to be learned.

Vision is a primary source of perception in robotics and provides different features that can be used to classify objects. In general, using a particular set of features can be adequate for particular tasks but inadequate for other tasks. In this work, objects are characterized by two complementary features: (i) SIFT features [7] and (ii) information about the silhouettes of objects. Other features could be used as well, but the main objective in this work is to show the different cases and possible confusions that can arise in the recognition of objects and merging of concepts, and how they can be addressed.

This article deals with the individual and collective representation of objects from visual information using a team of autonomous robots.

The rest of the paper is organized as follows. Section 2 reviews related work. Sections 3 and 4 introduce, respectively, the stages of individual learning and collective learning of concepts. Section 5 describes our experimental results, and Section 6 provides conclusions and future research work.

2 Related Work

Interesting experiments where physical mobile robots learn to recognize objects from visual information have been reported. First we review significant work developed for individual learning, and then we review learning approaches developed for robot teams.

In [8] different learning techniques to acquire automatically semantic and spatial information of the environment in a service robot scenario are applied. In that work, a mobile robot autonomously navigates in a domestic environment, builds a map, localizes its position in the map, recognizes objects and locates them in the map. Background subtraction techniques are applied for foreground object segmentation. Then objects are represented by SIFT points [7] and an appearance-based method is used for detecting objects named Receptive Field Co-occurrence Histograms. The authors developed a method for active object recognition which integrates both local and global information of objects.

In [9] authors applied an instance-based method to train a robot for object recognition purposes. The objects are represented by color histograms. Different representations are learned from different views of the same object using a mediator that can ask questions and provide feedback on the robot's answers. The recognition is performed by classifying new views of objects using the KNN algorithm [10].

In [11], a scheme for fast color invariant ball detection in the RoboCup context is presented. To ensure the color-invariance of the input images, a preprocessing stage is first applied for detecting edges using the Sobel filter, and specific thresholds for color removal. Then, windows are extracted from images and predefined spatial features, such as edges and lines, are identified in these windows. These features serve as input to an AdaBoost learning procedure that constructs a cascade of regression tree classifiers

(CART). The system is capable of detecting different soccer balls in RoboCup and other environments. The resulting approach is reliable and fast enough to classify objects in real time.

In [12] authors present an on-line method for learning objects for human-robot interaction. In this case, the robot's user is included in the learning framework as an instructor. For the learning method, authors used a lifelong incremental learning system that evolves with any new learned object based on one-class learning (OCLL). Objects are represented using normalized shape features. Aditionally, the authors proposed an experimental methodology for evaluating a word learning method, and for comparing the word learning capabilities of different agents as well as to asses research progress in similar works.

Concerning the problem of collective learning of objects using robot teams there are, as far as we know, very few works. In [13] the authors address the problem of mobile object recognition based on kinematic information. The basic idea is that if the same object is being tracked by two different robots, the trajectories and therefore the kinematic information observed by each robot must be compatible. Therefore, location and velocities of moving objects are the features used for object recognition instead of features such as color, texture, shape and size, more appropriate for static object recognition. Robots build maps containing the relative position of moving objects and their velocity at a given time. A Bayesian approach is then applied to relate the multiple views of an object acquired by the robots.

In [14], objects are represented with Principal Components (PC) learned from a set of global features extracted from images of objects. An object is first segmented and its global features such as color, texture, and shape are then extracted. Successive images in a sequence are related to the same object by applying a Kalman filter. Finally, a 3D reconstructed model of an object is obtained from the multiple views acquired by robots. For that purpose, a Shape From Silhouette based technique [15] is applied.

The main drawbacks of previous works can be summarized as follows: (i) Most approaches are able to cope with a limited number of learning objects, usually 3 to 12 objects, and (ii) the number of learning objects is known *a priori*. In contrast to previous works, in our method each robot learns on-line individual representations of objects without prior knowledge on the number or nature of the objects to learn. Individual concepts are represented as a combination of global and local features extracted autonomously by the robots from the training objects. A Bayesian approach is used to combine these features and used for classification. Individual concepts are shared among robots to improve their own concepts, combining information from other robots that saw the same object, and to acquire a new representation of an unnoticed object. We also analyzed and provide solutions to the different cases that can occur from information of two individual concepts.

3 Individual Learning of Concepts

The individual concepts are learned on-line by a robot team while traversing an environment without prior knowledge on the number or nature of the objects to learn. The individual learning of concepts consists of tree parts: object detection, feature extraction, and individual training.

Individual concepts of objects are represented by Principal Components (PC) over the information about the silhouettes of objects and by Scale Invariant Features (SIFT). Learned concepts are shared among robots.

3.1 Object Detection

Robots move through an environment and learn descriptions of objects that they encountered during navigation. Objects are detected using background substraction. In this paper we assume a uniform and static background. We performed morphological operations (closing and erode) to achieve better segmentation. Once an object is detected, it is segmented and scaled to a fixed size, to make the global PC features robust to changes in scale and position.

3.2 Feature Extraction and Individual Training

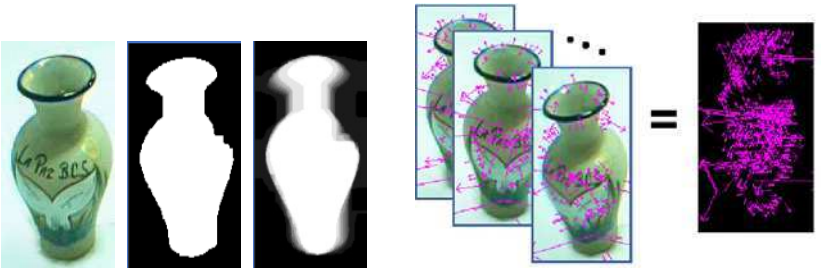
The segmented objects are grouped autonomously by the robots in sets of images containing the same object. Robots assume that they are observing the same object while it can be detected, and they finish to see it when they can not detect objects in the captured images. Only one object can be detected in an image at the same time. For each set of images, the robot obtains an individual concept that represents the object.

Training Using Global Features. We applied Principal Component Analysis (PCA) over the average silhouettes that are automatically extracted from the set of images of a particular object. The average provides a more compact representation of objects and reduces segmentation errors. Figure 1 (a) shows an image of the object *vase* used in the training phase, Figure 1 (b) shows its silhouette, and Figure 1 (c) illustrates the average silhouette obtained from a set of images that represent the object of Figure 1 (a). Once the robot has obtained an average silhouette, this is added by the robot to a set of known average silhouettes. After that, the robot uses PCA to reduce the dimensionality of all average silhouettes learned to get the PC features that represent them.

Training Using Local Features. Each robot extracts local SIFT features of each image of the set of images, and groups them in a final set which contains all the different SIFT features that represent an object. In Figure 1 (d) we show an example of the SIFT points obtained from a set of images of a *vase* and the final set of SIFT points obtained. The PC features and the SIFT features represent the individual concept of the observed object.

3.3 Sharing Concepts

The concepts learned by robots are shared among them to achieve collective learning. This can be done off-line or on-line. In the case of collective off-line learning the robots share their individual concepts once they have learned all the training objects. On the other hand, in the collective on-line learning the robots share their individual concept as soon as a new object is learned.



(a) (b) (c) (d)

Fig. 1. Examples of the silhouette (b) and average silhouette (c) of an object (a). Examples of the SIFT features extracted from a set of images and the final set of SIFT features (d).

4 Collective Learning of Concepts

Collective learning of concepts enables robots to improve individual concepts combining information from other robots that saw the same object, and to acquire a new representation of an object not seen by them. Therefore, a robot can learn to recognize more objects of what it saw and can improve their own concepts with additional evidence from other robots.

A robot has to decide whether the concept shared by another robot is of a new object or of a previously learned concept. A robot can face three possibilities: coincident, complementary or confused information. The shared concepts are fused depending on the kind of information detected, as described below.

4.1 Pre-analysis of Individual Concepts

The concept learned by a robot is defined as follows:

$$C_k^i = \left\{ Sil_k^i, SIFT_k^i \right\} \tag{1}$$

where C_k^i is the concept k learned by robot i, Sil_k^i is the average silhouette, and $SIFT_k^i$ is the set of SIFT features that form the concept k.

In order to determine if a shared concept is previously known or not to a robot, it evaluates the probabilities that the PC features and SIFT features are previously known by the robot. The probability vectors of PC features calculated by robot i, v_P^i, indicate the probability that a concept shared by robot j, C_k^j, is similar to the concepts known by robot i, $C_1^i, \ldots, C_{numObjs}^i$, given the global features. $numObjs$ is the number of concepts of objects known by robot i. The process to obtain the probability vector PCA is described as follows:

- A temporal training set of silhouettes is formed by adding the average silhouettes of concepts known by robot i or actual robot, $Sil_1^i, ..., Sil_{numObjs}^i$, and the average silhouette of the new or shared concept Sil_k^j.
- The PCA is trained using the temporal set of average silhouettes. The projection of the average silhouettes know by robot i is obtained as a matrix of projections, $matProys$. The projection of the average silhouette Sil_k^j is obtained in a vector, $vectProys$.

- The Euclidean distance (dE) is calculated between each vector of the matrix $matProys$ and the vector $vectProys$ as shown in formula 2, i.e, we obtain the distance between all the projections already computed and the projection of the new silhouette.

$$dE_l^i = \sqrt{\sum_{r=1}^{nEigens} (matProys_{(l,r)} - vectProys_{(1,r)})^2} \tag{2}$$

where $nEigens$ is the number of eigenvectors used during the PCA training ($nEigens = numObjs^i - 1$), and l is the index of the distance vector, where the maximum size of the vector dE^i is $numOb^i$.
- The distance value d^i is divided by a maximum distance value, T, determined experimentally to obtain a similarity metric also called the probability vector PCA, v_P^i, as shown in formula 3.

$$v_{P_l}^i = 1 - \frac{d_l^i}{T} \tag{3}$$

If d_l^i is bigger than the T value, then the probability will be fixed as shown in formula 4, which indicates that the projections of the object k and the one of the object l are completely different.

$$v_{P_l}^i = \frac{1}{numOb} \tag{4}$$

The value of the SIFT similarity metric also called the probability vector SIFT at the position $v_{S_l}^i$, is obtained calculating the number of coincident SIFT points, n_{coin}, between the individual SIFT concept S_l^i learned by robot i, and the individual SIFT concept S_k^j shared by robot j. If the number n_{coin} is bigger than an average of coincidences determined experimentally, $AverageCoin$, then the probability will be fixed to $v_{S_l}^i = 1.0$, which means that both concepts contain the same local SIFT features. Otherwise, the probability will be calculated using formula 5.

$$v_{S_l}^i = \frac{n_{coin}}{AverageCoin} \tag{5}$$

The constant $AverageCoin$ represents the average of coincidences between two sets of SIFT points of the same object from different perspectives.

4.2 Analysis and Fusion of Individual Concepts

This section describes how detection is achieved if the new or shared concept is one of the following: coincident, complementary or confused, and how the individual concepts are fused to form collective concepts depending on the kind of detected concept.

Coincident Concepts. A coincident concept is detected when one, two or more robots of the robot team learned individual concepts from similar views of the same object. A new or shared concept is classified as coincident if $v_{P_l}^i \geq \alpha$ and $v_{S_l}^i \geq \alpha$. That is, if both probabilities (PCA and SIFT) of a previously learned concept are greater than a

predefined threshold value (α). If a new or shared concept is determined as coincident it is merged with the most similar known concept as follows:

PCA Fusion. It is obtained by evaluating a new average silhouette from the average of the known Sil_l^i and new Sil_k^j silhouettes. After that, it is necessary to re-train the PCA substituting the concept Sil_l^i with the new average silhouette which contains information of the concept learned by robot j.

SIFT Fusion. It is obtained by adding the complementary SIFT points of concept $SIFT_k^j$ to the set of SIFT points of concept $SIFT_l^i$. Also, each pair of coincident SIFT points of both concepts is averaged in terms of position and their corresponding SIFT descriptors.

The main idea to fuse coincident concepts is to improve their representation.

Complementary Concepts. A concept C_k^j contains complementary information if it differs with all known concepts by robot i, i.e., if both shape and local features are different to all known concepts by robot i, $C_1^i, \ldots, C_{numObjs}^i$. That is, if $v_P^i < \alpha$ and $v_S^i < \alpha$.

A complementary concept C_k^j is fused with the collective concepts known by the robot i as follows:

PCA Fusion. The new average silhouette is added and the new PC features are obtained by re-training the PCA using the updated set of average silhouettes.

SIFT Fusion. The new SIFT features are simply added to the current set of SIFT concepts known by the robot i.

Confused Concepts. There are two types of confusion that can occur between concepts:

Different Shape and Similar Local Features. (type 1): This type of confusion occurs when the new concept C_k^j is complementary by shape, Sil_k^j, to all the concepts known by the robot i, $Sil_1^i, ..., Sil_{numObjs^i}^i$ but it is coincident by local SIFT features, $SIFT_k^j$, with at least one concept known by the robot i. That is, $v_{S_l}^i \geq \alpha$ and if $v_P^i < \alpha$.

Similar Shape and Different Local Features. (type 2): This type of information occurs when concept C_k^j is coincident by shape, Sil_k^j, to at least one concept known by the robot i, but it is complementary using its local SIFT features, $SIFT_k^j$. That is, if $v_{P_l}^i \geq \alpha$ and $v_S^i < \alpha$.

In both types of confusion, type 1 or type 2, there can be two options:

a) **Different Objects.** Both concepts correspond to different objects.
b) **Same Object.** Both concepts correspond to the same object but they were learned by robots from different points of view.

In our current approach, both types of confusions are solved as complementary objects. The reason is that robot i cannot distinguish with its current information between both, different objects or same object, using only the individual and the shared concepts. To solve the ambiguity, as future work each robot should build autonomously a map and locate its position in the map. In addition, for each learned object, robots will

Fig. 2. The robot team

locate them in the map. For confused objects a robot can move to the position of the object marked in the map to see the object from different perspectives in order to solve the conflict.

5 Experiments and Results

We performed several experiments to demonstrate the proposed algorithm. In section 5.1, we show the results of a general experiment that demonstrates the main features of the proposed approach. In section 5.2 we present the accuracy of the collective concepts versus the individual concepts.

In these experiments we used a robot team consisting of two homogeneous Koala robots (Figure 2) equipped with a video camera of 320×240 pixels, and an on board portable computer of 1 GB of RAM memory for the processing. For more than two robots our method can be applied straightforward. The only difference is that robots will need to consider the information from more than one robot, possibly reducing confused concepts.

5.1 Concept Acquisition and Testing

The mobile robots learn on-line representations of several objects while following a predefined trajectory without prior knowledge on the number or nature of the objects to learn. The idea of using pre-planned trajectories instead of making the robots wandering randomly, is that we can control the experimental conditions to show different aspects of the proposed methodology.

Each robot shares its individual concept as soon as it is learned to improve the representation of this concept or to include a new concept in the other robot. Figure 3 shows the training objects used in this experiment. As can be seen in the figure, some objects have the same shape but different texture, some have the same texture but different shape, some others are not symmetric in their shape. The objective of this experiment is to show the performance of the system to detect coincident, complementary and confused information under a wide variety of conditions.

In Figure 4 we present how the training objects were placed in the environment, and how the robots moved in the environment to see the different training objects. Robot 1 (R1) learned during individual training concepts for: *dolphin*, *can*, *water bottle* and *vase*. Robot 2 (R2) learned individual concepts for: *vase*, *soda bottle*, *bottle* and *cone*. Note that some objects are learned by both robots while others are only learned by one robot.

(a) (b) (c) (d) (e) (f) (g)

Fig. 3. Training objects. (a) *vase*, (b) *water bottle*, (c) *can*, (d) *dolphin*, (e) *soda bottle*, (f) *bottle* and (g) *cone*.

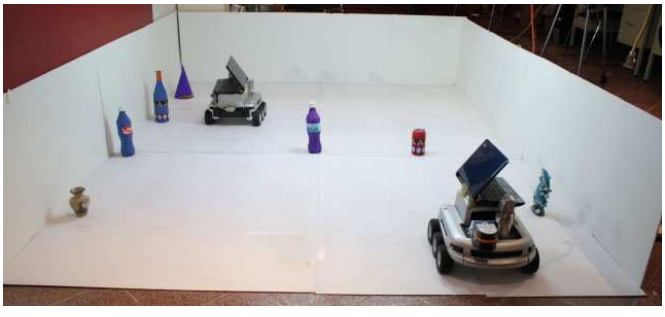


Fig. 4. Concept acquisition

While learning a new concept, each robot has to decide whether to fuse the current concept with a previously known concept or include it as a new one. Table 1 shows the probability vectors of the PCA features based on shape (v_P^1) and of the SIFT features (v_S^1) obtained by Robot 1. A similar table is obtained by Robot 2 (for details see [16]). In Table 1 the coincident information is represented in bold.

We used the defined criteria in Section 4.2 to recognize coincident, complementary or confused concepts, with $\alpha = 0.65$ as threshold value, and the probability vectors of Table 1.

For instance, in column labeled with PCA of Table 1 it is shown how the probabilities of objects of R1 are affected using only PCA over the shapes of objects, as both robots encounter and learn concepts while traversing the environment. In the first row, R1 learns about the concept *dolphin* and acquires it. In the second row, R2 then learns about *vase* and shares this concept to R1 (that is expressed with a subindex R2). The probability, according to the PCA features to be a *dolphin* is 0.19 (second row). R1 learns the object *can*, which has a probability of 0.31 to be a *dolphin* and a probability of 0.26 to be a *vase*, which was learned by R2 and shared to R1 (third row). In the fifth row, R1 learns about a *water bottle* but it confuses with the *soda bottle* learned and shared before by R2. As can be seen from Figure 3, both objects have the same shape and consequently the PCA features are not able to discriminate between these

Table 1. Probability vectors PCA (v_P^1) and SIFT (v_S^1) obtained by R1

	New (collective concepts R1)													
	PCA							SIFT						
Objects	*Dolphin*	*Vase*	*Can*	*Soda bottle*	*Water bottle*	*Bottle*	*Cone*	*Dolphin*	*Vase*	*Can*	*Soda bottle*	*Water bottle*	*Bottle*	*Cone*
Dolphin$_{R1}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vase$_{R2}$	0.19	-	-	-	-	-	-	0.09	-	-	-	-	-	-
Can$_{R1}$	0.31	0.26	-	-	-	-	-	0.12	0.12	-	-	-	-	-
Soda bottle$_{R2}$	0.36	0.28	0.58	-	-	-	-	0.28	0.11	0.40	-	-	-	-
Water bottle$_{R1}$	0.43	0.28	0.53	**0.73**	-	-	-	0.15	0.59	0.20	0.20	-	-	-
Bottle$_{R2}$	0.31	0.17	0, 56	0.61	0.58	-	-	0.08	0.15	**0.65**	0.04	0.12	-	-
Vase$_{R1}$	0.25	**0.69**	0.42	0.43	0.41	0.32	-	0.16	**1.00**	0.23	0.10	0.08	0.09	-
Cone$_{R2}$	0.31	0.01	0.28	0.28	0.33	0.43	-	0.05	0.28	0.43	0.10	0.14	0.09	-

two objects. This is not the case for the SIFT features, which prevent R1 to consider it as the same object. In the seventh row, R1 learns about *vase* which was already learned and shared by R2, and in this case both concepts are merged.

Robot 1 detects the three types of possible information and fuses them as mentioned in section 4.2. The first four learned objects (*dolphin, vase, can* and *soda bottle*) were detected as complementary because all values of their corresponding vectors $v^1_{P_{(x,l)}}$ and $v^1_{S_{(x,l)}}$ are less than α threshold. The next two objects (*water bottle* and *bottle*) were detected as confused (type 2 and type 1, respectively), because either their SIFT features, or their shape features are similar, according to the α threshold. Object *vase* learned by Robot 1 was detected as coincident with the learned object *vase* learned by Robot 2 and shared to Robot 1 ($v^1_{P_{(7,2)}} = 0.69$ and $v^1_{S_{(7,2)}} = 1.00$). Finally object *cone*, unnoticed for Robot 1, is detected as complementary.

To test the performance of the individual concepts and the collective concepts acquired by each robot, the concepts were used in an object recognition task. Each robot followed a predefined trajectory to recognize objects in the environment. The objects were detected by the robot team in the following order: *cone*, *water bottle*, *vase*, *bottle*, *soda bottle* and *dolphin*. Once an object is detected, the robot (i) evaluates its class using the PCA (v_P^i) and SIFT (v_S^i) probability vectors and combines both probabilities using a Bayesian approach, summarized in formula 6.

$$P^i_{B_l} = \frac{v^i_{P_l} \times v^i_{S_l} \times P_u}{\left(v^i_{P_l} \times v^i_{S_l} \times P_u\right) + \left((1 - v^i_{P_l}) \times (1 - v^i_{S_l}) \times (1 - P_u)\right)} \tag{6}$$

where P_u is a uniform probability distribution ($P_u = \frac{1}{numObjs^i}$), $v^i_P = p(PCA\ projection \mid Class = i)$, $v^i_S = p(SIFT\ matching \mid Class = i)$, P^i_B is the Bayesian probability vector ($p(Class = i \mid PCAprojection, SIFTmatching)$, and l is the

Table 2. Precision in the object recognition task using the individual and collective concepts acquired by each robot

	R1	R2	R1-R2	R2-R1
PCA	47.82 % (89.70 %)	52.12 % (98.00 %)	95.65 %	98.93 %
SIFT	50.00 % (93.87 %)	52.13 % (98.00 %)	95.65 %	98.93 %
Bayes	48.91 % (91.83 %)	52.12 % (98.00 %)	95.65 %	100.00 %

index of the Bayesian probability vector, where the maximum size of the probability vector is $numObjs^i$.

We show in Table 2 the precision of the object recognition task using the individual (columns labeled with R1 and R2) and collective (columns labeled with R1-R2 and R2-R1) concepts. Precision is calculated as the number of well classified images from the total number of test images that contain an object during one experiment. The precision is presented in two ways, one considering the total number of objects, and the other one taking into a count only the number of objects used during the individual training (reported in parentheses). As can be seen the collective concepts produce a significantly better precision.

5.2 Accuracy of the Individual and Collective Concepts

In this section we compare the results of the individual concepts with that of collective concepts. In each experiment, a different set of objects was used, and both robots learned the same set of objects. Therefore, all the shared concepts were coincident, that is, robots learned both individually and collectively the same number of concepts. At the end of each experiment the robots learned four concepts that were tested in a predefined sequence.

In Table 3, we present the average percentages of accuracy using the individual and collective concepts in an object recognition task. The accuracy indicates the average percentage of well classified images during the whole set of experiments, in this case six, when using the PCA and SIFT features and the Bayesian approach.

As it can be observed in Table 3, the accuracy that indicate the quantity of well classified images using the collective concepts for the object recognition task, is in general better than the accuracy using the individual concepts. For PCA, SIFT and Bayes there is an improvement in the accuracy of 2.56 %, 13.79 % and **20.62 %**, respectively. This demonstrates that the collective concepts have better coverage than the individual concepts because they contain information acquired from different points of view, which allows a better recognition of test objects.

In Table 4, we present the average percentages of false positives for both, the individual and the collective concepts acquired by the robots. From the results presented in Tables 3 and 4 we conclude that the collective concepts have better quality than the individual concepts.

In general for the individual and the collective concepts, we observed an improvement in the accuracy when using the Bayesian approach. The average profit in the percentages of classification using the Bayesian approach using the collective concepts with regard to the individual concepts is of **14.63 %**.

Table 3. Accuracy of individual and collective concepts

	PCA	SIFT	Bayes
Individual	84.94 %	67.88 %	80.18 %
Collective	87.18 %	81.12 %	94.81 %

Table 4. Average percentages of false positives of individual and collective concepts

	PCA	SIFT	Bayes
Individual	14.42 %	0.64 %	0.64 %
Collective	13.14 %	0.00 %	0.00 %

6 Conclusions and Future Work

In this paper we have introduced a new on-line learning framework for a team of robots. Some of the main features of the proposed scheme are:

- The robots do not know in advance how many objects they will encounter. This poses several problems as the robots need to decide if a new concept or a shared concept, is of a previously learned object or not.
- The representation of objects are learned on-line while the robots are traversing a particular environment. This is relevant for programming autonomous robots.
- Three possible cases in which to merge concepts and how to merge them were identified.

The detection of coincident concepts avoids producing multiple concepts for the same object. The detection of complementary concepts allows to detect and learn unknown objects not seen by a particular robot. The detection of confused concepts allows to fuse information: (i) when the objects have different shape and similar SIFT features, and (ii) when the objects have similar shape and different SIFT features. These cases are particularly difficult to deal with because the objects may be genuinely different or may be the same but seen from different points of view by the robots.

In general, the object recognition using the collective concepts had a better performance than using the individual concepts in terms of accuracy. This occurs because the collective concepts consider information from multiple points of view producing more general concepts.

As future work we propose to integrate schemes to object segmentation for dynamic environments. For instance, using an object segmentation based on distance as in [17]. Use a different set of features and identify possible conflicts between more that two kinds of features. We also plan to incorporate planning of trajectories to autonomously allocate the environment among robots. We also plan to add strategies to solve some confusions in shared concepts by taking different views from these objects. Finally, we plan to incorporate our algorithm for robot localization and search of objects, and to test our work for robot teams with three or more robots.

Acknowledgements. The first author was supported by the Mexican National Council for Science and Technology, CONACYT, under the grant numbers 212422 and 84162.

References

1. Asada, M., Uchibe, E., Noda, S., Tawaratsumida, S., Hosoda, K.: Coordination of Multiple Behaviors Acquired by Vision-Based Reinforcement Learning. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 917–924 (1994)
2. Matarić, M.J.: Reinforcement Learning in the Multi-Robot Domain. Autonomous Robots 4(1), 73–83 (1997)
3. Howard, A., Parker, L.E., Sukhatme, G.S.: Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection. International Journal of Robotics Research 25, 431–447 (2006)
4. Parker, L.E.: Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. Autonomous Robots 3(12), 231–255 (2002)
5. Wessnitzer, J., Melhuish, C.: Collective Decision-Making and Behaviour Transitions in Distributed Ad Hoc Wireless Networks of Mobile Robots: Target-Hunting. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 893–902. Springer, Heidelberg (2003)
6. Fernández, F., Borrajo, D., Parker, L.E.: A Reinforcement Learning Algorithm in Cooperative Multi-Robot Domains. Journal of Intelligent and Robotic Systems, 161–174 (2005)
7. Lowe, D.G.: Distinctive Image Features from Scale Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
8. Ekvall, S., Jensfelt, P., Kragic, D.: Integrating Active Mobile Robot Object Recognition and SLAM in Natural Environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5792–5797 (2006)
9. Steels, L., Kaplan, F.: AIBO's first words. The Social Learning of Language and Meaning. Evolution of Comunication 4(1), 3–32 (2001)
10. Mitchell, T.M.: Machine Learning. McGraw-Hill Science/Enginering/Math (1997)
11. Mitri, S., Pervolz, K., Surmann, H., Nuchter, A.: Fast Color Independent Ball Detection for Mobile Robots. In: Proceedings of the IEEE International Conference Mechatronics and Robotics, pp. 900–905 (2004)
12. Saebra Lopes, L., Chauhan, A.: How many words can my robot learn? Interaction Studies 8(1), 53–81 (2007)
13. Montesano, L., Montano, L.: Identification of Moving Objects by a Team of Robots Based on Kinematic Information. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 284–290 (2003)
14. O'Beirne, D., Schukat, M.: Exploration and Object Recognition Using Cooperative Robots. In: Proceedings of the International Conference on Imaging Science, Systems and Technology, pp. 592–598 (2004)
15. Cheung, G.K.M., Baker, S., Kanade, T.: Visual Hull Alignment and Refinement Across Time: A 3-D Reconstruction Algorithm Combining Shape-From-Silhouette with Stereo. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 375–382 (2003)
16. Palacios-García, A.C., Muñoz-Meléndez, A., Morales, E.F.: Collective Learning of Concepts using a Robot Team. In: Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (Robotics and Automation) vol. 2, pp. 79–88 (2010)
17. Méndez-Polanco, J.A., Muñoz-Meléndez, A., Morales, E.F.: People Detection by a Mobile Robot Using Stereo Vision in Dynamic Indoor Environments. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 349–359. Springer, Heidelberg (2009)

Active Method for Mobile Object Detection from an Embedded Camera, Based on *a Contrario* Clustering

Dora Luz Almanza-Ojeda, Michel Devy, and Ariane Herbulot

CNRS, LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS-CNRS, F-31077 Toulouse, France
{dlalmanz,michel,herbulot}@laas.fr

Abstract. This paper concerns the detection and the tracking of moving objects from images acquired by a single camera Embedded on a mobile robot. The proposed approach is based on the active extraction of salient features, feature tracking on some images, feature clustering as a way to detect moving objects.

