Mechatronic Systems, Simulation, Modelling and Control

Mechatronic Systems, Simulation, Modelling and Control

Edited by
Annalisa Milella, Donato Di Paola
and Grazia Cicirelli

In-Tech
intechweb.org

Published by In-Teh

In-Teh
Olajnica 19/2, 32000 Vukovar, Croatia

Abstracting and non-profit use of the material is permitted with credit to the source. Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. Publisher assumes no responsibility liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained inside. After this work has been published by the In-Teh, authors have the right to republish it, in whole or part, in any publication of which they are an author or editor, and the make other personal use of the work.

© 2010 In-teh
www.intechweb.org
Additional copies can be obtained from:
publication@intechweb.org

First published March 2010
Printed in India

Technical Editor: Sonja Mujacic
Cover designed by Dino Smrekar

Mechatronic Systems, Simulation, Modelling and Control,
Edited by Annalisa Milella, Donato Di Paola and Grazia Cicirelli

p. cm.
ISBN 978-953-307-041-4

Preface

Mechatronics, the synergistic blend of mechanics, electronics, and computer science, has evolved over the past twenty-five years, leading to a novel stage of engineering design. By integrating the best design practices with the most advanced technologies, mechatronics aims at realizing highquality products, guaranteeing, at the same time, a substantial reduction of time and costs of manufacturing. Mechatronic systems are manifold, and range from machine components, motion generators, and power producing machines to more complex devices, such as robotic systems and transportation vehicles. With its 15 chapters, which collect contributions from many researchers worldwide, this book provides an excellent survey of recent work in modelling and control of electromechanical components, and mechatronic machines and vehicles.

A brief description of every chapter follows. The book begins with eight chapters related to modelling and control of electromechanical machines and machine components. Chapter 1 presents an electromechanical model for a ring-type Piezoelectric Transformer (PT). The presented model provides a general framework capable of serving as a design tool for optimizing the configuration of a PT. Chapter 2 develops a current harmonic model for high-power synchronous machines. The use of genetic algorithm-based optimization techniques is proposed for optimal PWM. Chapter 3 deals with the control of a servo mechanism with significant dry friction. The proposed procedure for system structure identification, modelling, and parameter estimation is applicable to a wide class of servos. The solution is described in detail for a particular actuator used in the automotive industry, i.e., the electronic throttle. Chapter 4 proposes a diagram of H∞ regulation, linked to the field oriented control, that allows for a correct transient regime and good robustness against parameter variation for an induction motor. In Chapter 5, a pump-displacement-controlled actuator system with applications in aerospace industry is modelled using the bond graph methodology. Then, an approach is developed towards simplification and model order reduction for bond graph models. It is shown that using a bond graph model, it is possible to design fault detection and isolation algorithms, and to improve monitoring of the actuator. A robust controller for a Travelling Wave Ultrasonic Motor (TWUM) is described in Chapter 6. Simulation and experimental results demonstrate the effectiveness of the proposed controller in extreme operating conditions. Chapter 7 introduces a resonance frequency tracing system without the loop filter based on digital Phase Locked Loop (PLL). Ultrasonic dental scalar is presented as an example of application of the proposed approach. Chapter 8 presents the architecture of the Robotenis system composed by a robotic arm and a vision system. The system tests joint control and visual servoing algorithms. The main objective is to carry out tracking tasks in three dimensions and dynamical environments.

Chapters 9-11 deal with modelling and control of vehicles. Chapter 9 concerns the design of motion control systems for helicopters, presenting a nonlinear model for the control of a three-DOF helicopter. A helicopter model and a control method of the model are also presented and validated experimentally in Chapter 10. Chapter 11 introduces a planar laboratory testbed for the simulation of autonomous proximity manoeuvres of a uniquely control actuator configured spacecraft.The design of complex mechatronic systems requires the development and use of software tools, integrated development environments, and systematic design practices. Integrated methods of simulation and Real-Time control aiming at improving the efficiency of an iterative design process of control systems are presented in Chapter 12. Reliability analysis methods for an embedded Open Source Software (OSS) are discussed in Chapter 13. A new specification technique for the conceptual design of mechatronic and self-optimizing systems is presented in Chapter 14. The railway technology is introduced as a complex example, to demonstrate how to use the proposed technique, and in which way it may contribute to the development of future mechanical engineering systems. Chapter 15 provides a general overview of design specificities including mechanical and control considerations for micro- mechatronic structures. It also presents an example of a new optimal synthesis method, to design topology and associated robust control methodologies for monolithic compliant microstructures.

Annalisa Milella, Donato Di Paola and Grazia Cicirelli

Contents

Preface V

1. Electromechanical Analysis of a Ring-type Piezoelectric Transformer 001
   Shine-Tzong Ho

2. Genetic Algorithm–Based Optimal PWM in High Power Synchronous Machines and Regulation of Observed Modulation Error 017
   Alireza Rezazade, Arash Sayyah and Mitra Aflaki

3. Modelling and Control of Electromechanical Servo System with High Nonlinearity 045
   Grepl, R.

4. Robust Shaping Indirect Field Oriented Control for Induction Motor 059
   M. Boukhnifer, C. Larouci and A. Chaibet

5. Modeling and Fault Diagnosis of an Electrohydraulic Actuator System with a Multidisciplinary Approach Using Bond Graph 073
   M. H. Toufighi, S. H. Sadati and F. Najafi

6. Robust Control of Ultrasonic Motor Operating under Severe Operating Conditions 089
   Moussa Boukhnifer, Antoine Ferreira and Didier Aubry

7. Resonance Frequency Tracing System for Langevin Type Ultrasonic Transducers 105
   Yutaka Maruyama, Masaya Takasaki and Takeshi Mizuno

8. New visual Servoing control strategies in tracking tasks using a PKM 117
   A. Traslosheros, L. Angel, J. M. Sebastián, F. Roberti, R. Carelli and R. Vaca

9. Nonlinear Adaptive Model Following Control for a 3-DOF Model Helicopter 147
   Mitsuaki Ishitobi and Masatoshi Nishi

10. Application of Higher Order Derivatives to Helicopter Model Control 173
    Roman Czyba and Michal Serafin

11. Laboratory Experimentation of Guidance and Control of Spacecraft During On-orbit Proximity Maneuvers 187
    Jason S. Hall and Marcello Romano

12. Integrated Environment of Simulation and Real-Time Control Experiment for Control system 223
Kentaro Yano and Masanobu Koga

13. Reliability Analysis Methods for an Embedded Open Source Software 239
Yoshinobu Tamura and Shigeru Yamada

14. Architecture and Design Methodology of Self-Optimizing Mechatronic Systems 255
Prof. Dr.-Ing. Jürgen Gausemeier and Dipl.-Wirt.-Ing. Sascha Kahl

15. Contributions to the Multifunctional Integration for Micromechatronic Systems 287
M. Grossard Mathieu and M. Chaillet Nicolas

1

Electromechanical Analysis of a Ring-type Piezoelectric Transformer

Shine-Tzong Ho
Kaohsiung University of Applied Sciences
Taiwan

1. Introduction

The idea of a piezoelectric transformer (PT) was first implemented by Rosen (Rosen, 1956), as shown in Fig.1. It used the coupling effect between electrical and mechanical energy of piezoelectric materials. A sinusoidal signal is used to excite mechanical vibrations by the inverse piezoelectric effect via the driver section. An output voltage can be induced in the generator part due to the direct piezoelectric effect. The PT offers many advantages over the conventional electromagnetic transformer such as high power-to-volume ratio, electromagnetic field immunity, and nonflammable.

Due to the demand on miniaturization of power supplying systems of electrical equipment, the study of PT has become a very active research area in engineering. In literatures (Sasaki, 1993; Bishop, 1998), many piezoelectric transformers have been proposed and a few of them found practical applications. Apart from switching power supply system, a Roson-type PT has been adopted in cold cathode fluorescent lamp inverters for liquid-crystal display. The PT with multilayer structure to provide high-output power may be used in various kinds of power supply units. Recently, PT of ring (Hu, 2001) or disk (Laoratanakul, 2002) shapes have been proposed and investigated. Their main advantages are simple structure and small size. In comparing with the structure of a ring and a disk, the PZT ring offers higher electromechanical coupling implies that a ring structure is more efficient in converting mechanical energy to electrical energy, and vice versa, which is essential for a high performance PT.

Different from all the conventional PT, the ring-type PT requires only a single poling process and a proper electrode pattern, and it was fabricated by a PZT ring by dividing one of the electrodes into two concentric circular regions. Because of the mode coupling effect and the complexity of vibration modes at high frequency, the conventional lumped-equivalent circuit method may not accurately predict the dynamic behaviors of the PT.

In this chapter, an electromechanical model for a ring-type PT is obtained based on Hamilton's principle. In order to establish the model, vibration characteristics of the piezoelectric ring with free boundary conditions are analyzed in advance, and the natural frequencies and mode shapes are obtained. In addition, an equivalent circuit model of the PT is obtained based on the equations of the motion for the coupling electromechanical system. Furthermore, the voltage step-up ratio, input impedance, output impedance, input

power, output power, and efficiency for the PT will be conducted. Then, the optimal load resistance and the maximum efficiency for the PT will be calculated.

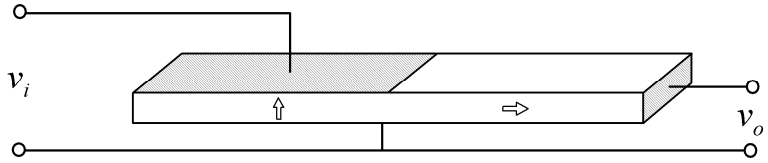


Fig. 1. Structure of a Rosen-type piezoelectric transformer.

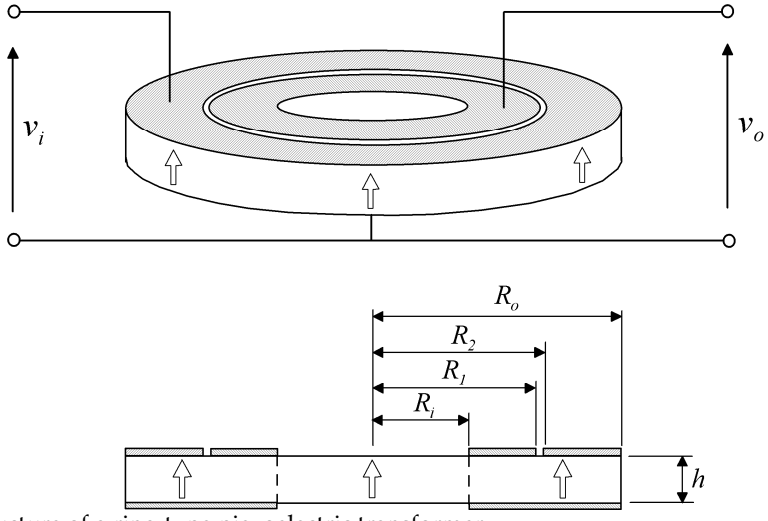


Fig. 2. Structure of a ring-type piezoelectric transformer.

2. Theoretical Analysis

2.1 Vibration Analysis of the Piezoelectric Ring

Fig.2 shows the geometric configuration of a ring-type PT with external radius R_o, internal radius R_i, and thickness h. The ring is assumed to be thin, $h << R_i$. The cylindrical coordinate system is adopted where the r-θ plane is coincident with the mid-plane of the undeformed ring, and the origin is in the center of the ring. The piezoelectric ring is polarized in the thickness direction, and two opposite surfaces are covered by electrodes. The constitutive equations for a piezoelectric material with crystal symmetry class C_{6v} can be expressed as follows.

$$\begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{\theta z} \\ \gamma_{zr} \\ \gamma_{r\theta} \end{Bmatrix} = \begin{bmatrix} s_{11}^E & s_{12}^E & s_{13}^E & 0 & 0 & 0 \\ s_{12}^E & s_{11}^E & s_{13}^E & 0 & 0 & 0 \\ s_{13}^E & s_{13}^E & s_{33}^E & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{44}^E & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{44}^E & 0 \\ 0 & 0 & 0 & 0 & 0 & s_{66}^E \end{bmatrix} \begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{\theta z} \\ \tau_{zr} \\ \tau_{r\theta} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & d_{31} \\ 0 & 0 & d_{31} \\ 0 & 0 & d_{33} \\ 0 & d_{15} & 0 \\ d_{15} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} E_r \\ E_\theta \\ E_z \end{Bmatrix} \quad (1a)$$

$$\begin{Bmatrix} D_r \\ D_\theta \\ D_z \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{15} & 0 \\ 0 & 0 & 0 & d_{15} & 0 & 0 \\ d_{31} & d_{31} & d_{33} & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \gamma_{\theta z} \\ \gamma_{zr} \\ \gamma_{r\theta} \end{Bmatrix} + \begin{bmatrix} \varepsilon_{11}^T & 0 & 0 \\ 0 & \varepsilon_{11}^T & 0 \\ 0 & 0 & \varepsilon_{33}^T \end{bmatrix} \begin{Bmatrix} E_r \\ E_\theta \\ E_z \end{Bmatrix} \quad (1b)$$

where σ_r, σ_θ, σ_z, $\tau_{\theta z}$, τ_{zr}, $\tau_{\theta r}$ are the components of the stress, ε_r, ε_θ, ε_z, $\gamma_{\theta z}$, γ_{zr}, $\gamma_{\theta r}$ are the components of the strain, and all the components are functions of r, θ, z, and t. ${s_{11}}^E$, ${s_{12}}^E$, ${s_{13}}^E$, ${s_{33}}^E$, ${s_{44}}^E$, ${s_{66}}^E$ are the compliance constants, d_{15}, d_{31}, d_{33} are the piezoelectric constants, ${\varepsilon_{11}}^T$, ${\varepsilon_{33}}^T$ are the dielectric constants, D_r, D_θ, D_z are the components of the electrical displacement, and E_r, E_θ, E_z are the components of the electrical field. The piezoelectric material is isotropic in the plane normal to the z-axis. The charge equation of electrostatics is represented as:

$$\frac{\partial D_r}{\partial r} + \frac{1}{r}\frac{\partial D_\theta}{\partial \theta} + \frac{1}{r}D_r + \frac{\partial D_z}{\partial z} = 0 \quad (2)$$

The electric field-electric potential relations are given by:

$$E_r = -\frac{\partial \varphi}{\partial r},\ E_\theta = -\frac{1}{r}\frac{\partial \varphi}{\partial \theta},\ E_z = -\frac{\partial \varphi}{\partial z}, \quad (3)$$

where φ is the electrical potential. The differential equations of equilibrium for three-dimensional problems in cylindrical coordinates are:

$$\frac{\partial \sigma_r}{\partial r} + \frac{1}{r}\frac{\partial \tau_{r\theta}}{\partial \theta} + \frac{\partial \tau_{zr}}{\partial z} + \frac{\sigma_r - \sigma_\theta}{r} = \rho \frac{\partial^2 u_r}{\partial t^2}, \quad (4a)$$

$$\frac{\partial \tau_{r\theta}}{\partial r}+\frac{1}{r}\frac{\partial \sigma_{\theta}}{\partial \theta}+\frac{\partial \tau_{\theta z}}{\partial z}+\frac{2\tau_{r\theta}}{r}=\rho\frac{\partial^2 u_{\theta}}{\partial t^2}, \tag{4b}$$

$$\frac{\partial \tau_{zr}}{\partial r}+\frac{1}{r}\frac{\partial \tau_{\theta z}}{\partial \theta}+\frac{\partial \sigma_{z}}{\partial z}+\frac{\tau_{zr}}{r}=\rho\frac{\partial^2 u_{z}}{\partial t^2}, \tag{4c}$$

where $u_r(r,\theta,z,t)$, $u_\theta(r,\theta,z,t)$, $u_z(r,\theta,z,t)$ are the displacements of the ring in the radial, tangential, and transverse direction, respectively. And ρ is the material density. The strain-displacement relations for three-dimensional problems in cylindrical coordinates are given by:

$$\varepsilon_r=\frac{\partial u_r}{\partial r},\ \varepsilon_\theta=\frac{u_r}{r}+\frac{1}{r}\frac{\partial u_\theta}{\partial \theta},\ \varepsilon_z=\frac{\partial u_z}{\partial z}, \tag{5a}$$

$$\gamma_{r\theta}=\frac{1}{r}\frac{\partial u_r}{\partial \theta}+\frac{\partial u_\theta}{\partial r}-\frac{u_\theta}{r}, \tag{5b}$$

$$\gamma_{\theta z}=\frac{\partial u_\theta}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial \theta}, \tag{5c}$$

$$\gamma_{zr}=\frac{\partial u_z}{\partial r}+\frac{\partial u_r}{\partial z}. \tag{5d}$$

Because the piezoelectric disk is thin and the deformation is small, the kirchoff assumption is made. The kirchoff assumptions are as follows:

$$u_r(r,\theta,z,t)=u_0(r,\theta,t)+z\frac{\partial w_0(r,\theta,t)}{\partial r}, \tag{6}$$

$$u_\theta(r,\theta,z,t)=v_0(r,\theta,t)+\frac{z}{r}\frac{\partial w_0(r,\theta,t)}{\partial r}, \tag{7}$$

$$u_z(r,\theta,z,t)=w_0(r,\theta,t), \tag{8}$$

where u_0, v_0, w_0 represent the radial, the tangential, and the transverse displacements of the middle surface of the plane, respectively. After inserting (6)-(8) into (5a), (5b), the strain-displacement relations can be obtained as:

$$\varepsilon_r = \frac{\partial u_r}{\partial r} = \frac{\partial u_0}{\partial r} + z\frac{\partial^2 w_0}{\partial r^2}, \tag{9a}$$

$$\varepsilon_\theta = \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = \frac{u_0}{r} + \frac{1}{r}\frac{\partial v_0}{\partial \theta} + z\left[\frac{\partial w_0}{\partial r} + \frac{\partial}{\partial \theta}\left(\frac{1}{r}\frac{\partial w_0}{\partial \theta}\right)\right], \tag{9b}$$

$$\begin{aligned}\gamma_{r\theta} = {} & \frac{1}{r}\frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} = \frac{1}{r}\frac{\partial u_0}{\partial \theta} + \frac{\partial v_0}{\partial r} - \frac{v_0}{r} \\ & + \frac{z}{r}\frac{\partial^2 w_0}{\partial r \partial \theta} + z\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial w_0}{\partial \theta}\right) - \frac{z}{r^2}\frac{\partial w_0}{\partial \theta}\end{aligned}, \tag{9c}$$

Since the ring is thin, stress σ_z can be neglected relative to the other stresses, and strain $\gamma_{\theta z}$, γ_{zr} can also be neglected. Thus, the constitutive equations of (1a), (1b) can be simplified as:

$$\sigma_r = \frac{\varepsilon_r + \nu\varepsilon_\theta}{s_{11}^E(1-\nu^2)} - \frac{d_{31}}{s_{11}^E(1-\nu)}E_z, \tag{10}$$

$$\sigma_\theta = \frac{\nu\varepsilon_r + \varepsilon_\theta}{s_{11}^E(1-\nu^2)} - \frac{d_{31}}{s_{11}^E(1-\nu)}E_z, \tag{11}$$

$$\tau_{r\theta} = \frac{\gamma_{r\theta}}{2s_{11}^E(1+\nu)}, \tag{12}$$

$$D_z = d_{31}(\sigma_r + \sigma_\theta) + \varepsilon_{33}^T E_z, \tag{13}$$

where v is the Poisson's ratio. In the piezoelectric transformer, the radial extensional vibration can be generated by driving the input electrode with AC voltage. The radial extensional vibration is supposed to be axisymmetric, and the radial extensional displacement of the middle plane can be assumed to be:

$$u_r(r,t) = U(r)e^{i\omega t} \tag{14}$$

The stress-displacement relations for the extensional vibration are given by:

$$\sigma_r = \frac{1}{s_{11}^E(1-\nu^2)}\left[\frac{dU}{dr} + \nu\frac{U}{r}\right] - \frac{d_{31}E_z}{s_{11}^E(1-\nu)} \tag{15}$$

$$\sigma_\theta = \frac{1}{s_{11}^E(1-\nu^2)}\left[\nu\frac{dU}{dr}+\frac{U}{r}\right]-\frac{d_{31}E_z}{s_{11}^E(1-\nu)} \tag{16}$$

Substituting (15),(16) into (4a), the governing equation of extensional vibrations can be obtained:

$$\frac{d^2U}{dr^2}+\frac{1}{r}\frac{dU}{dr}-\frac{U}{r^2}-\rho\omega^2 s_{11}^E(1-\nu^2)U=0 \tag{17}$$

The general solution of (17) is:

$$U(r)=C_1J_1(\beta r)+C_2Y_1(\beta r) \tag{18}$$

where J_1 is the Bessel function of first kind and first order, Y_1 is the Bessel function of second kind and first order, and

$$\beta^2=\rho s_{11}^E(1-\nu^2)\omega^2 \tag{19}$$

Because the stress-free boundary conditions must be satisfied at $r=R_i$ and $r=R_o$.

$$\int_{-h/2}^{h/2}\sigma_r dz=0 \tag{20}$$

Thus, the constants A and B can be found in (21) and (22).