Keywords: Moving obstacles, Detection, Tracking, Clustering, Monocular vision.

1 Introduction

A key function required for autonomous robot navigation, must cope with the detection of objects close to the robot trajectory, and the estimation of their states. This function has been studied by the robotic and the ITS (Intelligent Transportation Systems) communities, from different sensory data such as laser, radar, vision, among others. For driver assistance, many contributions concern laser-based obstacle detection and tracking [15]. In order to track obstacles, it is required to estimate from the same sensory data, the egomotion of the vehicle. The egomotion could be obtained from Inertial Measurement Units (IMU) or estimated by using the images and the epipolar geometry [7]. First solution needs that IMU be coupled rigidly to the camera which is normally not reached. In other hand, egomotion image-based is inaccurate in the case of noisy, low contrasted images. The disadvantages of both approaches are tackled in [9], where authors present an alternative camera-IMU system to acquire the images and accurately measure actual camera motion. However, this solution has low portability. So, in spite of numerous contributions, object tracking during robot navigation still remains a challenge when it is based only on vision. So this paper proposes a strategy to detect static points, and moreover to detect and cluster the moving ones in order to track mobile objects: it is the first step towards the full integration of a Visual SLAMMOT approach. It is proposed to reach this objective, using only a monocamera system.

There are various methods for feature detection and tracking using a single mobile camera. A.J.Davison [4] has proposed a spatio-temporal approach, in order to detect and track 2D points from an image sequence and then to reconstruct corresponding 3D points used to locate the camera. The system state at instant time t, is given by the 3D position and the speed of the camera and every tracked point selected by the Harris detector. From this state, every point position is predicted in the image acquired

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 267–280.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

at time $t + 1$, forming an elliptic zone where the point must be found if the global state is consistent. Then, each interest point is searched in its predicted zone by a similarity measurement based on correlation score using a $n \times n$ template around its first position.

A method widely used for robotics applications is based on the optical flow. If optical flow is only extracted for interest features, i.e. for a very small part of total image points, a tracking procedure of many objects characterized by some points, could be applied in real time. Such a sparse solution has been proposed by Shi-Tomasi [13] and it is commonly used in computer vision because of its simplicity and its low computational cost. Our own method is based also on this method as a valid and confirmed procedure, that can be applied in a real time context during navigation. Among others, let us cite the work presented in [10] where authors have used the optical flow field to leverage the difference in appearance between objects at close range and the same objects at more distant locations. This information allows them to interpret monocular video streams during off-road autonomous navigation and to propose an adaptive road following in unstructured environments. This method has been evaluated for the navigation of an intelligent vehicle in a desertic terrain.

However, once interest points and optical flow are extracted and tracked from an image sequence, it is so important to distinguish which of those tracked points represent moving objects. Clustering techniques are the first basic solution to this question, but unfortunately most of them require initial information about the scene as the number of clusters to find. The success of these methods highly depends on these parameters. T. Veit et. al [14] coped with the same issue for the analysis of short video sequences. They validate a clustering algorithm based on the *a contrario* method [6] which does not need parameter tuning or initial scene information for finding clusters of mobile features. This approach has been also used in [12] to detect moving objects in short sequences; additionally, authors obtain 3D components of feature points to better detect the correspondence between points and moving objects on which they have been extracted. This work presents experimental results on real images, acquired from fixed cameras, so that essential issues of autonomous navigation are not considered.

This paper proposes moving object detection and tracking on a robot navigation context based on KLT tracker and the a contrario clustering. Robot motion is obtained from the Inertial Measurement Units then compensated in order to obtain real velocity of the interest points. The resulted clusters of dynamic points are delimited by a B-spline contour then initialized as moving objects and tracked by a Kalman Filter. At each iteration, image locations are ponderated by a bi-variate pdf function when a point is tracked. This value represents the probability that the location contains a dynamical point, being 0 the most interesting location where to find an interest point.

Next section explains the proposed strategy. KLT procedure used to detect and track interest points is briefly described in section 3. Section 4 describes the main concepts of the *a contrario* theory. The global detection-clustering-tracking approach and the use of the occupancy map in a long sequence of images are presented in section 5. Experimental results on real images are presented and discussed in section 6. Finally last section concludes and explains future works.

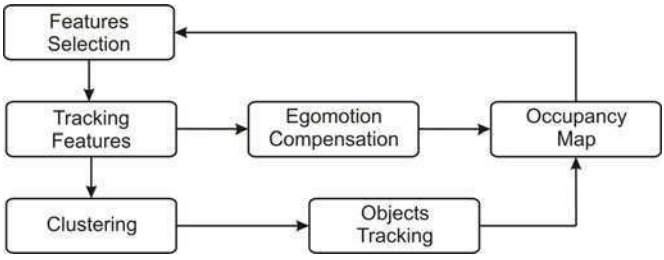


Fig. 1. Algorithm to detect and track multiple objects

2 Overall Detect, Cluster and Track Strategy

General block diagram in Fig. 1 describes the procedure carried out to detect and track multiple moving objects. N initial feature points are selected in the input image(t) using the Shi-Tomasi approach. Feature locations in next image are searched by the KLT tracker, based on correlation and optimization processes. Let us suppose first that no object is tracked; so the process loops on N_{im} successive images executing three task, feature tracking, ego-motion computation and update of the occupancy map. We will call *time of trail*, this set of N_{im} processed images, i.e. the number of images used to accumulate positions and apparent velocities of tracked features. In order to select new features, we seek into the occupancy map the most interesting zones in the image for searching new points. KLT process is executed continuously in this way, while the robot navigates in order to provide new visual information at each time of trail.

At the end of each *time of trail*, only moving KLT features are selected for being grouped by the a contrario clustering method. Once robot ego-motion is compensated, moving KLT features are characterized by a velocity vector of the higher than a threshold set to 1 pixel. Every resulted cluster is directly initialized as a moving object using the most external points in the cluster to calculate a contour. Then dynamic objects are tracked with a position and an apparent velocity estimated by a Kalman Filter. At every iteration, this object could be merged with another previous detected object based on similar velocity and close position. Implementation details of both clustering and merging process are presented in section 4. Finally, object current positions are stored in an occupancy map being the highest values of probability the zones of the image which are not interesting to look for new features.

3 Feature Selection and Tracking

Optical flow procedure used in this work is based on the initial technique proposed in [11] and on the well-known *Select Good Features to Track* algorithm [13]. This technique is largely used in the robotics community because it proposes to match the more salient points, minimizing the processing time. This is a sparse method since only few points are initially selected to describe image content.

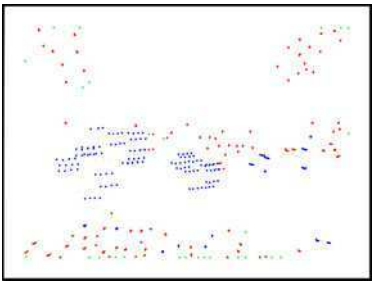


Fig. 2. Accumulated Optical flow in 5 successive images from the sequence of the Fig. 9

Detection of Moving Points. N distinctive feature points are initially extracted from image t_0 by the analysis of spatial image gradient in two orthogonal directions. Locations of these N points in next image are obtained by maximizing a correlation measurement over a small window. Iterative process is accelerated by constructing a pyramid with scaled versions of the input image. Furthermore, rotation, scaling and shearing are applied on each correlation window by optimizing a linear spatial transformation parameters during iterative process. Once displacement vectors are obtained for all initial features, apparent velocities are estimated based on displacement vectors.

Fig. 2 (frames 54 to 58 extracted from the sequence shown in the Fig 9) depicts $N = 150$ initial feature points detected and tracked during a *time of trail*. From these accumulated locations, blue points represent points with long optical flow displacements. For these points the assumption that they belong to a moving object has a high probability.

Perception on Moving Objects. By analyzing optical flow extracted from images acquired from a moving camera in dynamic environments, we deduce that larger is the *time of trail*, better will be the perception of moving objects, but also of the camera motion. In order to minimize the ego-motion impact, we propose to reduce this *time of trail* to 5 images which covers up ego-motion and also, points out independent movements. This can be verified in Fig. 2, a constant movement in the cloud of points can be seen in the middle of the image and the ego-motion is less remarkable. Furthermore, the best advantage of this choice $N_{im} = 5$ is the reduction of the waiting time before performing the *a contrario* clustering task. This strategy works well essentially when the robot velocity is slow so with the aim of reaching higher velocities during navigation we propose to compensate robot ego-motion using an Inertial Measurement Units mounted on the robot.

Once any moving object is detected, the number of its dynamic points is subtracted from the N points tracked permanently by KLT. Thus in following iterations, KLT will select less than N new points. This strategy allows to track only a fix number of N points between KLT and the object tracking process. This rigorous control on the number of points is important in our methodology because long image sequences will be evaluated and the performance directly depends on the number of processed points.

4 A Contrario Method and Merging Process

Visual perception is a complex function that requires the presence of "salient" or "evident" patterns to identify something that "breaks" the continuous motion due to the camera ego-motion. This "salient pattern" corresponds to "meaningful event" detected by the a contrario method [3]. Basic concepts of the a contrario clustering inspired by the Gestalt theory, are exposed in [5] and deeply in [6]. In general, Gestalt theory establishes that groups could be formed based on one or several common characteristics of their elements. In accord to this statement, an a contrario clustering technique (proposed by Veit, et. al. [14]) identifies one group as meaningful if all their elements show a different distribution than an established background random model. Contrary to most of clustering techniques, neither initial number of clusters is required nor parameter has to be tuned. In our context of unknown environment, these characteristics are very favorable.

4.1 Evaluation of a Background Model

We use the background model proposed in [14] which establishes a random organization of the observations. Hence, background model elements are independent identically distributed (iid) and follow a distribution p. The iid nature of random model components propose an organization with not coherent motion present.

Next, given an input vector $V(x, y, v, \theta)$ in R^4, first objective is to evaluate which elements in V show a particular distribution contrary to the established distribution p of the background model (that explains "a contrario" name). To avoid element by element evaluation, first, a binary tree with V elements is constructed using a single linkage method. Each node in the tree represents a candidate group G that will be evaluate in a set of given regions designed by $\mathcal{H}$. This set of regions is formed by different size of hyper-rectangles that will be used to test the distribution of several data groups. Each region H is centered at each element X in the group until finding the region H_X that contains all the group and at the same time makes minimal the probability of the background model distribution. Different size of hyper-rectangles are used in function of data range, in our experiments we use 20 different sizes by dimension. Final measure of meaningfulness (called Number of False Alarms NFA in referenced work) is given by eq 1.

$$NFA(G) = N^2 \cdot |\mathcal{H}| \min_{\substack{X \in G, \\ H \in \mathcal{H}, \\ G \subset H_X}} B(N-1, n-1, p(H_X)) \tag{1}$$

In this equation N represents the number of elements in vector V, $|\mathcal{H}|$ is the carnality of regions and n is the elements in group test G. The term which appears in the minimum function is the accumulated binomial law, this represents the probability that at least n points including X are inside the region test centered in X (H_X). Distribution p consists of four independent distributions, one for each dimension data. Point positions and velocity orientation follow a uniform distribution because object moving position and direction are arbitrary. In other hand, velocity magnitude distribution is obtained directly of the empirically histogram of the observed data. So that, joint distribution p will be the product of this four distributions. A group G is said to be meaningful if $NFA(G) \leq 1$.

Fig. 3. False dynamic object detected on the ground for higher motion velocity

Furthermore two sibling meaningful groups in the binary tree could belong to the same moving object, then a second evaluation for all the meaningful groups is calculated by Eq. 2.To obtain this new measure, we reuse region group information (dimensions and probability) and just a new region that contains both test groups G_1 and G_2 is calculated. New terms are $N' = N - 2$, number of elements in G_1 and G_2, respectively $n'_2 = n_1 - 1$ and $n'_2 = n_2 - 1$, and term $\mathcal{T}$ which represents the accumulated trinomial law.

$$NFA_G\left(G_1, G_2\right) = N^4 \cdot \left|\mathcal{H}\right|^2 \mathcal{T}\left(N', n'_1, n'_2, p_1, p_2\right) \tag{2}$$

Both mesures 1 and 2 represent the significance of groups in binary tree. Final clusters are found by exploring all the binary tree and comparing if it is more significant to have two moving objects G_1 and G_2 or to fusion it in a group G. Mathematically, $NFA(G) < NFA_G(G_1, G_2)$ where $G_1 \cup G_2 \subset G$.

4.2 Filtering False Dynamic Points by Estimated Camera Motion

This method has been first evaluated on numerous image sequences. Even if the method gave good results for slow camera motions [1], it was noted that false dynamic objects could appear for fast camera motions (especially rotations). For instance, on figure 3, a static region is detected as mobile on the ground. How to filter such bad detections? To overcome this problem it is proposed an approach based on the estimated camera motion, provided by inertial sensors fixed on the camera. For this strategy flat ground assumption is considered, so the apparent motions of points located on the ground could be compensated from their estimated motion calculated by projection geometry and IMU data.

First, IMU delivers the information of the 6 rigid motion parameters: 3 coordinates for translation and 3 angles for rotation $[X, Y, Z, \varphi, \alpha, \Psi]$. We assume that robot moves in the $X_W Y_W$ plane of the World reference frame denoted by *W* on Fig. 4. Note that the difference between World and Robot reference frames is only the vector translation. Important rotations in φ and Ψ angles are present between Camera and Image reference frames. Let us note H_{CR} the fixed transform between the Camera and the Robot reference frame, K the intrinsic matrix of the camera, and $M_{t-N,t}$ the estimated robot

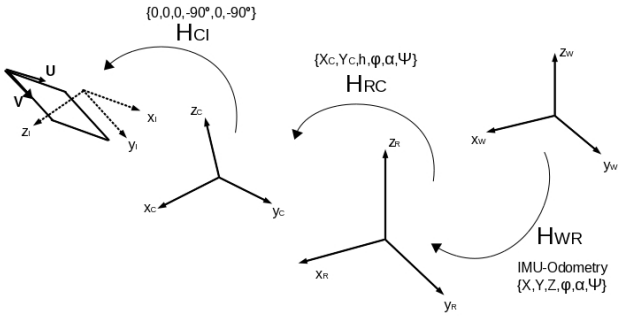


Fig. 4. Coordinate systems for the World (W), the Robot (R), the Camera (C) and the Image (I)

motion between images acquired at time $t-N$ and t. So the camera model is expressed at time $t-N$ in the robot frame by equation 3:

$$[su_{t-N}, sv_{t-N}, s]^T = KH_{CR}\,[X_{t-N}, Y_{t-N}, Z_{t-N}1]^T \tag{3}$$

Then the 3D point can be reconstructed if this point belongs to the ground plane ($Z_{t-N}=0$) or to the infinity plane ($X_{t-N}=\infty$, e.g. 100m). So if this point is static, its projection on the image acquired at time t can be predicted using the equation 4:

$$[s\hat{u}_t, s\hat{v}_t, s]^T = KH_{CR}M_{t-N,t}\,[X_{t-N}, Y_{t-N}, Z_{t-N}, 1]^T \tag{4}$$

in the two last equations s represents the scale. Then the tracked feature is considered as static, so not considered for the clustering process, if $\sqrt{((u_t-\hat{u}_t)^2+(v_t-\hat{v}_t)^2)}$ is less than one pixel.

4.3 Merging Groups

This function is executed only if moving objects have already been detected. O is a set of M objects that will contain all candidate objects for merging evaluation. That is, $O = O_T \cup O_C$ where O_T consists of $(1, 2, ..., k)$ moving objects tracked by Kalman filter, and O_C consist of $(1, 2, ..., l)$ new moving clusters, interpreted either as new moving objects, or part of existing ones . For each object in O, the velocity vector is modeled by the mean of their velocity components in X and Y, respectively represented by μ_{vX} and μ_{vY} . We use these models to evaluate eq.5 that let establish a decision constraint for merging.

$$\min_{\substack{i,\,j\,\in\,M,\\ i\,\neq\,j,\\ O_i,\,O_j\,\subset\,O}} \left(\begin{bmatrix} s(\mu_{vX}(O_i), \mu_{vX}(O_j)) \\ s(\mu_{vY}(O_i), \mu_{vY}(O_j)) \end{bmatrix} \right) < \begin{bmatrix} d_{vX} \\ d_{vY} \end{bmatrix} \tag{5}$$

We evaluate the similarity measure s which performs the subtraction among velocity models for each object in O. Parameters d_{vX} and d_{vY} are constant values set to one pixel in accord with the previous established threshold for detecting moving features in KLT process. This value is chosen in order to conserve the best trade off between

the threshold of moving points in KLT module and the expected bias among object velocities in the scene. This evaluation is carried out in a linked way, where merged groups are removed from O and added as a new object at the end of the list with, obviously, a new corresponding velocity model. This strategy allows the merging of the same objects previously detected with a more enriched model. See for example the image 7a in which the gray car that enters in the scene is detected as two different object, then several images later in image 7b the car is totally detected. From that case, even when both O_T and O_C object velocities are not resulted from the same process, they could be compared because both are based on pixel displacement in the scene (their optical flow).

5 Moving Objects Tracked by a Kalman Filter

Groups found by clustering technique are composed of point locations at different processing time. Hence, to confirm that moving object is still in the scene, only points present in last processed image are taken to initialize each cluster as a moving object in O_C. Therefore only the points located on the current image will be used to model the moving object.

To track moving objects in O_T along the image sequence, Kalman filter prediction evaluates a constant velocity model. To initialize this model a vector state is defined for each moving object detected. Vector state consists of the barycenter of object in X and Y and its velocities are set to μ_{vX} and μ_{vY} values, respectively. We assign the estimated position and velocity calculated by Kalman filter to baricenter of the object. This estimated position is used as the center of a window that will be extracted in the next image in order to search the object points. We called this window the zone of object and its size is a function of previous object limits and a security margin. Once object region is extracted in next image, we carried out a correlation process to find new object location.

5.1 Border Object Initialization and Tracking

The concept of object developed here covers the management of several points which have been evaluated as a part of a moving object. Rigid and non-rigid moving objects could be detected in the scene. As no general size and form of the objects is established, we calculated an initial contour with the dynamic points that forms the cluster. Thus, initial object model is defined by a linear combination of several B-spline functions and a set control points [2].

From the set of dynamic points in the cluster, we choose only the farthest points of the center, i.e. points in the perimeter, to define different B-spline functions of the contour. Figure 5a shows in blue the moving points detected on a dynamic rigid object, then in magenta the points found to describe the object border. An initial contour for a non-rigid object could be also defined by a B-spline function, this is shown in Fig. 5b[1].

[1] Image sequence downloaded from [8] courtesy of EC Funded CAVIAR project/IST 2001 37540.

(a) (b)

Fig. 5. Object contour initialization, for a rigid moving object, image (a), and non-rigid moving object, image (b)

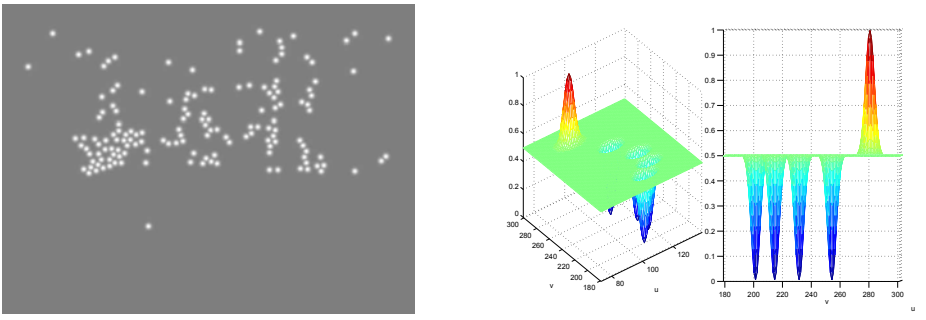
Contour evolution until object real limits is still in development, the main idea is to minimize the internal and external energy based on the intensity or in the color of selected object region. So, up to now the contour is updated in according to the new positions of its control points at each image, found by correlation as the rest of the points in the model object.

Feature Point-based Model. In this work, the task of tracking consists in following each dynamic point in the object with its particular appearance based only on their velocity and position. Thus, model object initialization consists in extracting a window patch (a template) around each point in the image where object was detected. The same number of templates are extracted around the estimated feature location in the next image. Appearance of initial templates in the current image is updated by an affine model. Feature points could be removed from the model if one or more of the following cases happen:

- Feature location is not found by correlation
- Location found is not inside of the object contour
- Displacement and velocity of the points found are not inside of the normally distribution of their respective mean data.

5.2 Occupancy Map

In our algorithm, it is important to add new points that complete the model of the detected moving object and at the same time detect new incoming objects. In a context of unknown environment, feature points should be initially chosen in all the image without highlighting some locations. To overcome this uniform point selection along robot navigation process, a space-time occupancy map is constructed by cells centered on tracked KLT points indicating occupied locations. This map develops the idea of an occupation grid in the sense of higher probabilities represent the locations in the image that are no important to seek for a new point, like *occupied* locations (see Fig. 6a). So, mainly locations in the image with lower probability values in the map will be first used to look for new points.



(a) (b)

Fig. 6. Image (a) shows the initial state of the occupancy map for an image of 640×480 pixels with 150 cells (in white) centered on the feature points. Image (b) shows simulated results of the occupancy map for a tracked point in 5 successive images. Inverse shapes represent the probability in the four previous images, higher shape depicts current point location.

First the probability for all map locations are initialized to $p_0 = 0.5$ that represents the initial fair selection of feature points, that is, $p(u, v, t = 0) = p_0$. When an interesting point is detected in (u, v) at $t \quad 0$, it becomes the center of an *occupied* cell. The probability for *occupied* cells is given by eqs. 6 and 7 where $\mathcal{G}(u, v|\mu_u, \mu_v, \sigma_u, \sigma_v)$ is a 2D Gaussian pdf with parameters $\mu_u = u$, $\mu_v = v$, $\sigma_u = \sigma_v = \frac{r}{3}$, being r the cell size. A cell becomes empty in function of interest point displacements at every iteration: if a tracked point leaves the cell it was using at time $t - 1$ its probability is updated and then this cell is labeled as empty in the current image t. The new cell in which tracked point is located at time t becomes now an occupied cell. The Fig. 6b depicts simulated results of this map construction along five successive images for a tracked point. A square cell of size 11×11 centered on (u, v) point location is used. The inverse effect of function $\alpha(u, v)$ applied to previous point positions is clearly seen in this image. With this probability map we enhance the assumption of new points of incoming object will appear behind its current detected point positions.

$$p(u, v)_t = \alpha(u, v)_t + \mathcal{G}(u, v|\mu_u, \mu_v, \sigma_u, \sigma_v) \tag{6}$$

where $\alpha(u, v)$ function describes the previous probability in the cell according to the previous state of location (u, v), that is:

$$\alpha(u, v)_t = \begin{cases} p_0 & \text{si Cell}(u, v) = \textit{occupied} \\ 1 - p(u, v)_{t-1} & \text{si Cell}(u, v) = \textit{empty} \end{cases} \tag{7}$$

To avoid that this map stores for long time the same probability values in some locations, the map is reset each 2 times of trail process, except for the current object tracking locations.

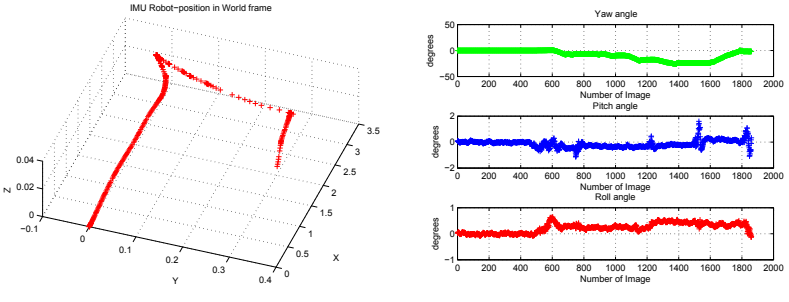
6 Experimental Results and Discussions

Proposed strategy have been implemented in C, C++ and TCL and included as a module into a framework for developing algorithms in robotics. Robot navigation was performed in indoor and outdoor context with a camera mounted on the robot (640×480 at $10Hz$). The number N of tracked points by KLT is set to 150. Robot odometry is obtained from a file for each image time, this information let to increase the robot velocity.

The Fig. 7 shows the algorithm performance with object occlusions. In this case the car which is closer to the robot is well detected and tracked. Some problems to perfectly detect the second car after the occlusion occur and image 7c shows that the algorithm divides it in 2 different moving objects.

(a) Image 106 (b) Image 136 (c) Image 176

Fig. 7. Algorithm results for occlusion objects. Gray car is detected by two different regions then it is totally detected several images later. A second car that crosses the scene is detected before and after its total occlusion.



(a) (b)

Fig. 8. IMU information obtained from image sequence of Fig. 9. Image (a) shows robot trajectory performed on the XYZ World reference frame. Image (b) shows information about the three rotation angles (*yaw*, *pitch*, *roll*) .

Second test was carried out on a sequence of 1900 images. Fig. 8 shows robot position in the world frame and its the *yaw*, *pitch* and *roll* angles given by the odometry measures. Note that rotation angles show negligible change values until image 400, so we have selected 6 representatives images from 600 image, in order to illustrated more

(a) Image 644 (b) Image 665 (c) Image 720

(d) Image 1121 (e) Image 1144 (f) Image 1197

Fig. 9. Experimental results in outdoor environment. First row shows moving object detection and tracking for a car that moves in the same sens of the robot, furthermore it does not cross the field of view of the robot. On the second row: moving object detection before the occlusion is successfully handle by our proposed method.

real and common robot motions like rotations and translations at the same time. Unlike to results shown in Fig. 7, the first row of Fig. 9 shows the detection and tracking of a car that not crosses the field of vision of the robot (*fov*). The car is initially detected on front right tire level, after when this part is hidden, the rear of the car is detected and tracked until its motion is not more perceived on the horizon. At the same time robot starts to turn left (see significant changes in the *yaw* angle) so this shows successfully detection results under real challenge situations. In the second row, on image 9d the car is moving towards the robot but it continues to appear only on the left side of the *fov* of the robot. After, the image 9e shows the detection of both cars before the occlusion. Finally, the dark color car is detected and tracked with success until it is out of the *fov*.