$$A=\frac{E_z d_{31}(1+\nu)R_o}{\Delta_1}[\beta R_i Y_0(\beta R_o)-\alpha(1-\nu)Y_1(\beta R_o)-\beta R_i Y_0(\beta R_i)+(1-\nu)Y_1(\beta R_i)], \tag{21}$$

$$B=\frac{E_z d_{31}(1+\nu)R_o}{\Delta_1}[\beta R_i J_0(\beta R_i)-(1-\nu)J_1(\beta R_i)-\beta R_i J_0(\beta R_o)+\alpha(1-\nu)J_1(\beta R_o)], \tag{22}$$

where $\alpha=R_i/R_o$, and Δ_1 is as follows.

$$\begin{aligned}\Delta_1=&[\beta R_i J_0(\beta R_i)-(1-\nu)J_1(\beta R_i)][\beta R_o Y_0(\beta R_o)-(1-\nu)Y_1(\beta R_o)]\\&-[\beta R_i Y_0(\beta R_i)-(1-\nu)Y_1(\beta R_i)][\beta R_o J_0(\beta R_o)-(1-\nu)J_1(\beta R_o)]\end{aligned} \tag{23}$$

2.2 Impedance of the Piezoelectric Transformer

In the output part of the PT, the output electrical current I_o for extensional vibrations can be developed as:

$$
\begin{aligned}
I_o &= \frac{\partial}{\partial t}\int_{S_o} D_z ds \\
&= j\omega\int_0^{2\pi}\int_{R_i}^{R_1}\left\{\frac{d_{31}(1+\nu)}{s_{11}^E(1-\nu^2)}\left[\frac{dU}{dr}+\frac{U}{r}\right]+\varepsilon_{33}^T(1-k_p^2)E_z\right\}rdrd\theta \\
&= j\omega\pi\varepsilon_{33}^T E_z\cdot\frac{(1+\nu)k_p^2\Delta_2-(k_p^2-1)(R_1^2-R_o^2)\Delta_1}{\Delta_1}
\end{aligned}
\tag{24}
$$

$$
\begin{aligned}
\Delta_2 &= \beta R_i R_o[Y_0(\beta R_o)-Y_o(\beta R_i)][R_1J_1(\beta R_1)-R_iJ_1(\beta R_i)] \\
&\quad+\beta R_i R_o[J_0(\beta R_i)-J_o(\beta R_o)][R_1Y_1(\beta R_1)-R_iY_1(\beta R_i)] \\
&\quad-(1-\nu)[R_iY_1(\beta R_o)-R_oY_1(\beta R_i)][R_1J_1(\beta R_1)-R_iJ_1(\beta R_i)] \\
&\quad-(1-\nu)[R_oJ_1(\beta R_i)-R_iJ_1(\beta R_o)][R_1Y_1(\beta R_1)-R_iY_1(\beta R_i)]
\end{aligned}
\tag{25}
$$

From (24), the resonant frequencies can be determined when the output current I_o approaches infinity. The characteristic equation of resonant frequencies for extensional vibrations is given by:

$$\Delta_1 = 0 \tag{26}$$

In the input part of the PT, the input electrical current I_i for extensional vibrations can be developed as:

$$
\begin{aligned}
I_i &= \frac{\partial}{\partial t}\int_{S_i} D_z ds \\
&= j\omega\int_0^{2\pi}\int_{R_2}^{R_o}\left\{\frac{d_{31}(1+\nu)}{s_{11}^E(1-\nu^2)}\left[\frac{dU}{dr}+\frac{U}{r}\right]+\varepsilon_{33}^T(1-k_p^2)E_z\right\}rdrd\theta \\
&= j\omega\pi\varepsilon_{33}^T E_z\cdot\frac{(1+\nu)k_p^2\Delta_3-(k_p^2-1)(R_o^2-R_2^2)\Delta_1}{\Delta_1}
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
\Delta_3 &= \beta R_i R_o[Y_0(\beta R_o)-Y_o(\beta R_i)][R_oJ_1(\beta R_o)-R_2J_1(\beta R_2)] \\
&\quad+\beta R_i R_o[J_0(\beta R_i)-J_o(\beta R_o)][R_oY_1(\beta R_o)-R_2Y_1(\beta R_2)] \\
&\quad-(1-\nu)[R_iY_1(\beta R_o)-R_oY_1(\beta R_i)][R_oJ_1(\beta R_o)-R_2J_1(\beta R_2)] \\
&\quad-(1-\nu)[R_oJ_1(\beta R_i)-R_iJ_1(\beta R_o)][R_oY_1(\beta R_o)-R_2Y_1(\beta R_2)]
\end{aligned}
\tag{28}
$$

From (27), the resonant frequencies can be determined when the input current I_i approaches infinity. The characteristic equation of resonant frequencies can be obtained, which is the same with (26). It is noted that the resonant frequencies of the PT can be obtained based on the measured impedance spectrum, and the same results will be obtained in spite of the measured electrodes are in the input part or in the output part. According to (19) and (26), the resonant frequencies for ring-type PT can be expressed as:

$$f = \frac{\beta}{2\pi\sqrt{\rho s_{11}^{E}(1-\nu^2)}} \tag{29}$$

3. Electromechanical Model

3.1 Electromechanical Model of the PT

The PT is not only a mechanical system but also electrical system. In this section, the electromechanical model for piezoelectrically coupled electromechanical systems will be derived. From Hagood's paper (Hagood, 1990), we have a generalized form of Hamilton's principle for a coupled electromechanical system:

$$\int_{t_1}^{t_2} \partial\left[T - U + W_1 + W_2\right]dt = 0 \tag{30}$$

where T is the kinetic energy, U is the potential energy of the system, W_1 is the applied electric energy in the driving portion, and W_2 is the applied electric energy in the receiving portion. T, U, W_1, W_2 can be written as

$$T = \frac{1}{2}\int_0^h \int_0^{2\pi} \int_{R_i}^{R_o} \rho \dot{u}_r^2(r,t) r dr d\theta dz\,, \tag{31}$$

$$U = \frac{1}{2}\int_0^h \int_0^{2\pi} \int_{R_i}^{R_o} [S^T T + E^T D] r dr d\theta dz\,, \tag{32}$$

$$\partial W_1 = -\partial\varphi_i \cdot q_i\,, \tag{33}$$

$$\partial W_2 = -\partial\varphi_o \cdot q_o\,, \tag{34}$$

where ρ is the density of the piezoelectric material. φ_i and q_i are the electric potential and the applied charge in the driving portion, respectively. φ_o and q_i are the electric potential and the applied charge in the receiving portion. By substituting Eqs.(31)-(34) into Eq.(30), the equations of motion for the PT can be written in Laplace transform as

$$(m_n s^2 + d_n s + k_n)X + A_o V_o = A_i V_i , \tag{35}$$

$$sA_i X + sC_i V_i = I_i , \tag{36}$$

$$sA_o X = sC_o V_o + I_o , \tag{37}$$

where V_i and I_i represent the input voltage and current in the driving portion, V_o and I_o represent the output voltage and current in the receiving port. The mass m_n , the stiffness k_n, input turn ratio A_i, output turn ratio A_o for the equivalent circuit of piezoelectric transformer can be obtained from the follows.

$$m_n = \int_0^h \int_0^{2\pi} \int_{R_i}^{R_o} \rho U^2(r) r dr d\theta dz , \tag{38}$$

$$k_n = \int_0^h \int_0^{2\pi} \int_{R_i}^{R_o} \left[c_{11}^E (\frac{\partial U}{\partial r})^2 + 2c_{12}^E \frac{\partial U}{\partial r}\frac{U}{r} + c_{22}^E \frac{U^2}{r^2} \right] r dr d\theta dz , \tag{39}$$

$$C_o = \int_0^h \int_0^{2\pi} \int_{R_i}^{R_1} \varepsilon_{33}^T (\frac{\partial \varphi}{\partial z})^2 r dr d\theta dz , \tag{40}$$

$$C_i = \int_0^h \int_0^{2\pi} \int_{R_2}^{R_o} \varepsilon_{33}^T (\frac{\partial \varphi}{\partial z})^2 r dr d\theta dz , \tag{41}$$

$$A_o = \int_0^h \int_0^{2\pi} \int_{R_i}^{R_1} e_{31} \frac{\partial \varphi}{\partial z} \cdot \left[\frac{\partial U}{\partial r} + \frac{U}{r} \right] r dr d\theta dz , \tag{42}$$

$$A_i = \int_0^h \int_0^{2\pi} \int_{R_2}^{R_o} e_{31} \frac{\partial \varphi}{\partial z} \cdot \left[\frac{\partial U}{\partial r} + \frac{U}{r} \right] r dr d\theta dz . \tag{43}$$

According to Eqs.(35)-(37), equivalent circuit model of the PT is shown in Fig.3. From the equivalent circuit model, we can see that Eq.(35) satisfy Kirchhoff's voltage law equation, which shows that the input voltage $A_i V_i$ is the sum of the output voltage $A_o V_o$ and the voltage difference $(m_n s^2 + d_n s + k_n)$X. Eq.(36) satisfy Kirchhoff's current law equation in the driving portion, which shows that the input current I_i is the sum of the current flowing through $(m_n s^2 + d_n s + k_n)$ and the current flowing through C_i. Eq.(37) satisfy Kirchhoff's current law equation in the receiving portion, which shows that the current flowing through $(m_n s^2 + d_n s + k_n)$ is the sum of the current flowing through C_o and the output current I_o.

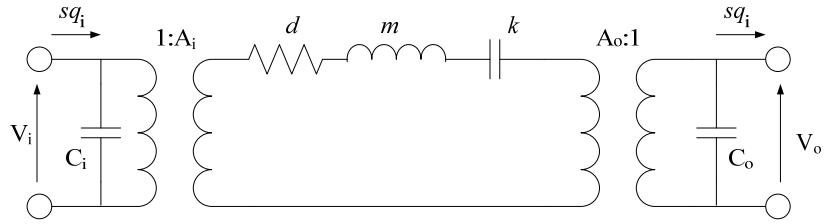


Fig. 3. Equivalent circuit of the piezoelectric transformer.

3.2 Characteristics of the PT

There is no output current in the receiving portion when the electrodes are open-circuited. Thus, voltage step-up ratio for the PT can be obtained based on Eqs.(35)(37). Substituting I_o =0 into Eq.(37) and eliminating $X(s)$ from Eqs. (35)(37) gives

$$\frac{V_o(s)}{V_i(s)} = \frac{A_o A_i}{(m_n s^2 + d_n s + k_n)C_o + A_o^2}. \tag{44}$$

When a load resistance R_L is connected between the electrodes in the receiving portion of the PT, Eq.(45) can be obtained by substituting $I_o=V_o/R_L$ into Eq.(37).

$$sA_o X = sC_o V_o + V_o / R_L \tag{45}$$

The voltage step-up ratio for the PT with a load resistance R_L in the receiving portion can be obtained based on Eqs.(35)(45) as the following.

$$\frac{V_o(s)}{V_i(s)} = \frac{sA_i A_o R_L}{(m_n s^2 + d_n s + k_n)(sC_o R_L + 1) + sA_o^2 R_L} \tag{46}$$

If the electrodes in the receiving portion of the PT is short-circuited, the voltage step-up ratio for the PT can be obtained as zero by substituting R_L=0 into Eq.(46). In addition, Eq.(46) shows that the higher the load resistance R_L, the higher the voltage step-up ratio. The maximum voltage step-up ratio can be obtained as Eq.(44) when the load resistance R_L approach infinite. On the other hand, the output power of the PT can be calculated by the power consumption of the load resistance R_L as the following:

$$P_o = |V_o|^2 / R_L \tag{47}$$

If the natural frequency is chosen as the operating frequency in the PT, then the voltage step-up ratio can be rewritten as

$$\frac{V_o}{V_i} = \frac{A_i A_o}{j\omega_n C_o d_n + d_n / R_L + A_o} \tag{48}$$

Therefore, the output power of the PT can be obtained by substituting Eq.(48) into Eq.(47).

$$P_o = |V_o|^2 / R_L = \frac{A_i^2 A_o^2 V_i^2}{R_L[(\omega_n C_o d_n)^2 + (d_n / R_L + A_o)^2]} \tag{49}$$

According to equivalent circuit of the PT shown as in Fig.3, the input power of the PT can be calculated by the sum of the power consumption of the damping d_n and that of the load resistance R_L. Eq.(37) shows that the current flowing through d_n is $(sC_oV_o+I_o)/A_o$, thus the input power of the PT can be obtained as

$$\begin{aligned} P_i &= \frac{d_n}{A_o^2}\left| j\omega_n C_o V_o + V_o / R_L \right|^2 + P_o \\ &= V_o^2[d_n(\omega_n C_o / A_o)^2 + d_n /(R_L A_o^2) + 1/R_L]. \end{aligned} \tag{50}$$

Therefore, the efficiency of the PT can be obtained as

$$\eta = \frac{P_o}{P_i} = \frac{1}{d_n R_L(\omega_n C_o / A_o)^2 + d_n /(A_o^2 R_L) + 1}. \tag{51}$$

The maximum efficiency can be calculated by the differential of Eq.(50). Thus, the maximum efficiency can be obtained when the optimal load resistance $R_{L,opt}$ is

$$R_{L,opt} = 1/(\omega_n C_o). \tag{52}$$

Substituting Eq.(52) into Eq.(51) gives the maximum efficiency.

$$\eta_{\max} = \frac{A_o^2}{2d_n\omega_n C_o + A_o^2} \tag{53}$$

It is note that the smaller the damping coefficient d_n, the higher the maximum efficiency.

4. Simulation and Experiment

4.1 Experimental Setup and the Impedance Measurements

To verify the electromechanical model, a ring-type PT with 16mm in outer diameter, 8mm in inner diameter, and 1mm in thickness was used. The PT is has silver electrodes on two

opposite surfaces and is poled along its thickness direction. One of the electrodes of the PT is split into two regions on the diameter of 11mm. The transformer structure was fabricated using the piezoelectric material APC840 by APC International, USA. The material properties provided by the supplier are listed in Table I. The displacement distributions of the mode shapes based on theoretical analysis for the PT are presented in Fig.4. Also, to easily realize the dynamic behavior of the PT, a finite element method analysis of the vibration of the PT is conducted. And the results of the extensional vibration modes of the PT are shown in Fig.5(a)(b)(c).

A HP 4194A Impedance Analyzer was used to measure the input impedance and output impedance, and results are shown in Fig.6. The input impedance was measured for the shorted electrodes in the receiving portion, and the output impedance was measured for the shorted electrodes in the driving portion. This transformer was designed to operate in the first vibration mode. For the input impedance of the PT, the first resonant frequency is 91.2 kHz, the first anti-resonant frequency is 94.05 kHz. For the output impedance of the PT, the first resonant frequency is 91.2 kHz, the first anti-resonant frequency is 93.6 kHz in the input impedance of the PT. It shows that nearly the same resonant frequency were obtained in spite of the impedance was measured from the driving portion or the receiving portion. The results are the same with theoretical analysis of Eqs. (24) and (27).

Basd on Eqs.(34)-(36), input impedance as a function of frequency at different load resistances are calculated and shown in Fig.7. And the experimental results are shown in Fig.8. In the input impedance of the PT with load resistance varied from short (R_L=0) to open (R_L=∞), it shows that the peak frequency is changed from 94.05 kHz to 97.85 kHz. The peak frequency is increased as the load resistance is increased. Also, there exists an optimal load resistance $R_{L,opt}$, which shows the maximum damping ratio in the input impedance when compared with the other different load resistances. We can also calculated the optimal load resistance $R_{L,opt}$ =2.6 kΩ from Eq.(52). It should be noted that efficiency of the PT approaches to the maximum efficiency when the load resistance R_L approaches the optimal load resistance $R_{L,opt}$.

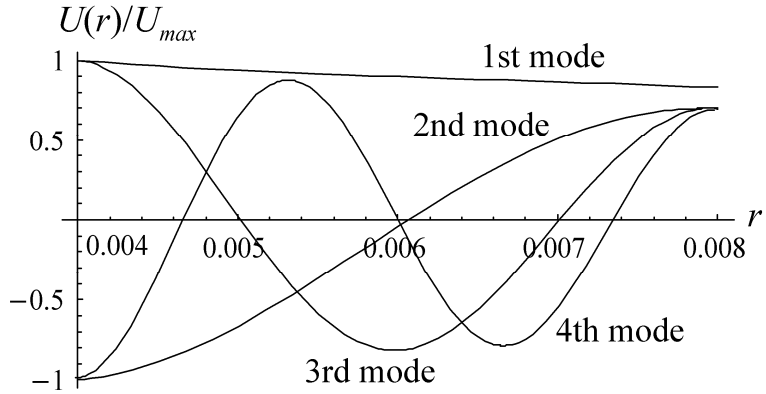
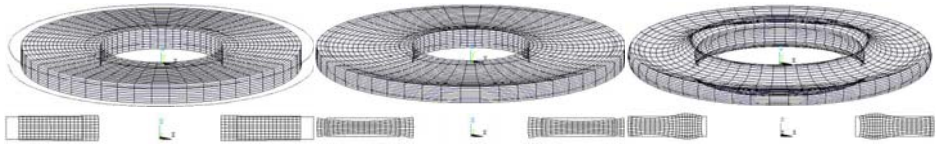


Fig. 4. Mode shapes of the piezoelectric transformer.



(a) 1st vibration mode (b) 2nd vibration mode (c) 3rd vibration mode

Fig. 5. Vibration modes of piezoelectric transformer.

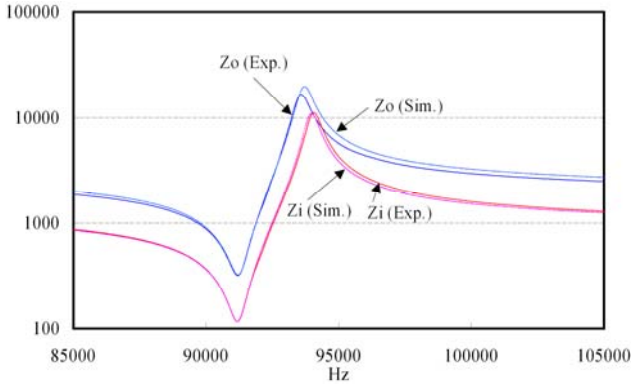


Fig. 6. Input and output impedance

4.2 Voltage Step-up Ratio, Output Power, and Efficiency

The experimental setup for the measurement of the voltage step-up ratio and output power of the PT is illustrated in Fig.9. A function generator (NF Corporation, WF1943) and a high frequency amplifier (NF Corporation, HSA4011) were used for driving power supply. The variation in electric characteristics with load resistance and driving frequency were measured with a multi-meter (Agilent 34401A). The voltage step-up ratios as a function of frequency at different load resistances were measured and compared with theoretical analysis, as shown in Fig.10. It shows that the experimental results are in a good agreement with the theoretical results, so the proposed electromechanical model for the PT was verified.

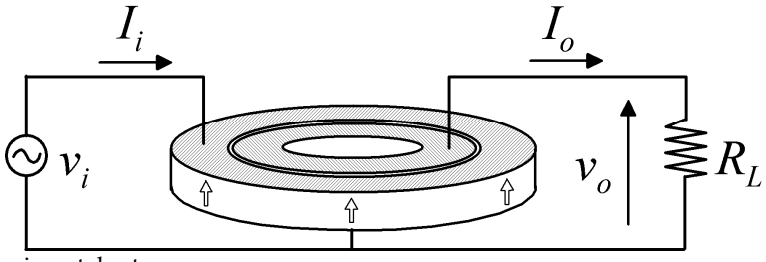


Fig. 7. Experimental setup

Piezoelectric coefficient d_{31}	-125×10^{-12} C/N
Coupling factor k_p	0.59
Mechanical quality factor Q_m	500
Dielectric constant $\varepsilon_{33}/\varepsilon_0$	1694
Density ρ	7600 g/cm^3
Young's modulus Y_{11}^E	8×10^{10} N/m^2

Table 1. Properties of piezoelectric material.