An extension of tests is carried out for evaluate the development of our algorithm during faster robot displacements. The Fig. 10 shows some images from a sequence acquired during robot navigation around the laboratory. The velocity of the robot is around 2 m/s, unlike the last two resulted sequences that were acquired at around 1 m/s. Middle image on this figure shows the successful detection of a truck that moves towards the robot. However, a noticeable ”dynamic” object is detected on the right side of the image. We have been interested in showing this image because that proves the ideal assumptions that we have established. That is, even when our ego-motion strategy (see section 4.2) has been proposed to filter out moving objects caused by the robot motion, this object is detected because it is located on the sidewalk, that is not at the same level of the robot XY plane. Thus, to overcome this restriction in our algorithm, a strategy that involves mobile 3D points projection of the KLT features will be carried out. Thus, camera motions could be corrected only *a posteriori*, using

(a) Image 159 (b) Image 169 (c) Image 189

Fig. 10. Outdoor experimental results at higher velocity. Two dynamic objects are detected: a false moving object on the ground and a real moving truck (see the text for a more detailed explanation).

a SLAMMOT module that reconstructs these 3D points, considered either as static or mobile. A Maximum Likelihood test will be then applied in order to select the more probable between these two hypothesis.

7 Conclusions

Experimental results show that even with few images, it is possible to detect rigid moving objects by a spatio-temporal analysis of features. Object model is enlarged thanks to prior knowledge managed by the proposed occupancy map. This map is successfully used during the active search of feature points because it mainly highlights zones that certainly contain new moving interesting points. Our tests are performed off line on recorded sequences; however, the global algorithm works fast and could process images at 10Hz. The clustering method is the highest time consuming in the global process; for that reason, the number of points to be grouped by the clustering method, should be no more to 150 points. Thus, the trade-off between image size and that number of points guarantees the highest performance in overall strategy. This work is being extended to change the current feature point-based model for an active region-model in which functions of higher level as object identification and classification could be performed. In the general case, it is not possible to remove camera motions from the apparent motions of tracked features. So a global strategy to avoid ego-motion detection and non rigid moving objects is being integrated based on monocamera SLAM approach. An interchange of 3D and 2D points information between SLAM and our MOT process will be continuously carried out giving a cooperative sense to our new proposed strategy.

Acknowledgements. This work has been performed in the context of the RINAVEC project funded by ANR, the french *Association Nationale de la Recherche*. It has been supported by the scholarship 183739 of the Consejo Nacional de Ciencia y Tecnología (CONACYT), the Secretaría de Educación Pública and by the mexican government.

References

1. Almanza-Ojeda, D., Devy, M., Herbulot, A.: Incremental detection and tracking of moving objects by optical flow and a contrario method. In: Proc. International Conference on Computer Vision Theory and Applications (VISAPP 2010), Angers, France (May 2010)

2. Brigger, P., Hoeg, J., Unser, M.: B-spline snakes: a flexible tool for parametric contour detection. IEEE Transactions on Image Processing 9(9), 1484–1496 (2000)
3. Cao, F., Delon, J., Desolneux, A., Musé, P., Sur, F.: A unified framework for detecting groups and application to shape recognition. Journal of Mathematical Imaging and Vision 27(2), 91–119 (2007)
4. Davison, A.: Real-time simultaneous localisation and mapping with a single camera. In: Int. Conf. on Computer Vision, pp. 1403–1410 (2003)
5. Desolneux, A., Moisan, L., Morel, J.M.: A grouping principle and four applications. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(4), 508–513 (2003)
6. Desolneux, A., Moisan, L., Morel, J.M.: From Gestalt Theory to Image Analysis A Probabilistic Approach, vol. 34. Springer, Heidelberg (2008)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004); ISBN: 0521540518
8. `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`: (Mai 2010)
9. Hwangbo, M., Kim, J., Kanade, T.: Inertial-aided KLT feature tracking for a moving camera. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, pp. 1909–1916 (2009)
10. Lookingbill, A., Lieb, D., Thrun, S. (eds.): Autonomous Navigation in Dynamic Environments, Springer Tracts in Advanced Robotics, vol. 35, pp. 29–44. Springer, Heidelberg (2007)
11. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (darpa). In: Proc. 1981 DARPA Image Understanding Workshop, pp. 121–130 (April 1981)
12. Poon, H.S., Mai, F., Hung, Y.S., Chesi, G.: Robust detection and tracking of multiple moving objects with 3D features by an uncalibrated monocular camera. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2009. LNCS, vol. 5496, pp. 140–149. Springer, Heidelberg (2009)
13. Shi, J., Tomasi, C.: Good features to track. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 593–600 (Juin 1994)
14. Veit, T., Cao, F., Bouthemy, P.: Space-time a contrario clustering for detecting coherent motion. In: IEEE Int. Conf. on Robotics and Automation, ICRA, Roma, Italy, pp. 33–39 (2007)
15. Vu, T.V., Aycard, O.: Laser-based detection and tracking moving objects using data-driven markov chain monte carlo. In: IEEE Int. Conf. on Robotics Automation (ICRA), Kobe, Japan (2009)

Part III

Signal Processing, Systems Modeling and Control

Adaptive and Robust Linearizing Control Strategies for Fed-Batch Cultures of Microorganisms Exhibiting Overflow Metabolism

Laurent Dewasme[1,⋆], Daniel Coutinho[2], and Alain Vande Wouwer[1]

[1] Automatic Laboratory, Faculty of Engineering of the University of Mons
Boulevard Dolez, 31, 7000 Mons, Belgium
[2] Department of Automation and Systems, Federal University of Santa Catarina
476 Florianopolis, 88040-900, Brazil
{Laurent.Dewasme,Alain.VandeWouwer}@umons.ac.be,
Coutinho@das.ufsc.br

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 the LMI framework so as to ensure robust stability and performance. Fed-batch cultures of yeast and bacteria are considered as application examples.

Keywords: Nonlinear robust control, Adaptive Control, Fermentation Process, Biotechnology.

1 Introduction

1.1 Context and Motivation

The culture of host recombinant microorganisms is nowadays a very important way of producing biopharmaceuticals. Fed-batch operation is popular in industrial practice since it is advantageous from an operational point of view to feed the cells (also called biomass) progressively as they grow, instead of overfeeding the bioreactor, which can inhibit the cell growth. The off-line determination of the feeding profile is usually suboptimal as some security margin has to be provided in order to avoid this possible excess of substrate leading to the accumulation of inhibitory byproducts (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 byproducts, a closed-loop solution is required, and a wide diversity of approaches, e.g., [1], [11], [4], [13], [12], [7], [18], [8] have been considered.

⋆ 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.

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 283–305.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

As discussed in [4], several control strategies can be applied to define an optimal feeding profile (usually considered as the sole manipulated variable in most of the published studies). In order to maintain the system in optimal conditions, it is possible to control the RQ (respiratory coefficient) of the cells using gas analyzer measurements, to keep the system at a very low critical substrate concentration, or to control the byproduct concentration at a sufficiently low level.

For example, in [18] and [8], two extremum-seeking strategies regulating the substrate level around the optimal critical value are investigated. This kind of approach can however only be applied when this critical value is measurable, which is generally difficult for yeasts and bacteria cultures considering the sensitivity of the probes currently available on the market.

In [1], a classical PID with gain scheduling is used to control the byproduct concentration. However, this investigation shows that a PID controller is unable to track an exponential trajectory as small tracking errors evolves exponentially.

In [12] and [7], a robust linear controller is applied to regulate the byproduct concentration. The main advantage of this controller is to require minimum a priori knowledge, i.e. one yield coefficient and the measurement of the byproduct concentration.

Among the published studies, linearizing control [2] is a very popular approach, which has been applied succesfully in a number of case studies (see [11],[4],[13]). However, linearizing control is by essence model-based and, therefore, requires the knowledge of an accurate model. As bioprocess models are generally uncertain, adaptation is required using observer-based or least-squares strategies. Whereas parametric adaptation is a simple approach, it does not guarantee stability in the presence of unmodeled dynamics or too high noise levels.

In this paper, another approach is considered, which is based on nonlinear robust control and the use of Linear Matrix Inequalities (LMIs) to design the free linear dynamics so as to ensure robust stability and performance in presence of model uncertainties. A comparison of the adpative and robust control approaches is provided in terms of implementation, and simulation tests show the respective advantages and limitations of both strategies.

2 Model and Control Objectives

2.1 Model Description

In this section, we first consider a generic mechanistic 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:

$$\text{Substrate oxidation :} \quad k_{S1}\mathrm{S} + k_{O1}\,\mathrm{O} \overset{r_1 X}{\rightarrow} k_{X1}\,\mathrm{X} + k_{C1}\,\mathrm{C} \tag{1a}$$

$$\text{Substrate fermentation :} \quad k_{S2}\mathrm{S} + k_{O2}\,\mathrm{O} \overset{r_2 X}{\rightarrow} k_{X2}\,\mathrm{X} + k_{P2}\,\mathrm{P} + k_{C2}\,\mathrm{C} \tag{1b}$$

$$\text{Byproduct oxidation :} \quad k_{P3}\mathrm{P} + k_{O3}\,\mathrm{O} \overset{r_3 X}{\rightarrow} k_{X3}\,\mathrm{X} + k_{C3}\,\mathrm{C} \tag{1c}$$

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

$$r_1 = \frac{\min\left(r_S, r_{S_{crit}}\right)}{k_{S1}} \tag{2a}$$

$$r_2 = \frac{\max\left(0, r_S - r_{S_{crit}}\right)}{k_{S2}} \tag{2b}$$

$$r_3 = \frac{\max\left(0, \frac{k_{os}(r_{S_{crit}} - r_S)}{k_{op}} \frac{P}{P+K_P}\right)}{k_{P3}} \tag{2c}$$

Note that these specific consumption rates are divided, for each reaction, by the corresponding substrate yield coefficient (k_{S1} and k_{S2} for the main substrate, usually glucose, in the first two reactions and k_{P3} for the substitute carbon source, the byproduct, in the third reaction) in order to normalize the consumption mechanism with respect to the substrate source. For instance, the substrate consumption rate of the first reaction is $k_{S1}r_1$ which is equal to r_S or $r_{S_{crit}}$ if the oxidative capacity is completely exploited, while the corresponding biomass growth rate is $k_{X1}r_1$ which is respectively equal to $\frac{k_{X1}}{k_{S1}}r_S$ or $\frac{k_{X1}}{k_{S1}}r_{S_{crit}}$. The yield coefficient ratio $\frac{k_{X1}}{k_{S1}}$ illustrates this normalization of the growth rate with respect to the substrate consumption. The kinetic terms associated with the substrate consumption r_S and the critical substrate consumption $r_{S_{crit}}$ (function of the cells oxidative or respiratory capacity r_O) are given by:

$$r_S = \mu_S \frac{\mathrm{S}}{\mathrm{S} + K_S} \tag{3a}$$

$$r_{S_{crit}} = \frac{r_O}{k_{os}} = \frac{\mu_O}{k_{os}} \frac{\mathrm{O}}{\mathrm{O} + K_O} \frac{K_{iP}}{K_{iP} + P} \tag{3b}$$

These expressions take the classical form of Monod laws where μ_S and μ_O are the maximal values of specific growth rates, K_S and K_O are the saturation constants of the corresponding element, and K_{iP} is the inhibition constant. k_{os} and k_{op} represent the coefficients characterizing respectively the yield between the oxygen and substrate consumptions, and the yield between the byproduct and oxygen consumptions. In order to illustrate the role of k_{os} and k_{op}, consider for instance the oxygen consumed in the first two reactions (1a) and (1b). As shown by Fig. 1, a certain substrate quantity equal to $k_{S1}r_1$ is oxidized using an equivalent oxygen quantity $k_{O1}r_{O1} = k_{S1}r_1$ where r_{O1} can be seen as the oxygen consumption rate in the first reaction. In a similar way, the equivalent substrate and oxygen quantities required by the second reaction are equal and respectively defined as $k_{S2}r_2$ and $k_{O2}r_{O2}$ where r_{O2} is the oxygen consumption rate in the second reaction. In order to link r_S to r_O, a global yield coefficient k_{os} is defined as $r_O = k_{os}r_S$. The introduction of k_{op} in the model follows therefore the same reasoning for the byproduct. Nevertheless, note that for particular cells which do not need oxygen in (1b), k_{os} is sometimes summarized to k_{O1} and, for analogous reasons, k_{op} to k_{O3}. As our aim is to provide a general representation of overflow metabolism, this notation is used in the following sections.

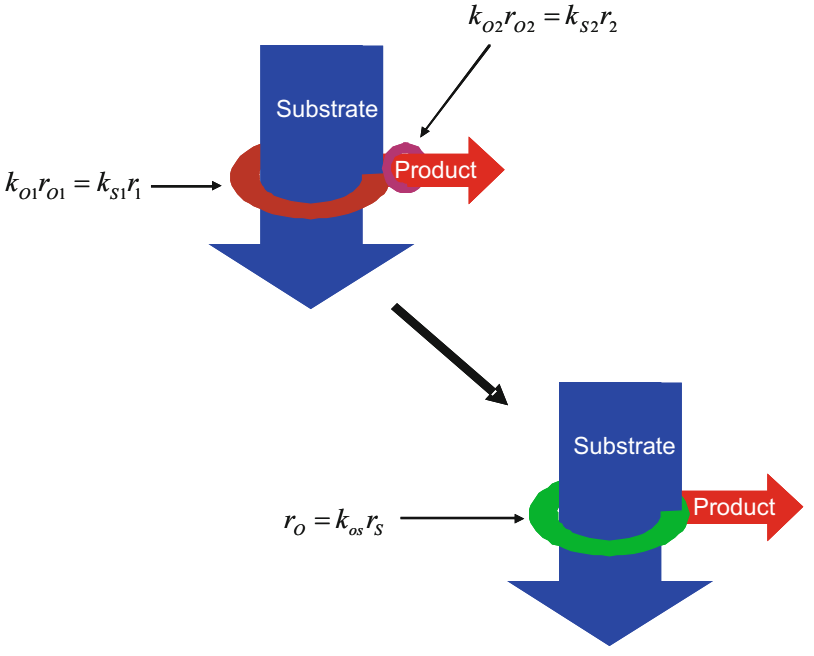


Fig. 1. Schematic representation of the simplified kinetic model using k_{os} as a global yield coefficient between the substrate and the oxygen consumptions

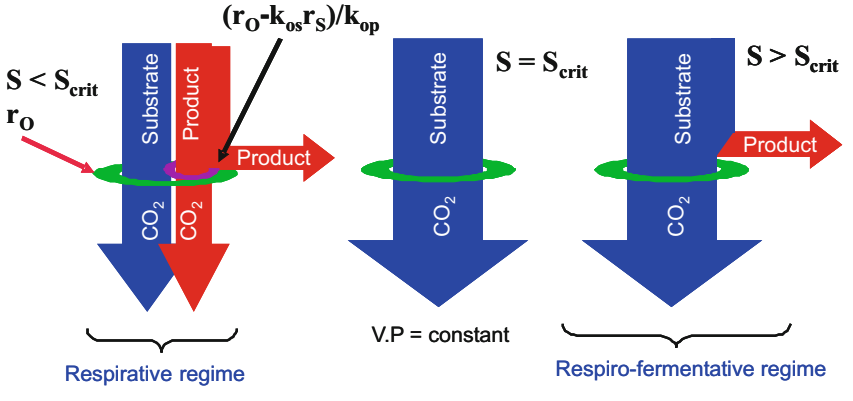


Fig. 2. Illustration of Sonnleitner's bottleneck assumption ([16]) for cells limited respiratory capacity

Kinetic model (2) is based on Sonnleitner's bottleneck assumption ([16]) which was applied to a yeast strain *Saccharomyces cerevisiae* (Figure 2). This model explains that during a culture, the cells are likely to change their metabolism because of their limited oxidative capacity. When the substrate is in excess (concentration $S > S_{crit}$ and the glucose consumption rate $r_S > r_{S_{crit}}$), the cells produce a byproduct P through the fermentative pathway, and the culture is said in (respiro-) fermentative (RF) regime.

On the other hand, when the substrate becomes limiting (concentration $S < S_{crit}$ and the glucose consumption rate $r_S < r_{S_{crit}}$), the available substrate (typically glucose), and possibly the byproduct 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:

$$\frac{dX}{dt} = (k_{X1}r_1 + k_{X2}r_2 + k_{X3}r_3)X - DX \tag{4a}$$

$$\frac{dS}{dt} = -(k_{S1}r_1 + k_{S2}r_2)X + DS_{in} - DS \tag{4b}$$

$$\frac{d}{d} = (k_{P2}r_2 - k_{P3}r_3)X - DP \tag{4c}$$

$$\frac{dO}{dt} = -(k_{O1}r_1 + k_{O2}r_2 + k_{O3}r_3)X - DO + OTR \tag{4d}$$

$$\frac{dC}{dt} = (k_{C1}r_1 + k_{C2}r_2 + k_{C3}r_3)X - DC - CTR \tag{4e}$$

$$\frac{dV}{dt} = F_{in} \tag{4f}$$

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:

$$OTR = k_L a(O_{sat} - O) \tag{5a}$$

$$CTR = k_L a(C - C_{sat}) \tag{5b}$$

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

The optimal operating conditions that maximize the biomass productivity are at the boundary of the two regimes. In these conditions, the fermentation and byproduct oxidation rates are equal to zero and, from (2):

$$r_1 = min(r_S, r_{S_{crit}}) \tag{6a}$$

$$r_2 = max(0, r_S - r_{S_{crit}}) \tag{6b}$$

where

$$r_{S_{crit}} = \frac{r_O}{k_{os}} \tag{7a}$$

$$r_S = \frac{S}{S + K_S} \tag{7b}$$

$$r_O = \mu_O \frac{O}{O + K_O} \frac{K_{iP}}{K_{iP} + P} \tag{7c}$$

the following relations hold:

$$r_1 = r_S = r_{S_{crit}} = \frac{r_O}{k_{os}} \tag{8a}$$

$$r_2 = 0 \tag{8b}$$

Expression (7a) shows that the respiratory capacity has an influence on the critical substrate concentration level. For illustration purposes, Fig.3 shows a simulation of a fed-batch yeast culture where the substrate concentration in the culture medium is regulated around a constant theoretical value $S_{sp} = 0.0226g/l$. This constant value is based on the assumption that the respiratory capacity would not be influenced by the ethanol level ($r_O = \mu_O \frac{O}{O+K_O}$ so that, following (8a), $r_1 = r_{S_{crit}} = r_S$, $r_2 = 0$ and $S_{sp} = S_{crit}$). As this assumption is not correct in practice, ethanol is produced during the batch, thus inhibiting the respiratory capacity and affecting the optimal glucose level, and the biomass growth rate is lower than expected ($r_O = \mu_O \frac{O}{O+K_O} \frac{K_{iE}}{K_{iE}+E}$ so that, following (6), $r_1 = r_{S_{crit}} < r_S$, $r_2 \neq 0$ and $S_{sp} \neq S_{crit}$). A simple regulation strategy, i.e., a regulation that does not adapt the glucose setpoint according to the respiratory capacity variations, does not allow to avoid the production of ethanol, leading to a poor level of productivity (while, as demonstrated in the following, more than $80\ g/l$ of biomass can be obtained within $30\ h$ with glucose setpoint adaptation, only $30\ g/l$ are obtained in Fig. 3).

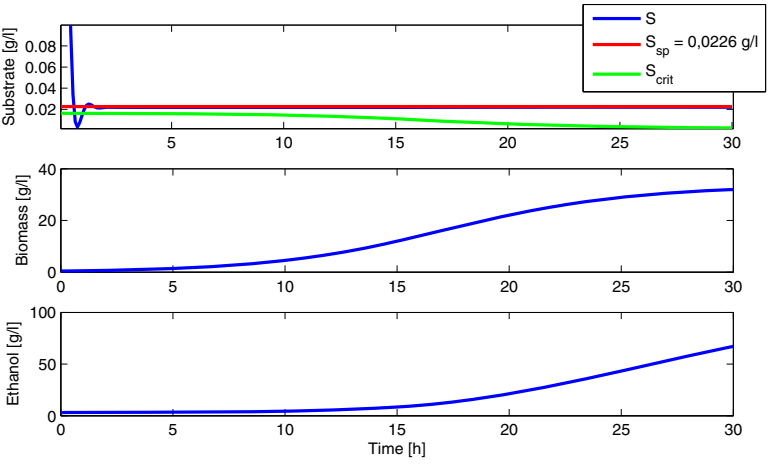


Fig. 3. Simulation of a fed-batch process controlled at a constant S_{sp} value

Consequently, after a trivial mathematical manipulation of (8a) using the Monod law $r_S = \mu_S \frac{S}{S+K_S}$, a relation between the critical substrate concentration level and the cell respiratory capacity is obtained as:

$$S_{crit} = \frac{K_S r_O}{k_{os}\mu_S - r_O} \tag{9}$$

Fig. 4 shows a plot of this relation where the point $[0, 0]$ corresponds to a totally inhibited respiratory capacity, preventing any growth, and the point $[r_{O_{max}}, S_{crit_{max}}]$ corresponds to maximum productivity (i.e. absence of metabolite product in the culture medium and a sufficient level of oxygenation). Obviously, the presence of the byproduct in the culture medium can decrease the respiratory capacity and in turn the value of the critical substrate concentration. Moreover, the estimation of the critical substrate

level S_{crit} requires additional measurements (P, O) and a perfect knowledge of K_S, k_{os}, μ_S, K_O, μ_O and K_{iO}, which are generally uncertain.

In order to maintain the system at the edge between the respirative and respiro-fermentative regimes, it would be necessary to determine on-line an estimation of the biological threshold S_{crit} and to control the substrate concentration in the culture medium around a setpoint S_{sp} ideally equal to S_{crit} in order to reach the optimal operating conditions [7].

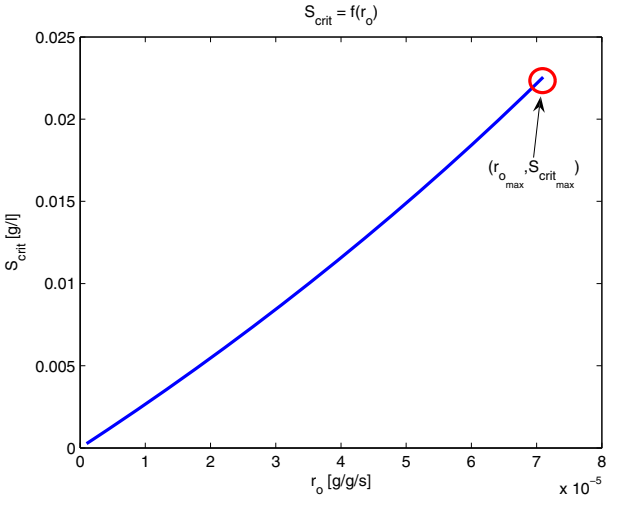


Fig. 4. Critical substrate level (S_{crit}), separating the two regimes, as a function of the respiratory capacity (r_O)

2.2 A Practical 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 byproduct constant in a fed-batch process means that the byproduct 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 byproduct probe. For those practical limitations, a suboptimal strategy is elaborated through the control of the byproduct concentration around a low value P^* depending on the sensitivity of commercially available probes (for instance, a general order for ethanol probe is $0.1g/l$), 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 byproduct at a constant low setpoint, leading to a self-optimizing control in the sense of [14] 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.

3 Linearizing Control Strategy

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

$$\dot{x} = Kr(x)X + Ax - ux + B(u) \tag{10}$$

where $x = [X\ S\ P\ O\ C\ V]'$ is the state vector, $r(x) = [r_1\ r_2\ r_3]'$ 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:

$$K = \begin{bmatrix} k_{X1} & k_{X2} & k_{X3} \\ -k_{S1} & -k_{S2} & 0 \\ 0 & k_{P2} & -k_{P3} \\ -k_{O1} & -k_{O2} & -k_{O3} \\ k_{C1} & k_{C2} & k_{C3} \\ 0 & 0 & 0 \end{bmatrix},\ B(u) = \begin{bmatrix} 0 \\ S_{in}\,u \\ 0 \\ k_La\,O_{sat} \\ k_La\,P_{sat} \\ 0 \end{bmatrix}, \tag{11}$$

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

A feedback linearizing controller is illustrated in Figure 5. In a first step, this controller is derived assuming a perfect process knowledge. The basic idea is to derive a nonlinear controller, which allows a linearization of the process behavior ([4,11]).

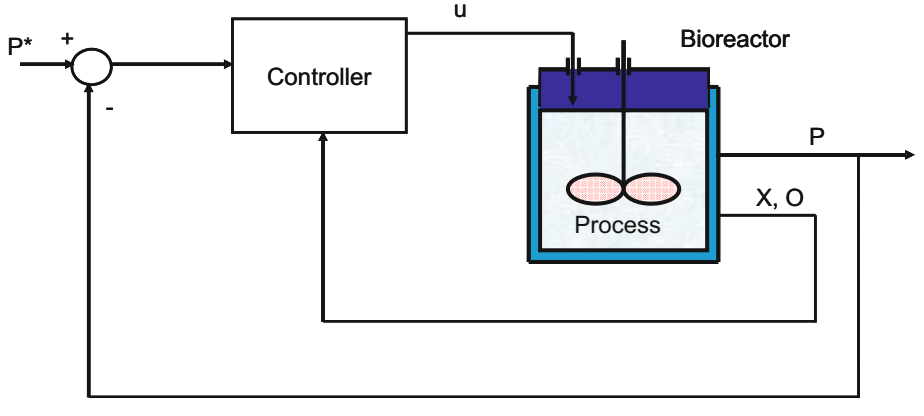


Fig. 5. Linearizing control scheme

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 (4b) becomes:

$$k_{S2} r_2 X = -k_{S1} r_1 X + S_{in} u \tag{12}$$

where r_1 and r_2 are nonlinear functions of S and O as given by (2a-2b).

Replacing $r_2 X$ by (12) in the mass balance equation for P (4c), we obtain:

$$\dot{P} = -\frac{k_{P2} k_{S1}}{k_{S2}} r_1 X - k_{P3} r_3 X - u \left(P - \frac{k_{P2}}{k_{S2}} S_{in} \right) \tag{13}$$

A first-order linear reference model is imposed:

$$\frac{d(P^* - P)}{dt} = -\lambda (P^* - P)\ ,\ \lambda > 0 \tag{14}$$

and a constant setpoint is considered so that:

$$\frac{dP}{dt} = \lambda (P^* - P)\ ,\ \lambda > 0 \tag{15}$$

Equating (15) and (13), the following control law is obtained:

$$F_{in} = V \frac{\lambda (P^* - P) + (\frac{k_{P2} k_{S1}}{k_{S2}} r_1 + k_{P3} r_3) X}{\frac{k_{P2}}{k_{S2}} S_{in} - P} \tag{16}$$

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

3.1 A Classical Adaptive Strategy

In [4], 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 (16).

The main operating assumptions are summarized as follows:

- The byproduct concentration (in this case, ethanol) and the dissolved oxygen and carbon dioxide concentrations are available;
- An exhaust gas analysis (adding 3 new measurements: OTR, CTR and Q_E, the gaseous ethanol outflow rate) is available;
- The stoichiometric parameters are known;
- Kinetics are unknown.

In this earlier work, the biomass is not measured on-line. Nevertheless, nowadays, many biomass probes are readily available and not as expensive as a gas analyzer could be (for instance, a turbidimetric probe or a conductance probe) so that, in this paper, the

biomass concentration measurement X replaces the gas analysis and the dissolved carbon dioxide measurement. Note that in [4], these measurements are used by an asymptotic observer to estimate X. The following adaptive scheme is therefore a simplified version of the original algorithm.

$$F_{in} = V\frac{\lambda(P^* - P) + \hat{\theta}X}{\frac{k_{P2}}{k_{S2}}S_{in} - P} \tag{17}$$

A direct adaptive scheme as described in [2] is used. Consider the following Lyapunov function candidate:

$$V(t) = \frac{1}{2}\left(\tilde{P}^2 + \frac{\tilde{\theta}^2}{\gamma}\right) \tag{18}$$

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:

$$\frac{dV}{dt} = \frac{d\tilde{P}}{dt}\tilde{P} + \tilde{\theta}\frac{d\tilde{\theta}}{dt}\frac{1}{\gamma} \tag{19}$$

Considering (15) and a possible parameter mismatch ($\hat{\theta} \neq \theta$):

$$\frac{d\tilde{P}}{dt} = -\lambda\tilde{P} - \tilde{\theta}X \tag{20}$$

so that (19) becomes:

$$\frac{dV}{dt} = -\lambda\tilde{P}^2 - \tilde{P}\tilde{\theta}X - \tilde{\theta}\frac{d\hat{\theta}}{dt}\frac{1}{\gamma} \tag{21}$$

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

$$\frac{d\hat{\theta}}{dt} = \gamma X\tilde{P} \tag{22}$$

Notice that the adaptive law is asymptotically stable as the negativeness of $\frac{dV}{dt}$ in (19) is guaranteed by (22).

3.2 A Robust Strategy

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

$$\delta = \bar{\theta} - \theta \tag{23}$$

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 (17) using the new expression taken from (23), we obtain:

$$u = F_{in} = V\frac{\lambda(P^* - P) + \bar{\theta}X - \delta X}{\frac{k_{P2}}{k_{S2}}S_{in} - P} \tag{24}$$

which corresponds to the perturbed reference system:

$$\dot{P} = \lambda(P^* - P) - \delta X \tag{25}$$

Borrowing the ideas of the *Quasi-LPV* approach [10], 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:

$$\mathcal{M} : \begin{cases} \dot{P} = -\lambda z - \delta X \\ z = P^* - P \end{cases} \tag{26}$$

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

Let $w = \begin{bmatrix} P^* & X \end{bmatrix}' \subset \mathcal{L}_{2,[0,T]}$ be the disturbance input to the system $\mathcal{M}$, $a(\lambda, \delta) = \begin{bmatrix} \lambda & -\delta \end{bmatrix}$ and $c = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The closed-loop system (26) can be rewritten:

$$\mathcal{M} : \begin{cases} \dot{P} = -\lambda P + a(\lambda, \delta)w \\ z = -P + c\,w\,,\ \delta \in \Delta \end{cases} \tag{27}$$

Consider the finite horizon (for instance, between the instant 0 and the time T) $\mathcal{L}_2$-gain of system $\mathcal{M}$ [9], 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:

$$\|\mathcal{M}_{wz}\|_{\infty,[0,T]} = \sup_{\delta \in \Delta, 0 \neq w \subset \mathcal{L}_{2,[0,T]}} \frac{\|z\|_{2,[0,T]}}{\|w\|_{2,[0,T]}} \tag{28}$$

Thus, the parameter λ is designed based on the $\mathcal{H}_\infty$ control theory [9,15]. Let $\alpha > 0$ be an upper limiting of $\|\mathcal{M}_{wz}\|_{\infty,[0,T]}$. The problem is to find α such that:

$$\min_{\lambda, \delta \in \Delta} \alpha : \|\mathcal{M}_{wz}\|_{\infty,[0,T]} \leq \alpha \tag{29}$$

while ensuring the robust stability of system (27).

This optimization problem can be written in terms of linear matrix inequalities ($LMIs$) and solved using readily available toolboxes, e.g., SeDuMi [17] can be applied to solve the problem. These constraints can be easily obtained via a quadratic Lyapunov function [3]

$$V(P) = P'QP \tag{30}$$

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

The minimization in (29) is then equivalent to:

$$\min\ \alpha\ :\ V(P) \succ 0\ ,\ \dot{V}(P) + \frac{1}{\alpha} z'z - \alpha w'w \prec 0 \tag{31}$$

where, using (27) and (30), the time derivative of $V(P)$ is given by:

$$\begin{aligned}\dot{V}(P) &= \dot{P}'QP + P'Q\dot{P} \\ &= (-\lambda P + aw)'QP + P'Q(-\lambda P + aw) \\ &= -\lambda P'QP + (aw)'QP - \lambda P'QP + P'Qaw \\ &= -2\lambda P'QP + a'w'QP + P'Qaw \end{aligned} \tag{32}$$

Using (32) in (31), the following expression is obtained:

$$\begin{bmatrix} P \\ w \end{bmatrix}' \begin{bmatrix} -2m & Qa \\ a'Q & -\alpha I_{n_w} \end{bmatrix} \begin{bmatrix} P \\ w \end{bmatrix} - \frac{1}{\alpha} zz' \prec 0 \tag{33}$$

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 (27) and Lemma 1, the optimization problem in (29) can be written as follows:

$$\begin{aligned} &\min_{Q,m} \alpha\ :\ \alpha > 0\ ,\ Q = Q' > 0 \text{ and} \\ &\begin{bmatrix} -2m & m & -\delta Q & -1 \\ m & -\alpha & 0 & 1 \\ -\delta Q & 0 & -\alpha & 0 \\ -1 & 1 & 0 & -\alpha \end{bmatrix} \prec 0 \end{aligned} \tag{34}$$

If there exists a feasible solution to the above optimization problem for all δ evaluated at the vertices of Δ, then (29) is satisfied and $\lambda = mQ^{-1}$.

Remark 1: Quadratic Lyapunov functions may be conservative for assessing the stability of parameter-dependent systems [5]. 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 [6] to deal with the nonlinearities at the cost of a larger computational effort.

4 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: $X_0 = 0.4g/l$, $S_0 = 0.5g/l$, $E_0 = 0.8g/l$, $O_0 = O_{sat} = 0.035g/l$, $C_0 = C_{sat} = 1.286g/l$, $V_0 = 6.8l$, $S_{in} = 350g/l$. The second simulation set is dedicated to bacteria cultures with initial and operating conditions: $X_0 = 0.4g/l$, $S_0 = 0.05g/l$, $A_0 = 0.8g/l$, $O_0 = O_{sat} = 0.035g/l$, $C_0 = C_{sat} = 1.286g/l$, $V_0 = 3.5l$, $S_{in} = 250g/l$.

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 [16]. For the bacteria model, parameters values are taken from [13] 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 3 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 (25) via the optimization problem (29), the parameters K_S, K_P, K_O and K_{i_P}, μ_S, μ_O are assumed to be respectively varying with standard

Table 1. Yield coefficients values of Sonnleitner and Käppeli for *S. cerevisiae* model [16]

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 [16]

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	$g\ of\ E/l$
K_{iE}	10	$g\ of\ E/l$

Table 3. Yield coefficients values of Rocha's *E.coli* model [13]

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 [13]

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	$g\ of\ A/l$
K_{iA}	6,952	$g\ of\ A/l$

deviations of 500% and 35% of their nominal values. Simulating the operating conditions of the control strategy in (24), we may infer that $\overline{\delta} = 0.5/3600\ s^{-1}$ and $\underline{\delta} = 0\ s^{-1}$ for yeast cultures and $\overline{\delta} = 0.1/3600\ s^{-1}$ and $\underline{\delta} = 0\ s^{-1}$ for bacteria cultures. In light of (27) and (29), these constraints lead to $\lambda = 0.0056$ and $\lambda = 0.0046$ for yeasts and bacteria, respectively.

Concerning the adaptive control law, $\lambda = 1$ and $\gamma = 0.05$ for yeast cultures while $\lambda = 2$ and $\gamma = 0.25$ for bacteria cultures. Note also that the sampling period is chosen equal to $0.1\ h$.

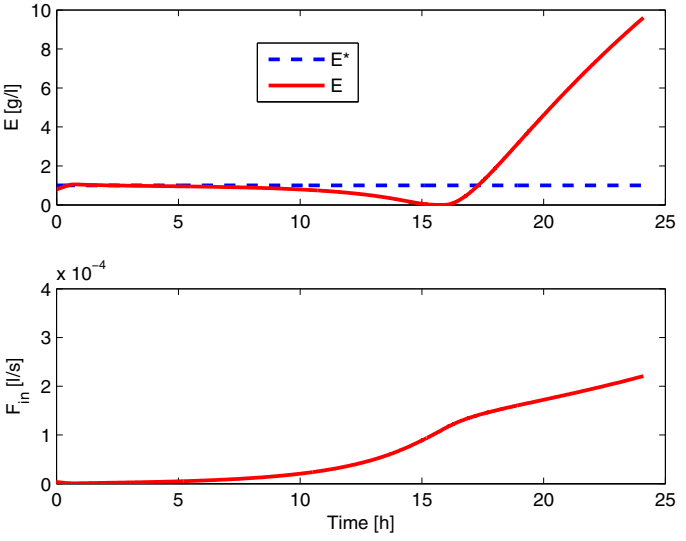


Fig. 6. Yeast cultures – ethanol concentration and feed rate when the controller is designed using a plain linearizing control approach (no adaptation and no robustification) in the presence of modeling errors

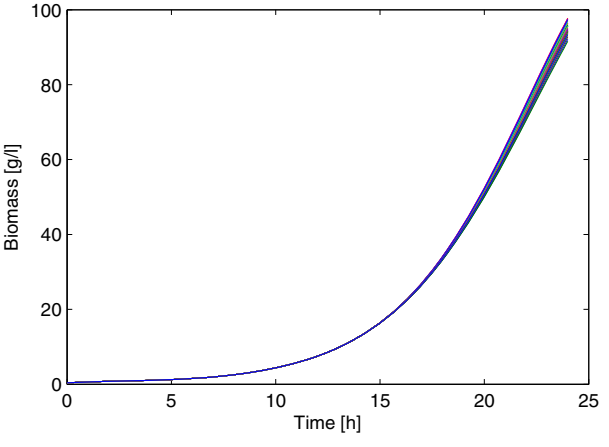


Fig. 7. Yeast cultures – biomass concentrations of 50 runs with random parameter variations and a noise standard deviation of $0.15\ [g/l]$ using the robust control strategy (i.e., $\lambda = 0.0056$)

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 6 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 7 shows now the closed-loop response of biomass X and figure 8 ethanol E concentrations, for 50 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 $0.15\ [g/l]$ and the culture is considered as always evolving in the optimal operating conditions in which $r_1 = \frac{\mu_O}{k_{O1}}$ and $r_3 = 0$ so that the hypothetical parameter $\bar{\theta}$ in (24) is taken as

$$\bar{\theta} = \frac{k_{P2}\tilde{k}_{S1}}{k_{S2}} r_1 + \tilde{k_{P3}} r_3 \approx \frac{\frac{k_{P2}k_{S1}}{k_{S2}}\mu_O}{k_{O1}} \tag{35}$$

In Figure 7, the different curves are more or less indistinguishable 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 even if, in figure 8, ethanol concentrations may vary from 0.5 to $1.5\ g/l$ when the high biomass dynamics is coupled to important model errors. In order to enhance the idea of a negligible influence of these last variations, figure 9 shows the different productivity levels of each run. It is obvious that, from an operating point of view, the results are satisfactory as the productivity remains higher than $98\ \%$ of the optimal value.

Figures 10 and 11 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 12 and 13 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. A way to overcome part of this sensitivty problem is proposed in [4]. The dynamics of ethanol is increasing as the biomass grows, so that the linearizing reference model should take these dynamics into account under the following form:

$$\frac{d(P^* - P)}{dt} = -(\lambda_1 + \lambda_2 X)(P^* - P) \quad \lambda_1, \lambda_2 > 0 \tag{36}$$

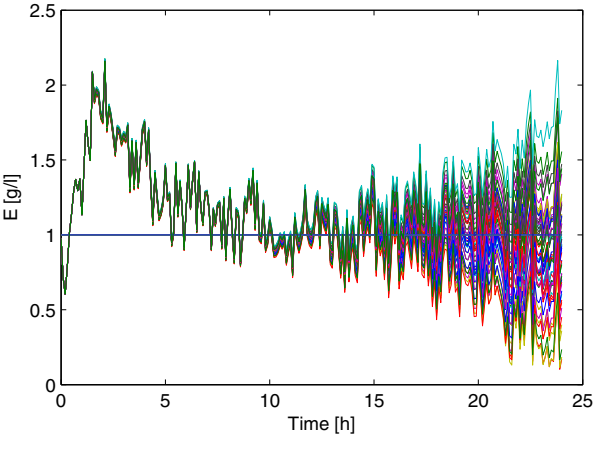


Fig. 8. Yeast cultures – ethanol concentrations of 50 runs with random parameter variations and a noise standard deviation of $0.15\ [g/l]$ using the robust control strategy (i.e., $\lambda = 0.0056$)

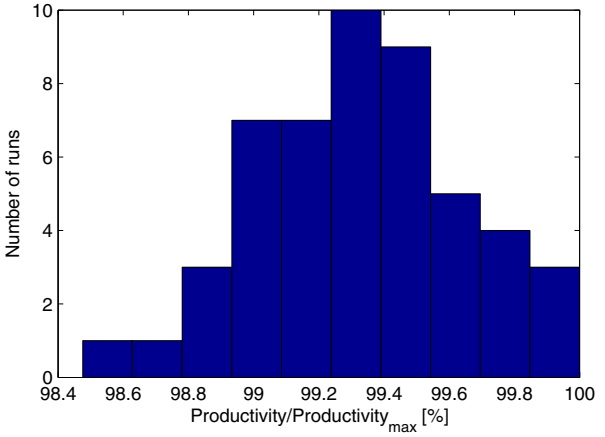


Fig. 9. Yeast cultures – productivity levels of the 50 runs with random parameter variations using the robust control strategy (i.e., $\lambda = 0.0056$)

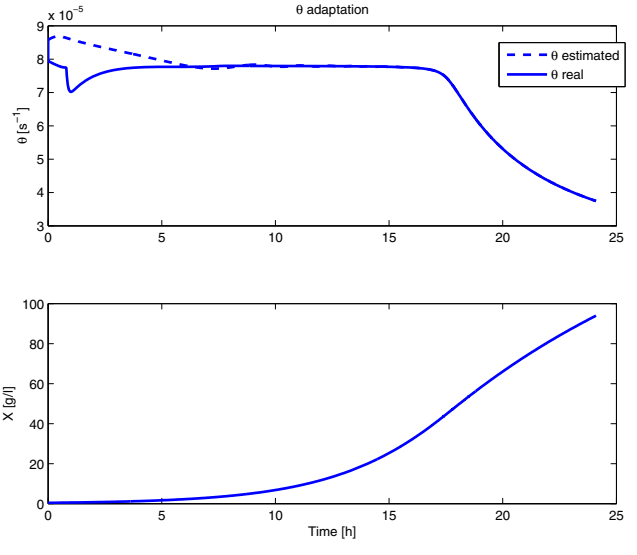


Fig. 10. Yeast cultures – θ adaptation and biomass concentration – adaptive control strategy – no measurement noise

In expression (36), the time constant decreases as the system dynamics (represented by the biomass growth) increases. The new linearizing control law becomes:

$$u = F_{in} = V \frac{(\lambda_1 + \lambda_2 X)(P^* - P) + \theta X}{\frac{k_{P2}}{k_{S2}} S_{in} - P} \tag{37}$$

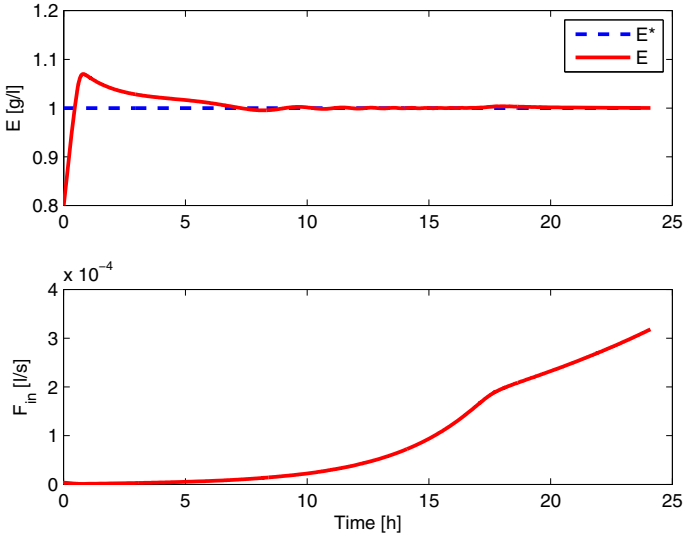


Fig. 11. Yeast cultures – ethanol concentration and feed flow rate – adaptive control strategy – no measurement noise

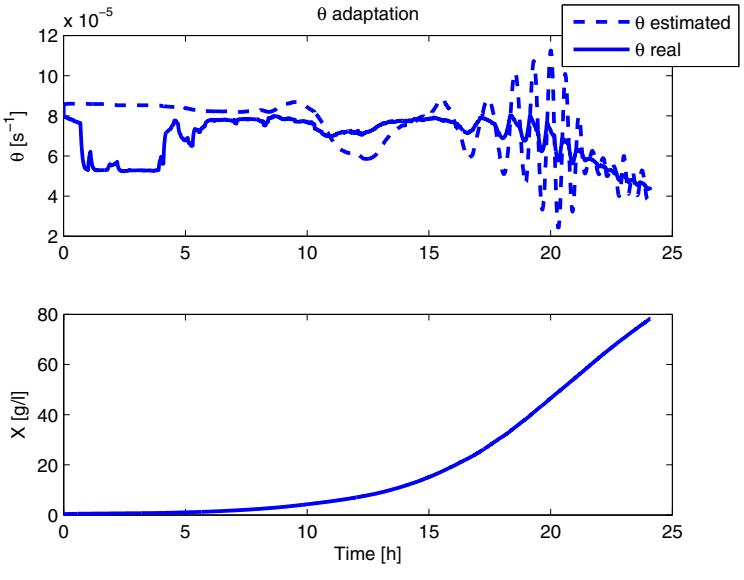


Fig. 12. Yeast cultures – θ adaptation and biomass concentration – adaptive control strategy – noise standard deviation of 0.05 $[g/l]$.

Nevertheless, Figure 14 shows that the adaptive method is still sensitive to the noise level as simulation failures occur for standard deviations higher than $0.125g/l$ (i.e., below the noise level used for the robust strategy).

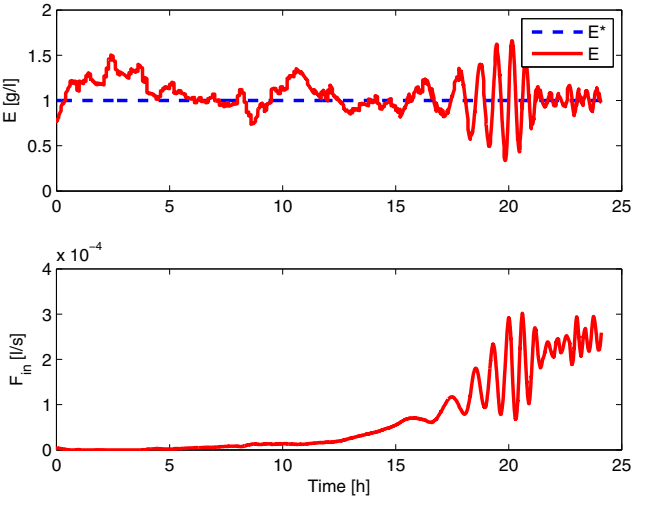


Fig. 13. Yeast cultures – ethanol concentration and feed flow rate – adaptive control strategy – noise standard deviation of 0.05 $[g/l]$

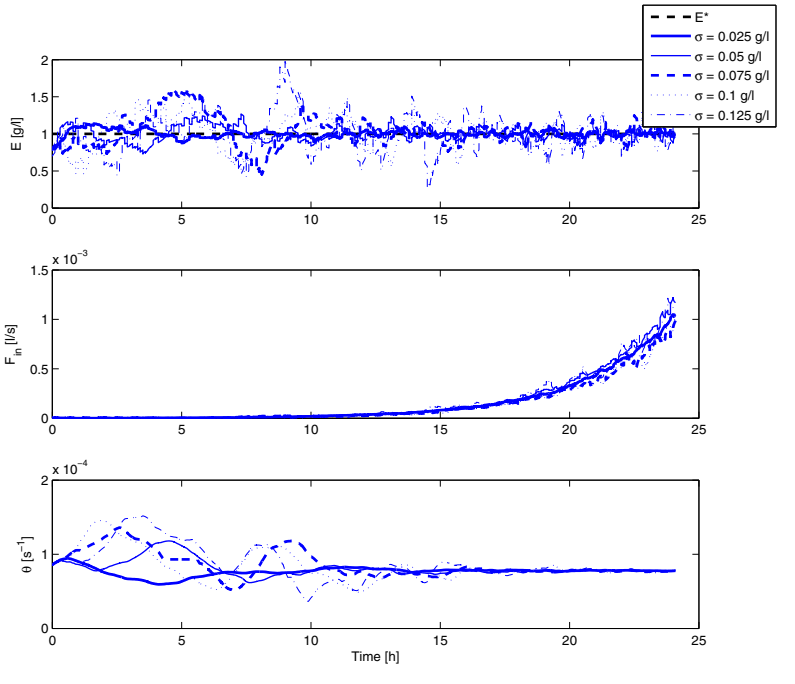


Fig. 14. Yeast cultures – ethanol concentration, feed flow rate and parameter adaptation for 5 different noise levels going from 0.025 to 0.125 g/l. $\lambda_1 = 1$ and $\lambda_2 = 0.2$

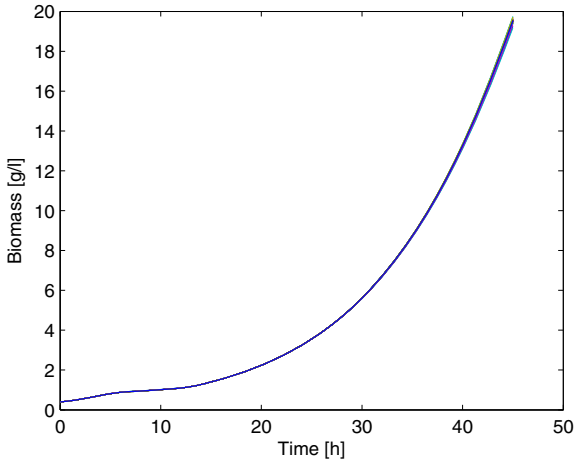


Fig. 15. Bacteria cultures – biomass concentrations of 50 runs with random parameter variations and a noise standard deviation of $0.15\ [g/l]$ using the robust control strategy (i.e., $\lambda = 0.0046$)

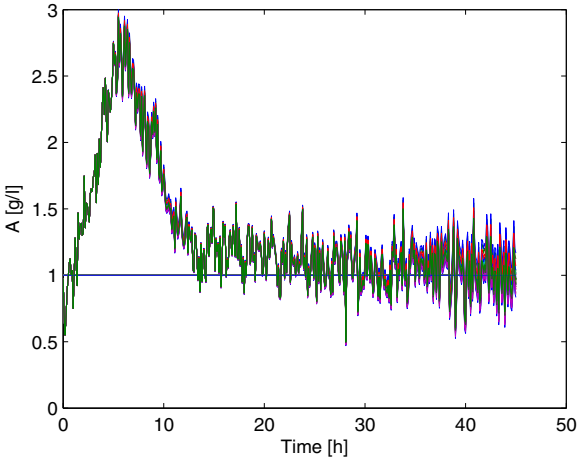


Fig. 16. Bacteria cultures – acetate concentrations of 50 runs with random parameter variations and a noise standard deviation of $0.15\ [g/l]$ using the robust control strategy (i.e., $\lambda = 0.0046$)

However, 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 $90\ g/l$ is obtained within 24 hours.

Figures 15, 16 and 17 show the closed-loop response of biomass X and acetate A concentrations, and the productivity levels for 50 different values of the kinetic parameters which are randomly chosen, in the bacteria cultures under a robust control strategy (noise level of $\sigma = 0.15\ g/l$).

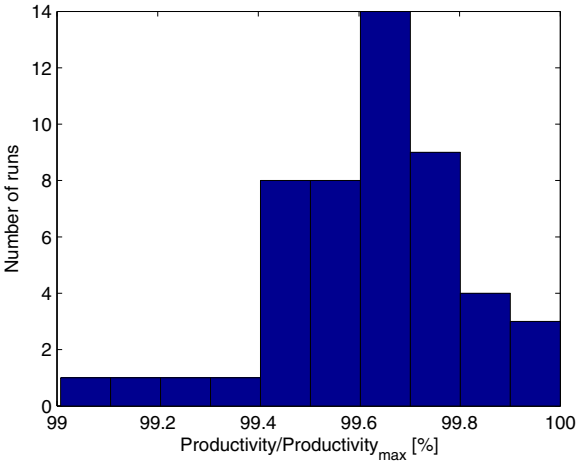


Fig. 17. Bacteria cultures – productivity levels of the 50 runs with random parameter variations using the robust control strategy (i.e., $\lambda = 0.0046$)

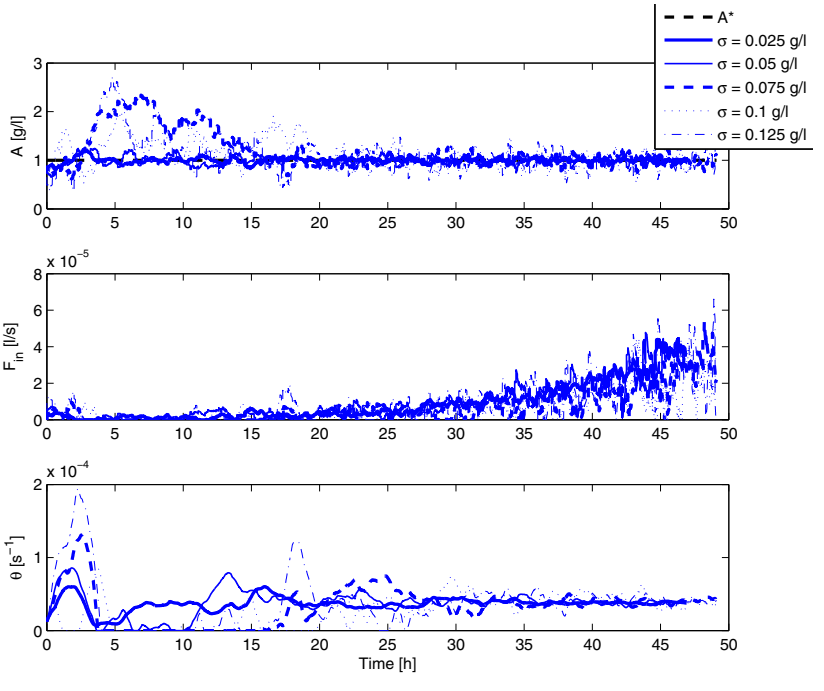


Fig. 18. Bacteria cultures – acetate concentration, feed flow rate and parameter adaptation for 5 different noise levels going from 0.025 to 0.125 g/l. $\lambda_1 = 2$ and $\lambda_2 = 0.4$

Figure 18 show similar simulation runs as in figure 14 with the adaptive strategy where the noise level increases from $\sigma = 0.025$ g/l to $\sigma = 0.125$ g/l. The same comments as in the yeast case concerning the noise sensitivity apply.