Input piezoelectric capacitance C_i	1.5nF
Output piezoelectric capacitance C_o	671.5pF
Input turn ratio A_i	0.1198
Output turn ratio A_o	0.07545
Effective mass m_1	4.773×10^{-4} kg
Effective damping d_1	1.868 N-s/m
Effective stiffness k_1	1.569×10^{8} N/m

Table 2. Parameters of the equivalent circuit

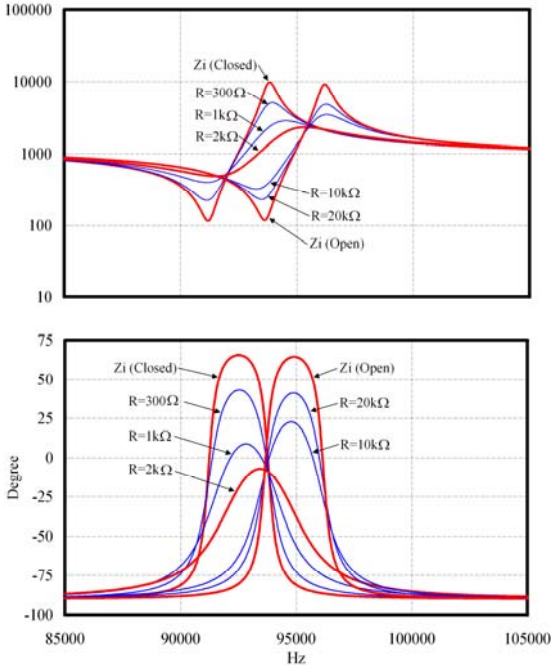


Fig. 8. Calculated input impedance

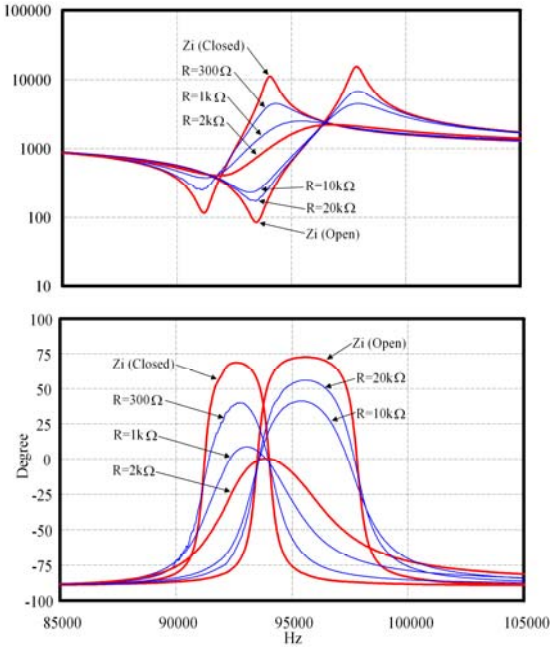


Fig. 9. Measured input impedance

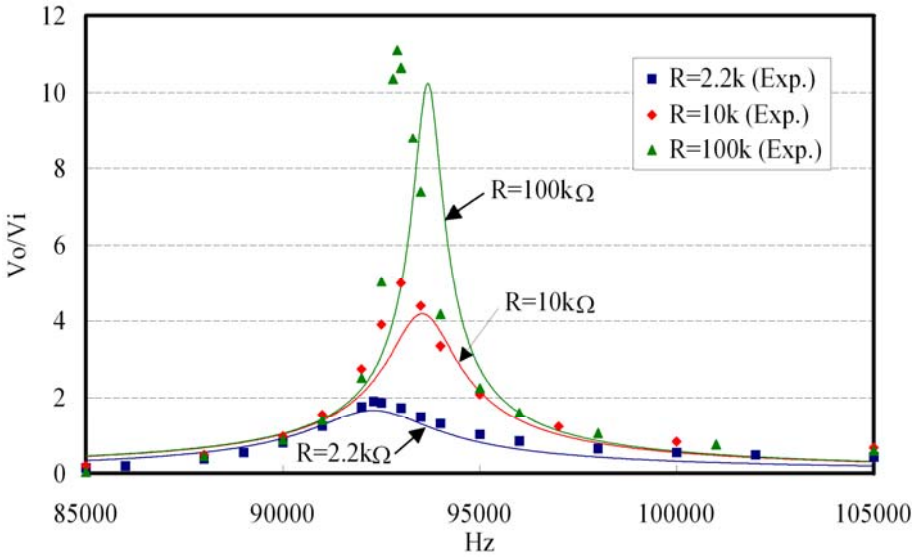


Fig. 10. Voltage step-up ratio

5. Conclusion

In this chapter, an electromechanical model for ring-type PT is presented. An equivalent circuit of the PT is shown based on the electromechanical model. Also, the voltage step-up ratio, input impedance, output impedance, and output power of the PT are calculated, and the optimal load resistance and the maximum efficiency for the PT have been obtained. In the last, some simulated results of the electromechanical model are compared with the experimental results for verification. The model presented here lays foundation for a general framework capable of serving a useful design tool for optimizing the configuration of the PT.

6. References

Bishop, R. P. (1998). Multi-Layer Piezoelectric Transformer, *US Patent No.5834882.*

Hagood, N. W. Chung, W. H. Flotow, A. V. (1990). Modeling of Piezoelectric Acatuator Dynamics for Active Structural Control. *Intell. Mater. Syst. And Struct.,* Vol.1, pp. 327-354, ISSN:1530-8138.

Hu, J. H. Li, H. L. Chan, H. L. W. Choy, C. L. (2001). A Ring-shaped Piezoelectric Transformer Operating in the third Sysmmetric Extenxional Vibration Mode. *Sensors and Actuators,* A., No.88, pp. 79-86, ISSN:0924-4247.

Laoratanakul, P. Carazo, A. V. Bouchilloux P. Uchino, K. (2002). Unipoled Disk-type Piezoelectric Transformers. *Jpn. J. Appl. Phys.,* Vol.41, No., pp. 1446-1450, ISSN:1347-4065.

Rosen, C. A. (1956). Ceramic Transformers and Filters, *Proceedings of Electronic Comp.*, pp. 205-211.

Sasaki, Y. Uehara, K. Inoue, T. (1993). Piezoelectric Ceramic Transformer Being Driven with Thickness Extensional Vibration, *US Patent No.5241236.*

2

Genetic Algorithm–Based Optimal PWM in High Power Synchronous Machines and Regulation of Observed Modulation Error

Alireza Rezazade
Shahid Beheshti University G.C.
Arash Sayyah
University of Illinois at Urbana-Champaign
Mitra Aflaki
SAIPA Automotive Industries Research and Development Center

1. Introduction

UNIQUE features of synchronous machines like constant-speed operation, producing substantial savings by supplying reactive power to counteract lagging power factor caused by inductive loads, low inrush currents, and capabilities of designing the torque characteristics to meet the requirements of the driven load, have made them the optimal choices for a multitude of industries. Economical utilization of these machines and also increasing their efficiencies are issues that should receive significant attention. At high power rating operation, where high switching efficiency in the drive circuits is of utmost importance, optimal PWM is the logical feeding scheme. That is, an optimal value for each switching instant in the PWM waveforms is determined so that the desired fundamental output is generated and the predefined objective function is optimized (Holtz , 1992). Application of optimal PWM decreases overheating in machine and results in diminution of torque pulsation. Overheating resulted from internal losses, is a major factor in rating of machine. Moreover, setting up an appropriate cooling method is a particularly serious issue, increasing in intricacy with machine size. Also, from the view point of torque pulsation, which is mainly affected by the presence of low-order harmonics, will tend to cause jitter in the machine speed. The speed jitter may be aggravated if the pulsing torque frequency is low, or if the system mechanical inertia is small. The pulsing torque frequency may be near the mechanical resonance of the drive system, and these results in severe shaft vibration, causing fatigue, wearing of gear teeth and unsatisfactory performance in the feedback control system.

Amongst various approaches for achieving optimal PWM, harmonic elimination method is predominant (Mohan et al., 2003), (Chiasson et al., 2004), (Sayyah et al., 2006), (Sun et al., 1996), (Enjeti et al., 1990). One of the disadvantages associated with this method originates from this fact that as the total energy of the PWM waveform is constant, elimination of low-order harmonics substantially boosts remaining ones. Since copper losses are fundamentally

determined by current harmonics, defining a performance index related to undesirable effects of the harmonics is of the essence in lieu of focusing on specific harmonics (Bose BK, 2002). Herein, the total harmonic current distortion (THCD) is the objective function for minimization of machine losses. The fundamental frequency is necessarily considered constant in this case, in order to define a sensible optimization problem (i.e. "Pulse width modulation for Holtz, J. 1996").

In this chapter, we have strove to propose an appropriate current harmonic model for high power synchronous motors by thorough inspecting the main structure of the machine (i.e. "The representation of Holtz, J 1995"), (Rezazade et al.,2006), (Fitzgerald et al., 1983), (Boldea & Nasar, 1992). Possessing asymmetrical structure in direct axis (d- axis) and quadrature axis (q-axis) makes a great difference in modelling of these motors relative to induction ones. The proposed model includes some internal parameters which are not part of machines characteristics. On the other hand, machines d and q axes inductances are designed so as to operate near saturation knee of magnetization curve. A slight change in operating point may result in large changes in these inductances. In addition, some factors like aging and temperature rise can influence the harmonic model parameters.

Based on gathered input and output data at a specific operating point, these internal parameters are determined using online identification methods (Åström & Wittenmark, 1994), (Ljung & Söderström, 1983). In light of the identified parameters, the problem has been redrafted as an optimization task, and optimal pulse patterns are sought through genetic algorithm (GA) (Goldberg, 1989), (Michalewicz, 1989), (Fogel, 1995), (Davis, 1991), (Bäck, 1996), (Deb, 2001), (Liu, 2002). Indeed, the complexity and nonlinearity of the proposed objective function increases the probability of trapping the conventional optimization methods in suboptimal solutions. The GA provided with salient features can effectively cope with shortcomings of the deterministic optimization methods, particularly when decision variables increase. The advantages of this optimization are so remarkable considering the total power of the system. Optimal PWM waveforms are accomplished up to 12 switches (per quarter period of PWM waveform), in which for more than this number of switching angles, space vector PWM (SVPWM) method, is preferred to optimal PWM approach. During real-time operation, the required fundamental amplitude is used for addressing the corresponding switching angles, which are stored in a read-only memory (ROM) and served as a look-up table for controlling the inverter.

Optimal PWM waveforms are determined for steady state conditions. Presence of step changes in trajectories of optimal pulse patterns results in severe over currents which in turn have detrimental effects on a high-performance drive system. Without losing the feed forward structure of PWM fed inverters, considerable efforts should have gone to mitigate the undesired transient conditions in load currents. The inherent complexity of synchronous machines transient behaviour can be appreciated by an accurate representation of significant circuits when transient conditions occur. Several studies have been done for fast current tracking control in induction motors (Holtz & Beyer, 1991), (Holtz & Beyer, 1994), (Holtz & Beyer, 1993), (Holtz & Beyer, 1995). In these studies, the total leakage inductance is used as current harmonic model for induction motors. As mentioned earlier, due to asymmetrical structure in d and q axes conditions in synchronous motors, derivation of an appropriate current harmonic model for dealing with transient conditions seems indispensable which is covered in this chapter. The effectiveness of the proposed method for fast tracking control has been corroborated by establishing an experimental setup, where a

field excited synchronous motor in the range of 80 kW drives an induction generator as the load. Rapid disappearance of transients is observed.

2. Optimal Synchronous PWM for Synchronous Motors

2.1 Machine Model

Electrical machines with rotating magnetic field are modelled based upon their applications and feeding scheme. Application of these machines in variable speed electrical drives has significantly increased where feed forward PWM generation has proven its effectiveness as a proper feeding scheme. Furthermore, some simplifications and assumptions are considered in modelling of these machines, namely space harmonics of the flux linkage distribution are neglected, linear magnetic due to operation in linear portion of magnetization curve prior to experiencing saturation knee is assumed, iron losses are neglected, slot harmonics and deep bar effects are not considered. In light of mentioned assumptions, the resultant model should have the capability of addressing all circumstances in different operating conditions (i.e. steady state and transient) including mutual effects of electrical drive system components, and be valid for instant changes in voltage and current waveforms. Such a model is attainable by Space Vector theory (i.e. "On the spatial propagation of Holtz, J 1996").

Synchronous machine model equations can be written as follows:

$$\boldsymbol{u}_S^R = \boldsymbol{r}_S \boldsymbol{i}_R^S + j\omega \boldsymbol{\Psi}_S^R + \frac{d\boldsymbol{\Psi}_S^R}{d\tau}, \tag{1}$$

$$0 = \boldsymbol{R}_D \boldsymbol{i}_D + \frac{d\boldsymbol{\Psi}_D}{d\tau}, \tag{2}$$

$$\boldsymbol{\Psi}_S^R = \boldsymbol{l}_S \boldsymbol{i}_R^S + \boldsymbol{\Psi}_m^R, \tag{3}$$

$$\boldsymbol{\Psi}_m^R = \boldsymbol{l}_m \left(\boldsymbol{i}_D + \boldsymbol{i}_F \right), \tag{4}$$

$$\boldsymbol{\Psi}_D = \boldsymbol{l}_D \boldsymbol{i}_D + \boldsymbol{l}_m \left(\boldsymbol{i}_S + \boldsymbol{i}_F \right), \tag{5}$$

where:

$$\boldsymbol{l}_S = \boldsymbol{l}_{lS} + \boldsymbol{l}_m = \begin{pmatrix} l_d & 0 \\ 0 & l_q \end{pmatrix}, \qquad \boldsymbol{i}_F = \begin{pmatrix} 1 \\ 0 \end{pmatrix} i_F, \tag{6}$$

$$\boldsymbol{l}_m = \begin{pmatrix} l_{md} & 0 \\ 0 & l_{mq} \end{pmatrix}, \qquad \boldsymbol{l}_D = \begin{pmatrix} l_{Dd} & 0 \\ 0 & l_{Dq} \end{pmatrix} \tag{7}$$

where l_d and l_q are inductances of the motor in *d* and *q* axes; i_D is damper winding current; $\boldsymbol{u}_S^R$ and $\boldsymbol{i}_S^R$ are stator voltage and current space vectors, respectively; l_D is the damper

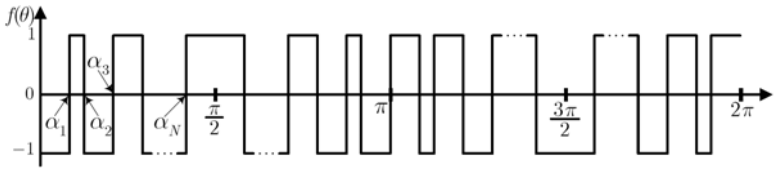


Fig. 2. One Line-to-Neutral PWM structure.

Owing to the symmetries in PWM waveform of Figure 2, only the odd harmonics exist. As such, $f(\theta)$ can be written with the Fourier series as

$$f(\theta) = \sum_{k=1,3,5,\ldots} u_k \sin(k\theta) \tag{8}$$

with

$$\begin{aligned} u_k &= \frac{4}{\pi}\int_0^{\frac{\pi}{2}} f(\theta)\sin(k\theta) \\ &= \frac{4}{k\pi}\left(-1+2\sum_{i=1}^{N}(-1)^{i-1}\cos(k\alpha_i)\right). \end{aligned} \tag{9}$$

2.3 THCD Formulation

The total harmonic current distortion is defined as follows:

$$\sigma_i = \sqrt{\frac{1}{T}\int_T \left[\boldsymbol{i}_S(t) - \boldsymbol{i}_{S1}(t)\right]^2 dt}, \tag{10}$$

where i_{S1} is the fundamental component of stator current.

Assuming that the steady state operation of machine makes a constant exciting current, the dampers current in the system can be neglected. Therefore, the equation of the machine model in rotor coordinates can be written as:

$$\boldsymbol{u}_S^R = \boldsymbol{r}_S \boldsymbol{i}_S^R + j\omega \boldsymbol{l}_S \boldsymbol{i}_S^R + j\omega \boldsymbol{l}_m \boldsymbol{i}_F + \boldsymbol{l}_S \frac{d\boldsymbol{i}_S^R}{d\tau} \tag{11}$$

With the Park transformation, the equation of the machine model in stator coordinates (the so called α-β coordinates) can be written as:

$$\begin{aligned} \boldsymbol{u}_{\alpha\beta} = R_S \boldsymbol{i}_{\alpha\beta} &+ \omega\left(l_d - l_q\right)\begin{pmatrix} -\sin 2\theta & \cos 2\theta \\ \cos 2\theta & \sin 2\theta \end{pmatrix}\boldsymbol{i}_{\alpha\beta} + \frac{l_d + l_q}{2}\frac{d\boldsymbol{i}_{\alpha\beta}}{d\tau} \\ &+ \frac{l_d - l_q}{2}\begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}\frac{d\boldsymbol{i}_{\alpha\beta}}{d\tau} + \omega l_{md}\begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} i_F, \end{aligned} \tag{12}$$

where θ is the rotor angle. Neglecting the ohmic terms in (12), we have:

$$\boldsymbol{u}_{\alpha\beta} = \frac{d}{d\tau}\left\{\boldsymbol{l}_S(\theta)\boldsymbol{i}_{\alpha\beta}\right\} + l_{md}\frac{d}{d\tau}\left\{\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F\right\}, \tag{13}$$

where:

$$\boldsymbol{l}_S(\theta) = \frac{l_d + l_q}{2}\boldsymbol{I}_2 + \frac{l_d - l_q}{2}\begin{pmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{pmatrix}. \tag{14}$$

$\boldsymbol{I}_2$ is the 2×2 identity matrix. Hence:

$$\boldsymbol{i}_{\alpha\beta} = \boldsymbol{l}_S^{-1}(\theta).\left\{\int \boldsymbol{u}_{\alpha\beta}d\tau - l_{md}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F\right\} =$$

$$\begin{pmatrix}\frac{l_d + l_q}{2l_d l_q} - \frac{l_d - l_q}{2l_d l_q}\cos 2\theta & -\frac{l_d - l_q}{2l_d l_q}\sin 2\theta\\ -\frac{l_d - l_q}{2l_d l_q}\sin 2\theta & \frac{l_d + l_q}{2l_d l_q} + \frac{l_d - l_q}{2l_d l_q}\cos 2\theta\end{pmatrix}.\left\{\int \boldsymbol{u}_{\alpha\beta}d\tau - l_{md}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F\right\} \tag{15}$$

$$= \left\{\frac{l_d + l_q}{2l_d l_q}I_2 - \frac{l_d - l_q}{2l_d l_q}\begin{pmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{pmatrix}\right\}.\left\{\int \boldsymbol{u}_{\alpha\beta}d\tau - l_{md}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F\right\}$$

With further simplification, we have $i_{\alpha\beta}$ can be written as:

$$\boldsymbol{i}_{\alpha\beta} = \frac{l_d + l_q}{2l_d l_q}\int \boldsymbol{u}_{\alpha\beta}d\tau \underbrace{-l_{md}\frac{l_d + l_q}{2l_d l_q}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F + l_{md}\frac{l_d - l_q}{2l_d l_q}\begin{pmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{pmatrix}.\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}}_{J_1} \tag{16}$$

$$\underbrace{-\frac{l_d - l_q}{2l_d l_q}\begin{pmatrix}\cos 2\theta & \sin 2\theta\\ \sin 2\theta & -\cos 2\theta\end{pmatrix}\int \boldsymbol{u}_{\alpha\beta}d\tau}_{J_2}.$$

Using the trigonometric identities, $\cos(\theta_1 - \theta_2) = \cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2$ and $\sin(\theta_1 - \theta_2) = \sin\theta_1\cos\theta_2 - \cos\theta_1\sin\theta_2$ the term J_1 in Equation (16) can be simplified as:

$$J_1 = -l_{md}\frac{l_d + l_q}{2l_d l_q}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F + l_{md}\frac{l_d - l_q}{2l_d l_q}\begin{pmatrix}\cos 2\theta.\cos\theta + \sin 2\theta.\sin\theta\\ \sin 2\theta.\cos\theta - \cos 2\theta.\cos\theta\end{pmatrix} i_F$$
$$= -l_{md}\frac{l_d + l_q}{2l_d l_q}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F + l_{md}\frac{l_d - l_q}{2l_d l_q}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F \tag{17}$$
$$= \frac{l_{md}}{l_d}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} i_F .$$

On the other hand, writing the phase voltages in Fourier series: $u_A = \sum_{s\in S_3} u_{2s+1}\sin\big((2s+1)\theta\big)$, $u_B = \sum_{s\in S_3} u_{2s+1}\sin\left((2s+1)\left(\theta - \frac{2\pi}{3}\right)\right)$ and $u_C = \sum_{s\in S_3} u_{2s+1}\sin\left((2s+1)\left(\theta - \frac{4\pi}{3}\right)\right)$; then using 3-phase to 2-phase transformation, we have:

$$\begin{pmatrix}u_\alpha\\ u_\beta\end{pmatrix} = \begin{pmatrix}u_A\\ \frac{1}{\sqrt{3}}(u_B - u_C)\end{pmatrix} = \begin{pmatrix}\sum_{s\in S_3} u_s \sin(s\theta)\\ \sum_{s\in S_3} u_s \sin\left(s\left(\theta - \frac{2\pi}{3}\right) + \varphi_s\right)\end{pmatrix} \tag{18}$$

in which:

$$\varphi_s = \begin{cases}\dfrac{\pi}{6} & for \quad s = 1,7,13,...\\ -\dfrac{\pi}{6} & for \quad s = 5,11,17,...\end{cases} \tag{19}$$

As such, we have:

$$\boldsymbol{u}_{\alpha\beta} = \begin{pmatrix}\sum_{l=0}^{\infty}\left[u_{6l+1}\sin\big((6l+1)\theta\big) + u_{6l+5}\sin\big((6l+5)\theta\big)\right]\\ \sum_{l=0}^{\infty}\left[u_{6l+1}\sin\left((6l+1)\left(\theta - \frac{2\pi}{3}\right) + \frac{\pi}{6}\right) + u_{6l+5}\sin\left((6l+5)\left(\theta - \frac{2\pi}{3}\right) - \frac{\pi}{6}\right)\right]\end{pmatrix}. \tag{20}$$

Integration of $u_{\alpha\beta}$ yields:

$$\int \boldsymbol{u}_{\alpha\beta} d\tau = -\frac{1}{\omega}\begin{pmatrix} \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l+1)\theta\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta\right)\right] \\ \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l+1)\theta-4\pi l-\frac{\pi}{2}\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta-4\pi l-\frac{3\pi}{2}\right)\right] \end{pmatrix}.$$
$$= -\frac{1}{\omega}\begin{pmatrix} \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l+1)\theta\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta\right)\right] \\ \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\sin\left((6l+1)\theta\right)-\frac{u_{6l+5}}{6l+5}\sin\left((6l+5)\theta\right)\right] \end{pmatrix}. \quad (21)$$

By substitution of $\int u_{\alpha\beta} d\tau$ in Equation (16), the term J_2 can be written as:

$$J_2 = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} . \int \boldsymbol{u}_{\alpha\beta} d\tau$$
$$= -\frac{1}{\omega}.\begin{pmatrix} \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\left[\cos\left((6l+1)\theta\right).\cos(2\theta)+\sin\left((6l+1)\theta\right).\sin(2\theta)\right]\right] \\ \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\left[\cos\left((6l+1)\theta\right).\sin(2\theta)-\sin\left((6l+1)\theta\right).\cos(2\theta)\right]\right] \end{pmatrix}.+$$
$$\begin{pmatrix} \sum_{l=0}^{\infty}\left[\frac{u_{6l+5}}{6l+5}\left[\cos\left((6l+5)\theta\right).\cos(2\theta)-\sin\left((6l+5)\theta\right).\sin(2\theta)\right]\right] \\ \sum_{l=0}^{\infty}\left[\frac{u_{6l+5}}{6l+5}\left[\cos\left((6l+5)\theta\right).\sin(2\theta)+\sin\left((6l+5)\theta\right).\cos(2\theta)\right]\right] \end{pmatrix} \quad (22)$$
$$= -\frac{1}{\omega}\begin{pmatrix} \sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l-1)\theta\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+7)\theta\right)\right] \\ \sum_{l=0}^{\infty}\left[-\frac{u_{6l+1}}{6l+1}\sin\left((6l-1)\theta\right)+\frac{u_{6l+5}}{6l+5}\sin\left((6l+7)\theta\right)\right] \end{pmatrix}.$$

Considering the derived results, we can rewrite $i_A = i_\alpha$ as:

$$i_A = -\frac{l_d + l_q}{2l_d l_q \omega}\sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l+1)\theta\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta\right)\right]$$
$$+\frac{l_d - l_q}{2l_d l_q \omega}\sum_{l=0}^{\infty}\left[\frac{u_{6l+1}}{6l+1}\cos\left((6l-1)\theta\right)+\frac{u_{6l+5}}{6l+5}\cos\left((6l+7)\theta\right)\right] \quad (23)$$
$$-\frac{l_{md}}{l_d} i_F \cos\theta.$$

Using the appropriate dummy variables $l = l' + 1$ and $l = l'' - 1$, we have:

$$\begin{aligned} i_A &= -\frac{l_d + l_q}{2l_d l_q \omega}\left\{\sum_{l=-1}^{\infty}\frac{u_{6l+1}}{6l+1}\cos\left((6l+5)\theta\right) + \sum_{l=1}^{\infty}\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta\right)\right\} \\ &+ \frac{l_d - l_q}{2l_d l_q \omega}\left\{\sum_{l'=0}^{\infty}\frac{u_{6l'+7}}{6l'+7}\cos\left((6l'+1)\theta\right) + \sum_{l''=0}^{\infty}\frac{u_{6l''-1}}{6l''-1}\cos\left((6l''+1)\theta\right)\right\} - \frac{l_{md}}{l_d} i_F \cos\theta \\ &= -\frac{l_d + l_q}{2l_d l_q \omega}\left\{\sum_{l=0}^{\infty}\frac{u_{6l+1}}{6l+1}\cos\left((6l+1)\theta\right) + \sum_{l=0}^{\infty}\frac{u_{6l+5}}{6l+5}\cos\left((6l+5)\theta\right)\right\} \\ &+ \frac{l_d - l_q}{2l_d l_q \omega}\left\{\sum_{l=0}^{\infty}\frac{u_{6l+7}}{6l+7}\cos\left((6l+5)\theta\right) + \sum_{l=0}^{\infty}\frac{u_{6l-1}}{6l-1}\cos\left((6l+1)\theta\right) + u_1\cos\theta\right\} - \frac{l_{md}}{l_d} i_F \cos\theta \end{aligned} \tag{24}$$

Thus, we have i_A as:

$$\begin{aligned} i_A = &-\frac{1}{2l_d l_q \omega}\left\{\sum_{l=0}^{\infty}\left[\left(l_d + l_q\right)\frac{u_{6l+1}}{6l+1} - \left(l_d - l_q\right)\frac{u_{6l-1}}{6l-1}\right].\cos\left((6l+1)\theta\right)\right. \\ &\left. -\left(l_d - l_q\right)u_1\cos(\theta) + \sum_{l=0}^{\infty}\left[\left(l_d + l_q\right)\frac{u_{6l+5}}{6l+5} - \left(l_d - l_q\right)\frac{u_{6l+7}}{6l+7}\right].\cos\left((6l+1)\theta\right)\right\} \\ &-\frac{l_{md}}{l_d} i_F \cos\theta. \end{aligned} \tag{25}$$

Removing the fundamental components from Equation (25), the current harmonic is introduced as:

$$\begin{aligned} i_{Ah} = &-\frac{1}{2l_d l_q \omega}.\left\{\sum_{l=1}^{\infty}\left[\left\{\left(l_d + l_q\right)\frac{u_{6l+1}}{6l+1} - \left(l_d - l_q\right)\frac{u_{6l-1}}{6l-1}\right\}.\cos\left((6l+1)\theta\right)\right] + \right. \\ &\left.\sum_{l=0}^{\infty}\left[\left\{\left(l_d + l_q\right)\frac{u_{6l+5}}{6l+5} - \left(l_d - l_q\right)\frac{u_{6l+7}}{6l+7}\right\}.\cos\left((6l+5)\theta\right)\right]\right\} \\ = &-\frac{1}{2l_d l_q \omega}.\left\{\sum_{l=0}^{\infty}\left[\left\{\left(l_d + l_q\right)\frac{u_{6l+7}}{6l+7} - \left(l_d - l_q\right)\frac{u_{6l+5}}{6l+5}\right\}.\cos\left((6l+7)\theta\right)\right] + \right. \\ &\left.\sum_{l=0}^{\infty}\left[\left\{\left(l_d + l_q\right)\frac{u_{6l+5}}{6l+5} - \left(l_d - l_q\right)\frac{u_{6l+7}}{6l+7}\right\}.\cos\left((6l+5)\theta\right)\right]\right\}. \end{aligned} \tag{26}$$

On the other hand, σ_l^2 can be written as:

$$\sigma_l^2 = \left(\left(l_d + l_q\right)\frac{u_{6l+7}}{6l+7} - \left(l_d - l_q\right)_q \frac{u_{6l+5}}{6l+5} \right)^2 + \left(\left(l_d + l_q\right)\frac{u_{6l+5}}{6l+5} - \left(l_d - l_q\right)\frac{u_{6l+7}}{6l+7} \right)^2$$
$$= 2\left(l_d{}^2 + l_q{}^2\right)\left(\frac{u_{6l+7}}{6l+7}\right)^2 + 2\left(l_d{}^2 + l_q{}^2\right)\left(\frac{u_{6l+5}}{6l+5}\right)^2 - 4\left(l_d{}^2 - l_q{}^2\right)\frac{u_{6l+5}u_{6l+7}}{(6l+5)(6l+7)}. \quad (27)$$

With normalization of σ_l^2; i.e. $\tilde{\sigma}_l^2 = \frac{\sigma_l^2}{l_d^2 + l_q^2}$ and also the definition of the total harmonic current distortion as $\sigma_i^2 = \sum_{l=0}^{\infty} \tilde{\sigma}_l^2$, it can be simplified as:

$$\sigma_i^2 = \sum_{l=0}^{\infty} \left\{ \left(\frac{u_{6l+5}}{6l+5}\right)^2 + \left(\frac{u_{6l+7}}{6l+7}\right)^2 - 2\frac{l_d^2 - l_q^2}{l_d^2 + l_q^2}\left(\frac{u_{6l+5}}{6l+5}\right).\left(\frac{u_{6l+7}}{6l+7}\right) \right\} \quad (28)$$

Considering the set $S_3 = \{5,7,11,13,...\}$ and with more simplification, σ_i in high-power synchronous machines can be explicitly expressed as:

$$\sigma_i = \sqrt{\sum_{k \in S_3}\left(\frac{u_k}{k}\right)^2 - 2\frac{l_d^2 - l_q^2}{l_d^2 + l_q^2}\sum_{l=1}^{\infty}\left(\frac{u_{6l-1}}{6l-1}\right).\left(\frac{u_{6l+1}}{6l+1}\right)}. \quad (29)$$

As mentioned earlier, THCD in high-power synchronous machines depends on l_d and l_q, the inductances of *d* and *q* axes, respectively. Needless to say, switching angles: $\alpha_1, \alpha_2, ..., \alpha_N$ determine the voltage harmonics in Equation (29). Hence, the optimization problem consists of identification of the l_q / l_d for the under test synchronous machine; determination of these switching angles as decision variables so that the σ_i is minimized. In addition, throughout the optimization procedure, it is desired to maintain the fundamental output voltage at a constant level: $u_1 = M$. *M*, the so-called the modulation index may be assumed to have any value between 0 and $4/\pi$. It can be shown that α_N is dependent on modulation index and the rest of *N-1* switching angles. As such, one decision variable can be eliminated explicitly. More clearly:

Minimize $$\sigma_i^2 = \sum_{k \in S_3}\left(\frac{u_k}{k}\right)^2 - 2\frac{1 - \left(l_q/l_d\right)^2}{1 + \left(l_q/l_d\right)^2}\sum_{l=1}^{\infty}\left(\frac{u_{6l-1}}{6l-1}\right).\left(\frac{u_{6l+1}}{6l+1}\right) \quad (30)$$

Subject to

$$0 < \alpha_1 < \alpha_2 < ... < \alpha_{N-1} < \frac{\pi}{2} \qquad \text{and}$$

$$\left\{\frac{(\quad)}{\qquad}\qquad(\quad)\qquad(\quad)\;\frac{}{\qquad}\qquad\right\}$$

The transducer was vertically contacted with a transparent object (an acrylic resin) with a contact load. The vibrometer laser beam was irradiated through the resin, as shown in Fig. 3. Through the measurement, operating frequency was swept with constant amplitude of driving voltage. Frequency responses of the vibration amplitude with the change of contact load are shown in Fig. 4. Admittance phases are also shown in the figure. Local peaks of the amplitude mean resonance frequencies. It can be seen that resonance frequency shifts to the higher frequency region according to the increase of the contact load. Admittance phase changes dramatically around the resonance and meets a certain value (around 0 [deg]: same as the previous result) at the resonance frequency.

3. Resonance Frequency Tracing System

3.1 Overview

As described in the previous section, resonance frequency of the Langevin type ultrasonic transducer changes according to the various reasons. To keep strong vibration, resonance frequency should be traced during the operation of the transducer. In this research, tracing system based on admittance phase measurement is proposed. A microcomputer was applied for the measurement. Tracing algorithm was also embedded in the computer. Other intelligent functions such as communication with other devices can be installed to the computer. Therefore, this system has extensibility according to functions of the computer. Overview of the fabricated resonance frequency tracing system is shown in Fig. 5. The system consists of a computer unit, an amplifier, voltage/current detecting circuit and a wave forming circuit. The computer unit includes a microcomputer (SH-7045F), a DDS and a COM port. The computer is connected to a LCD, a PS/2 keyboard and an EEPROM to execute intelligent functions.

3.2 Oscillating unit

To oscillate driving voltage, sinusoidal wave, the DDS is used. The synthesizer outputs digital wave amplitude data directly at a certain interval, which is much shorter than the cycle of the sinusoidal wave. The digital data is converted to analog signal by an AD converter inside. The frequency of the wave is decided by a parameter stored in the synthesizer. The parameter can be modified by an external device through serial

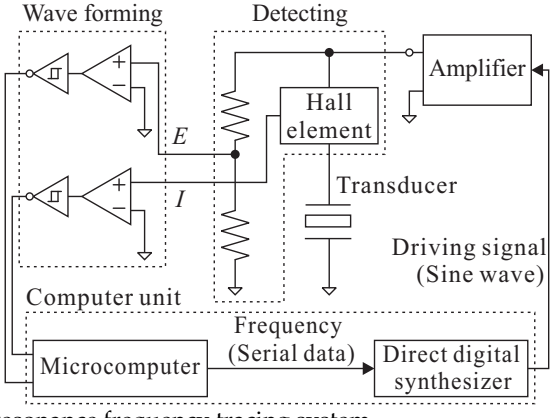


Fig. 5. Overview of resonance frequency tracing system.

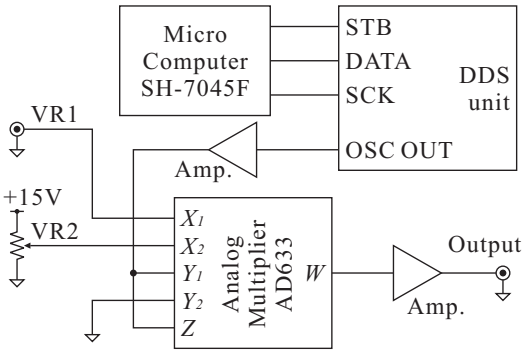


Fig. 6. A direct digital synthesizer and volume control.

communication. In this system, the synthesizer is connected to the microcomputer through three wires, as shown in Fig. 6. The DDS unit includes the AD converter and a LPF. Voltage of the generated sinusoidal wave is amplified and arranged by an analog multiplier (AD633). To control the voltage, the unit has an external DC input VR1 and a volume VR2. Multiplying result W is described as

$$W = \frac{(X_1 - X_2)}{10}(Y_1 - Y_2) + Z \,. \tag{1}$$

The result is amplified. Arranging the DC voltage of VR1 and VR2, the amplitude of the oscillated sinusoidal wave can be controlled continuously.

3.3 Detecting unit

To measure phase difference between applied voltage and current, amplified driving voltage is supplied to the ultrasonic transducer through a detecting unit illustrated in Fig. 7. The applied voltage is divided by a variable resistance, filtered and transformed in to rectangular wave by a comparator. The comparative result is transformed into TTL level pulses. The current flowing to the transducer is detected by a hall element. Output signal from the element is filtered and transformed in the same manner as the voltage detecting. Phase difference of these pulse trains is counted by the microcomputer. To monitor amplitude of the voltage and the current, half-wave rectification circuits and smoothing circuits are installed in the unit. AD converters of the microcomputer sample voltages of the output signals.

3.4 Control unit

A control unit comprises the microcomputer, a keyboard, a LCD, a COM port and an EEPROM, as described in Fig. 8. Commands to control the computer can be typed using the keyboard. Status of the system is displayed on the LCD. A target program executed in the computer is written through the COM port. Control parameters can be stored in the EEPROM. The system has such intelligent functions. The pulses transformed in the detecting unit are input to a multifunction timer unit (MTU) of the microcomputer. T_c (the

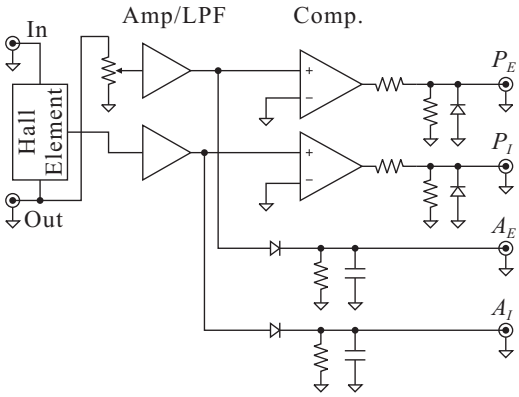


Fig. 7. Voltage/current detecting unit.

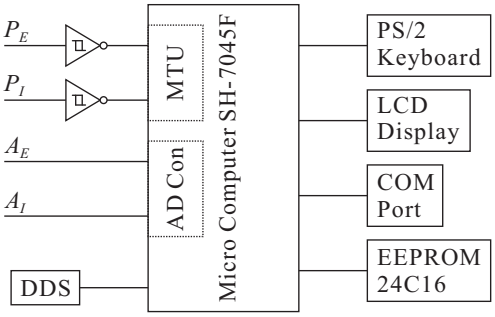


Fig. 8. Control unit with a microcomputer.

time between rising edges of P_E) and T_I (the time between rising edge of P_E and trailing edge of P_I) are measured by the unit, as shown in Fig. 10. The phase difference is calculated from

$$\varphi = 180 \times \frac{T_C - 2T_I}{T_C}. \tag{2}$$

This value is measured as average in averaging factor N_a cycles of pulse signal P_E. Thus, the operating frequency is updated every N_a cycles of the driving signals. The updated operating frequency f_{n+1} is given by

$$f_{n+1} = f_n - K_p(\phi_r - \phi), \tag{3}$$

where f_n is the operating frequency before update, ϕ_r is the admittance phase at resonance, ϕ is the calculated admittance phase from eq. (2) (at a frequency of f_n), K_p is a proportional feedback gain. To stabilize the tracing, K_p should be selected as following inequality is satisfied.

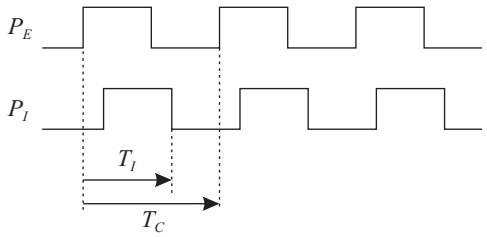


Fig. 9. Measurement of cycle and phase diference.

$$K_p < \frac{2}{S}, \tag{4}$$

where S is the slope of the admittance phase vs. frequency curve at resonanse frequency. The updated frequency is transmitted to the DDS. Repeating this routine, the operating frequency can approach resonance frequency of transducer.

4. Application for Ultrasonic Dental Scaler

4.1 Ultrasonic dental scaler

Ultrasonic dental scaler is an equipment to remove dental calculi from teeth. the scaler consists of a hand piece as shown in Fig. 10 and a driver circuit to excite vibration. A Langevin type ultrasonic transducer is mounted in the hand piece. the structure of the transducer is shown in Fig. 11. Piezoelectric elements are clamped by a tail block and a hone block. A tip is attached on the top of the horn. The blocks and the tip are made of stainless steel. The transducer vibrates longitudinally at first-order resonance frequency. One vibration node is located in the middle. To support the node, the transducer is bound by a silicon rubber.

To carry out the following experiments, a sample scaler was fabricated.Frequency response of the electric charactorristics of the transducer was observed with no mechanical load and input voltage of 20 $V_{p\text{-}p}$. The result is shown in Fig. 12. From this result, the resonance

Fig. 10. Example of ultrasonic dental scalar hand piece.

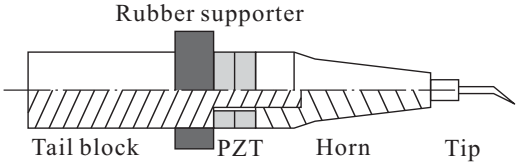


Fig. 11. Structure of transducer for ultrasonic dental scalar.

frequency was 31.93 kHz, admittance phase coincided with 0 at the resonance frequency, electorical Q factor was 330 and the admittance phase response had a slope of -1 [deg/Hz] in the neighborhood of the resonanse frequency.

4.2 Tracing test

Dental calculi are removed by contact with the tip. The applied voltage is adjusted according to condition of the calculi. Temparature rises due to high applied voltage. Therefore, during the operation, the resonance frequency of the transducer is shifted with the changes of contact condition, temperature and amplitude of applied voltage. The oscillating frequency was fixed in the conventional driving circuit. Consequently, vibration amplitude was reduced due to the shift. The resonance frequency tracing system was apllied to the ultrasonic dental scaler.

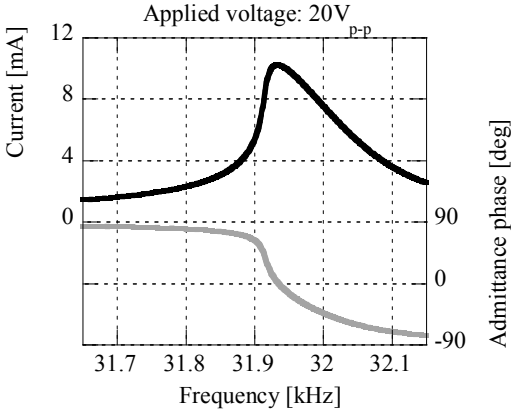


Fig. 12. Electric frequency response of the transducer for ultrasonic dental scalar.

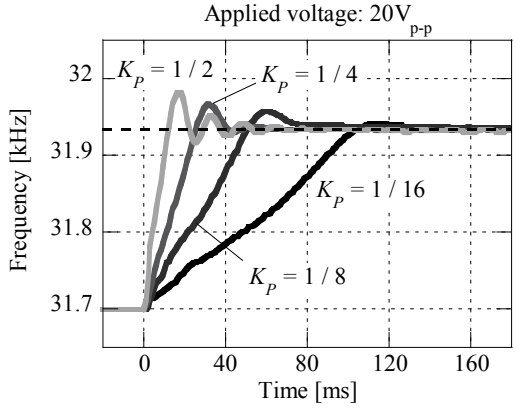


Fig. 13. Step responses of the resonance frequency tracing system with the transducer for ultrasonic dental scaler.