Note that, as shown in figure 15, the productivity is lower in the bacteria 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 of the robust strategy are satisfactory in both cases.

5 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.

References

1. Axelsson, J.P., Mandenius, C.F., Holst, O., Hagander, P., Mattiasson, B.: Experience in using an ethanol sensor to control molasses feed-rates in baker's yeast production. Bioprocess Engineering 3, 1–9 (1988)
2. Bastin, G., Dochain, D.: On-Line Estimation and Adaptive Control of Bioreactors, Process Measurement and Control, vol. 1. Elsevier, Amsterdam (1990)
3. Boyd, S., El-Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)
4. Chen, L., Bastin, G., van Breusegem, V.: A case study of adaptive nonlinear regulation of fed-batch biological reactors. Automatica 31(1), 55–65 (1995)
5. Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Robust analysis of lfr systems through homogeneous polynomial lyapunov functions 49(7), 1211–1215 (2004)
6. Coutinho, D., Fu, M., Trofino, A., Danes, P.: L2-gain analysis and control of uncertain nonlinear systems with bounded disturbance inputs. Int'l J. Robust Nonlinear Contr. 18(1), 88–110 (2008)
7. Dewasme, L., Richelle, A., Dehottay, P., Georges, P., Remy, M., Bogaert, P., Vande Wouwer, A.: Linear robust control of s. cerevisiae fed-batch cultures at different scales. Biochemical Engineering Journal (October 2009) (article in press), doi:10.1016/j.bej.2009.10.001
8. Dewasme, L., Vande Wouwer, A., Srinivasan, B., Perrier, M.: Adaptive extremum-seeking control of fed-batch cultures of micro-organisms exhibiting overflow metabolism. In: Proceedings of the ADCHEM Conference in Istanbul, Turkey (July 2009)
9. Green, M., Limebeer, D.: Linear Robust Control. Prentice Hall, Englewood Cliffs (1994)
10. Leith, D., Leithead, W.: Survey of gain-scheduling analysis and design. International Journal of Control 73, 1001–1025 (2000)
11. Pomerleau, Y.: Modélisation et commande d'un procédé fed-batch de culture des levures á pain. Ph.D. thesis, Département de génie chimique. Université de Montréal (1990)
12. Renard, F., Vande Wouwer, A.: Robust adaptive control of yeast fed-batch cultures. Comp. and Chem. Eng. 32, 1238–1248 (2008)
13. Rocha, I.: Model-based strategies for computer-aided operation of recombinant E. coli fermentation. Ph.D. thesis, Universidade do Minho (2003)
14. Skogestad, S.: Control structure design for complete chemical plants. Computers and Chemical Engineering 28(1-2), 219–234 (2004)

15. Skogestad, S., Postlethwaite, I.: Multivariable Feedback Control - Analysis and Design. John Wiley & Sons, New York (2001)
16. Sonnleitner, B., Käppeli, O.: Growth of saccharomyces cerevisiae is controlled by its limited respiratory capacity. Formulation and verification of a hypothesis. Biotechnol. Bioeng. 28, 927–937 (1986)
17. Sturm, J.F., Romanko, O., Pólik, I.: SeDuMi, version 1.1R3 (2006), http://sedumi.mcmaster.ca/
18. Titica, M., Dochain, D., Guay, M.: Adaptive extremum-seeking control of fed-batch bioreactors. European Journal of Control 9(6), 618–631 (2003)

A New Framework for Motion Estimation in Image Sequences Using Optimal Flow Control

Annette Stahl* and Ole Morten Aamo

Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
O.S. Bragstads plass 2D, N-7491 Trondheim, Norway
Annette.Stahl@itk.ntnu.no,
aamo@ntnu.no

Abstract. Application of tools from optimal flow control to the field of computer vision and image sequence processing, has recently led to a new and promising research direction. We present an approach to image motion estimation that uses an optimal flow control formulation subject to a physical constraint. Motion fields are forced to satisfy appropriate equations of motion. Although the framework presented is flexible with respect to selection of equations of motion, we employ the Burgers equation from fluid mechanics as physical prior knowledge in this study. To solve the resulting time-dependent optimisation problem we introduce an iterative method to uncouple the derived state and adjoint equations. We perform numerical experiments on synthetic and real image sequences and compare our results with other well-known methods to demonstrate performance of the optimal control formulation in determining image motion from video and image sequences. The results indicate improved performance.

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

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* [9]. 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 [10]. Variational optical flow methods represent global optimisation

* 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).

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 307–320.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

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 [9] 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* [12,15,1] or *flow-driven* [16,6] way, where the flow is prevented from smoothing across object or motion boundaries, respectively. A systematic classification of these approaches can be found in [19].

Most of the variational approaches incorporate a purely *spatial* regularisation of the flow. However, some efforts have been made to incorporate *temporal* smoothness [13]. The work of [20] 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 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 [7].

The ideas of two existing control approaches that are related to motion computation of image sequences are presented by [14] and [3]. 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 [3] 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 [9] 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 [7] 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 [9] 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 Rerm. We incorporate the regularisation term from [9]

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

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 [18] 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

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

has been studied and successfully applied for many decades in aero- and fluid dynamics [4,8] 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).

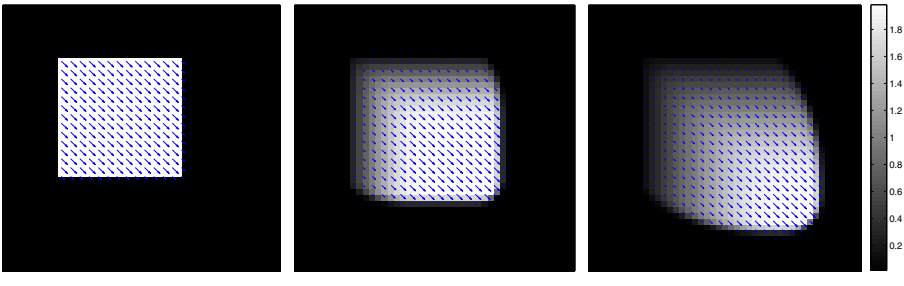


Fig. 1. Illustration of the transportation of a vector field with equation (3) at times $t = 0, 5, 10$. Gray values visualise vector magnitudes. Fictive particles move along a shock front in the lower right direction. In the absence of any further external information, a region of rarefaction arises 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 [7].

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

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

subject to the equations of motion

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

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

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

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.

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

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

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

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.

In the next section we describe how the optimisation problem can be solved.

2.3 Solution of the Optimality System

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. To solve this system we introduce an iterative method which decouples the state and adjoint computation. This results in a gradient descent method which 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. The step length is adaptively adjusted to ensure that the actual energy of the objective functional (4) is smaller than in the previous iteration. Note that we choose the start value for f to be zero in the very first iteration.

Gradient of the Objective Functional. To determine the optimal state and control we use - as mentioned above - a gradient descent method. Therefore, we have to determine the gradient of our objective functional (4) with respect to the control f. The state equation (7) is solved to determine the state u as a function of the control f so that the functional $J(u,f) = J(u(f),f)$ is a function of only the control f. Applying the chain rule to $J(u(f),f)$ we obtain the total derivative of the functional (4):

$$\partial_f J = \frac{\partial J}{\partial u}\frac{du}{df} + \frac{\partial J}{\partial f}. \tag{10}$$

We specify in detail the gradient $\frac{du}{df}$. This term defines the so-called *sensitivities* and describes the change of the state when the control variable changes. To obtain this dependency we can use a variation of the distributed control $f + \varepsilon\tilde{f}$ which is assumed to correspond to the changes of the state u to $u + \varepsilon u_f$. The change u_f is determined by the state system:

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

One obtains for $\varepsilon \to 0$ the sensitivity equation

$$\begin{cases} \partial_t u_f + (u_f \cdot \nabla)u + (u \cdot \nabla)u_f = \tilde{f} & \text{in } \Omega \times [0,T], \\ \partial_n u_f = 0 & \text{on } \Gamma \times [0,T], \end{cases} \tag{11}$$

which describes the fact that an infinitesimal variation of the control in the direction of $\tilde{f}$ induces an infinitesimal variation in the direction of the local velocity u_f.

The change in the functional $J(u, f)$ of (4), effected by an infinitesimal change in the direction $\tilde{f}$ in the control f leads to the gradient of the objective functional:

$$\partial_f J = w + \beta f. \tag{12}$$

This gradient is used within our iterative gradient descent algorithm to update the search direction. Below we sketch the implementation of the gradient method.

2.4 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.

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 [5].

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,

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

```
set u = 0, ε = 10^−8, and s := s_0 (initial step)
repeat
    solve the adjoint equation (8) for w
    update f: f_m = f_{m−1} − s(βf_{m−1} + w)
    solve the state equation (7) for u
    if E(u, f_m) ≥ E(u, f_{m−1}) then
        s := 0.5s
        GOTO 4
    else
        s := 1.5s
    end if
until |E(u, f_m) − E(u, f_{m−1})|/|E(u, f_m)| < ε
```

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 [9] and the dynamic optical flow approach from [17].

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

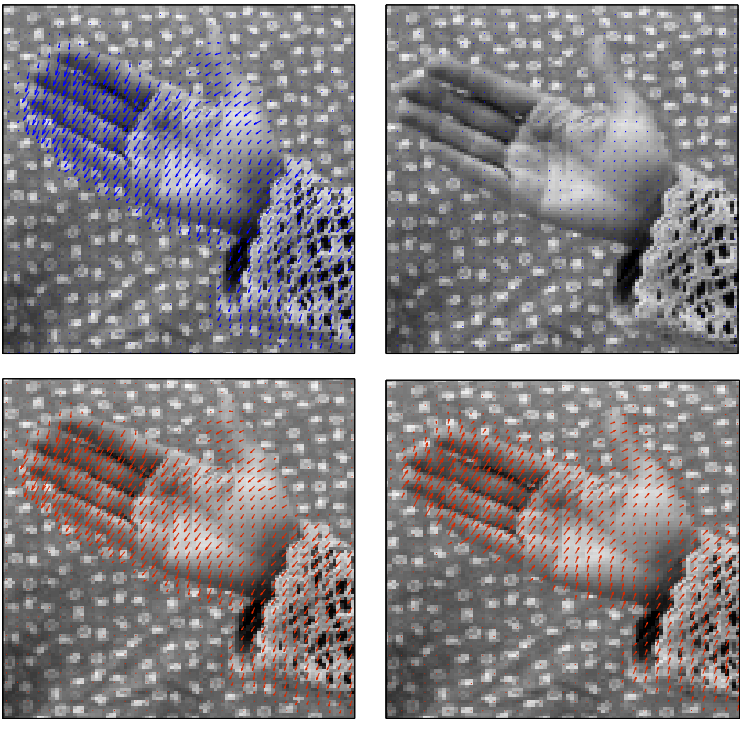


Fig. 2. "Waving hand" sequence: Unexpected events. **Top:** A waving hand stops. The estimated optical flow field u for a starting (left) and stopping (right) event is depicted in blue. **Bottom:** The corresponding control field f is shown in red. The force acts when the hand starts to move (left) and 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: $\alpha = 0.01$, $\beta = 0.0001$).

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.

In particular we evaluate our approach on the grey value versions of the following two image sequences: the "rotating sphere" sequence [11] and the "Yosemite" sequence (available at `ftp://ftp.csd.uwo.ca/pub/vision`).

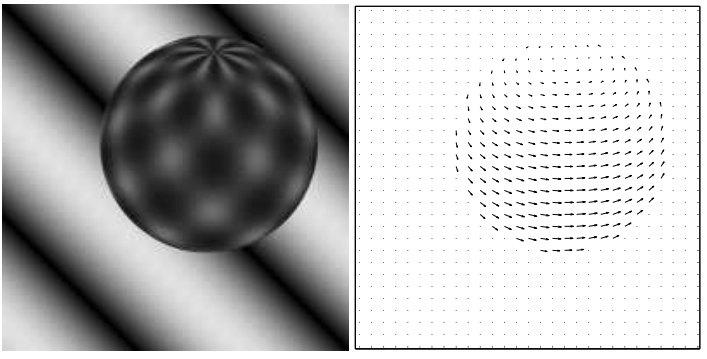


Fig. 3. The synthetic "rotating sphere" sequence. The sphere rotates in front of a stationary background. **Left:** Gray value version of frame 6 that is used in our computations. **Right:** Vector plot of the ground truth data.

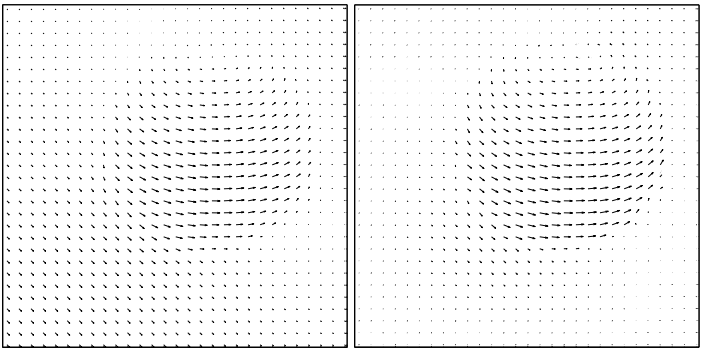


Fig. 4. The synthetic "rotating sphere" sequence. Computational Results for the Horn and Schunck approach and the control based approach. **Left:** Result Horn and Schunck (RMSE = 0.395). **Right:** Result control based approach (RMSE = 0.192).

The "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.

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.

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.

Fig. 5. Left: Yosemite sequence. **Centre:** We added Gaussian noise with standard deviation $\sigma = 40$. **Right:** The shown high quality optical flow field is obtained by the control based optical flow approach (4) - (5) (parameters: $\alpha = 0.05$ and $\beta = 0.000003$).

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 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)

$$RMSE(u_o, u_e) = \frac{1}{|\Omega|} \int_{\Omega} \sqrt{(u_o - u_e)^2} dx \tag{13}$$

and the average angular error (AAE)

$$AAE(u_o, u_e) = \frac{1}{|\Omega|} \int_{\Omega} \arccos\left(\frac{u_o \cdot u_e}{|u_o||u_e|}\right) dx, \tag{14}$$

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 [2]). 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 [9], the spatio-temporal dynamic image motion approach from [17], 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 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

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

noise	approach	α	β	RMSE	AAE
$\sigma = 0$	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°**
$\sigma = 10$	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°**
$\sigma = 20$	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°**
$\sigma = 40$	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°**

error values with increasing noise levels. In contrast to the spatio-temporal dynamic image motion approach [17] 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 incorporates also future knowledge of the image sequence instead of using only past information with a prediction as in [17].

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 [9] and the approach from [17] 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

1. Alvarez, L., Esclarìn, J., Lefebure, M., Sànchez, J.: A PDE model for computing the optical flow. In: Proceedings of CEDYA XVI, pp. 1349–1356 (1999)
2. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. Int. J. of Computer Vision 12(1), 43–77 (1994)
3. Borzi, A., Ito, K., Kunisch, K.: Optimal control formulation for determining optical flow. SIAM J. Sci. Comput. 24(3), 818–847 (2002)
4. Burgers, J.M.: A mathematical model illustrating the theory of turbulence. Adv. Appl. Mech. 1, 171–199 (1948)
5. Colella, P., Puckett, E.G.: Modern Numerical Methods for Fluid Flow. Lecture Notes, Dep. of Mech. Eng., Uni. of California, Berkeley, CA (1998), `http://www.rzg.mpg.de/~bds/numerics/cfd-lectures.html`
6. Deriche, R., Kornprobst, P., Aubert, G.: Optical-flow estimation while preserving its discontinuities: A variational approach. In: ACCV, pp. 71–80 (1995)
7. Gunzburger, M.: Perspectives in Flow Control and Optimization. Society for Industrial and Applied Mathematics (2002)
8. Hirsch, C.: Numerical Computation of Internal and External Flows, vol. I+II. John Wiley & Sons, Chichester (2000)
9. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
10. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (darpa). In: Proc. of the 1981 DARPA Image Understanding Workshop, pp. 121–130 (1981)
11. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow. Comput. Vis. Image Underst. 84(1), 126–143 (2001)
12. Nagel, H.H.: Constraints for the estimation of displacement vector fields from image sequences. In: Int. Joint Conf. on Artificial Intelligence, pp. 945–951 (1983)
13. Nagel, H.H.: Extending the oriented smoothness constraint into the temporal domain and the estimation of derivatives of optical flow. In: Faugeras, O. (ed.) ECCV 1990. LNCS, vol. 427, pp. 139–148. Springer, Heidelberg (1990)
14. Ruhnau, P., Schnörr, C.: Optical Stokes flow: An imaging-based control approach. Experiments in Fluids 42, 61–78 (2007)
15. Schnörr, C.: Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. Int. J. of Computer Vision 6(1), 25–38 (1991)
16. Schnörr, C.: Segmentation of visual motion by minimizing convex non-quadratic functionals. In: 12th Int. Conf. on Pattern Recognition, vol. A, pp. 661–663. IEEE Computer Society Press, Los Alamitos (1994)
17. Stahl, A., Ruhnau, P., Schnörr, C.: A Distributed Parameter Approach to Dynamic Image Motion. In: Int. Workshop on The Representation and Use of Prior Knowledge in Vision, ECCV Workshop (2006)
18. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for TV-11 optical flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Statistical and Geometrical Approaches to Visual Motion Analysis. LNCS, vol. 5604, pp. 23–45. Springer, Heidelberg (2009)

19. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. Int. J. of Computer Vision 45(3), 245–264 (2001)
20. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. J. Math. Imaging and Vision 14(3), 245–255 (2001)

A New Design Procedure of the Pole-Placement and the State Observer for Linear Time-Varying Discrete Systems

Yasuhiko Mutoh

Sophia University, Department of Engineering and Applied Scienses
7-1 Kioicho, Chiyoda-ku, Tokyo, Japan
y_mutou@sophia.ac.jp

Abstract. This paper presents a new design procedure of the pole placement and the Luenberger observer for linear time-varying discrete systems. First, a simple calculation method to derive the pole placement state feedback gain 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. In general, the equivalent transformation of the time-varying linear system does not preserve the stability. The paper also shows the condition of the stability of the closed loop system. Then, this method is applied to the design of the observer. For this purpose, the anti-causal system is introduced. Using the dual property of this system, the observer gain vector is calculated by the proposed simple design technique of the pole placement control.

Keywords: Pole-placement, Observer, Linear time-varying discrete system.

1 Introduction

The observer based pole placement control is the well established basic strategy of linear control systems [1] [2]. As for time-invariant systems, the pole placement state feedback gain for continuous linear time-varying systems can be obtained by tranforming the system into the controllability standard form [3] [4]. However, in the time-varying case, this calculation procedure is complicated. So, for the simplicity, Ackermann-like pole placement algorithm was proposed in [5]. Similarly, the derivation of Ackermann-like algorithm from the point of the relative degree and its application to the design of the observer was also proposed [9].

This paper considers the simplified design procedure of the pole placement control and the observer for linear time-varying discrete systems. This method can be regarded as "a discrete Ackermann-like algorithm". The basic problem is to find a time-varying state feedback gain for linear time-varying discrete systems, so that the closed loop system is equivalent to some time-invariant system with desired constant closed loop poles. It will be 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

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 321–333.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

this output is equal to the order of the system. Then, the feedback gain vector can be calculated directly from the system parameters without transforming the system into any standard form. It should be noted that, in general, even some linear time-varying system is stable, its equivalent system is not necessarily stable. To preserve the stability, the Lyapunov transformation is needed. The paper shows the condition for the pole placement closed loop system to be stable.

The simple design procedure of the design of the observer is also presented in this paper. It will be shown that, as for the time-invariant case, the observer can be obtained by solving the pole placement problem for the particular anti-causal time-varying systems. Because of the duality, the simplified pole placement technique can be directly applied to the design of the state observer for linear time-varying discrete systems.

In the sequel, some basic properties of linear time-varying discrete systems are presented in Section 2. The simple pole placement technique is proposed in Section 3, and then, this method is used to the observer design problem in Section 4.

2 Preliminary

Consider the following linear time-varying discrete system with a single input and a single output.

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{1}$$

$$y(k) = c^T(k)x(k) \tag{2}$$

Here, $x \in R^n$, $u \in R^1$ and $y \in R^1$ are the state variable, the input signal and the output signal respectively. $A(k) \in R^{n\times n}$, $b(k) \in R^n$ and $c(k) \in R^n$ are time-varying bounded parameter matrices.

The transition matrix from $k = j$ to $k = i$ of this system, $\Phi(i, j)$, is defined by

$$\Phi(i,j) = A(i-1)A(i-2)\cdots A(j) \quad i > j. \tag{3}$$

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 $x(k+n) = x_1$.

Definition 2. $U_R(k)$ defined by the following equation is called the controllability matrix.

$$U_R(k) = [\, b(k+n-1), \Phi(k+n, k+n-1)b(k+n-2), \cdots, \Phi(k+n, k+1)b(k)\,] \tag{4}$$

Lemma 1. The system (1) is completely reachable in step n from the origin, if and only if

$$rank\, U_R(k) = n, \qquad {}^{\forall}k. \tag{5}$$

Definition 3. The system (1),(2) is called completely observable 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.

Definition 4. $U_o(k)$ defined by the following equation is called the controllability matrix.

$$U_o(k) = \begin{bmatrix} c^T(k) \\ c^T(k+1)\Phi(k+1,k) \\ \vdots \\ c^T(k+n-1)\Phi(k+n-1,k) \end{bmatrix} \tag{6}$$

Lemma 2. The system (1),(2) is completely observable in step n, if and only if

$$rank\, U_o(k) = n, \quad \forall k. \tag{7}$$

Let $T(k)$ be a non-singular matrix for all k. The change of variable

$$x(k) = T(k)w(k) \tag{8}$$

transforms the system

$$x(k+1) = A(k)x(k) \tag{9}$$

into the following equivalent system.

$$\begin{aligned} w(k+1) &= T^{-1}(k+1)A(k)T(k)w(k) \\ &= \hat{A}(k)w(k) \end{aligned} \tag{10}$$

As for the continuous system, the stability of the time-varying system does not necessarily implies the stability of the equivalent system [6] [7] [10].

Definition 5. If the transformation matrix $T(k)$, and $T^{-1}(k)$ are bounded, (8) is called Lyapunov transformation.

Lemma 3. Lyapunov transformation preserves the stability of the system.

3 Pole Placement of Linear Time-Varying Discrete Systems

Consider the system (1), that is

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{11}$$

The problem is to find the state feedback

$$u(k) = h^T(k)x(k) \tag{12}$$

which makes the closed loop system equivalent to the time-invariant linear system with arbitrarily stable constant poles.

Now, consider the problem of finding a new output signal $\tilde{y}(k)$ such that the relative degree from $u(k)$ to $\tilde{y}(k)$ is n. Here, $\tilde{y}(k)$ has the following form.

$$\tilde{y}(k) = \tilde{c}^T(k)x(k) \tag{13}$$

Then, the problem is to find a vector $\tilde{c}(k) \in R^n$ that satisfies this condition.

Lemma 4. The relative degree from $u(k)$ to $\tilde{y}(k)$ defined by (13) is n, if and only if $\tilde{c}^T(k)$ satisfies the following equations for all k.

$$\begin{aligned}
&\tilde{c}^T(k+1)b(k) = 0 \\
&\tilde{c}^T(k+2)\Phi(k+2,k+1)b(k) = 0 \\
&\quad\vdots \\
&\tilde{c}^T(k+n-1)\Phi(k+n-1,k+1)b(k) = 0 \\
&\tilde{c}^T(k+n)\Phi(k+n,k+1)b(k) = 1
\end{aligned} \tag{14}$$

(Here, $\tilde{c}^T(k+n)\Phi(k+n,k+1)b(k) = 1$ without loss of generality.) $\nabla\nabla$
Proof : Using (14), $\tilde{y}(k+1), \cdots, \tilde{y}(k+n)$ can be calculated as follows.

$$\begin{aligned}
\tilde{y}(k) &= \tilde{c}^T(k)x(k) \\
\tilde{y}(k+1) &= \tilde{c}^T(k+1)\Phi(k+1,k)x(k) \\
&\quad\vdots \\
\tilde{y}(k+n-1) &= \tilde{c}^T(k+n-1)\Phi(k+n-2,k)x(k) \\
\tilde{y}(k+n) &= \tilde{c}^T(k+n)\Phi(k+n-1,k)x(k) + u(k)
\end{aligned} \tag{15}$$

This implies that the relative degree from $u(k)$ to $\tilde{y}(k)$ is n.

To obtain $\tilde{c}^T(k)$, (14) can be modified as follows.

$$\begin{aligned}
\tilde{c}^T(k)\,[\, b(k-1),\ \Phi(k,k-1)b(k-2),\ \cdots,\ \Phi(k,k+1-n)b(k-n)\,] \\
= \tilde{c}^T(k)U_R(k-n) \\
= [0,\ \cdots, 0,\ 1]
\end{aligned} \tag{16}$$

From this, we have the following Theorem.

Theorem 1. If the system (11) is completely reachable in step n, there exists a vector $\tilde{c}(k)$ such that the relative degree from $u(k)$ to $\tilde{y}(k) = \tilde{c}^T(k)x(k)$ is n. And, such a vector, $\tilde{c}^T(k)$, is obtained by

$$\tilde{c}^T(k) = [0,\ \cdots,\ 0,\ 1]U_R^{-1}(k-n) \tag{17}$$

The next step is to derive the state feedback for the arbitrary pole placement. The problem is to find the time-varying state feedback gain vector so that the closed loop system is equivalent to some linear time-invariant system which has arbitrarily stable constant poles.

Let $q(z)$ be a desired closed loop characteristic polynomial of z-operator, i.e.,

$$q(z) = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_0 \tag{18}$$

which has its roots, $\lambda_i (i = 1, \cdots, n)$, as desired stable closed loop poles.

Since the new output, $\tilde{y}(k) = \tilde{c}^T(k)x(k)$, with $\tilde{c}(k)$ obtained by (17), satisfies (14), by multiplying $\tilde{y}(k+i)$ by α_i $(i = 0, \cdots, n)(\alpha_n = 1)$ and then summing them up, the following equation is obtained from (14).

$$q(p)\tilde{y}(k) = d^T(k)x(k) + u(k) \tag{19}$$

Here, $d(k) \in R^n$ is defined by the following.

$$d^T(k) = [\, \alpha_0, \alpha_1, \cdots, \alpha_{n-1}, 1 \,] \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k+1)\Phi(k+1,k) \\ \vdots \\ \tilde{c}^T(k+n)\Phi(k+n,k) \end{bmatrix} \tag{20}$$

Hence, the state feedback,

$$u = -d^T(k)x(k) \tag{21}$$

makes the closed loop system as follows.

$$q(z)\tilde{y}(k) = 0 \tag{22}$$

This control system can be summarized as follows. The given system and the pole placement state feedback are as follows.

$$x(k+1) = A(k)x(k) + b(k)u(k) \tag{23}$$
$$u(k) = -d^T(k)x(k) \tag{24}$$

Then, the closed loop system becomes

$$x(k+1) = (A(k) - b(k)d^T(k))x(k). \tag{25}$$

Let $T(k)$ be a time varying matrix defined by

$$T(k) = \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k+1)\Phi(k+1,k) \\ \vdots \\ \tilde{c}^T(k+n-1)\Phi(k+n-1,k) \end{bmatrix} \tag{26}$$

and define the new state variable $w(k)$ by the following equations.

$$x(k) = T(k)w(k), \qquad w(k) = \begin{bmatrix} \tilde{y}(k) \\ \tilde{y}(k+1) \\ \vdots \\ \tilde{y}(k+n-1) \end{bmatrix} \tag{27}$$

Then, (22) implies that the change of variable (27) transforms the closed loop system (25) into the following system.

$$\begin{aligned} w(k+1) &= T^{-1}(k+1)(A(k) - b(k)d^T(k))T(k)w(k) \\ &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ \vdots & & & 1 \\ -\alpha_0 & \cdots & \cdots & -\alpha_{n-1} \end{bmatrix} w(k) \\ &= A^* w(k) \end{aligned} \tag{28}$$

This implies that the closed loop system is equivalent to the time-invariant linear system which has the desired closed loop poles $\lambda_i (i = 1, \cdots, n)$ $(\det(zI - A^*) = q(z))$.

Non-singularity of the transformation matrix and the stability of the pole placement control system are established in the following Theorems.

Theorem 2. If the system (11) is completely reachable in step n, then, the matrix for the change of variable, $T(k)$, given by (26) is nonsingular for all k.
Proof : Since, if $U_R(k)$ is nonsingular for all k, equations in (14) are satisfied for all k, we have the following equation.

$$T(k)U_R(k-n) = \begin{bmatrix} 0 & & 1 \\ & \diagup & \\ 1 & & * \end{bmatrix} \tag{29}$$

Here, the right hand side is an anti-triangular matrix with 1 in the anti-diagonal position, then, $T(k)$ is nonsingular.

Theorem 3. The pole placement control system obtained above is stable if $T(k)$ is Lyapunov transformation matrix.

The calculation procedure for pole placement feedback gain is as follows.

[Calculation Procedure of the Pole Placement]

Step 1. Calculate the state transition matrix and the controllability matrix by (3) and (4).
Step 2. Calculate $\tilde{c}^T(k)$ by (17).
Step 3. Define the desired closed loop characteristic polynomial by (18) and calculate the state feedback by (20) and (21).
Step 4. Check if $T(k)$ in (26) is Lyapunov transformation matrix.

Example 1
Consider the following unstable system.

$$\begin{aligned} x(k+1) &= A(k)x(k) + b(k)u(k) \\ y(k) &= c^T(k)x(k) \end{aligned} \tag{30}$$

where

$$\begin{aligned} A(k) &= \begin{bmatrix} 1 & 2 + cos0.1k \\ 2 + sin0.2k & 2 \end{bmatrix} \\ b(k) &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ c^T(k) &= \begin{bmatrix} 3 + sin0.5k & 0 \end{bmatrix} \end{aligned} \tag{31}$$

Here, we define $\lambda_1(k)$, $\lambda_2(k)$ and $\lambda_3(k)$ by the followings.

$$\begin{aligned} \gamma_1(k) &= 2 + cos0.1k \\ \gamma_2(k) &= 2 + sin0.2k \\ \gamma_3(k) &= 3 + sin0.5k \end{aligned} \tag{32}$$

From (17), $\tilde{c}^T(k)$ is obtained as follows.

$$\begin{aligned}\tilde{c}^T(k) &= [\ 0,\ 1\ ][\ b(k-1),\ A(k-1)b(k-2)\ ]^{-1} \\ &= \left[\ \frac{1}{\gamma_1(k-1)},\ 0\ \right]\end{aligned}$$

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

$$q(z) = z^2 + 0.9z + 0.2.$$

The pole placement feedback is $u(k) = -d^T(k)x(k)$, and, from (20), $d(k)$ is calculated by

$$\begin{aligned}d^T(k) &= [\ 0.2,\ 0.9,\ 1\ ] \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k+1)A(k) \\ \tilde{c}^T(k+2)A(k+1)A(k) \end{bmatrix} \\ &= \left[\, d_1(k)\ d_2(k) \,\right] \end{aligned} \tag{33}$$

where, $d_1(k)$ and $d_2(k)$ are given by

$$\begin{aligned} d_1(k) &= \frac{0.2}{\gamma_1(k-1)} + \frac{0.9}{\gamma_1(k)} + \frac{1}{\gamma_1(k+1)} + 2 + \gamma_2(k) \\ d_2(k) &= 0.9 + \frac{\gamma_1(k)}{\gamma_1(k+1)} + 2 \end{aligned}$$

It is readily checked that the transformation matrix

$$T(k) = \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k+1)A(k) \end{bmatrix}$$

is the Lyapunov transformation.

Fig.1 shows the state response of this unstable plant with the initial condition, $x_1(1) = x_2(1) = 1$. The simulation result of the plant with the pole placement state feedback obtained above is in Fig.2 with the same initial condition.

4 State Observer

In this section, we consider the design of the observer for the system (1),(2), that is the following linear time-varying system with a single input and a single output.

$$\begin{aligned} x(k+1) &= A(k)x(k) + b(k)u(k) \\ y(k) &= c^T(k)x(k) \end{aligned} \tag{34}$$

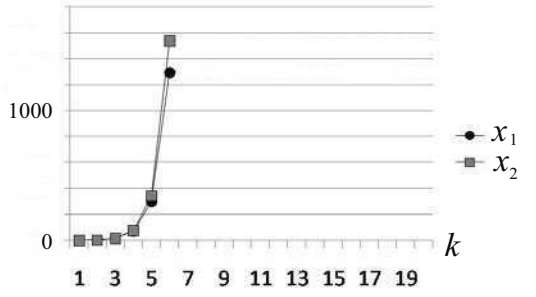


Fig. 1. Responce of the unstable plant

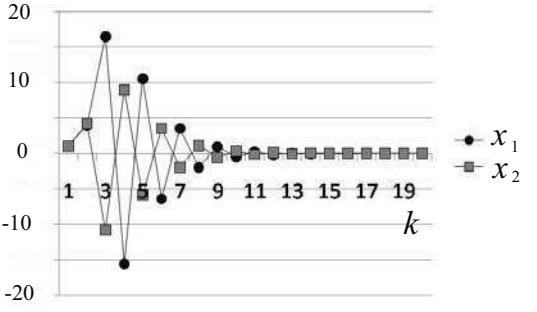


Fig. 2. Responce of the state variable (x) of pole placement feedback control

The problem is to design the full order state observer for (34). Consider the following system as a candidate of the observer.

$$\begin{aligned}\hat{x}(k+1) &= F(k)\hat{x}(k) + b(k)u(k) + h(k)y(k) \\ &= F(k)\hat{x}(k) + b(k)u(k) + h(k)c^T(k)x(k)\end{aligned} \tag{35}$$

where $F(k) \in R^{n\times n}$, and $h(k) \in R^n$. Define the state estimation error $e(k) \in R^n$ by

$$e = x(k) - \hat{x}(k) \tag{36}$$

Then, $e(k)$ satisfies the following error equation.

$$e(k+1) = F(k)e(k) + (A(k) - F(k) - h(k)c^T(k))x(k) \tag{37}$$

Hence, as well known, (35) is a state observer of (39) if $F(k)$ and $h(k)$ satisfy the following condition.

$$\begin{aligned}&F(k) = A(k) - h(k)c^T(k) \\ &F(k) \ : \ \text{arbitrarily stable matrix}\end{aligned} \tag{38}$$

Then, the problem is to find $h(k)$ such that $F(k)$ is equivalent to some constant matrix F^* with arbitrarily desired stable eigenvalues. To apply the pole placement technique to this problem, consider the pole placement control problem of the following anti-causal system.

$$\xi(k-1) = A^T(k)\xi(k) + c(k)v(k) \tag{39}$$

where $\xi(k) \in R^n$ and $v(k) \in R^1$ are the state variable and an input signal.

The transition matrix, $\Psi(i,j)$, of the anti-causal system (39) is the following.

$$\Psi(i,j) = A^T(i+1)\cdots A^T(j-1)A^T(j) \qquad (i<j) \tag{40}$$

From the property of the duality of the time varying discrete system, if the pair $(A(k), c^T(k))$ is completely observable in step n, the pair $(A^T(k), c(k))$ of the anti-causal system (39) is completely reachable in step n.

Then, if the given system (34) is completely observable in step n, the system (39) has a state feedback

$$v(k) = h^T(k)\xi(k) \tag{41}$$

such that the anti-causal closed loop system is equivalent to the linear time-invariant anti-causal system with arbitrarily constant stable poles.

This implies that for some state transformation matrix, $P^T(k) \in R^n$,

$$P^T(k-1)^{-1}(A^T(k) - c(k)h^T(k))P^T(k) = F^{*T} \tag{42}$$

where, F^{*T} is a constant matrix with arbitrarily stable eigenvalues. From this, we have the following equation.

$$P^{-1}(k)(A(k) - h(k)c^T(k))P(k-1) = F^* \tag{43}$$

Hence, using this $h(k)$, the state observer for the system (34) is obtained. By applying the design procedure stated in the previous section, the calculation procedure for the observer is as follows.

[Calculation Procedure of the Observer Gain]

Step 1. Using the transition matrix (40), calculate the controllability matrix of the anti-causal system (39), $V_R(k)$, by the following.

$$\begin{aligned} V_R(k) = \ [\ & c(k-n+1), \Psi(k-n, k-n+1)c(k-n+2), \\ & \cdots, \Psi(k-n, k-1)c(k)\ ] \end{aligned} \tag{44}$$

Step 2. Using the (17), calculate $\tilde{c}^T(k)$ for the anti-causal system (39) by the following.

$$\tilde{c}^T(k) = [0,\ \cdots,\ 0,\ 1]V_R^{-1}(k+n) \tag{45}$$

Step 3. Define the stable characteristic polynomial of backward shift operator, z^{-1}, in the similar way of (18).

$$q(z^{-1}) = z^{-n} + \alpha_{n-1}z^{-(n-1)} + \cdots + \alpha_0 \tag{46}$$

Step 4. In the similar way of (20), calculate $h(k)$ by

$$h^T(k) = [\, \alpha_0, \alpha_1, \cdots, \alpha_{n-1}, 1 \,] \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k-1)\Psi(k-1,k) \\ \vdots \\ \tilde{c}^T(k-n)\Psi(k-n,k) \end{bmatrix} \tag{47}$$

and the observer gain vector is $h(k)$.

Step 5. Define the transformation matrix, $P^T(k)$, by

$$P^T(k) = \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k-1)\Psi(k-1,k) \\ \vdots \\ \tilde{c}^T(k-n+1)\Psi(k-n+1,k) \end{bmatrix} \tag{48}$$

and check if $P(k)$ is Lyapunov transformation matrix.

As for the linear time-invariant case, the observer based pole placement control system can be described by

$$\begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A(k) - b(k)d^T(k) & b(k)d^T(k) \\ 0 & A(k) - h(k)c^T(k) \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} \tag{49}$$

where $e(k) := x(k) - \hat{x}(k)$. Using the Lyapunov transformation matrix

$$\begin{bmatrix} T(k) & 0 \\ 0 & P(k-1) \end{bmatrix} \tag{50}$$

and, also from (28) and (43), the system matrix of (48) is equivalent to

$$\begin{bmatrix} A^*, & T^{-1}(k+1)b(k)d^T(k)P(k-1) \\ 0, & F^* \end{bmatrix} \tag{51}$$

which implies that the observer based pole placement control of linear time-varying system satisfies the separation principle.

Example 2

Consider the system (30) and (31) of Example 1, again. Suppose that the state,$x(k)$, is not accessible. Then, we design the pole placement feedback using the state observer. In this example we show the calculation procedure to obtain the observer gain. We use $\lambda_1(k)$, $\lambda_2(k)$ and $\lambda_3(k)$ as the same definitions as in Example 1.

First, note that this plant is completely observable in step n, because the observability matrix is as follows.

$$\begin{bmatrix} c^T(k) \\ c^T(k+1)A(k) \end{bmatrix} = \begin{bmatrix} \gamma_3(k) & 0 \\ \gamma_3(k) & \gamma_3(k)\gamma_1(k) \end{bmatrix} \tag{52}$$

The system matrices of the anti-causal and dual system are as follows.

$$A^T(k) = \begin{bmatrix} 1 & \gamma_2(k) \\ \gamma_1(k) & 2 \end{bmatrix}$$
$$c(k) = \begin{bmatrix} \gamma_3(k) \\ 0 \end{bmatrix} \tag{53}$$

From (45), $\tilde{c}^T(k)$ for the new output matrix is obtained as

$$\begin{aligned} \tilde{c}^T(k) &= [\,0,\ 1\,][\,c(k+1),\ \ A^T(k+1)c(k+2)\,]^{-1} \\ &= \frac{1}{\delta(k)}\left[0,\ \gamma_3(k+1)\right] \end{aligned} \tag{54}$$

where,

$$\delta(k) = \gamma_3(k+1)\gamma_3(k+2)\gamma_1(k+2) \tag{55}$$

The purpose is to design the state feedback for the dual system matrices, (39), so that the anti-causal closed loop system is equivalent to the linear time-invariant system with $\lambda_1 = 0.2$ and $\lambda_2 = -0.1$ as its closed loop poles. This implies that the desired closed loop characteristic polynomial is

$$q(z) = z^{-2} - 0.1z^{-1} - 0.02.$$

From (47), $h^T(k)$ is calculated as follows.

$$\begin{aligned} h^T(k) &= [\,-0.02,\ -0.1,\ 1\,] \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k-1)A^T(k) \\ \tilde{c}^T(k-2)A^T(k-1)A^T(k) \end{bmatrix} \\ &= \left[h_1(k)\ h_2(k)\right] \end{aligned} \tag{56}$$

Here, $h_1(k)$ and $h_2(k)$ are

$$\begin{aligned} h_1(k) &= -\frac{0.1\gamma_3(k)\gamma_1(k)}{\delta(k-1)} \\ &\quad + \frac{\gamma_3(k-1)\gamma_1(k-1) + 2\gamma_3(k)\gamma_1(k)}{\delta(k-2)} \\ h_2(k) &= -\frac{0.02\gamma_3(k+1)}{\delta(k)} - \frac{0.2\gamma_3(k)}{\delta(k-1)} \\ &\quad + \frac{\gamma_3(k-1)\gamma_1(k-1)\gamma_2(k) + 4\gamma_3(k)}{\delta(k-2)} \end{aligned}$$

Hence, the observer gain vector is $h(k)$.

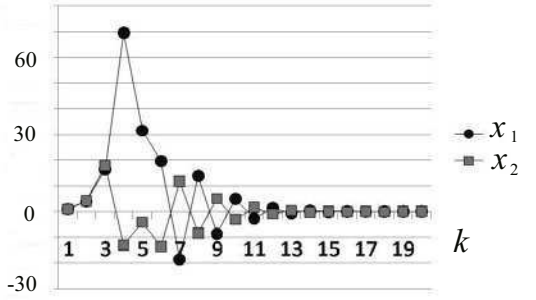


Fig. 3. Response of the pole placement with the observer ($x(k)$)

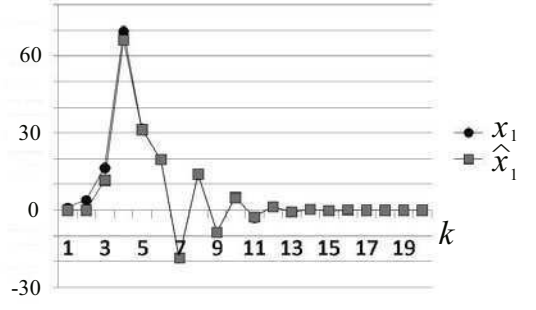


Fig. 4. Response of $x_1(k)$ and $\hat{x}_1(k)$

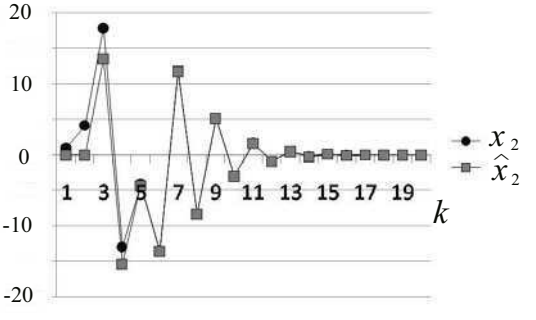


Fig. 5. Response of $x_2(k)$ and $\hat{x}_2(k)$

It is also readily checked that the transformation matrix

$$P^T(k) = \begin{bmatrix} \tilde{c}^T(k) \\ \tilde{c}^T(k-1)A^T(k) \end{bmatrix}$$

is the Lyapunov transformation.

From the above, the observer is

$$\hat{x}(k+1) = \{A(k) - h(k)g^T(k)\}\hat{x}(k) + b(k)u(k) + h(k)y(k) \tag{57}$$

and, the pole placement feedback using this observer becomes

$$u(k) = -d^T(k)\hat{x}(k) \tag{58}$$

where $d(k)$ is defined in (33) in Example 1.

Fig.3 $\sim$ 5 show the simulation results of the pole placement feedback system with the observer. The initial condition of the plant is $x_1(1) = x_2(1) = 1$.

5 Conclusions

This paper proposed the simple derivation method of the pole placement state feedback gain for liner time-varying discrete systems. 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.

References

1. Kwakaernaak, H.C.: Linear Optimal Control Systems. Wiley-Interscience, Hoboken (1972)
2. Kailath, T.C.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
3. Nguyen, C.: Arbitrary eigenvalue assignments for linear time-varying multivariable control systems. Internal Journal of Control 45(3), 1051–1057 (1987)
4. Valasek, M., Olgac, N.: Efficient pole placement technique for linear time-variant SISO systems. In: IEE Proc. Control Theory Appl., vol. 142(5), pp. 451–458 (1995)
5. Lee, H.C., Choi, J.W.: Ackermann-like eigenvalue assignment formulae for linear time-varying systems. In: IEE Proc. Control Theory Appl., vol. 152(4), pp. 427–434 (2005)
6. Rugh, W.J.: Linear System Theory, 2nd edn. Prentice-Hall, Englewood Cliffs (1995)
7. Chi-Tsong Chen, C.: Linear System Theory and Design, 3rd edn. Oxford University Press, Oxford (1999)
8. Tse, E., Athans, M.: Optimal Minimal-Order Observer-Estimators for Discrete Linear Time-Varying System. IEEE Transaction on AC AC-15(4), 416–426 (1970)
9. Mutoh, Y.: Simple Design of the State Observer for Linear Time-Varying Systems. In: 6th ICINCO (2009)
10. Halanay, A., Ionescu, V.: Time-Varying Discrete Linear Systems, Birkhauser (1994)

Optimal Controller Gain Tuning for Robust Stability of Spacecraft Formation

Esten I. Grøtli[1], Antoine Chaillet[2], Elena Panteley[3], and J. Tommy Gravdahl[1]

[1] Dept. of Eng. Cybernetics, O.S. Bragstads plass 2D, NTNU, 7491 Trondheim, Norway
[2] Univ. Paris Sud 11 - L2S - EECI - Supélec, 3 rue Joliot-Curie, 91192 Gif Sur Yvette, France
[3] CNRS - L2S, Supélec, 3 rue Joliot-Curie, 91192 Gif Sur Yvette, France

Abstract. The spacecraft formation control problem sets high demands to the performance, especially with respect to positional accuracy. The problem is further complicated due to scarce fuel resources and limited actuation effects, in addition to the many sources of disturbances. This paper addresses the problem of finding the optimal gains of spacecraft formation controllers. By optimal, we mean the gains that minimizes a cost functional which penalizes both the control efforts and the state deviation, while still guaranteeing stability of the closed-loop systems in the presence of disturbances.

Keywords: Robustness, ISS, Moving average of disturbances, Spacecraft formation, gain tuning.

1 Introduction

Formations of spacecraft are mainly motivated by their flexibility and increased baseline length compared to monolithic spacecraft. They are complex systems which require precise control to maintain relative trajectories, even in the presence of disturbances. These disturbances are often only described by the statistical or averaged characteristics of the perturbing signals (*e.g.* amplitude, energy, average energy, etc.), and are due to for instance intervehicle interference, solar wind and radiation and gravitational effects. To account for the limited fuel resources, the control law should minimize the fuel consumption, while still guaranteeing a predefined accuracy.

Ignoring the disturbances, an optimal control, that is the control input that minimizes some sensible cost function, can be derived using the Hamilton-Jacobi-Bellman equation or Pontryagin's maximum principle. With the introduction of $\mathcal{H}_\infty$ methods, by [19], one can specify the level of plant uncertainty and the signal gain from disturbance inputs to error outputs [4]. Although the $\mathcal{H}_\infty$ control problem was initially stated for systems described by transfer matrices, the ideas were soon translated into a state space setting [18], by realizing that the $\mathcal{H}_\infty$ norm is the $\mathcal{L}_2$ induced norm in the time domain. The nonlinear equivalent of the $\mathcal{H}_\infty$ problem was shown by [13] to be determined by the Hamilton-Jacobi-Isaacs (HJI) equation or inequality. An analytic solution to the HJI equation is general difficult to find. With the introduction of the inverse optimal control problem by [5], the performance index is *a posteriori* determined rather than *a priori*. In [10] this idea was applied to systems with disturbances, and it was shown that input-to-state stabilizability is a necessary and sufficient condition for what is known as the

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 335–347.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

inverse optimal gain assignment problem to be solvable. The sufficiency part is relying on the Sontag type control law introduced in [16]. By solving the inverse optimal control problem solutions to a whole family of HJI equations are found. In [8] the link between optimality and closed-loop stability was further investigated by relating the cost function to a Lyapunov function, with the purpose of an optimal selection of the controller design parameters.

The problem of study in this paper is similar to the problem in [8], however we do not use the Lyapunov function as our cost function. Instead we introduce a new cost function, and solve the optimization problem with the stability constraints based on the Lyapunov analysis. The Lyapunov analysis of was performed in a previous paper by the authors, [6], and a guarantee for a *hard bound* on the state norm for input-to-state stable (ISS) systems in presence of signals with limited moving average were found. Furthermore, it was shown that the Lyapunov function can give an explicit estimate of the maximum of the disturbances' moving average that can be tolerated for a given precision. For the sake of completeness, some of the results are repeated here. In this paper we address the problem of finding the optimal gains, subject to constraints imposed by the Lyapunov analysis to achieve the above mentioned precision and stability. We consider a cost functional that is quadratic in both state and control variables, and the optimization is performed using `fmincon`, because of its ability to handle nonlinear constraints. We emphasize that we are not doing optimal control in usual sense, as the the controller and observer has already been designed without necessarily optimality in mind. It is obvious that the performance is restricted by parameterization of the chosen control law.

Notation and Terminology. A continuous function $\alpha : \mathbb{R}_{\geq} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ ($\alpha \in \mathcal{K}$), if it is strictly increasing and $\alpha(0) = 0$. If, in addition, $\alpha(s) \rightarrow \infty$ as $s \rightarrow \infty$, then α is of class $\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty$). A continuous function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ 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 $\delta \geq 0$ 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 $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^p$. For a signal $u \in \mathcal{U}$, $\|u\|_\infty := \text{ess sup}_{t \geq 0} |u(t)|$. The maximum and minimum eigenvalues of a symmetric matrix A are denoted by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$, respectively. I_n and 0_n denote the identity and zero matrices of $\mathbb{R}^{n \times n}$ respectively. We use E for the expectancy operator.

2 ISS Systems and Signals with Low Moving Average

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 $f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is locally Lipschitz and satisfies $f(0, 0) = 0$.

Definition 1. *Let δ be a nonnegative constant and $\mathcal{W} \subset \mathcal{U}$. The ball $\mathcal{B}_\delta$ is said to be* globally exponentially stable (GES) *for (1) with respect to $\mathcal{W}$ if there exists some positive constants k_1 and k_2 such that the solution of (1) satisfies*

$$|x(t; x_0, u)| \leq \delta + k_1 |x_0| e^{-k_2 t}, \quad \forall t \geq 0. \tag{2}$$

for all $x_0 \in \mathbb{R}^n$ and all $u \in \mathcal{W}$.

We next recall the definition of ISS, originally introduced in [15].

Definition 2. *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 [12] and thus extending the original characterization proposed by Sontag in [17].

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 $u \in \mathbb{R}^p$,*

$$\underline{\alpha}(|x|) \leq V(x) \leq \overline{\alpha}(|x|) \tag{4}$$

$$\tfrac{\partial V}{\partial x}(x) f(x, u) \leq -\kappa V(x) + \gamma(|u|)\,. \tag{5}$$

γ is then called a supply rate *for (1).*

The input signals we consider in this paper are slightly more restrictive than those in for instance [1], but the advantage is that a *hard bound* on the state norm can be guaranteed. Namely, we consider input signals with bounded moving average.

Definition 3. *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 are taken into account in the control design. More parsimonious control laws than those based on the disturbances' 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. Robustness to this class of signals thus constitutes an extension of the typical properties of ISS systems.

With input signals with bounded moving average, the following result of [6] guarantees global *exponential* stability of some neighborhood of the origin.

Corollary 1. *Assume there exists a continuously differentiable function* $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$*, class* $\mathcal{K}_\infty$ *function* γ*, functions* $\underline{\alpha}(s) = \underline{c}s^p$ *and* $\overline{\alpha}(s) = \overline{c}s^p$*, with* $\underline{c}, \overline{c}, p$ *positive constants, and a positive constant* κ *such that (4) and (5) hold for all* $x \in \mathbb{R}^n$ *and all* $u \in \mathbb{R}^p$*. 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 Spacecraft Formation Control

The results of Section 2 were in [6] exploited to demonstrate robustness of a spacecraft formation control in a leader-follower configuration. We will here only include the parts necessary to keep this section self-contained. For further details, the reader is referred to the original paper.