The transducer was driven by the tracing system, where averaging factor N_a was set to 8. To evaluate the system characteristic, step responses of the oscillating frequency were observed in the same condition as the measurement of the electric frequency response. In this measurement, initial operating frequency was 31.70 kHz. the frequency was differed from the resonance frequency (31.93 kHz). At a time of 0 sec, the tracing was started. Namely, the terget frecuency was changed, as a step input, to 31.93 kHz from 31.7 kHz. The transient response of the oscillating frequency was observed. The oscillating frequency was measured by a modulation domain analyzer in real time. Figure 13 shows the measurement results of the responces. With each K_p, the oscillating frequency in steady state was 31.93 kHz. the frequency coincided with the resonance frequency. A settling time was 40 ms with K_p of 1/4. The settling time was evaluated from the time settled within ±2 % of steady state value. The response speed is enough for the application to the dental scaler. Contact load does not change faster than the response speed since the scaler is wielded by human. The temperature and the amplitude of applied voltage also do not change so fast in normal operation.

4.3 Dental diagnosis

When the transducer is contacted with an object, the natural frequency of the transdcer is shifted. A value of the shift depends on stiffness and damping factor of the object (Nishimura et. al, 1994). The contact model can be discribed as shown in Fig. 14. In this model, the natural angular frequency of the transducer with contact is presented as

$$\omega = \sqrt{\frac{1}{m}\left(\frac{AE}{l} + K_C\right) - 2\left(\frac{C_C}{2m}\right)^2}\ , \tag{5}$$

where m is the equivalent mass of the transducer, A is the section area of the transducer, E is the elastic modulus of the material of the transducer, l is the half length of the transducer, K_c is the stiffness of the object and C_c is the damping coefficient of the object. Equation (5) indicates that the combination factor of the damping factor and the stiffness can be estimated from the natural frequency shift. The shift can be observed by the proposed resonance frequency tracing system in real time. If the correlation between the combination factor and the material properties is known, the damping factor or the stiffness of unknown material can be predicted. For known materials, the local stiffness on the contacting point can be estimated if the damping factor is assumed to be constant and known. Geometry also can be evaluated from the estimated stiffness. For a dental health diagnosis, the stiffness

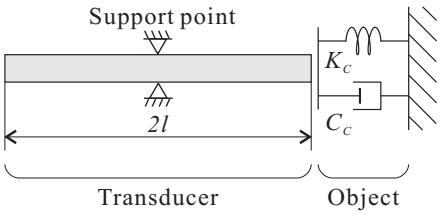


Fig. 14. Contact model of the transducer.

estimation can be applied. To discuss the possibility of the diagnosis, the frequency shifts were measured using the experimental apparatus as shown in Fig. 15. A sample was supported by an aluminum disk through a silicon rubber sheet. The transducer was fed by a z-stage and contacted with the sample. The contact load was measured by load cells under the aluminum disk. This measuring configuration was used in the following experiments.
The combination factor were observed in various materials. The natural frequency shifts in contact with various materials were measured with the change of contact load. The shape and size of the sample was rectangular solid and 20 mm x 20 mm x 5 mm except the $LiNbO_3$ sample. the size of the $LiNbO_3$ sample was 20 mm x 20 mm x 1 mm. The results are plotted in Fig. 16. the natural frequency of the transducer decreased with the increase of contact load in the case of soft material with high damping factor such as rubber. The natural frequency did not change so much in the case of silicon rubber. The natural frequency increased in the case of other materials. Comparing steel (SS400) and aluminum, stiffness of steel is higher than that of aluminum. Frequency shift of $LiNbO_3$ is larger than that of steel

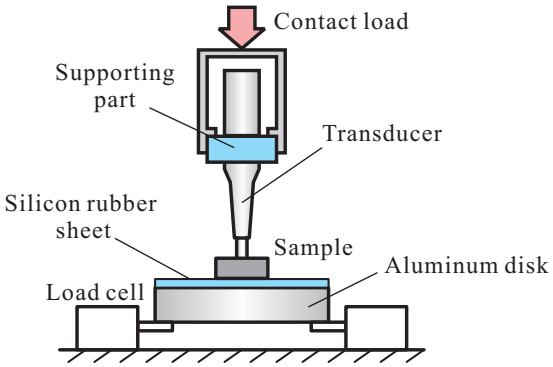


Fig. 15. Experimental apparatus for measurement of the frequency shifts with contact.

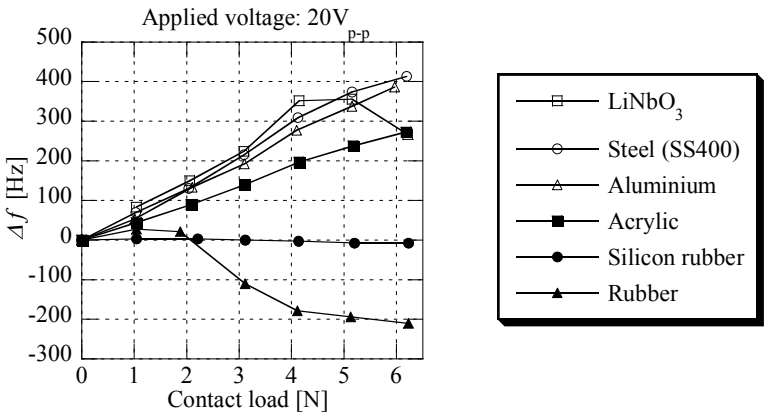


Fig. 16. Measurement of natural frequency shifts with the change of contact load in contact with various materials.

within 4 N though stiffness of $LiNbO_3$ is approximately same as that of steel. This means that mechanical Q factor of $LiNbO_3$ is higher than that of steel, namely, damping factor of $LiNbO_3$ is lower. Frequency shift of $LiNbO_3$ was saturated above 5 N. The reason can be considered that effect of the silicon rubber sheet appeared in the measuring result due to enough acoustic connection between the transducer and the $LiNbO_3$.

The geometry was evaluated from local stiffness. The frequency shifts in contact with aluminum blocks were measured with the change of contact load. The sample of the aluminum block is shown in Fig. 17 (a). Three samples were used in the following experiments. One of the samples had no hole, another had thickness t = 5 mm and the other had the thickness t = 1 mm. Measured frequency shifts are shown in Fig. 17 (b). The frequency shifts tended to be small with decrease of thickness t. These results show that the

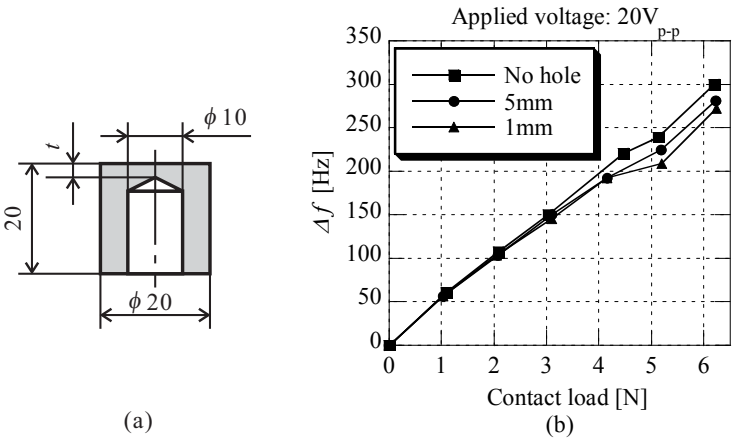


Fig. 17. Measurement of natural frequency shifts with the change of contact load in contact with aluminum blocks.

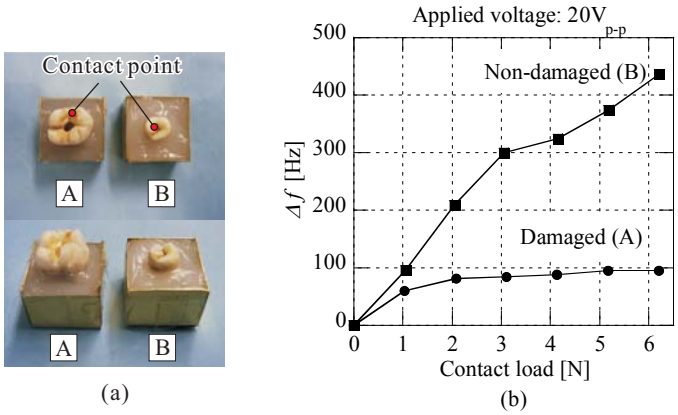


Fig. 18. Measurement of natural frequency shifts with the change of contact load in contact with teeth.

hollow in the contacted object can be investigated from the frequency shift even though there is no difference in outward aspect.
Such elastic parameters estimation and the hollow investigation were applied for diagnosis of dental health. The natural frequency shifts in contact with real teeth were also measured on trial. Figure 18 (a) shows the teeth samples. Sample A is damaged by dental caries and B is not damaged. The plotted points in the picture indicate contact points. To simulate real environment, the teeth were supported by silicon rubber. Measured frequency shifts are shown in Fig. 18 (b). It can be seen that the natural frequency shift of the damaged tooth is smaller than that of healthy tooth.
Difference of resonance frequency shifts was observed. To conclude the possibility of dental health diagnosis, a large number of experimental results were required. Collecting such scientific date is our future work.

5. Conclusions

A resonance frequency tracing system for Langevin type ultrasonic transducers was built up. The system configuration and the method of tracing were presented. The system does not included a loop filter. This point provided easiness in the contoller design and availability for various transducers.
The system was applied to an ultrasonic dental scaler. The traceability of the system with a transducer for the scaler was evaluated from step responses of the oscillating frequency. The settling time was 40 ms. Natural frequency shifts under tip contact with various object, materials and geometries were observed. The shift measurement was applied to diagnosis of dental health. Possibility of the diagnosis was shown.

6. References

Ide, M. (1968). Design and Analysis of Ultorasonic Wave Constant Velocity Control Oscillator, *Journal of the Institute of Electrical Engineers of Japan,* Vol.88-11, No.962, pp.2080-2088.

Si, F. & Ide, M. (1995). Measurement on Specium Acousitic Impedamce in Ultrsonic Plastic Welding, *Japanese Journal of applied physics,* Vol.34, No.5B, pp.2740-2744.

Shimizu, H., Saito, S. (1978). Methods for Automatically Tracking the Transducer Resonance by Rectified-Voltage Feedback to VCO, *IEICE Technical Report*, Vol.US78, No.173, pp.7-13.

Hayashi, S. (1991). On the tracking of resonance and antiresonance of a piezoelectric resonator, *IEEE Transactions on. Ultrasonic, Ferroelectrics and Frequency Control,* Vol.38, No.3, pp.231-236.

Hayashi, S. (1992). On the tracking of resonance and antiresonance of a piezoelectric resonator. II. Accurate models of the phase locked loop, *IEEE Transactions on. Ultrasonic, Ferroelectrics and Frequency Control,* Vol.39, No.6, pp.787-790.

Aoyagi, R. & Yoshida, T. (2005), Unified Analysis of Frequency Equations of an Ultrasonic Vibrator for the Elastic Sensor, *Ultrasonic Technology*, Vol.17, No.1, pp. 27-32.

Nishimura, K. et al., (1994), Directional Dependency of Sensitivity of Vibrating Touch sensor, Proceedings of Japan Society of Precision Engineering Spring Conference, pp. 765-766.

8

New visual Servoing control strategies in tracking tasks using a PKM

[i]A. Traslosheros, [ii]L. Angel, [i]J. M. Sebastián,
[iii]F. Roberti, [iii]R. Carelli and [i]R. Vaca
[i]DISAM, Universidad Politécnica de Madrid, Madrid, España, [ii] Facultad de Ingeniera Electrónica Universidad Pontificia Bolivariana Bucaramanga, Colombia, [iii]Instituto de Automática, Universidad Nacional de San Juan, San Juan, Argentina

1. Introduction

Vision allows a robotic system to obtain a lot of information on the surrounding environment to be used for motion planning and control. When the control is based on feedback of visual information is called Visual Servoing. Visual Servoing is a powerful tool which allows a robot to increase its interaction capabilities and tasks complexity. In this chapter we describe the architecture of the Robotenis system in order to design two different control strategies to carry out tracking tasks. Robotenis is an experimental stage that is formed of a parallel robot and vision equipment. The system was designed to test joint control and Visual Servoing algorithms and the main objective is to carry out tasks in three dimensions and dynamical environments. As a result the mechanical system is able to interact with objects which move close to 2m=s. The general architecture of control strategies is composed by two intertwined control loops: The internal loop is faster and considers the information from the joins, its sample time is 0:5ms. Second loop represents the visual Servoing system and it is an external loop to the first mentioned. The second loop represents the main study purpose, it is based in the prediction of the object velocity that is obtained from visual information and its sample time is 8:3ms. The robot workspace analysis plays an important role in Visual Servoing tasks, by this analysis is possible to bound the movements that the robot is able to reach. In this article the robot jacobian is obtained by two methods. First method uses velocity vector-loop equations and the second is calculated from the time derivate of the kinematical model of the robot. First jacobian requires calculating angles from the kinematic model. Second jacobian instead, depends on physical parameters of the robot and can be calculated directly. Jacobians are calculated from two different kinematic models, the first one determines the angles each element of the robot. Fist jacobian is used in the graphic simulator of the system due to the information that can be obtained from it. Second jacobian is used to determine off-line the work space of the robot and it is used in the joint and visual controller of the robot (in real time). The work space of the robot is calculated from the condition number of the jacobian (this is a topic that is not studied in article). The dynamic model of the mechanical system is based on Lagrange multipliers, and it uses forearms and end effector platform of non-negligible inertias for the

development of control strategies. By means of obtaining the dynamic model, a nonlinear feed forward and a PD control is been applied to control the actuated joints. High requirements are required to the robot. Although requirements were taken into account in the design of the system, additional protection is added by means of a trajectory planner. the trajectory planner was specially designed to guarantee soft trajectories and protect the system from exceeding its Maximum capabilities. Stability analysis, system delays and saturation components has been taken into account and although we do not present real results, we present two cases: Static and dynamic. In previous works (Sebastián, et al. 2007) we present some results when the static case is considered.
The present chapter is organized as follows. After this introduction, a brief background is exposed. In the third section of this chapter several aspects in the kinematic model, robot jacobians, inverse dynamic and trajectory planner are described. The objective in this section is to describe the elements that are considered in the joint controller. In the fourth section the visual controller is described, a typical control law in visual Servoing is designed for the system: Position Based Visual Servoing. Two cases are described: static and dynamic. When the visual information is used to control a mechanical system, usually that information has to be filtered and estimated (position and velocity). In this section we analyze two critical aspects in the Visual Servoing area: the stability of the control law and the influence of the estimated errors of the visual information in the error of the system. Throughout this section, the error influence on the system behaviour is analyzed and bounded.

2. Background

Vision systems are becoming more and more frequently used in robotics applications. The visual information makes possible to know about the position and orientation of the objects that are presented in the scene and the description of the environment and this is achieved with a relative good precision. Although the above advantages, the integration of visual systems in dynamical works presents many topics which are not solved correctly yet. Thus many important investigation centers (Oda, Ito and Shibata 2009) (Kragic and I. 2005) are motivated to investigate about this field, such as in the Tokyo University ((Morikawa, et al. 2007), (Kaneko, et al. 2005) and (Senoo, Namiki and Ishikawa 2004)) where fast tracking (up to $6m=s$ and $58m=s^2$) strategies in visual servoing are developed. In order to study and implementing the different strategies of visual servoing, the computer vision group of the UPM (Polytechnic University of Madrid) decided to design the Robotenis vision-robot system. Robotenis system was designed in order to study and design visual servoing controllers and to carry out visual robot tasks, specially, those involved in tracking where dynamic environments are considered. The accomplishment of robotic tasks involving dynamical environments requires lightweight yet stiff structures, actuators allowing for high acceleration and high speed, fast sensor signal processing, and sophisticated control schemes which take into account the highly nonlinear robot dynamics. Motivated by the above reasons we proposed to design and built a high-speed parallel robot equipped with a vision system.

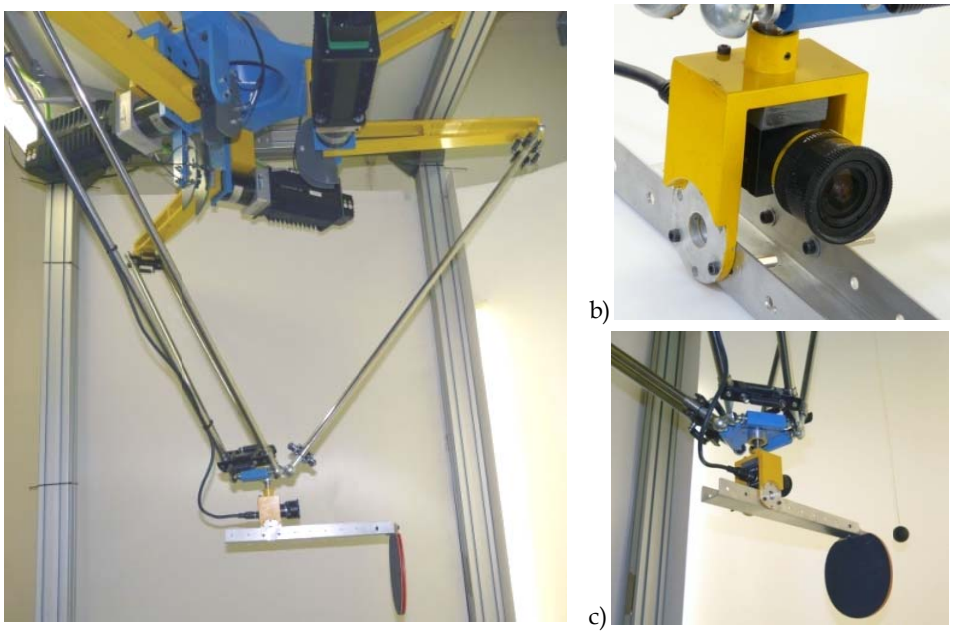


Fig. 1. Robotenis system and its environment: Robot, camera, background, ball and paddle.

The Robotenis System was created taking into account mainly two purposes. The first one is the development of a tool in order to use in visual servoing research. The second one is to evaluate the level of integration between a high-speed parallel manipulator and a vision system in applications with high temporary requirements. The mechanical structure of Robotenis System is inspired by the DELTA robot (Clavel 1988) (Stamper and Tsai 1997) and the vision system is based in one camera allocated at the end effector of the robot. The reasons that motivate us the choice of the robot is a consequence of the high requirements on the performance of the system, especially with regard to velocity, acceleration and the precision of the movements. The kinematic analysis and the optimal design of the Robotenis System have been presented by Angel, et al. (Angel, et al. 2005). The structure of the robot has been optimized from the view of both kinematics and dynamics respectively. The design method solved two difficulties: determining the dimensions of the parallel robot and selecting the actuators. In addition, the vision system and the control hardware was also selected.

3. Robotenis description

Basically, the Robotenis platform (Fig. 1.a) is formed by a parallel robot and a visual acquisition system. The parallel robot is based on a DELTA robot and its maximum end-effector speed is 4m=s. The visual system is based on a camera in hand and its objective in this article resides in tracking a black ping pong ball. Visual control is designed by considering static and dynamic case. Static case considers that the desired distance between the camera and the ball is constant. Dynamic case considers that the desired distance

between the ball and the camera can be changed at any time. Image processing is conveniently simplified using a black ball on white background. The ball is moved through a stick (Fig. 1.c) and the ball velocity is close to 2m=s. The visual system of the Robotenis platform is formed by a camera located at the end effector (Fig. 1.b) and a frame grabber (SONY XC-HR50 and Matrox Meteor 2-MC/4 respectively) The motion system is formed by AC brushless servomotors, Ac drivers (Unidrive) and gearbox.