3.1 Spacecraft Models

The model for the leader spacecrafts motion with respect to a moving coordinate frame is given by

$$\ddot{p} + C(\dot{\nu}_o)\dot{p} + D(\dot{\nu}_o, \ddot{\nu}_o)p + n(r_o, p) = F_l \tag{6}$$

while the model for follower spacecraft with respect to the leader is given by

$$\ddot{\rho} + C(\dot{\nu}_o)\dot{\rho} + D(\dot{\nu}_o, \ddot{\nu}_o)\rho + n(r_o + p, \rho) = F_f - F_l, \tag{7}$$

where $F_l := (u_l + d_l)/m_l$, $F_f := (u_f + d_f)/m_f$, and where subscripts l and f stand for the leader and follower spacecraft respectively. p is the position of the leader spacecraft with respect to a coordinate frame in a Keplerian orbit, where as ρ is the position of the follower spacecraft with respect to the leader. m_l and m_f are the spacecraft' masses, u_l and u_f are the control inputs, and d_l and d_f denote all exogenous perturbations acting on the spacecraft. Furthermore,

$$C(\dot{\nu}_o) := 2\dot{\nu}_o\bar{C}, \quad \bar{C} := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$D(\dot{\nu}_o, \ddot{\nu}_o) := \dot{\nu}_o^2\bar{D} + \ddot{\nu}_o\bar{C}, \quad \bar{D} := \mathrm{diag}(-1, -1, 0),$$

and

$$n(r_o, p) := \mu\left(\frac{r_o + p}{|r_o + p|^3} - \frac{r_o}{|r_o|^3}\right),$$

with r_o being the origin of the coordinate reference frame, and ν_o the true anomaly. We make the following assumptions on the first and second derivatives of ν_o:

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* $\beta_{\ddot{\nu}_o}$*.*

3.2 Controller and Observer Design

In [6] the proposed controller of the leader spacecraft were:

$$u_l = m_l \left[\ddot{p}_d + C\left(\dot{\nu}_o\right) \dot{p}_d + D\left(\dot{\nu}_o, \ddot{\nu}_o\right) p + n\left(r_o, p\right) - k_l \left(\dot{p}_0 - \dot{p}_r\right) \right] \tag{8}$$

with p_d being the reference trajectory, $e_l = p - p_d$ the tracking error, $\tilde{p} = p - \hat{p}$ the estimation error, $\dot{p}_r = \dot{p}_d - \ell_l e_l$, $\dot{p}_0 = \dot{\hat{p}} - \ell_l \tilde{p}$, and with velocity estimate provided by

$$\dot{\hat{p}} = a_l + \left(l_l + \ell_l\right) \tilde{p} \tag{9}$$

$$\dot{a}_l = \ddot{p}_d + l_l \ell_l \tilde{p} \,. \tag{10}$$

The controller for the follower spacecraft has a similar structure, given by

$$\begin{aligned} u_f = & m_f \Big[\ddot{p}_d + \ddot{\rho}_d + C\left(\dot{\nu}_o\right)\left(\dot{p}_d + \dot{\rho}_d\right) + D\left(\dot{\nu}_o, \ddot{\nu}_o\right)\left(p + \rho\right) + n\left(r_o + p, \rho\right) + n\left(r_o, p\right) \\ & - k_l \left(\dot{p}_0 - \dot{p}_r\right) - k_f \left(\dot{\rho}_0 - \dot{\rho}_r\right) \Big] \end{aligned} \tag{11}$$

with ρ being the reference trajectory, $e_f = \rho - \rho_d$ the follower spacecraft tracking error, $\tilde{\rho} = \rho - \hat{\rho}$ the estimation error, auxiliary signals $\dot{\rho}_r = \dot{\rho}_d - \ell_f e_f$ and $\dot{\rho}_0 = \dot{\hat{\rho}} - \ell_f \tilde{\rho}$, and with the observer

$$\dot{\hat{\rho}} = a_f + \left(l_f + \ell_f\right) \tilde{\rho} \tag{12}$$

$$\dot{a}_f = \ddot{\rho}_d + l_f \ell_f \tilde{\rho} \tag{13}$$

The parameters k_l, l_l, ℓ_l, k_f, l_f and ℓ_f denote some positive tuning gains. The closed-loop system may now be summarized into the equations

$$\dot{X} = A(\dot{\nu}_o(t), \theta) X + B d \,, \tag{14}$$

where $X := (X_l^\top, X_f^\top)^\top$ with $X_l := (e_l^\top, \dot{e}_l^\top, \tilde{p}^\top, \dot{\tilde{p}}^\top)^\top$ and $X_f := (e_f^\top, \dot{e}_f^\top, \tilde{\rho}^\top, \dot{\tilde{\rho}}^\top)^\top$, $d := (d_l^\top, d_f^\top)^\top$, $\theta := (k_l, \ell_l, l_l, k_f, \ell_f, l_f)^\top$ and $A := \mathrm{blckdiag}(A_l, A_f)$ with

$$A_i(\dot{\nu}_o) := \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 \\ a_{21} & a_{22}(\dot{\nu}_o) & a_{23} & a_{24} \\ 0_3 & 0_3 & 0_3 & I_3 \\ a_{41} & a_{42}(\dot{\nu}_o) & a_{43} & a_{44} \end{bmatrix} , \tag{15}$$

for $i \in \{l, f\}$. Out of notational compactness, the following matrices have been used: $a_{21} := a_{41} := -k_i \ell_i I_3$, $a_{22} := a_{42} := -C(\dot{\nu}_o) - k_i I_3$, $a_{23} := k_i \ell_i I_3$, $a_{24} := k_i I_3$, $a_{43} := (k_i - l_i)\ell_i I_3$ and $a_{44} := (k_i - l_i - \ell_i) I_3$. Finally, $B := (B_l^\top, B_f^\top)^\top$ with

$$B_l := \frac{1}{m_l} \begin{bmatrix} 0_3 & 0_3 \\ I_3 & 0_3 \\ 0_3 & 0_3 \\ I_3 & 0_3 \end{bmatrix} \quad \text{and} \quad B_f := \frac{1}{m_l m_f} \begin{bmatrix} 0_3 & 0_3 \\ -m_f I_3 & m_l I_3 \\ 0_3 & 0_3 \\ -m_f I_3 & m_l I_3 \end{bmatrix} .$$

3.3 Robustness of the Overall Formation

The following result establishes 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 (8)-(10) and the controller of the follower spacecraft be given by (11)-(13) with, for each $i \in \{l, f\}$, $l_i \geq 2k_i$, $k_i > 2k_i^\star$, where*

$$k_i^\star := \begin{cases} \ell_i + \tilde{\beta}_i & \text{if } k_i - \ell_i \leq k_i \ell_i^2 \\ \tilde{\beta}_i / \ell_i^2 & \text{otherwise,} \end{cases}$$

with

$$\tilde{\beta}_i := \beta_{\dot{\nu}_o} \sqrt{2\ell_i^2 + 1} + \left(1 + \frac{m_f^2}{m_l^2}\right) \frac{\left(l_i^2 + 1\right)}{m_i^2} \,. \tag{16}$$

Given any precision $\delta > 0$ and any time window $T > 0$, consider any average energy satisfying

$$E \leq \frac{1}{4} \min_{i \in \{l,f\}} \left\{ \ell_i^2 - \frac{1}{2}\sqrt{4\ell_i^4 + 1} + \frac{1}{2} \right\} \delta^2 \frac{e^{\kappa T} - 1}{2e^{\kappa T} - 1} \,, \tag{17}$$

where

$$\kappa := \frac{\min_{i \in \{l,f\}} k_i^\star / \max_{i \in \{l,f\}} \left\{ \frac{k_i}{\ell_i} \right\}}{\max_{i \in \{l,f\}} \left\{ \ell_i^2 + \frac{1}{2}\sqrt{4\ell_i^4 + 1} + \frac{1}{2} \right\}} \,. \tag{18}$$

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 (14).

The proof of the proposition is found in [6].

As already stressed, the results recalled in Section 2 allow to expect more parsimonious solicitation of the actuators than classical ISS-based reasonings. However, such an improvement will only be made practical if the gains of Proposition 2 are tuned in an adequate manner. The rest of the document focuses on this issue.

4 Optimal Controller and Observer Gain Tuning

Of simplicity we will in the following consider the unperturbed version of (14),

$$\dot{X} = A(\dot{\nu}_o(t), \theta) X \,, \tag{19}$$

since d is only characterized by its moving average and therefore difficult to incorporate in the optimization problem we will set forth in the sequel. One possibility to incorporate the disturbance into the optimization problem would be to look at the worst case scenario, which is typical of differential game problems or the inverse optimal gain assignment problem mentioned in the introduction. The worst case excitation due to the disturbances with limited moving average, can for a linear system be found using the

results of [7]. We however, aim at finding the parameters θ, which are the best compromise between fast state convergence and limited control efforts, while ignoring the effect of the disturbance. If the problem is posed as an optimization problem, a common choice is to use a cost function which quadratically penalizes control and state deviation, i.e. a cost function of the form

$$J(\theta) = \int_{t_0}^{t_h} X(\tau)^\top Q X(\tau) + \overline{u}(\tau)^\top R \overline{u}(\tau) d\tau \tag{20}$$

with weighting matrices $Q \in \mathbb{R}^{24\times 24}$ and $R \in \mathbb{R}^{6\times 6}$ both being symmetric and positive definite, optimization horizon t_h and $\overline{u} = (\overline{u}_l^\top, \overline{u}_f^\top)^\top$, where

$$\overline{u}_l = m_l \Big[D\left(\dot{\nu}_o, \ddot{\nu}_o\right) e_l - k_l \left(\dot{p}_0 - \dot{p}_r\right) \Big] \tag{21}$$

and

$$\overline{u}_f = m_f \Big[D\left(\dot{\nu}_o, \ddot{\nu}_o\right) \left(e_l + e_f\right) - k_l \left(\dot{p}_0 - \dot{p}_r\right) - k_f \left(\dot{\rho}_0 - \dot{\rho}_r\right) \Big] \tag{22}$$

are the terms of (8) and (11), respectively, that we want to penalize. It can be noticed, that we have chosen not to penalize the feed-forward terms from the reference and the gravity compensation in (8) and (11). The cost (20), can be written as:

$$J(\theta) = \int_{t_0}^{t_h} X^\top(\tau)(Q + (T + \Xi)^\top R (T + \Xi)) X(\tau) d\tau \tag{23}$$

with

$$T := m_f \begin{bmatrix} \frac{m_l}{m_f} k_l \ell_l I_3 & \frac{m_l}{m_f} k_l I_3 & -\frac{m_l}{m_f} k_l \ell_l I_3 & -\frac{m_l}{m_f} k_l I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ k_l \ell_l I_3 & k_l I_3 & -k_l \ell_l I_3 & -k_l I_3 & k_f \ell_f I_3 & k_f I_3 & -k_f \ell_f I_3 & -k_f I_3 \end{bmatrix}, \tag{24}$$

and

$$\Xi := \begin{bmatrix} m_l D(\dot{\nu}_o, \ddot{\nu}_o) & 0_3 \\ 0_3 & m_f D(\dot{\nu}_o, \ddot{\nu}_o) \end{bmatrix} \begin{bmatrix} I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ I_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 \end{bmatrix}. \tag{25}$$

We now introduce the state transition matrix $\Phi(t, t_o)$ which is the unique solution of

$$\frac{\partial}{\partial t} \Phi(t, t_0) = A(\dot{\nu}_o(t), \theta) \Phi(t, t_0), \tag{26}$$

with initial condition

$$\Phi(t_0, t_0) = I_{24}. \tag{27}$$

The solution to (19) is

$$X(t) = \Phi(t, t_0) X_0, \tag{28}$$

and is used to rewrite the cost function into the following form

$$J(\theta) = X_0^\top \int_{t_0}^{t_h} \Phi^\top(\tau, t_0)(Q + (T + \Xi(\tau))^\top R (T + \Xi(\tau))) \Phi(\tau, t_0) d\tau X_0. \tag{29}$$

We will avoid dependency of the cost function on initial conditions, and will assume that X_0 are randomized variables with zero mean, and covariance matrix equal to identity, that is $\mathrm{E}\{X_0 X_0^\top\} = I_{24}$. By defining

$$P(t_h, t_0) := \int_{t_0}^{t_h} \Phi^\top(\tau, t_0)(Q + (T + \Xi(\tau))^\top R(T + \Xi(\tau)))\Phi(\tau, t_0)d\tau \tag{30}$$

and using the fact that

$$\begin{aligned}\mathrm{E}\{X_0^\top P(t_h, t_0)X_0\} &= \mathrm{E}\{\mathrm{trace}\, X_0^\top P(t_h, t_0)X_0\} \\ &= \mathrm{E}\{\mathrm{trace}\, X_0 X_0^\top P(t_h, t_0)\} \\ &= \mathrm{trace}\, \mathrm{E}\{X_0 X_0^\top\} P(t_h, t_0) \\ &= \mathrm{trace}\, P(t_h, t_0)\,,\end{aligned}$$

our optimization problem can be formulated as

$$\min_\theta \mathrm{E}\{J(\theta)\} = \min_\theta \mathrm{trace}\, P(t_h, t_0) \tag{31}$$

subject to

$$\frac{\partial}{\partial t}\Phi(t, t_0) = A(\dot{\nu}_o(t), \theta)\Phi(t, t_0)\,, \tag{32}$$

with $\Phi(t_0, t_0) = I_{24}$,

$$\frac{\partial}{\partial t}P(t, t_0) = \Phi^\top(t, t_0)(Q + (T + \Xi(t))^\top)R(T + \Xi(t)))\Phi(t, t_0)\,, \tag{33}$$

with $P(t_0, t_0) = (Q + (T + \Xi(t_0))^\top R(T + \Xi(t_0)))$, for each $i \in \{l, f\}$,

$$l_i \geq 2k_i \tag{34}$$

$$k_i > 2k_i^\star \tag{35}$$

where

$$k_i^\star := \begin{cases} \ell_i + \tilde{\beta}_i & \text{if } k_i - \ell_i \leq k_i \ell_i^2 \\ \tilde{\beta}_i / \ell_i^2 & \text{otherwise,} \end{cases} \tag{36}$$

with

$$\tilde{\beta}_i := \beta_{\dot{\nu}_o}\sqrt{2\ell_i^2 + 1} + \left(1 + \frac{m_f^2}{m_l^2}\right)\frac{(l_i^2 + 1)}{m_i^2}\,, \tag{37}$$

and

$$k_l, \ell_l, k_f, \ell_f \in [0, 2] \tag{38}$$

$$l_l \in [0, 10] \tag{39}$$

$$l_f \in [0, 15]\,. \tag{40}$$

The constraints (38)-(40) are introduced to more efficiently solve the problem. The problem was solved using the interior point algorithm of the function `fmincon` in MATLAB.

The true anomaly, ν_o, of the reference frame can be obtained by numerical integration of the equation

$$\dot{\nu}_o(t) = \frac{\sqrt{\mu}\left(1 + e_o \cos \nu_o(t)\right)^2}{\left(\frac{1}{2}(r_a + r_p)\left(1 - e_o^2\right)\right)^{3/2}}, \tag{41}$$

where radius of perigee and apogee is chosen as $r_p = 10^7 m$ and $r_a = 3 \times 10^7 m$, respectively. The true anomaly rate and rate of change are calculated using (41) and

$$\ddot{\nu}_o(t) = \frac{-2\mu e_o \left(1 + e_o \cos \nu_o(t)\right)^3 \sin \nu_o(t)}{\left(\frac{1}{2}(r_p + r_a)\left(1 - e_o^2\right)\right)^3}. \tag{42}$$

The eccentricity can be calculated from r_a and r_p to be $e_o = 0.5$, and we see from (42), that the constant $\beta_{\ddot{\nu}_o}$ in Assumption 1 can be chosen as $\beta_{\ddot{\nu}_o} = 4 \times 10^{-7}$. The constant $\beta_{\dot{\nu}_o}$ in Assumption 1 can be chosen as $\beta_{\dot{\nu}_o} = 8 \times 10^{-4}$ as seen from (41). We have assumed that the reference frame is initially at perigee, that is $\nu_o(t_0) = 0$ and $\dot{\nu}_o(t_0) = v_p / r_p$, where

$$v_p = \sqrt{2\mu\left(\frac{1}{r_p} - \frac{1}{(r_p + r_a)}\right)}. \tag{43}$$

The horizon of the problem was chosen as $t_h = 30$, and the optimization problem was solved using $R = I_{24}$ and Q with ones along the diagonal except that $Q(2,2) = Q(3,3) = Q(14,14) = Q(15,15) = 20$. It can be argued that choosing the elements of R and Q is as hard as finding the parameters θ in the original problem. However, with the common choice of diagonal Q and R, choosing Q and R is more intuitive in the authors opinion. The optimization problem, was solved with all combinations of initial conditions $\theta_i \in \{0, 1, 2\}$, and the results are summarized in Table 1. The reason for

Table 1. Optimization result of 729 iterations, covering all combinations of initial conditions $\theta_i \in \{0, 1, 2\}$. By *Best value*, we mean the parameters which gives the lowest cost function value.

Parameter	k_l	ℓ_l	l_l	k_f	ℓ_f	l_f
Best value	0.3382	0.2658	2.0048	0.3738	0.3302	1.7644
Mean	0.3280	0.2809	1.6541	0.3855	0.3154	1.8660

solving the optimization problem with several different initial values, is to find, if not a global minimum, then at least a good local minimum. Looking at mean values in Table 1, we see that the parameters l_l and l_f are most affected by different initial guesses for the parameters.

The parameters giving the lowest cost function were then used as initial guess, and the problem was solved with different values for the horizon t_h. The results in Table 2 suggests that the found parameters represent a good local minimum.

Table 2. Optimization result with different values of the horizon length t_h

Horizon	k_l	ℓ_l	l_l	k_f	ℓ_f	l_f
20	0.3195	0.2743	1.5802	0.3829	0.3200	1.8948
50	0.3382	0.2658	2.0048	0.3738	0.3302	1.7644
100	0.3382	0.2658	2.0048	0.3738	0.3302	1.7644
300	0.3415	0.3089	1.9524	0.3631	0.3245	1.7608
1000	0.3382	0.2658	2.0048	0.3738	0.3302	1.7644

4.1 Simulations

We will now provide simulations of the spacecraft formation, using the gains achieved from the optimization procedure in the previous section. 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 reference orbit is generated by numerical integration of

$$\ddot{r}_o = -\frac{\mu}{|r_o|^3} r_o, \tag{44}$$

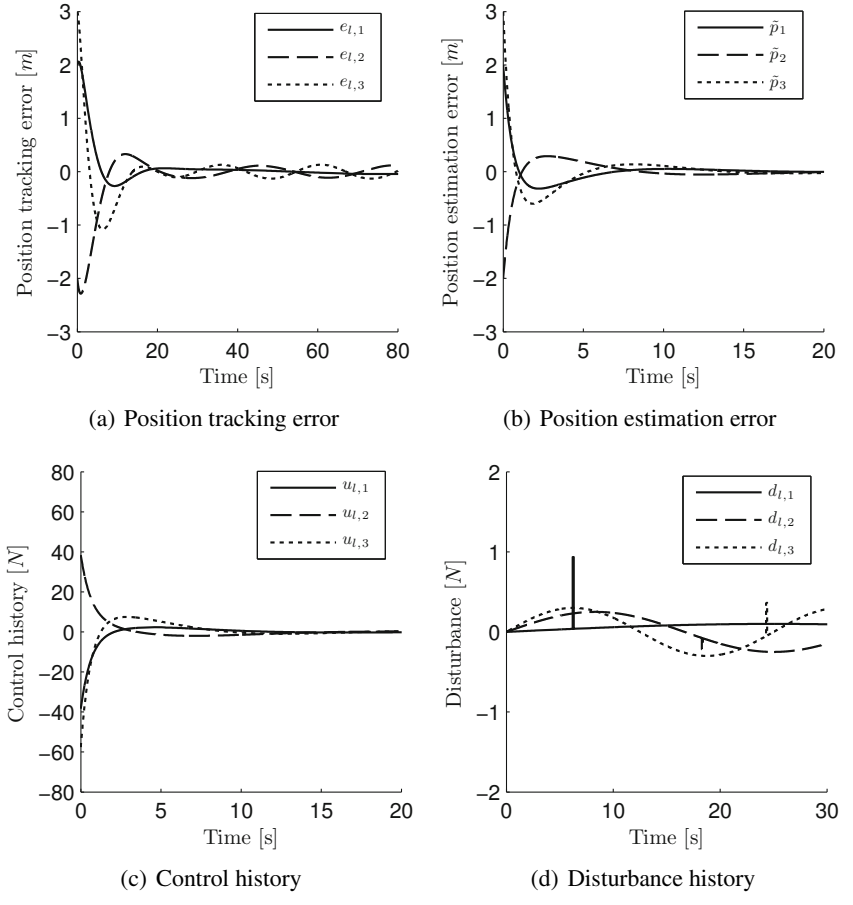
with $r_o(0) = (r_p, 0, 0)$ and $\dot{r}_o(0) = (0, v_p, 0)$, with v_p as in (43). Initial values of the leader spacecraft are $p(0) = (2,-2,3)^\top$ and $\dot{p}(0) = (0.4,-0.8,-0.2)^\top$. The initial values of the observer are chosen as $\hat{p}(0) = (0,0,0)^\top$ and $a_l(0) = (0,0,0)^\top$.

The reference trajectory of the follower spacecraft are chosen as the solutions of a special case of the Clohessy-Wiltshire equations, cf. [3]. We use

$$\rho_d(t) = \begin{bmatrix} 10\cos\nu_o(t) \\ -20\sin\nu_o(t) \\ 0 \end{bmatrix}. \tag{45}$$

This choice imposes that the two spacecraft evolve in the same orbital plane, and that the follower spacecraft makes a full rotation about the leader spacecraft at each 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.

With θ as the best value in Table 1, we find from (18) that $\kappa = 0.0901$. Over a 10 second interval (*i.e.* T=10), the average excitation must satisfy $E(T,\delta) \leq 0.0061\,\delta^2$, according to (17). 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 at most 0.1s. The continuous disturbances are 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. Notice from (7)



(a) Position tracking error (b) Position estimation error

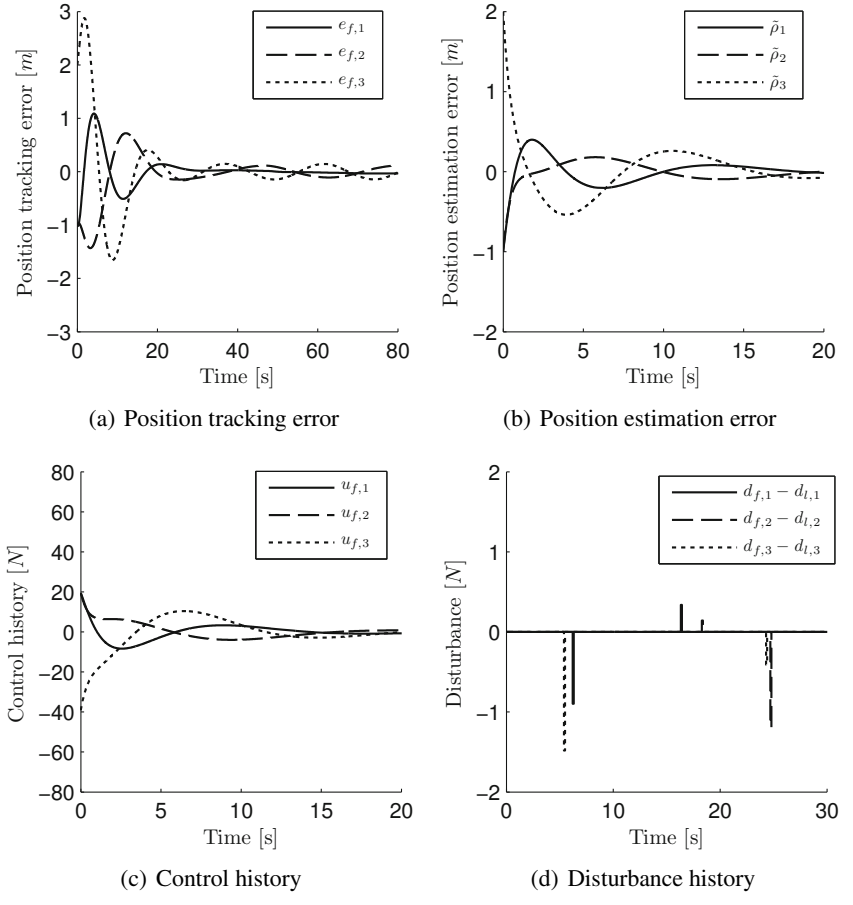
(c) Control history (d) Disturbance history

Fig. 1. Leader spacecraft

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 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 1(a), 1(b) and 1(c) show the position tracking error, position estimation error and control history of the leader spacecraft, whereas Figure 2(a), 2(b) and 2(c) are the equivalent figures for the follower spacecraft. Figure 1(d) and 2(d) show the effect of d_l and $d_f - d_l$ acting on the formation. Notice in Figure 2(d) 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



(a) Position tracking error (b) Position estimation error

(c) Control history (d) Disturbance history

Fig. 2. Follower spacecraft

the effect of the impacts has increased compared to the effect of the impacts on the leader spacecraft. Since $E = 1.42$, and should satisfy $E \leq 0.0061\delta^2$, this gives a very large δ. As can be seen from Figure 1(a), 1(b), 2(a), 2(b) the actual precision reached, is much better than the theoretical expectations. The reason for this is that the constraints on the control gains are based on Lyapunov analysis, which in general yields very conservative results, and also conservative estimates of the disturbances the control system is able to handle. However, we stress that the constraints on the gains based on moving average of the disturbance are much more relaxed than those obtained through a classical ISS approach, *i.e.* relying on the disturbance magnitude. Figure 1(c) and Figure 2(c), shows that the conservative estimates results in large transients in the actuation, but by finding the optimal parameters, the control efforts are heavily reduced compared to [6], while the tracking errors are kept at a reasonable level.

References

1. Angeli, D., Nešić, D.: Power characterizations of input-to-state stability and integral input-to-state stability. IEEE Transactions on Automatic Control 48, 1298–1303 (2001)
2. Brockett, R.W.: Finite Dimensional Linear Systems. Wiley, New York (1970)
3. Clohessy, W.H., Wiltshire, R.S.: Terminal guidance system for satellite rendezvous. Journal of Aerospace Sciences 27, 9 (1960)
4. Doyle, J.: Robust and optimal control. In: Proc. of the 35th Conference on Decision and Control (1996)
5. Freeman, R.A., Kokotović, P.V.: Robust nonlinear control design: state-space and Lyapunov techniques, Birkhuser (1996)
6. Grøtli, E.I., Chaillet, A., Panteley, E., Gravdahl, J.T.: Robustness of ISS systems to inputs with limited moving average, with application to spacecraft formations. In: Proc. of the 7th International Conference on Informatics in Control, Automation and Robotics (2010)
7. Houska, B., Diehl, M.: Robust nonlinear optimal control of dynamic systems with affine uncertainties. In: Proc. of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (2009)
8. Kazantizis, N., Kravaris, C., Tseronis, C., Wright, R.A.: Optimal controller tuning for nonlinear processes. Automatica 41, 79–86 (2005)
9. Kristiansen, R., Loría, A., Chaillet, A., Nicklasson, P.J.: Adaptive output feedback control of spacecraft relative translation. In: Proc. of the 45th Conference on Decision & Control, pp. 6010–6015 (2006)
10. Krstić, M., Li, Z.H.: Inverse optimal design of input-to-state stabilizing nonlinear controllers. IEEE Transactions on Automatic Control 43, 336–350 (1998)
11. Ploen, S.R., Scharf, D.P., Hadaegh, F.Y., Acikmese, A.B.: Dynamics of earth orbiting formations. In: Proc. of AIAA Guidance, Navigation and Control Conference (2004)
12. Praly, L., Wang, Y.: 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 (1996)
13. van der Schaft, A.J.: $\mathcal{L}_2$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$ control. IEEE Transactions on Automatic Control 37, 770–784 (1992)
14. Serrani, A.: Robust coordinated control of satellite formations subject to gravity perturbations. In: Proc. of the American Control Conference, pp. 302–307 (2003)
15. Sontag, E.D.: Smooth stabilization implies coprime factorization. IEEE Transactions on Automatic Control 34, 435–443 (1989)
16. Sontag, E.D.: A 'universal' construction of Artstein's theorem on nonlinear stabilization. Systems & Control Letters 13, 117–123 (1989)
17. Sontag, E.D., Wang, Y.: On characterizations of the input-to-state stability property. Systems & Control Letters 24, 351–359 (1995)
18. Tadmor, G.: Worst-case design in the time domain: The maximum principle and the standard $\mathcal{H}_\infty$ problem. Math. Control Signals Systems 3 (1989)
19. Zames, G.: Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses. IEEE Transactions on Automatic Control 26, 301–320 (1981)

Stability and Performance of Wireless Sensor Networks during the Tracking of Dynamic Targets

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

Abstract. The performance of Wireless Sensor Networks (WSNs) during the tracking of dynamic targets is addressed in this paper. The strategy outlined in this paper uses a Distributed implementation of a Kalman Filter to track dynamic targets. In contrast to the results reported in the literature, the approach in this paper 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. The tracking problem in WSNs is formulated mathematically and the stability and tracking error of the proposed strategy is rigorously analyzed. Numerical simulations are then used to demonstrate the utility of the proposed technique. 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.

Keywords: Distributed kalman filter, Wireless sensor networks, Target tracking.

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 [1, 2, 3, 4, 5, 6] Diffusion and directed diffusion approaches have been proposed to address coverage, routing, discovering, and sensing fusion issues in WSNs [7]. The application of WSNs in surveillance and monitoring of target areas have also been widely researched [8]. 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 [9, 10, 11, 12, 13, 14, 15]. The convergence analysis of extened Kalman Filters also proposed in [16, 17]. Both centralized and distributed implementation of the Kalman Filter was proposed to

J.A. Cetto et al. (Eds.): Informatics in Control, Automation and Robotics, LNEE 89, pp. 349–362.
springerlink.com © Springer-Verlag Berlin Heidelberg 2011

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 [15] where every sensor node takes measurements and communicates with the other nodes while simultaneously computing 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 [9, 13, 14] 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 this paper, the distributed Kalman Filter is proposed to estimate the position of the target. The 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 reported elsewhere in the literature. 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. This results in reduced communication costs in the entire sensor network and complexity of the tracking algorithm. Consequently, the performance of the distributed Kalman Filter is as good as that of the centralized Kalman Filter.

This approach is validated through mathematical analysis and simulation examples. 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. Another aspect of the proposed algorithm 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. Thus, nodes further away from the target are inactive and only a small subset of the nodes participates in the sensing. Prior to the start of the tracking, the knowledge of the maximum target velocity can be used, the boundary nodes of the sensor field are activated in round robin fashion discussed in [6] to save energy.

The rest of the paper is organized as follows: Section 2 is the problem formulation. Section 3 is the convergence analysis of the distributed Kalman Filter. Section 4 is discussion. In section 5, we show the numerical simulation. Section 6 and 7 is conclusion and appendix.

2 Problem Formulation

A closed and bounded sensor field in three dimensional (3D) Cartesian coordinate system is densely deployed with stationary sensor nodes. There is an intruder entering the sensor field with unknown nonlinear trajectory, the problem is how to the sensor network tracks the target correctly. It is assumed that each node has omnidirectional sensing capability to measure the distance between the target and itself.

At time k, the tracking system is governed the following equations

$$x_{k+1} = f(x_k) + w_k \tag{1}$$

$$y_{k+1} = h(x_{k+1}) + n_{k+1} \tag{2}$$

$x_k = [v_k, p_k]^T$ is the state of the target at time k where $v_k \in \mathbb{R}^3$ and $p_k \in \mathbb{R}^3$ are the velocity and position respectively in 3D coordinate system. $y_k \in \mathbb{R}^3$ is the measured position of the target which is calculated by trilateration algorithm (3). $w_k \in \mathbb{R}^6$ and $n_k \in \mathbb{R}^3$ are Gaussian distributed process noise with covariance matrix Q and measurement noise with covariance matrix N respectively. Q and N are assumed to be symmectric and positive semidefinite. $f(x_k)$ and $h(x_k)$ are assumed to be continuous with respect to time and differentiable with respect to x_k. Suppose that the maximum velocity of tracking targets is known. Thus, the domain $V \in \mathbb{R}^6$ of $f(x_k)$ in (1) is a compact and connected set.

The target, represented by ⋆ symbol shown in Fig. 1, is moving in the direction of vector $\overrightarrow{OE}$. The region $\boldsymbol{R}$ is defined by the sphere of 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. Fig. 1 is the projection of the region R onto a plane that is parallel to the moving direction of the target. All the sensor nodes inside the region of activation $\boldsymbol{R}$ are activated, while the

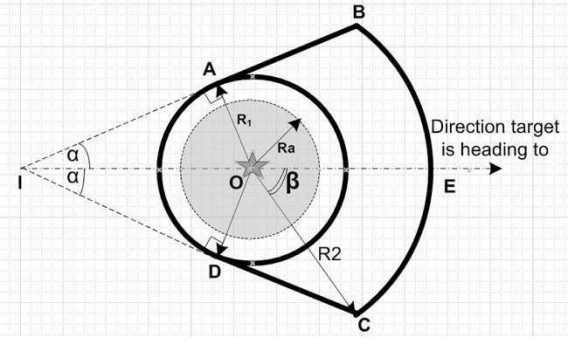


Fig. 1. The target represented by ⋆ at point O. The boundary of the region of activation *R* is limited by line AB, curve BC, line CD and curve CA (the bold line above). The curve BC is formed by part of the sphere$(\boldsymbol{O}, \boldsymbol{R_2})$. No nodes outside sphere $(\boldsymbol{O}, \boldsymbol{R_1})$ can sense the target. All the nodes inside *R* are activated. However, only the sensors inside the sphere $(\boldsymbol{O}, \boldsymbol{R_a})$ are actively taking measurement.

nodes outside the region are in sleep mode to save power. All the nodes inside the sphere (O, R_1) can sense the target while no node outside can detect the target. However, only nodes inside the sphere $(O,\ R_a)$ are actively taking measurements and reporting the data to the master node.

A master node is selected depending on two criteria: the distance to the target and residual power. The sensors inside the sphere with radius R_a take measurements and transfer the measured data to the master node. The master node runs the extended Kalman Filter for the system (1) and (2) in the distributed sense 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 $\boldsymbol{R}$.

2.1 Position Calculation

Suppose that at time k there are n sensor nodes that are actively taking measurements whose coordinates are $(x_{k1},\ y_{k1},\ z_{k1})$, $(x_{k2},\ y_{k2},\ z_{k2})$,.. $(x_{kn},\ y_{kn},\ z_{kn})$, and measured distances from each nodes to the target are $d_1,\ d_2,\ ..d_n$ respectively. These measurements are sent to the master node. Consequently, master nodes run the trilateration algorithm using least squares method. The solution of the target's coordinate $y_k = (x_{kt},\ y_{kt},\ z_{kt})$ in (2) is given by

$$y_k = \begin{bmatrix} x_{kt} \\ y_{kt} \\ z_{kt} \end{bmatrix} = (A^T A)^{-1} Ab \tag{3}$$

where A and b are in the following forms

$$A = \begin{bmatrix} 2(x_{k2} - x_{k1}) & 2(y_{k2} - y_{k1}) & 2(z_{k2} - z_{k1}) \\ 2(x_{k3} - x_{k2}) & 2(y_{k3} - y_{k2}) & 2(z_{k3} - z_{k2}) \\ \vdots & \vdots & \vdots \\ 2(x_{k1} - x_{kn}) & 2(y_{k1} - y_{kn}) & 2(z_{k1} - z_{kn}) \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} (\boldsymbol{d_1^2} - \boldsymbol{d_2^2}) + (\boldsymbol{x_2^2} + \boldsymbol{y_2^2} + \boldsymbol{z_2^2}) - (\boldsymbol{x_1^2} + \boldsymbol{y_1^2} + \boldsymbol{z_1^2}) \\ (\boldsymbol{d_2^2} - \boldsymbol{d_3^2}) + (\boldsymbol{x_3^2} + \boldsymbol{y_3^2} + \boldsymbol{z_3^2}) - (\boldsymbol{x_2^2} + \boldsymbol{y_2^2} + \boldsymbol{z_2^2}) \\ \vdots \\ (\boldsymbol{d_n^2} - \boldsymbol{d_1^2}) + (\boldsymbol{x_1^2} + \boldsymbol{y_1^2} + \boldsymbol{z_1^2}) - (\boldsymbol{x_n^2} + \boldsymbol{y_n^2} + \boldsymbol{z_n^2}) \end{bmatrix}$$

Since the measurement y_k in (3) is the position of the target, the measurement equation (2) becomes linear equation

$$h(x_{k+1}) = Hx_{k+1} \text{ where } H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

2.2 Distributed Kalman Filter

The distributed Kalman filter of the system (1), (2) and (4) is run at only a given master node.

$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - H\hat{x}_{k+1|k}) \tag{5}$$

$$K_{k+1} = P_{k+1|k}H^T(HP_{k+1|k}H^T + N)^{-1} \tag{6}$$

$$P_{k+1} = (I - K_{k+1}H)P_{k+1|k} \tag{7}$$

$$\hat{x}_{k+1|k} = f(\hat{x}_k) \tag{8}$$

$$P_{k+1|k} = FP_kF^T + Q \tag{9}$$

$$F_{\mathrm{k}} = \frac{\partial \mathrm{f}(\mathrm{x_k})}{\partial \mathrm{x_k}}|_{x_k=\widehat{x_k}} \tag{10}$$

Equations (5)-(7) reflect measurement update, and equations (9)-(10) reflect time update. F is the value of Jacobian matrix of function f in (1) at time k. The initial value P_0 of P_k in (9) is symmetric positive semidefinite matrix. If the sampling time is fixed Δt, then F_k in (10) a constant matrix.

$$F = \frac{\partial \mathrm{f}(\mathrm{x_k})}{\partial \mathrm{x_k}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \Delta t & 0 & 0 & 1 & 0 & 0 \\ 0 & \Delta t & 0 & 0 & 1 & 0 \\ 0 & 0 & \Delta t & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

3 Performance Analysis

3.1 Assumptions

Before doing the stability analysis, the following assumptions are made.

(i) Function $f(x_k)$ in (1) is two time differentiable with respect to x_k.

(ii) The state error is Gaussian distributed and the covariance matrix Q is uniformly bounded.

(iii) The range measurement error is Gaussian distributed.

(iv) The sensor nodes are deployed dense enough that the joint measurement error is close to Gaussian with zero mean and the joint measurement covariance matrix N is uniformly bounded.

(v) The sampling frequency is high enough.

(vi) There is no delay when there is changing in master nodes.

(vii) Every measured sensor node is within one hop to the target

3.2 Stability Analysis

The state of equation (1) can be linearized as follows.

$$x_{k+1} = Fx_k + \Delta F_k x_k + w_k = a_k F x_k + w_k \tag{12}$$

Where ΔF_k and a_k are the diagonal matrix represented the nonlinear terms. If the system (1) is linear, $\Delta F_k = 0$ and a_k is equal to identity matrix. From (3), the equation (2) becomes

$$y_{k+1} = h(x_{k+1}) = Hx_{k+1} + n_k \tag{13}$$

The Lyapunov function candidate is chosen as

$$V_{k+1} = \tilde{x}_{k+1}^T P_{k+1}^{-1} \tilde{x}_{k+1} \text{ where } \tilde{x}_{k+1} = x_{k+1} - \hat{x}_{k+1} \tag{14}$$

From (20) we have

$$\tilde{x}_{k+1|k} = a_k F \tilde{x}_k \text{ and } e_{k+1} = y_{k+1} - Fx_{k+1|k} = H\tilde{x}_{k+1|k} \tag{15}$$

Then, from (23)-(27) and assumption (v) we have

$$\begin{aligned} V_{k+1} &= \left(\tilde{x}_{k+1|k} - P_{k+1}HN^{-1}e_{k+1}\right)^T P_{k+1}^{-1}\left(\tilde{x}_{k+1|k} - P_{k+1}HN^{-1}e_{k+1}\right) \\ &= \tilde{x}_{k+1|k}^T P_{k+1}^{-1}\tilde{x}_{k+1|k} - e_{k+1}^T[N^{-1}H^T P_{k+1}HN^{-1} - 2N^{-1}]e_{k+1} \end{aligned} \tag{16}$$

From (6), (7) and (9) we have $P_{k+1}^{-1} = P_{k+1|k}^{-1} + HN^{-1}H^T$

$$V_k = V_{k+1} - V_k = \tilde{x}_k^T[F^T a_k^T(FP_kF^T + Q)^{-1}a_kF - P_k^{-1}]\tilde{x}_k e_{k+1}^T N^{-1}(HP_{k+1}H^T N)N^{-1}e_{k+1}$$

$$= (F\tilde{x}_k)^T[a_k^T(FP_kF^T + Q)^{-1}a_k - (FP_kF^T)^{-1}]F\tilde{x}_k + e_{k+1}^T N^{-1}(HP_{k+1}H^T - N)N^{-1}e_{k+1}$$

For any vector m we have

$$\begin{aligned} &m^T[a_k^T(FP_kF^T + Q)^{-1}a_k - (FP_kF^T)^{-1}]m \\ &\qquad \le \lambda_{max}((FP_kF^T + Q)^{-1})\|a_k\|^2\|m\|^2 - \lambda_{min}((FP_kF^T)^{-1})\|m\|^2 \end{aligned}$$

Thus, if

$$HP_{k+1}H^T - N \le 0 \text{ and} \tag{17}$$

$$\lambda_{max}(a_k) \le \frac{\lambda_{min}((FP_kF^T)^{-1})}{\lambda_{max}((FP_kF^T + Q)^{-1}} \tag{18}$$

Then $\Delta V_k \le 0$.

On the other hand, the matrix

$$O_k = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \text{ where } \begin{bmatrix} H \\ HF \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \Delta t & 0 & 0 & 1 & 0 & 0 \\ 0 & \Delta t & 0 & 0 & 1 & 0 \\ 0 & 0 & \Delta t & 0 & 0 & 1 \end{bmatrix}$$

Clearly O_k is full rank then F_k and H_k are observable. By LaSalle lamma, we conclude that

$$\lim_{k\to\infty} \tilde{x}_k = 0 \tag{19}$$

4 Discussion

The range measurements using in (10) normally have noise. If the noise statistic is assumed to be Gaussian, then the jointly noise distribution in (2) or in (9) is neither Gaussian nor zero mean in general. In this subsection, it is stated that if the sensors are deployed dense enough, the jointly measurement error statistic v_k (2) is bounded and closed to the Gaussian distribution. The jointly distribution depends not only on the range measurement noise statistic but also on the geometry of the sensor nodes and the target.

Suppose that the maximum sensing radius of a sensor node is R_S and the uncertainty in range measurement error is u. It means that the measurement error is smaller than u with a specified probability of p.

If the target is covered by only one sensor, the target is in the shading donut shape (Fig. 2a) with the probability of p. The maximum uncertainty of the target is $R_S + u$ in this case. If the target is covered by two sensor, then the target is in either two shading area in Fig. 2b. The maximum uncertainty is

$$l = \sqrt[2]{\left(\frac{d}{2}\right)^2 - R_s^2} \tag{20}$$

where $d = S_1S_2 \leq 2R_S$ is the distance between two sensor nodes. The probability the target is either in open of the two shading region in Fig. 3b is p^2 the following theorem yields a bound on the area of each shading region.

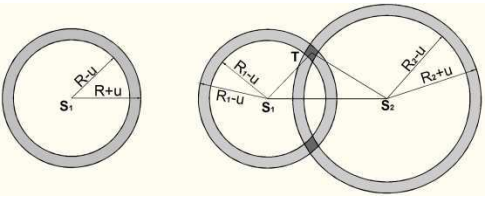


Fig. 2. The uncertainty in range measurement. The target is covered by one sensor (a) and two sensors (b).

Theorem 1. Suppose that the target T is covered by 2 sensor nodes S_1 and S_2, and the angle at T formed by 2 vertices S_1, and S_2 is $\widehat{S_1TS_2} = \propto$ The maximum uncertainty of the target in one region is (one shading region Fig. 2b)

$$l \leq \min\left\{\frac{2\sqrt{2}u}{sin\frac{\propto}{2}}; R+u\right\}, \tag{21}$$

The first term in (22) can be easily verified by fundumental geometry, and the the second term is the maxium untertainty of a target in Fig. 2a.

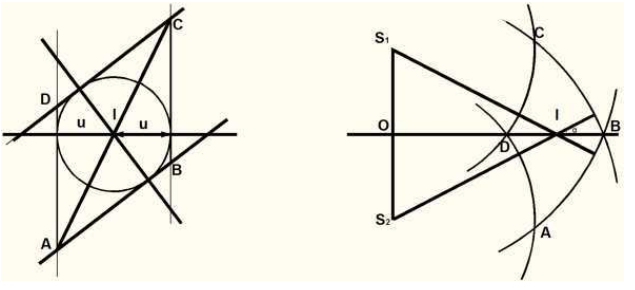


Fig. 3. The uncertainty in range measurement. The target is covered by one sensor (a) and two sensors (b).

In Fig. 3a, as R goes to ∞, the upper bond of uncertainty approaches

$$l = \frac{2u}{sin\frac{\propto}{2}} \tag{22}$$

In short, the jointly measurement uncertainty depends on both sensor measured uncertainty u and the angle α. Thus, if given u is fixed, sensor nodes can be deployed densely enough such that $sin\frac{\propto}{2}$ is within a certain bound. Moreover, the jointly measurement noise distribution is close to Gaussian. Hence, the measurement noise covariance matrix N is uniformly bounded.

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 Numerical Examples

5.1 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 [6]. From region **R**, the number of sensor nodes inside the

sphere radius R_a is n_a. n_i is the number of sensor nodes outside the sphere with radius of R_a, but inside the region $\boldsymbol{R}$. 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.

$$P_{meas} = n_a(P_{Tx} + P_{Rx}) \tag{23}$$

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.

$$P_{broadcast} = (n_a + n_i)P_{Rx} + P_{Tx} \tag{24}$$

Each active node, except measurement nodes, consumes an amount of the idle energy

$$P_{idle} = (n_b + n_i)P_i \tag{25}$$

The other nodes are sleeping, and the total power consumed by these nodes is

$$P_{sleep} = (n - n_a - n_i - n_b)P_s \tag{26}$$

Then total consumed power is

$$P_w = P_{meas} + P_{broadcast} + P_{idle} + P_{sleep} \tag{27}$$

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 disktributed. 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. The sensor field is assumed to be a square of the dimension 10×10 units as seen in Fig. 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.

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ z_k &= Hx_k + v_k \end{aligned} \tag{28}$$

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 Fig. 3.

Table 1. Performance analysis

Average measured sensors	Average active sensors	Error variance without Kalman Filter ($\times10^{-3}$)	Error variance with Kalman Filter ($\times10^{-3}$)	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

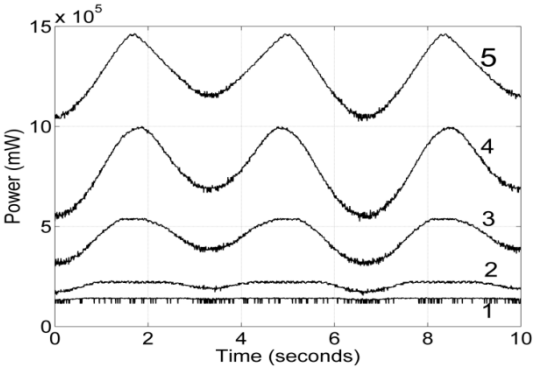


Fig. 4. Without the Kalman Filter, the line number 1, 2, 3, 4, and 5 have average measured sensor nodes of 4.5, 17.5, 60.4, 130.8, and 279 respectively. For the line number 3 to 5, the total power consumption is fluctuated because when the target moves close to the boundary the number of active sensors 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.

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 Fig. 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.

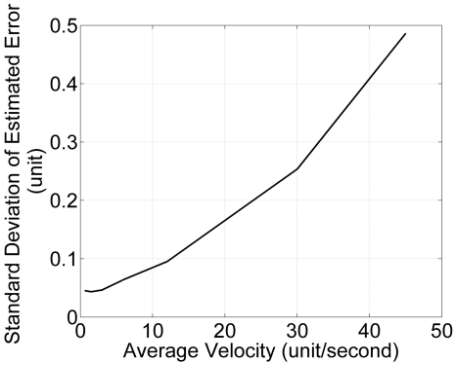


Fig. 5. Average velocity increases as the estimated error has a larger standard deviation

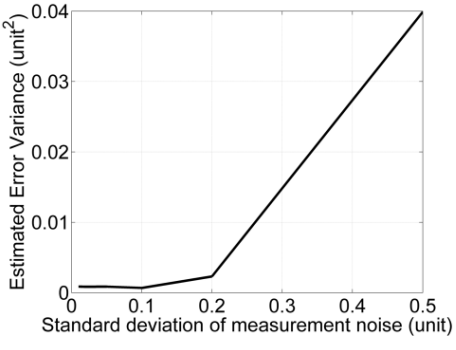


Fig. 6. When the distance measurement is subjected to a larger noise, the variance of estimated tracking error becomes bigger

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 Fig. 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.

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

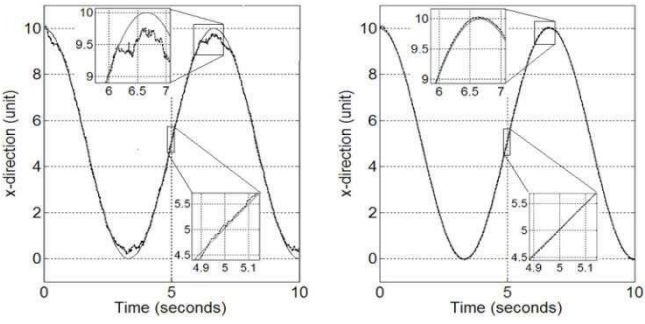


Fig. 7. The true and the estimated trajectory with different measurement noise levels. The standard deviation of measurement noise is 0.5 in the left side while it is 0.04 on the right side. (- True trajectory; -- Estimated trajectory).

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 Fig. 8. When there is a change in the master node, the Kalman Filter requires some extra time steps to converge.

Scenario 6. As shown in Fig. 10, 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.

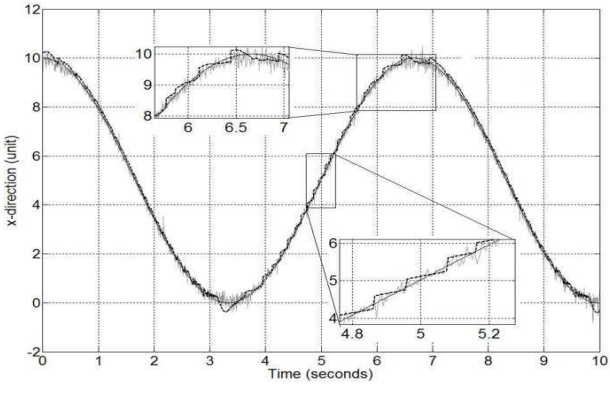


Fig. 8. Without sharing the state vector and covariance matrix to the subsequently master node, each master node has to start the Kalman Filter from scratch. The measurement noise standard deviation is 0.2, while the number of average measured sensor nodes is 7.6. (- True trajectory; -- Estimated trajectory).

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.

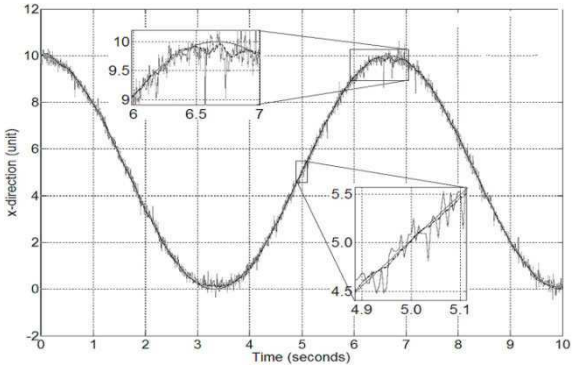


Fig. 9. Without sharing the state vector and covariance matrix to the subsequently master node, each master node has to start the Kalman Filter from scratch. The measurement noise standard deviation is 0.2, while the number of average measured sensor nodes is 7.6.

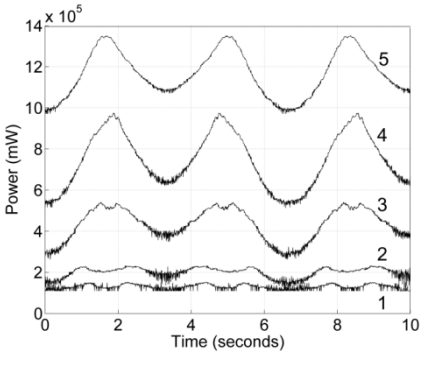


Fig. 10. Power consumption of one sampling cycle in random deployment. There are 441 sensor nodes deployed in the sensor field of 10×10. The line number 1, 2, 3, 4 and 5 have average measured sensors of 3.4, 15.7 59.5, 127.2, and 259.7 respectively. . (- True trajectory; - - Estimated trajectory).

6 Conclusions

In this paper, a distributed computation approach is proposed for the tracking dynamic targets using a Wireless Sensor Network (WSN). The tracking problem is mathematically formulated and the tracking error of the distributed Kalman Filter is rigorously analyzed. It is shown that the proposed algorithm is stable and can track

the target with predetermined error bounds on the performance. The algorithm is also robust to changes in the velocity of the target and measurement noises. It is shown that the algorithm reduces the total power consumption in the network compared to similar algorithms reported in literature. The theoretical proofs are augments by numerical simulations that demonstrate the flexibility and power of the proposed tracking algorithm.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., et al.: A survey on sensor network. IEEE Communications Magazine 8, 102–114 (2002)
2. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications 11(6), 6–28 (2004)
3. Cardei, M., Thai, M.T., Li, Y., et al.: Energy-efficient target coverage in wireless sensor networks
4. Chiang, C., Wu, H., Liu, W., et al.: Routing In Clustered Multihop, Mobile Wireless Networks With Fading Channel, pp. 197–211
5. Watfa, M.K., Commuri, S.: The 3-Dimensional Wireless Sensor Network Coverage Problem, pp. 856–861
6. Watfa, M.K., Commuri, S.: Optimal sensor placement for Border Perambulation
7. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks
8. Chen, M., Gonzalez, S., Leung, V.C.M.: Applications and design issues for mobile agents in wireless sensor networks. IEEE Wireless Communications 14(6), 20–26 (2007)
9. Cattivelli, F.S., Lopes, C.G., Sayed, A.H.: Diffusion strategies for distributed kalman filtering: Formulation and performance analysis
10. Hashemipour, H.R., Roy, S., Laub, A.J.: Decentralized structures for parallel Kalman filtering. IEEE Transaction on Automatic Control 33(1), 88–94 (1998)
11. Kim, J.-H., West, M., Scholte, E., et al.: Multiscale consensus for decentralized estimation and its application to building systems, pp. 888–893
12. Mutambara, G.O.: Decentralized estimation and control for multisensor systems. CRC Press, Boca Raton (1998)
13. Olfati-Saber, R.: Distributed Kalman filtering for sensor networks, pp. 5492–5498
14. Olfati-Saber, R., Shamma, J.S.: Consensus Filters for Sensor Networks and Distributed Sensor Fusion, pp. 6698–6703
15. Rao, B.S., Durrant-Whyte, H.F.: Fully decentralized algorithm for multisensor Kalman filtering. IEE Proceedings-D Control Theory & Application 138(5), 413–420 (1991)
16. Boutayeb, M., Rafaralahy, H., Darouach, M.: Covergence analysis of the Extended Kalman Filter as an observer for non-linear discrete-time system, pp. 1555-1560
17. Song, Y., Grizzle, J.W.: The Extended Kalman Filter as a Local Asymptotic Observer for Nonlinear Discrete-Time Systems. Journal of Mathematical Systems, Estimation and Control 5(1), 59–78 (1995)

Author Index