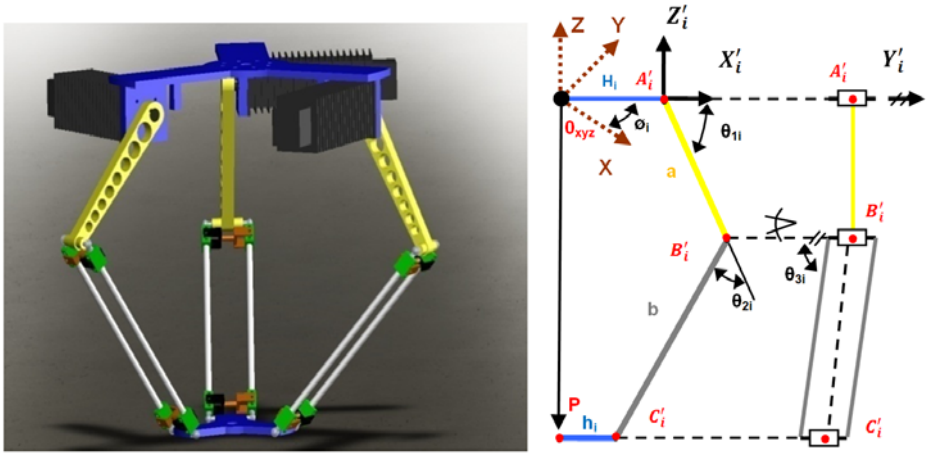


Fig. 2. Cad model and sketch of the robot that it is seen from the side of the i-arm
In section 3.1

3.1 Robotenis kinematical models

A parallel robot consists of a fixed platform that it is connected to an end effector platform by means of legs. These legs often are actuated by prismatic or rotating joints and they are connected to the platforms through passive joints that often are spherical or universal. In the Robotenis system the joints are actuated by rotating joints and connexions to end effector are by means of passive spherical joints. If we applied the Grüble criterion to the Robotenis robot, we could note that the robot has 9 DOF (this is due to the spherical joints and the chains configurations) but in fact the robot has 3 translational DOF and 6 passive DOF. Important differences with serial manipulators are that in parallel robots any two chains form a closed loop and that the actuators often are in the fixed platform. Above means that parallel robots have high structural stiffness since the end effector is supported in several points at the same time. Other important characteristic of this kind of robots is that they are able to reach high accelerations and forces, this is due to the position of the actuators in the fixed platform and that the end effector is not so heavy in comparison to serial robots. Although the above advantages, parallel robots have important drawbacks: the work space is generally reduced because of collisions between mechanical components and that singularities are not clear to identify. In singularities points the robot gains or losses degrees of freedom and is not possible to control. We will see that the Jacobian relates the actuators velocity with the end effector velocity and singularities occur when the Jacobian rank drops.

Nowadays there are excellent references to study in depth parallel robots, (Tsai 1999), (Merlet 2006) and recently (Bonev and Gosselin 2009).
For the position analysis of the robot of the Robotenis system two models are presented in order to obtain two different robot jacobians. As was introduced, the first jacobian is utilized in the Robotenis graphic simulator and second jacobian is utilized in real time tasks. Considers the Fig. 2, in this model we consider two reference systems. In the coordinate system Σ_{Oxyz} are represented the absolute coordinates of the system and the position "P" of the end effector of the robot. In the local coordinate system Σ'_{Oxyz} (allocated in each point A_i) the position and coordinates (A', B', C') of the i-arm are considered. The first kinematic model is calculated from Fig. 2 where the loop-closure equation for each limb is:

$$\overline{A'_i B'_i} + \overline{B'_i C'_i} = \overline{O_{x\,y\,z} P} + \overline{P C'_i} - \overline{O_{x\,y\,z} A'_i} \tag{1}$$

Expressing (note that $s(x) = sin(x)$ and $c(x) = cos(x)$ the eq. (1) in the coordinate system attached to each limb is possible to obtain:

$$\begin{bmatrix} C'_{i\,x} \\ C'_{i\,y} \\ C'_{i\,z} \end{bmatrix} = \begin{bmatrix} a\,c(\theta_{1i}) + b\,s(\theta_{3i})\,c(\theta_{1i}+\theta_{2i}) \\ b\,c(\theta_{3i}) \\ a\,s(\theta_{1i}) + b\,s(\theta_{3i})\,s(\theta_{1i}+\theta_{2i}) \end{bmatrix} \tag{2}$$

Where P and C_i are related by

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} \mathrm{c}(\phi_i) & -\mathrm{s}(\phi_i) & 0 \\ \mathrm{s}(\phi_i) & \mathrm{c}(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} C'_{ix} \\ C'_{iy} \\ C'_{iz} \end{bmatrix} + \begin{bmatrix} -h_i + H_i \\ 0 \\ 0 \end{bmatrix} \right) \tag{3}$$

In order to calculate the inverse kinematics, from the second row in eq. (2), we have:

$$\theta_{3i} = \mathrm{c}^{-1}\left(\frac{C'_{i\,y}}{b_i} \right) \tag{4}$$

θ_{2i} can be obtained by summing the squares of C_{ix}, C_{iy} and C_{iz} of the eq. (2).

$$C'^2_{i\,x} + C'^2_{i\,y} + C'^2_{i\,z} - a^2 - b^2 = 2a\,b\,s(\theta_{3i})\,c(\theta_{2i}) \quad \rightarrow \quad \theta_{2i} = -\mathrm{c}^{-1}\left(\frac{C'^2_{ix} + C'^2_{iy} + C'^2_{iz} - a^2 - b^2}{2a\,b\,s(\theta_{3i})} \right) \tag{5}$$

By expanding left member of the first and third row of the eq. (2) by using trigonometric identities and making $\Psi_i = b_i \sin(\theta_{2i}) \sin(\theta_{3i})$ and $\Upsilon_i = a_i + b_i \cos(\theta_2) \sin(\theta_3)$:

$$\begin{aligned} \Upsilon_i c(\theta_{1i}) - \Psi_i s(\theta_{1i}) = C'_{ix} \\ \Upsilon_i s(\theta_{1i}) - \Psi_i c(\theta_{1i}) = C'_{iz} \end{aligned} \tag{6}$$

Note that from (6) we can obtain:

$$s(\theta_{1i}) = \left(\frac{\Upsilon_i C'_{iz} - \Psi_i C'_{ix}}{\Upsilon_i^2 + \Psi_i^2} \right) \quad \text{and} \quad c(\theta_{1i}) = \left(\frac{\Upsilon_i C'_{ix} + \Psi_i C'_{iz}}{\Upsilon_i^2 + \Psi_i^2} \right) \tag{7}$$

Equations in (7) can be related to obtain θ_{1i} as:

$$\theta_{1i} = \tan^{-1}\left(\frac{\Upsilon_i C'_{iz} - \Psi_i C'_{ix}}{\Upsilon_i C'_{ix} + \Psi_i C'_{iz}} \right) \tag{8}$$

In the use of above angles we have to consider that the "Z" axis that is attached to the center of the fixed platform it is negative in the space that the end effector of the robot will be operated. Taking into account the above consideration, angles are calculated as:

$$\theta_{3i} = -c^{-1}\left(\frac{C'_{iy}}{b} \right) \quad \theta_{2i} = -c^{-1}\left(\frac{C'^2_{ix} + C'^2_{iy} + C'^2_{iz} - a^2 - b^2}{2a\, b\, s(\theta_{3i})} \right) \quad \theta_{1i} = -\tan^{-1}\left(\frac{\Upsilon_i C'_{iz} - \Psi_i C'_{ix}}{\Upsilon_i C'_{ix} + \Psi_i C'_{iz}} \right) \tag{9}$$

Second kinematic model is obtained from Fig. 3.

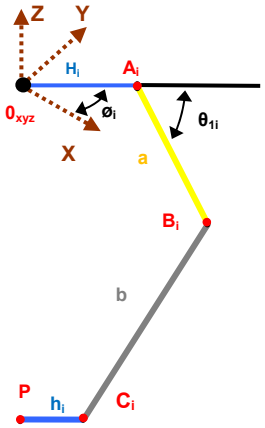


Fig. 3. Sketch of the robot taking into account an absolute coordinate reference system.

If we consider only one absolute coordinate system in Fig. 3, note that the segment $\overline{B_i C_i}$ is the radius of a sphere that has its center in the point B_i and its surface in the point C_i, (all points in the absolute coordinate system). Thus sphere equation as:

$$\Gamma_i = \left(C_{ix} - B_{ix}\right)^2 + \left(C_{iy} - B_{iy}\right)^2 + \left(C_{iz} - B_{iz}\right)^2 - b^2 = 0 \tag{10}$$

From the Fig. 3 is possible to obtain the point $B_i = {}^{O_{xyz}} B_i$ in the absolute coordinate system.

$$ {}^{O_{xyz}}\begin{bmatrix} B_{ix} \\ B_{iy} \\ B_{iz} \end{bmatrix} = \begin{bmatrix} \left(H_i + a\,c\left(\theta_i\right)\right) c\left(\phi_i\right) \\ \left(H_i + a\,c\left(\theta_i\right)\right) s\left(\phi_i\right) \\ a\,s\left(\theta_i\right) \end{bmatrix} \qquad \text{where } \mu_i = \mu_{1i} \tag{11}$$

Replacing eq. (11) in eq. (10) and expanding it the constraint equation $\boldsymbol{\Gamma_i}$ is obtained:

$$\begin{aligned} \Gamma_i = & -H_i^2 - a^2 + b^2 - C_{ix}^2 - C_{iy}^2 - C_{iz}^2 + 2H_i C_{ix} c\left(\phi_i\right) + 2H_i C_{iy} s\left(\phi_i\right) - 2H_i a\,c\left(\theta_i\right) + 2a\,C_{ix} c\left(\theta_i\right) c\left(\phi_i\right) \\ & - 2a\,C_{iz} s\left(\theta_i\right) + 2a\,C_{iy} c\left(\theta_i\right) s\left(\phi_i\right) = 0 \end{aligned} \tag{12}$$

In order to simplify, above can be regrouped, thus for the i-limb:

$$E_i\, s\left(\theta_i\right) + F_i c\left(\theta_i\right) + M_i = 0 \tag{13}$$

Where:

$$\begin{aligned} F_i &= 2a\left(C_{ix} c\left(\phi_i\right) + C_{iy} s\left(\phi_i\right) - H_i\right) & E_i &= 2a\,C_{iz} \\ M_i &= b^2 - a^2 - C_{ix}^2 - C_{iy}^2 - C_{iz}^2 - H_i^2 + 2H_i\left(C_{ix} c\left(\phi_i\right) + C_{iy} s\left(\phi_i\right)\right) \end{aligned} \tag{14}$$

The following trigonometric identities can be replaced into eq. (13):

$$s\left(\theta_i\right) \equiv \frac{2\tan\left(\frac{1}{2}\theta_i\right)}{1 + \tan^2\left(\frac{1}{2}\theta_i\right)} \qquad \text{and} \qquad c\left(\theta_i\right) \equiv \frac{1 - \tan^2\left(\frac{1}{2}\theta_i\right)}{1 + \tan^2\left(\frac{1}{2}\theta_i\right)} \tag{15}$$

And we can obtain the following second order equation:

$$\left(M_i - F_i\right)\tan^2\left(\frac{1}{2}\theta_i\right) + 2E_i \tan\left(\frac{1}{2}\theta_i\right) + M_i + F_i = 0 \tag{16}$$

And the angle θ_i can be finally obtained as:

$$\theta_i = 2\left[\tan^{-1}\left(\frac{-E_i \pm \sqrt{E_i^2 + \left(F_i^2 - M_i^2\right)}}{\left(M_i - F_i\right)}\right)\right] \tag{17}$$

Where C_{ix}, C_{iy} and C_{iz} are:

$$\begin{bmatrix} C_{ix} \\ C_{iy} \\ C_{iz} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} - \begin{bmatrix} c(\phi_i) & -s(\phi_i) & 0 \\ s(\phi_i) & c(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -h_i \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

3.2 Robot Jacobians

In robotics, the robot Jacobian can be seen as the linear relation between the actuators velocity and the end effector velocity. In Fig. 4. direct and inverse Jacobian show them relation with the robot speeds. Although the jacobian can be obtained by other powerful methods (screws theory (Stramigioli and Bruyninckx 2001) (Davidson and Hunt Davidson 2004) or motor algebra (Corrochano and Kähler 2000)), conceptually the robot jacobian can be obtained as the derivate of the direct or the inverse kinematic model. In parallel robots the obtaining of the Jacobian by means of the screws theory or motor algebra can be more complicated. This complication is due to its non actuated joints (that they are not necessary passive joints). The easier method to understand, but not to carry out, is to derivate respect the time the kinematic model of the robot.

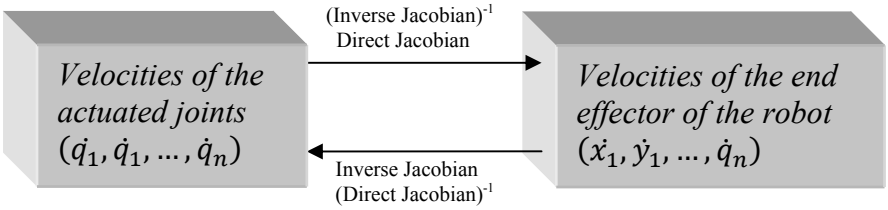


Fig. 4. Direct and indirect Jacobian and its relation with the robot velocities

Sometimes is more complex to obtain the inverse or direct Jacobian from one kinematic model than other thus, in some practical cases is possible to obtain the inverse Jacobian by inverting the direct Jacobian and vice versa, Fig. 4. Above proposal is easy to describe but does not analyze complications. For example if we would like to calculate the inverse Jacobian form the direct Jacobian, we have to find the inverse of a matrix that its dimension is normally 6 × 6. This matrix inversion it could be very difficult because components of the Jacobian are commonly complex functions and composed by trigonometric functions. Alternatively it is possible to calculate the inverse Jacobian or the direct Jacobian by other methods. For example if we have the algebraic form of the direct Jacobian, we could calculate the inverse of the Jacobian by means of inverting the numeric direct Jacobian (previously evaluated at one point). On the other hand the Jacobian gives us important

information about the robot, from the Jacobian we can determinate when a singularity occurs. There are different classifications for singularities of a robot. Some singularities can be classified according to the place in the space where they occurs (singularities can present on the limit or inside of the workspace of the robot). Another classification takes into account how singularities are produced (produced from the direct or inverse kinematics).
Suppose that the eq. (19) describes the kinematics restrictions that are imposed to mechanical elements (joints, arms, lengths, etc.) of the robot.

$$f(x,q)=0 \tag{19}$$

Where $\boldsymbol{x} \in \mathfrak{R}^{6\times 1}$ is the position and orientation of the end effector of the robot and $\boldsymbol{q} \in \mathfrak{R}^{n\times 1}$ are the joint variables that are actuated. Note that if $\mathbf{n} > 6$ the robot is redundant and if $\mathbf{n} < 6$ the robot cannot fully orientate ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$) or displace (along $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) in the 3D space. Although sometimes a robot can be specially designed with other characteristics, in general a robot has the same number of DOF that its number of actuators.
Consider the time derivative of the eq. (19) in the following equation.

$$J_x\dot{x}=J_q\dot{q} \quad \text{where} \quad J_x=\frac{\partial f(x,q)}{\partial x} \quad \text{and} \quad J_q=-\frac{\partial f(x,q)}{\partial q} \tag{20}$$

Note that $\dot{x}$ and $\dot{q}$ are the time derivate of x and q respectely. The direct and the inverse Jacobian can be obtained as the following equations.

$$\dot{x}=J_D\dot{q} \quad \text{and} \quad \dot{q}=J_I\dot{x} \quad \text{where} \quad J_D=J_x^{-1}J_q \quad \text{and} \quad J_I=J_q^{-1}J_x \tag{21}$$

A robot singularity occurs when the determinant of the Jacobian is cero. Singularities can be divided in three groups: singularities that are due to the inverse kinematics, those that are due to the direct kinematics and those that occurs when both above singularities take place at the same time (combined singularities). For a non redundant robot (the Jacobian is a square matrix), each one of above singularities happens when: $Det(J_q)=0$, $Det(J_x)=0$ and when $Det(J_q)=Det(J_x)=0$. Singularities can be interpreted differently in serial robots and in parallel robots. When we have that $Det(J_q)=0$, it means that the null space of J_q is not empty. That is, there are values of $\dot{q}$ that are different from cero and produce an end effector velocity that is equal to cero $\dot{x}=0$. In this case, the robot loses DOF because there are infinitesimal movements of the joints that do not produce movement of the end effector; commonly this occurs when robot links of a limb are in the same plane. Note that when an arm is completely extended, the end effector can supported high loads when the action of the load is in the same direction of the extended arm. On the other hand when $Det(J_x)=0$ we have a direct kinematics singularity, this means that the null space of J_x is not empty. The above means that there are values of $\dot{x}$ that are different from cero when the actuators are blocked $\dot{q}=0$. Physically, the end effector of the robot gains DOF. When the end effector gains DOF is possible to move infinitesimally although the actuators would be blocked. The third type of singularity it is a combined singularity and can occurs in parallel robots with special architecture or under especial considerations. Sometimes singularities can be

identified from the Jacobian almost directly but sometimes Jacobian elements are really complex and singularities are difficult to identify. Singularities can be identified in easier manner depending on how the Jacobian is obtained. The Robotenis Jacobian is obtained by two methods, one it is obtained from the time derivate of a closure loop equation (1) and the second Jacobian is obtained from the time derivate of the second kinematic model. Remember that the jacobian obtained from the eq. (1) requires solving the kinematic model in eqs. (9). This jacobian requires more information of each element of the robot and it is used in the graphic simulator of the system. In order to obtain the jacobian consider that $O = O_{xyz}$ and that the eq. (1) is rearranged as:

$$\overline{O\,P}+R(\phi,z)\,\overline{P\,C'_i}=R(\phi_i,z)\left[\overline{O\,A'_i}+\overline{A'_i B'_i}-\overline{B'_i C'_i}\right] \tag{22}$$

Where $R(\emptyset,z)$ is a $\emptyset\ rad$ rotation matrix around the Z axis in the absolute coordinate system.

$$R(\phi,z)=\begin{bmatrix} c(\phi) & -s(\phi) & 0 \\ s(\phi) & c(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

By obtaining the time derivate of the equation (22) and multiplying by $R^T(\emptyset,z)$ we have:

$$R^T(\phi_i,z)\,\dot{P}=\omega_{1i}\times\mathbf{a_i}+\omega_{2i}\times\mathbf{b_i} \tag{24}$$

Where $\dot{P}$ is velocity of the end effector in the XYZ coordinate system, $a_i=\overrightarrow{A'_iB'_i}$, $b_i=\overrightarrow{B'_iC'_i}$ and w_{1i}, w_{2i} are the angular velocities of the links 1 and 2 of the i limb. Observe that θ_{i2} and θ_{i3} are passive variables (they are not actuated) thus to eliminate the passive joint speeds (w_{2i}) we dot multiply both sides of eq. (24) by $\boldsymbol{b_i}$. By means of the properties of the triple product ($\vec{b}\cdot(\vec{\omega}\times\vec{a})=\ \vec{\omega}\cdot(\vec{a}\times\vec{b})$ and $\vec{b}\cdot(\vec{\omega}\times\vec{b})=\ \vec{\omega}\cdot(\vec{b}\times\vec{b})=0$) is possible to obtain:

$$b\cdot\left[R^T(\phi_i,z)\,\dot{P}\right]=\omega_{1i}\cdot(\mathbf{a}_i\times\mathbf{b}_i) \tag{25}$$

From Fig. 2 elements of above equation are:

$$\mathbf{a_i}=a\begin{bmatrix} c(\theta_{1i}) \\ 0 \\ s(\theta_{1i}) \end{bmatrix};\qquad \mathbf{b_i}=b\begin{bmatrix} s(\theta_{3i})c(\theta_{1i}+\theta_{2i}) \\ c(\theta_{3i}) \\ s(\theta_{3i})s(\theta_{1i}+\theta_{2i}) \end{bmatrix}\quad\text{and}\quad \omega_{1i}=\begin{bmatrix} 0 \\ -\dot{\theta}_{1i} \\ 0 \end{bmatrix} \tag{26}$$

All of them are expressed in the i −coordinate system. Substituting equations in (26) into (25) and after operating and simplifying we have:

$$\begin{bmatrix} m_{1x} & m_{1y} & m_{1z} \\ m_{2x} & m_{2y} & m_{2z} \\ m_{3x} & m_{3y} & m_{3z} \end{bmatrix} \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = a \begin{bmatrix} s(\theta_{21})s(\theta_{31}) & 0 & 0 \\ 0 & s(\theta_{22})s(\theta_{32}) & 0 \\ 0 & 0 & s(\theta_{23})s(\theta_{33}) \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \tag{27}$$

Where

$$\begin{aligned} m_{ix} &= c(\theta_{1i}+\theta_{2i})s(\theta_{3i})c(\phi_i) - c(\theta_{3i})s(\phi_i) \\ m_{iy} &= c(\theta_{1i}+\theta_{2i})s(\theta_{3i})s(\phi_i) + c(\theta_{3i})c(\phi_i) \\ m_{iz} &= s(\theta_{1i}+\theta_{2i})s(\theta_{3i}) \end{aligned} \tag{28}$$

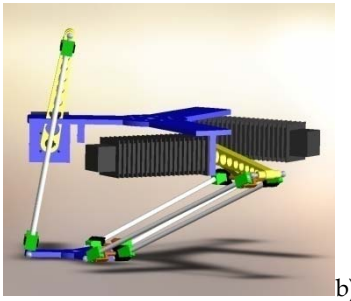
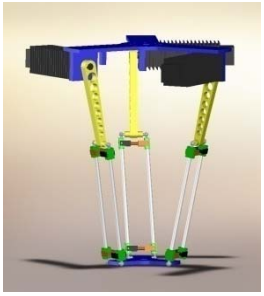
Note that the right and left part of the eq. (27) represents the inverse and direct Jacobians respectively. An inverse kinematic singularity occurs when $\theta_{2i} = 0\ or\ \pi$ or $\theta_{3i} = 0\ or\ \pi$, see Fig. 5 a) and b). On the other hand direct kinematic singularities occur when rows of the left matrix become linearly dependent. The above is:

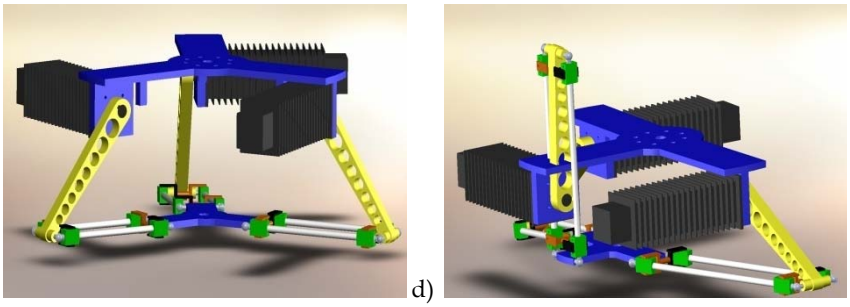
$$k_1[m_1]+k_2[m_2]+k_3[m_3]=0 \qquad \text{Where } k_1, k_2, k_3 \in \Re \text{ and not all are cero} \tag{29}$$

Equation (29) is not as clear as the right part of the equation (27) but we can identify a group of direct kinematic singularities when the last column in the three rows is cero, this is:

$$s(\theta_{11}+\theta_{21})s(\theta_{31}) = s(\theta_{12}+\theta_{22})s(\theta_{32}) = s(\theta_{13}+\theta_{23})s(\theta_{33}) = 0 \tag{30}$$

When $\theta_{1i}+\theta_{2i}=0$ or $=\pi \quad \forall i=1,2,3$ or when $\theta_{3i}=0$ or $=\pi \quad \forall i=1,2,3$

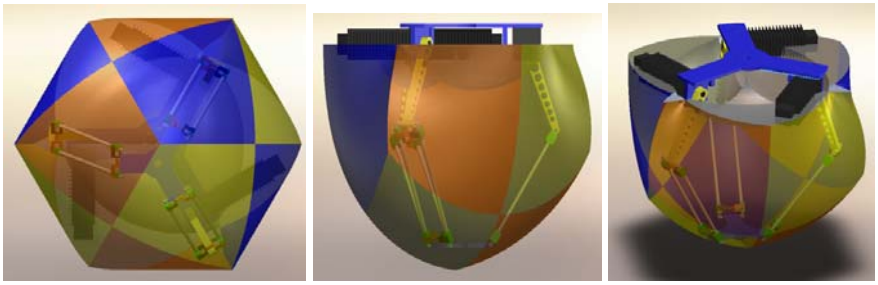
a)  b) 



c) d)

Fig. 5. a) Inverse kinematic singularities if $\boldsymbol{\theta_{21} = \pi}$. b) Inverse kinematic singularities where $\boldsymbol{\theta_{2i} = 0}$; $\boldsymbol{\forall\, i = 1,2,3}$. c) Direct kinematic singularity if $\boldsymbol{\theta_{1i} + \theta_{2i} = \pi}$; $\boldsymbol{\forall\, i = 1,2,3}$. d) Combined kinematic singularity if $\boldsymbol{\theta_{21} = \pi}$; $\boldsymbol{\theta_{32} = 0}$; $\boldsymbol{\theta_{33} = 0}$; $\boldsymbol{\theta_{12} + \theta_{22} = \pi}$; $\boldsymbol{\theta_{13} + \theta_{23} = 0}$ and $\boldsymbol{\phi_2 + \phi_3 = 0\ or\ \pi}$. Note that the robot presents a combined singularity if three angles $\boldsymbol{\theta_{3i} = \pi}$; $\boldsymbol{\forall\, i = 1,2,3}$ consequently case c) is a combined singularity ($\boldsymbol{\theta_{32} = 0}$; $\boldsymbol{\theta_{33} = 0}$). Note that the design of the robot plays a very important role because singularities can even avoid. For example in figure c) the singularity is present because lengths of the forearm allows to be in the same plane that the end effector platform and in the figure d) a combined singularity is present because $\boldsymbol{\phi_2 + \phi_3 = 0}$. In all figures we suppose that the limb $\boldsymbol{i = 1}$ is the limb situated to the left of the images. Note that collisions between mechanical elements are not taken into account.

By considering (27) to (30), direct kinematic singularities present when the end effector platform is in the same plane as the parallelograms of the 3 limbs, in this configuration the robot cannot resist any load in the Z direction, see Fig. 5 c). Note that singularities like above depend on the lengths and angles of the robot when it was designed Fig. 5 c), such is the case of the above configuration where singularity can present when $\boldsymbol{a + H \geq b +}$, other singularities can present in special values of $\boldsymbol{\phi_i}$ Fig. 5 d).
Analysis of singularities of the work space is important for Visual controller in order to bound the workspace an avoid robot injures. Above analysis is useful because some singularities are given analytically. Different views of the work space of the CAD model of the Robotenis system is shown in Fig. 6



a) b) c)

Fig. 6 Work space of the Robotenis system. a) Work space is seen from bottom part of the robot, b) it shows the workspace from side. c) The isometric view of the robot is shown.

As was mentioned a second jacobian is obtained to use in real time tasks, by the condition number of the jacobian (Yoshikawa 1985) is possible to know how far a singularity is. Condition number of the jacobian y checked before carry out any movement of the robot, if a large condition number is present, then the movement is not carried out.
In order to obtain the second jacobian consider that we have the inverse kinematic model of a robot in given by eq. (31).

$$\begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} f_1(x, y, z, \alpha, \beta, \gamma) \\ \vdots \\ f_n(x, y, z, \alpha, \beta, \gamma) \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J_I \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \qquad \text{Where:} \qquad J_I = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \cdots & \frac{\partial f_n}{\partial \gamma} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x} & \cdots & \frac{\partial f_n}{\partial \gamma} \end{bmatrix} \tag{32}$$

Note that the kinematic model of the Robotenis system is formed by three equations in eq. (17) (the end effector cannot orientate) and this model has the form of the eq. (31). Consequently to obtain the jacobian we have to find the time derivate of the kinematic model. Thus to simplify operations we suppose that.

$$\psi_i(t) = \frac{-E_i \pm \sqrt{E_i^2 + \left(F_i^2 - M_i^2\right)}}{\left(M_i - F_i\right)} \tag{33}$$

And that in terms of $\psi$the time derivate of (17) is:

$$\dot{\theta}_i = 2\left(\frac{\dot{\psi}_i}{1+\psi_i^2}\right) \tag{34}$$

Where $\dot{\psi}_i$ is

$$\dot{\psi}_i = \left[\frac{-\dot{E}}{M-F} + \frac{P\left(\dot{M}-\dot{F}\right)}{(M-F)^2} \pm \left[\frac{E\dot{E} - M\dot{M} + F\dot{F}}{(M-F)\sqrt{E^2 - \left(M^2 - F^2\right)}} + \frac{-\sqrt{E^2 - \left(M^2 - F^2\right)}\left(\dot{M}-\dot{F}\right)}{(M-F)^2} \right] \right]_i \tag{35}$$

Considering that $\eta_1 = \frac{1}{M-F}$ and $\eta_2 = \frac{1}{\sqrt{E^2-(M^2-F^2)}}$ can be replaced in (35).

$$\dot{\psi}_i = \left[-\eta_1 \dot{E} + \eta_1^2 E\dot{M} - \eta_1^2 E\dot{F} \pm \left[\eta_1\eta_2 E\dot{E} - \eta_1\eta_2 M\dot{M} + \eta_1\eta_2 F\dot{F} - \frac{\eta_1^2\dot{M}}{\eta_2} + \frac{\eta_1^2\dot{F}}{\eta_2} \right] \right]_i \tag{36}$$

On the other hand we know that:

$$\begin{aligned}\dot{P}_i &= 2a\dot{C}_{ix}\mathrm{c}(\phi_i)+2a\dot{C}_{iy}\mathrm{s}(\phi_i)\\ \dot{M}_i &= -2C_{ix}\dot{C}_{ix}-2C_{iy}\dot{C}_{iy}-2C_{iz}\dot{C}_{iz}+2H\dot{C}_{ix}\mathrm{c}(\phi_i)+2H\dot{C}_{iy}\mathrm{s}(\phi_i)\\ \dot{E}_i &= 2a\dot{C}_{iz}\end{aligned} \tag{37}$$

By rearranging terms in eq. (36) and considering terms in (37) is possible to obtain ψ in terms of the velocity of the end effector $\dot{C}_{xyz}$.

$$\dot{\psi}_i = 2\left(d_{ix}\dot{C}_{ix}+d_{iy}\dot{C}_{iy}+d_{iz}\dot{C}_{iz}\right) \tag{38}$$

Where:

$$d_{ix}=\left[\begin{array}{l}\eta_1^2 P\left[-C_x+H\mathrm{c}(\phi)-a\mathrm{c}(\phi)\right]\pm\\ \left[\eta_1\eta_2\left[MC_x+Fa\mathrm{c}(\phi)-MH\mathrm{c}(\phi)\right]+\right.\\ \left.\frac{\eta_1^2}{\eta_2}\left[a\mathrm{c}(\phi)+C_x-H\mathrm{c}(\phi)\right]\right]\end{array}\right]_i,\quad d_{iy}=\left[\begin{array}{l}\eta_1^2 P\left[-C_y+H\mathrm{s}(\phi)-a\mathrm{s}(\phi)\right]\pm\\ \left[\eta_1\eta_2\left[Fa\mathrm{s}(\phi)+MC_y-MH\mathrm{s}(\phi)\right]+\right.\\ \left.\frac{\eta_1^2}{\eta_2}\left[a\mathrm{s}(\phi)+C_y-H\mathrm{s}(\phi)\right]\right]\end{array}\right]_i,$$

$$d_{iz}=\eta_1\left[\begin{array}{l}-a-\eta_1 EC_z\pm\\ \left[\eta_2\left(Pa+MC_z\right)+\frac{\eta_1}{\eta_2}C_z\right]\end{array}\right]_i \tag{39}$$

Then replacing eq. (38) in (34) we have:

$$\dot{\theta}_i = 4\frac{d_{ix}\dot{C}_{ix}+d_{iy}\dot{C}_{iy}+d_{iz}\dot{C}_{iz}}{1+\left(\left.\frac{-E\pm\sqrt{E^2+\left(F^2-M^2\right)}}{2(M-F)}\right)_i^2\right.} \qquad \text{and} \qquad \dot{\theta}_i = 4\left[D_{ix}\quad D_{iy}\quad D_{iz}\right]\begin{bmatrix}\dot{C}_{ix}\\ \dot{C}_{iy}\\ \dot{C}_{iz}\end{bmatrix} \tag{40}$$

Note that the actuator speed is in terms of the velocity of the point C_i and the time derivate of C_i is:

$$\begin{bmatrix}\dot{C}_{ix}\\ \dot{C}_{iy}\\ \dot{C}_{iz}\end{bmatrix}=\begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\end{bmatrix}+\frac{d}{dt}\begin{bmatrix}\cos(\phi_i) & \sin(\phi_i) & 0\\ -\sin(\phi_i) & \cos(\phi_i) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}h_i\\ 0\\ 0\\ 1\end{bmatrix} \quad \text{where } \phi \text{ is constant and} \quad \begin{bmatrix}\dot{C}_{ix}\\ \dot{C}_{iy}\\ \dot{C}_{iz}\end{bmatrix}=\begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\end{bmatrix} \tag{41}$$

Substituting the above equation in (40) and the expanding the equation, finally the inverse Jacobian of the robot is given by:

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}=4\begin{bmatrix}D_{1x} & D_{1y} & D_{1z}\\ D_{2x} & D_{2y} & D_{2z}\\ D_{3x} & D_{3y} & D_{3z}\end{bmatrix}\begin{bmatrix}\dot{P}_x\\ \dot{P}_y\\ \dot{P}_z\end{bmatrix} \tag{42}$$

Note that the robot Jacobian in eq. (42) has the advantage that is fully expressed in terms of physical parameters of the robot and is not necessary to solve previously any kinematic model. Terms in eq. (42) are complex and this make not easy to detect singularities by only inspecting the expression. In the real time controller, the condition number of the jacobian is calculated numerically to detect singularities and subsequently the jacobian is used in the visual controller.

3.3 Robotenis inverse dynamical model

Dynamics plays an important role in robot control depending on applications. For a wide number of applications the dynamical model it could be omitted in the control of the robot. On the other hand there are tasks in which dynamical model has to be taken into account. Dynamic model is important when the robot has to interact with heavy loads, when it has to move at high speed (even vibrating), when the robot structure requires including dynamical model into its analysis (for example in wired and flexible robots), when the energy has to be optimized or saved. In our case the dynamical model make possible that the end effector of the robot reaches higher velocities and faster response. The inverse dynamics, (given the trajectory, velocities and accelerations of the end effector) determine the necessary joint forces or torques to reach the end-effector requirements. The direct dynamics, being given the actuators joint forces or torques, determine the trajectory, velocity and acceleration of the end effector. In this work the inverse dynamical is retrofitted to calculate the necessary torque of the actuator to move the end effector to follow a trajectory at some velocity and acceleration. We will show how the inverse dynamics is used in the joint controller of the robot. Robotenis system is a parallel robot inspired in the delta robot, this parallel robot is relatively simple and its inverse dynamics can be obtained by applying the Lagrangian equations of the first type. The Lagrangian equations of the Robotenis system are written in terms of coordinates that are redundant, this makes necessary a set of constraint equations (and them derivates) in order to solve the additional coordinates. Constraint equations can be obtained from the kinematical constraints of the mechanism in order to generate the same number of equations that the coordinates that are unknown (generalized and redundant coordinates). Lagrangian equations of the first type can be expressed:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right)-\frac{\partial L}{\partial q_j}=Q_j+\sum_{i=1}^{k}\lambda_i\frac{\partial \Gamma_i}{\partial q_j} \qquad j=1,\ 2,\ \ldots,\ n \tag{43}$$

Where $\boldsymbol{\Gamma_i}$ is the $\boldsymbol{i}$ constraint equation, $\boldsymbol{\lambda_i}$ is the Lagrangian multiplier, $\boldsymbol{k}$ is the number of constraint equation, $\boldsymbol{n}$ is the number of coordinates (Note that $\boldsymbol{Degrees\ of\ freedom}$ =

$\boldsymbol{n-k}$ and in our case DOF = number of actuated joints), $\boldsymbol{Q}$ contains the external applied forces $\hat{\boldsymbol{Q}}_{\boldsymbol{k}}$ and the actuator torques or forces $\boldsymbol{Q_{n-k}}$ ($\boldsymbol{Q}=[\hat{\boldsymbol{Q}}_{\boldsymbol{j}},\boldsymbol{Q_j}]=[\boldsymbol{Q_{j=1,2,\ldots,k}},\boldsymbol{Q_{j=k+1,\ldots,n}}]$). By means of following considerations, the equations in (43) can be arranged in two sets of equations. Consider that the first $\boldsymbol{k}$ equations are associated with the redundant coordinates and the $\boldsymbol{n-k}$ equations are associated with the actuated joint variables, consider that for the inverse dynamics external forces are given or measured. Thus the first set of equations can be arranged as:

$$\sum_{i=1}^{k}\lambda_i\frac{\partial\Gamma_i}{\partial q_j}=\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{q}_j}\right)-\frac{\partial L}{\partial q_j}-\hat{Q}_j \qquad j=1,\ 2,\ \ldots,\ k \tag{44}$$

Where the right side is known and for each redundant coordinate yields a set of k linear equations that can be solved for the k Lagrangian multipliers $\lambda_{1,\ldots,k}$. Finally the second set of equations uses the k Lagrangian multipliers to find the actuator forces or torques. Second set of equations can be grouped in:

$$\tilde{Q}_j=\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{q}_j}\right)-\frac{\partial L}{\partial q_j}-\sum_{i=1}^{k}\lambda_i\frac{\partial\Gamma_i}{\partial q_j} \qquad j=k+1,\ \ldots,\ n \tag{45}$$

Applying the above to the Robotenis system, we have that $\boldsymbol{\theta_{11}}$, $\boldsymbol{\theta_{12}}$ and $\boldsymbol{\theta_{13}}$ can define the full system and can be chosen as generalized coordinates moreover to simplify the Lagrange expression and to solve the Lagrangian by means of Lagrange multipliers we choose 3 additional redundant coordinates $\boldsymbol{P_x}$, $\boldsymbol{P_y}$ and $\boldsymbol{P_z}$. Thus the generalized coordinates are: $\boldsymbol{P_x},\boldsymbol{P_y},\boldsymbol{P_z},\boldsymbol{\theta_{11}},\boldsymbol{\theta_{12}}$ and $\boldsymbol{\theta_{13}}$. External forces and position, velocity and acceleration of the end effector (mobile platform) are known, thus the six variables are: the three Lagrangian multipliers (they correspond to the three constraint equations) and the three actuators torque. Three constraint equations are obtained from the eq. (10) when points $\boldsymbol{C_{ixyz}}$ are substituted by $\boldsymbol{P_{xyz}}$ by means of eq. (18).

$$\Gamma_i=\left(P_x+\left(h_i-H_i\right)c\left(\phi_i\right)-a\,c\left(\phi_i\right)c\left(\theta_i\right)\right)^2+\left(P_y+\left(h_i-H_i\right)s\left(\phi_i\right)-a\,s\left(\phi_i\right)c\left(\theta_i\right)\right)^2+\left(P_z-a\,s\left(\theta_i\right)\right)^2-b^2=0 \tag{46}$$

In the above equation $i=1,2,3$ and to simplify considers that $\theta_{1i}=\ \theta_i$ (this angles are the actuated joint angles) and that $H=H_i$, $h=h_i$; $i=1,2,3$. The Lagrangian equation is obtained from the kinetics and potential energy, thus some considerations are done to simplify the analysis. m_a is the half of the mass of the input link and is supposed to be concentrated at two points (A and B), I is the axial moment of inertia of the input shaft (and the half of the input link), m_b is the half of the mass of the second link (thus m_b is supposed that is concentrated in two points, in the point B and in the point C), m_p is the mass of the end effector and is supposed being concentrated at the point P_{xyz}. Regarding that the translational kinetic energy of a rigid body is: $K_t=\frac{mv^2}{2}$ and if the rigid body is rotating

around its center of mass the kinetic energy is: $K_r = \frac{I\omega^2}{2}$, where v is the translational velocity, m is the mass of the body in the center of mass, I is the moment of inertia and ω is the body's angular velocity. Thus the total kinetic energy of the robot is (mobile platform, 3 input links and 3 input shafts, and 3 parallelogram links):

$$K=\frac{1}{2}\left[m_p\left(\dot{p}_x^2+\dot{p}_y^2+\dot{p}_z^2\right)+\left(m_a a^2+I\right)\left(\dot{\theta}_1^2+\dot{\theta}_2^2+\dot{\theta}_3^2\right)+m_b a^2\left(\dot{\theta}_1^2+\dot{\theta}_2^2+\dot{\theta}_3^2\right)+3m_b\left(\dot{p}_x^2+\dot{p}_y^2+\dot{p}_z^2\right)\right] \tag{47}$$

The Potential energy is energy depends on the elevation of the elements ($V = mgP_z$), m is the mass, g is the constant of gravity and P_z is the s the altitude of the gravitated object. In the robot the potential energy of the platform, the input links and the parallelogram links is:

$$V=g\left[m_p P_z+m_a a\left(s\left(\theta_1\right)+s\left(\theta_2\right)+s\left(\theta_3\right)\right)+m_b\left(3P_z+a\left(s\left(\theta_1\right)+s\left(\theta_2\right)+s\left(\theta_3\right)\right)\right)\right] \tag{48}$$

Therefore the Lagrangian function ($L = K - V$) is:

$$\begin{aligned} L=&\frac{1}{2}\left(m_p+3m_b\right)\left(\dot{p}_x^2+\dot{p}_y^2+\dot{p}_z^2\right)+\frac{1}{2}\left(m_a a^2+I+m_b a^2\right)\left(\dot{\theta}_1^2+\dot{\theta}_2^2+\dot{\theta}_3^2\right)\\ &-g\left(m_p+3m_b\right)P_z-a g\left(m_a+m_b\right)\left(s\left(\theta_1\right)+s\left(\theta_2\right)+s\left(\theta_3\right)\right)\end{aligned} \tag{49}$$

Taking the partial derivatives of the Lagrangian with respect to the generalized coordinates, we have.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{P}_x}\right)=\left(m_p+3m_b\right)\ddot{P}_x \qquad \frac{\partial L}{\partial P_x}=0$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{P}_y}\right)=\left(m_p+3m_b\right)\ddot{P}_y \qquad \frac{\partial L}{\partial P_y}=0$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{P}_z}\right)=\left(m_p+3m_b\right)\ddot{P}_z \qquad \frac{\partial L}{\partial P_x}=-g\left(m_p+3m_b\right)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right)=\left(m_a a^2+I+m_b a^2\right)\ddot{\theta}_1 \qquad \frac{\partial L}{\partial \theta_1}=-a g\left(m_a+m_b\right)c\left(\theta_1\right)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right)=\left(m_a a^2+I+m_b a^2\right)\ddot{\theta}_2 \qquad \frac{\partial L}{\partial \theta_2}=-a g\left(m_a+m_b\right)c\left(\theta_2\right)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_3}\right)=\left(m_a a^2+I+m_b a^2\right)\ddot{\theta}_3 \qquad \frac{\partial L}{\partial \theta_3}=-a g\left(m_a+m_b\right)c\left(\theta_3\right)$$

Taking the partial derivatives of the constraint equations (46) with respect to the generalized coordinates, we have.

$$\frac{\partial \Gamma_i}{\partial P_x} = 2\left(P_x + \left(h - H - a\,c\left(\theta_i\right)\right)\right)c\left(\phi_i\right), \qquad \text{i} = 1,2,3$$

$$\frac{\partial \Gamma_i}{\partial P_y} = 2\left(P_y + \left(h - H - a\,c\left(\theta_i\right)\right)\right)s\left(\phi_i\right), \qquad \text{i} = 1,2,3$$

$$\frac{\partial \Gamma_i}{\partial P_z} = 2\left(P_z - a\,s\left(\theta_i\right)\right), \qquad \text{i} = 1,2,3$$

$$\frac{\partial \Gamma_1}{\partial \theta_1} = 2a\left(\left(c\left(\phi_1\right)P_x + c\left(\phi_1\right)P_y + h - H\right)s\left(\theta_1\right) - P_z\,c\left(\theta_1\right)\right) \qquad \frac{\partial \Gamma_2}{\partial \theta_2} = \frac{\partial \Gamma_3}{\partial \theta_3} = 0$$

$$\frac{\partial \Gamma_2}{\partial \theta_2} = 2a\left(\left(c\left(\phi_2\right)P_x + c\left(\phi_2\right)P_y + h - H\right)s\left(\theta_2\right) - P_z\,c\left(\theta_2\right)\right) \qquad \frac{\partial \Gamma_1}{\partial \theta_1} = \frac{\partial \Gamma_3}{\partial \theta_3} = 0$$

$$\frac{\partial \Gamma_3}{\partial \theta_3} = 2a\left(\left(c\left(\phi_3\right)P_x + c\left(\phi_3\right)P_y + h - H\right)s\left(\theta_3\right) - P_z\,c\left(\theta_3\right)\right) \qquad \frac{\partial \Gamma_1}{\partial \theta_1} = \frac{\partial \Gamma_2}{\partial \theta_2} = 0$$

Once we have the derivatives above, they are substituted into equation (44) and the Lagrangian multipliers are calculated. Thus for $j = 1, 2, 3$.

$$\begin{aligned}
&2\left[\lambda_1\left(P_x + \left(h - H + a\,c\left(\theta_1\right)\right)c\left(\phi_1\right)\right) + \lambda_2\left(P_x + \left(h - H + a\,c\left(\theta_2\right)\right)c\left(\phi_2\right)\right) + \lambda_3\left(P_x + \left(h - H + a\,c\left(\theta_3\right)\right)c\left(\phi_3\right)\right)\right] = \\
&\qquad \left(m_p + 3m_b\right)\ddot{p}_x - F_{P\,x} \\
&2\left[\lambda_1\left(P_y + \left(h - H + a\,c\left(\theta_1\right)\right)s\left(\phi_1\right)\right) + \lambda_2\left(P_y + \left(h - H + a\,c\left(\theta_2\right)\right)s\left(\phi_2\right)\right) + \lambda_3\left(P_y + \left(h - H + a\,c\left(\theta_3\right)\right)s\left(\phi_3\right)\right)\right] = \\
&\qquad \left(m_p + 3m_b\right)\ddot{p}_y - F_{P\,y} \\
&2\left[\lambda_1\left(P_z - a\,s\left(\theta_1\right)\right) + \lambda_2\left(P_z - a\,s\left(\theta_2\right)\right) + \lambda_3\left(P_z - a\,s\left(\theta_3\right)\right)\right] = \left(m_p + 3m_b\right)\ddot{p}_z + g\left(m_p + 3m_b\right) - F_{P\,z}
\end{aligned} \tag{50}$$

Note that F_{Px}, F_{Py} and F_{Pz} are the components ($Q_j,\ j = 1,2,3$) of an external force that is applied on the mobile platform. Once that the Lagrange multipliers are calculated the (45) is solved (where $j = 4,5,6$) and for the actuator torques ($\tau_{ck} = Q_{k+3},\ k = 1,2,3$).

$$\begin{aligned}
\tau_{c1} &= \left(m_a a^2 + I + m_b a^2\right)\ddot{\theta}_1 + \left(m_a + m_b\right)g\,a\,c\left(\theta_1\right) - 2\,a\,\lambda_1\left[\left(P_x c\left(\phi_1\right) + P_y\,s\left(\phi_1\right) + h - H\right)s\left(\theta_1\right) - P_z\,c\left(\theta_1\right)\right] \\
\tau_{c2} &= \left(m_a a^2 + I + m_b a^2\right)\ddot{\theta}_2 + \left(m_a + m_b\right)g\,a\,c\left(\theta_2\right) - 2\,a\,\lambda_2\left[\left(P_x c\left(\phi_2\right) + P_y\,s\left(\phi_2\right) + h - H\right)s\left(\theta_2\right) - P_z\,c\left(\theta_2\right)\right] \\
\tau_{c3} &= \left(m_a a^2 + I + m_b a^2\right)\ddot{\theta}_3 + \left(m_a + m_b\right)g\,a\,c\left(\theta_3\right) - 2\,a\,\lambda_3\left[\left(P_x\,c\left(\phi_3\right) + P_y\,s\left(\phi_3\right) + h - H\right)s\left(\theta_3\right) - P_z\,c\left(\theta_3\right)\right]
\end{aligned} \tag{51}$$

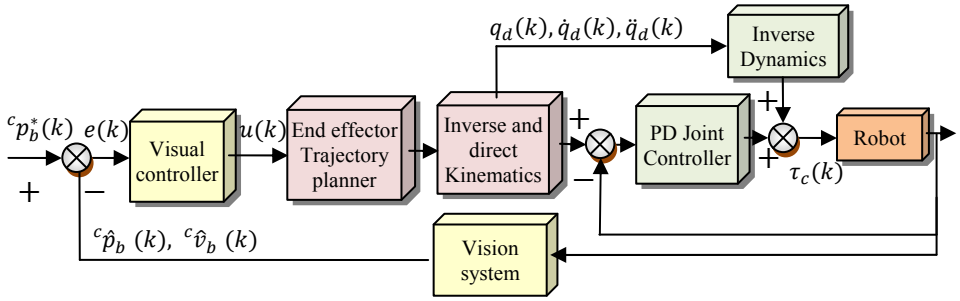


Fig. 7. Basic architecture of the control system of the Robotenis platform.

The results above are used in real time to control each joint independently. The joint controller is based in a classical computed-torque controller plus a PD controller (Santibañez and Kelly 2001). The objective of the computed-torque controller is to Feedback a signal that cancels the effects of gravity, friction, the manipulator inertia tensor, and Coriolis and centrifugal force, see in Fig. 7.

3.4 Trajectory planner

The structure of the visual controller of the Robotenis system is called dynamic position-based on a look-and-move structure (Corke 1993). The above structure is formed of two intertwined control loops: the first is faster and makes use of joints feedback, the second is external to the first one and makes use of the visual information feedback, see in Fig. 7.

Once that the visual control loop analyzes the visual information then, this is sent to the joint controller as a reference. In other words, in tracking tasks the desired next position is calculated in the visual controller and the joint controller forces to the robot to reach it. Two control loops are incorporated in the Robotenis system: the joint loop is calculated each 0.5 ms; at this point dynamical model, kinematical model and PD action are retrofitted. The external loop is calculated each 8.33 ms and it was mentioned that uses the visual data. As the internal loop is faster than the external, a trajectory planner is designed in order to accomplish different objectives: The first objective is to make smooth trajectories in order to avoid abrupt movements of the robot elements. The trajectory planner has to guarantee that the positions and its following 3 derivates are continuous curves (velocity, acceleration and jerk). The second objective is to guarantee that the actuators of the robot are not saturated and that the robot specifications are not exceeded, the robot limits are: MVS= maximum allowed velocity, MAS= maximum allowed acceleration and MJS= maximum allowed jerk (maximum capabilities of the robot are taken from the end effector). In the Robotenis system maximum capabilities are: $S = 0.8\ \frac{m}{s^3}$, $MAS = 20\ \frac{m}{s^2}$, and $MVS = 4\ \frac{m}{s}$.

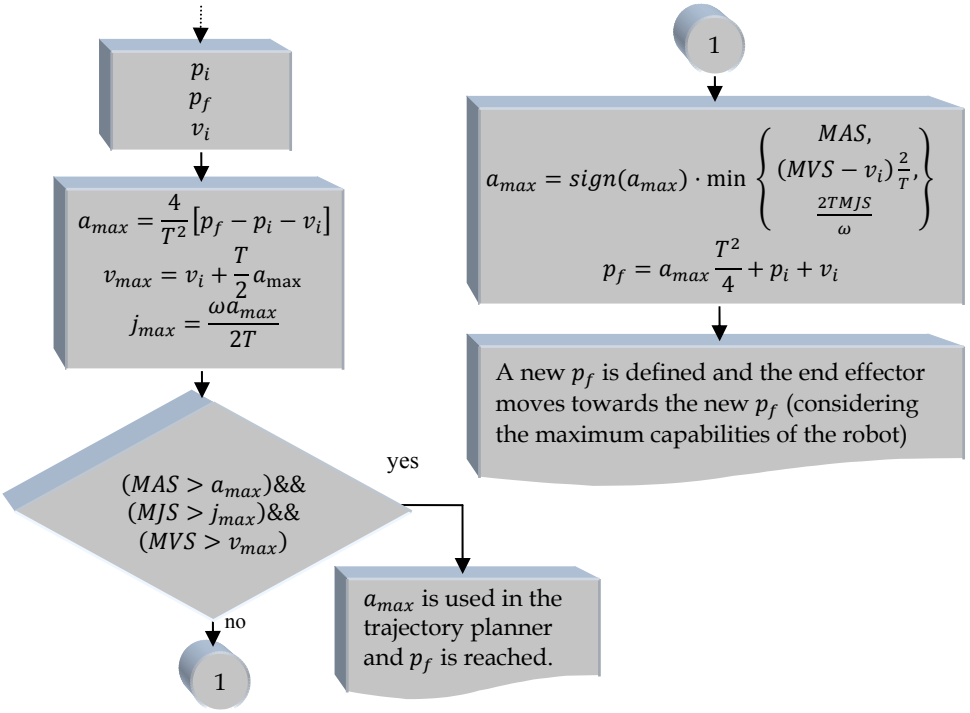


Fig. 8. Flowchart of the Trajectory planner.

And the third objective is to guarantee that the robot is in prepared to receive the next reference, in this point the trajectory planner imposes a cero jerk and acceleration at the end of each trajectory. In order to design the trajectory planner it has to be considered the system constrains, the maximum jerk and maximum acceleration. As a result we have that the jerk can be characterized by:

$$j(\tau) = \frac{k}{T^3}\text{sen}(\omega\tau) = j_{\max}\text{sen}(\omega\tau) \tag{52}$$

Where j_{max} is the maximum allowed jerk, $\omega = 2\pi$, $\tau \in [0,1] = \frac{t-t_i}{t_f-t_i}$, t is the real time clock, t_i and t_f represent the initial and final time of the trajectory. Supposing that the initial and final acceleration are cero and by considering that the acceleration can be obtained from the integral of the eq. (52) and that if $T = 8.333\,ms = t_f - t_i$ then $\frac{d\tau}{dt} = \frac{1}{T}$ we have:

$$a(\tau) = \frac{T j_{\max}}{\omega}[1 - \cos(\omega\tau)] \tag{53}$$

By supposing that the initial velocity (v_i) is different of cero, the velocity can be obtained from the convenient integral of the eq. (53).

$$v(\tau) = v_i + \frac{T^2 j_{max}}{\omega}\left[\tau - \frac{\sin(\omega\tau)}{\omega}\right] \tag{54}$$

Finally, supposing p_i as the initial position and integrating the eq. (54) to obtain the position:

$$p(\tau) = p_i + T v_i \tau + \frac{T^3 j_{max}}{\omega}\left[\frac{\tau^2}{2} + \frac{\cos(\omega\tau)}{\omega^2} - \frac{1}{\omega^2}\right] \tag{55}$$

We can see that the final position p_f is not defined in the eq. (**55**). p_f is obtained by calculating not to exceed the maximum jerk and the maximum acceleration. From eq. (53) the maximum acceleration can be calculated as:

$$a_{max} = \frac{T j_{max}}{\omega}\left[1 - \cos(\omega\tau)\right]\Big|_{\tau=\frac{1}{2}} \quad \Rightarrow \quad j_{max} = \frac{\omega a_{max}}{2T} \tag{56}$$

The final position of the eq. (55) is reached when $\tau = 1$, thus substituting eq. (56) in eq. (55) when $\tau = 1$, we have:

$$a_{max} = \frac{4}{T^2}\left[p_f - p_i - T v_i\right] \tag{57}$$

By means of the eq. (57) a_{max} can be calculated but in order to take into account the maximum capabilities of the robot. Maximum capabilities of the robot are the maximum speed, acceleration and jerk. By substituting the eq. (56) in (54) and operating, we can obtain the maximum velocity in terms of the maximum acceleration and the initial velocity.

$$v_{max} = v_i + \frac{T^2 j_{max}}{\omega}\left[\tau - \frac{\sin(\omega\tau)}{\omega}\right]\Big|_{\tau=1} = v_i + \frac{T a_{max}}{2} \tag{58}$$

Once we calculate a_{max} from eq. (57) the next is comparing the maximum capabilities from equations (56) and (58). If maximum capabilities are exceeded, then the final position of the robot is calculated from the maximum capabilities and the sign of a_{max} (note that in this case the robot will not reach the desired final position). See the Fig. 8. The time history of sample trajectories is described in the Fig. 9 (in order to plot in the same chart, all curves are normalized). This figure describes when the necessary acceleration to achieve a target, is bigger than the maximum allowed. It can be observed that the fifth target position (83:3ms) is not reached but the psychical characteristics of the robot actuators are not exceeded.

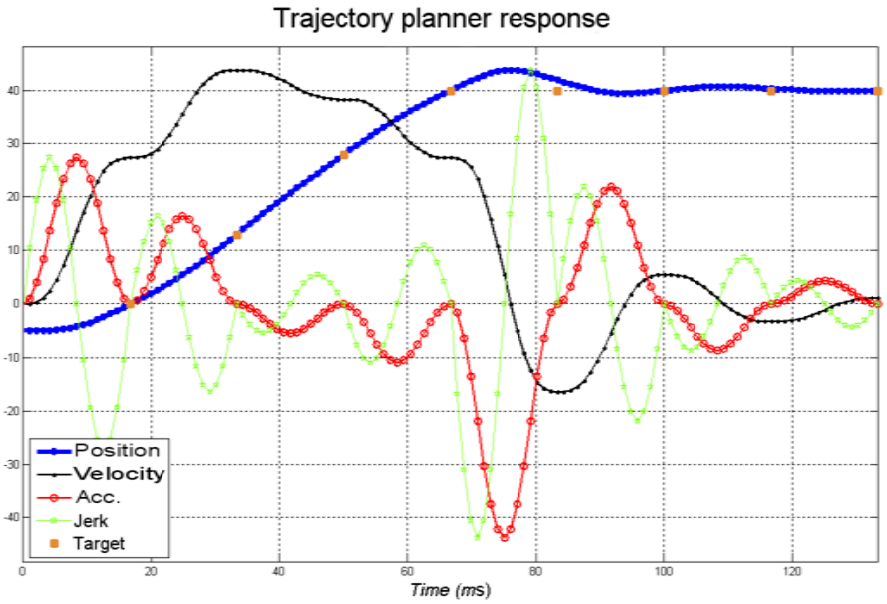


Fig. 9. Example of the time response of the trajectory planner, note that the target position is not reached (when $t = 83{:}3ms$) but robot capabilities are not exceeded.

4. Description of the visual controller.

Coordinated systems are shown in the Fig. 10 and are $\S_w$, $\S_e$, and $\S_c$ which represent the word coordinate system, the end-effector robot system and the camera coordinate system respectively. Other notations defined are: ${}^{c}p_b$ represents the position of the ball in the camera coordinate system, ${}^{w}p_e$ represents the position of the robot end effector in the word coordinate system. ${}^{w}p_e$ is obtained by means of the direct kinematical model. Transformation matrices are ${}^{w}R_e$, ${}^{w}R_c$ and ${}^{e}R_c$ where ${}^{w}R_e$ is calculated from the kinematical model and ${}^{e}R_c$ is obtained from the camera calibration. The position of the ball is calculated by means of the mass center of the projection of the ball on the image ( ) and by means of the diameter of the ball ( ). Diameter of the ball is principally critical and its calculation requires sub-pixel precision techniques.

Although there are advanced controllers that have been proposed by (Chaumette and Hutchinson, Visual servo control. II Advanced approaches 2007), the controller selected is based in position (Chaumette and Hutchinson, Visual servo control. I. Basic approaches 2006). Schematic control can be appreciated in the Fig. 7 and Fig. 10, the error function is obtained though the difference between the reference position ( ) and the measured position ( ). In the present article the control signal is obtained as a result of considering when desired position is fixed (static case) and when it is variable (dynamic case). Once the error is obtained, the controller calculates the desired velocity of the end effector. By means of the trajectory planner and the Jacobian matrix, all the joint motions are

calculated. Signals are supposed as known in the instant kT, where T is the sample time in the visual control loop (in order to simplify we suppose kT as k).

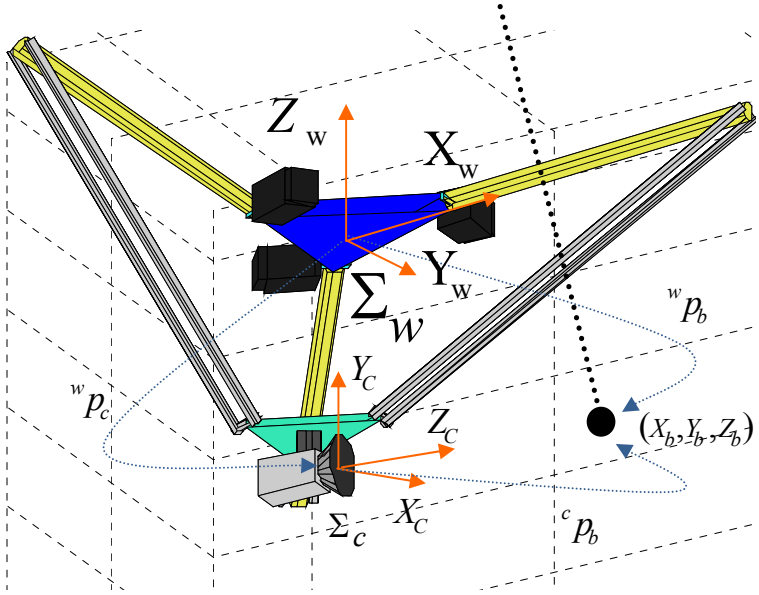


Fig. 10. Coordinated systems that are considered in the controller.

4.1 Static case

In the Fig. 7 can be observed that the position error can be expressed as follows:

$$e(k)={}^{c}p_{b}^{*}-{}^{c}p_{b}(k) \tag{59}$$

In this section ${}^{c}p_{b}^{*}(k)$ is the desired position of the ball in the camera coordinate system and in this section is considered as constant and known. ${}^{c}p_{b}\ (k)$ is the position of the ball in the camera coordinate system. Thus, by considering the position in the word coordinate system:

$${}^{c}p_{b}(k)={}^{c}R_{w}\left({}^{w}p_{b}(k)-{}^{w}p_{c}(k)\right) \tag{60}$$

If **(60)** is substituted into (**59**) then we obtain:

$$e(k)={}^{c}p_{b}^{*}-{}^{c}R_{w}({}^{w}p_{b}(k)-{}^{w}p_{c}(k)) \tag{61}$$

The system (robot) is supposed stable and in order to guarantee that the error will decrease exponentially we choose:

$$\dot{e}(k) = -\lambda e(k) \quad where \quad \lambda > 0 \tag{62}$$

Deriving **(60)** and supposing that ${}^{c}p_{b}^{*}$ and ${}^{c}R_{w}$ are constant, we obtain:

$$\dot{e}(k) = -{}^{c}R_{w}\left({}^{w}v_{b}(k) - {}^{w}v_{c}(k)\right) \tag{63}$$

Substituting **(61)**and **(63)**into **(62)**, we obtain:

$${}^{w}v_{c}(k) = {}^{w}v_{b}(k) - \lambda\, {}^{c}R_{w}^{T}\left({}^{c}p_{b}^{*} - {}^{c}p_{b}(k)\right) \tag{64}$$

Where ${}^{w}v_{c}$ and ${}^{w}v_{b}$ represent the camera and ball velocities (in the word coordinate system) respectively. Since $\left[{}^{w}v_{e}\ (k) = {}^{w}v_{c}\ (k)\right]$ the control law can be expressed as:

$$u(k) = {}^{w}v_{b}(k) - \lambda\, {}^{c}R_{w}^{T}\left[{}^{c}p_{b}^{*} - {}^{c}p_{b}(k)\right] \tag{65}$$

The equation (65) is composed by two components: a component which predicts the position of the ball (${}^{w}v_{b}\ (k)$) and the other contains the tracking error ($\left[{}^{c}p_{b}^{*} - {}^{c}p_{b}\ (k)\right]$).The ideal control scheme (65) requires a perfect knowledge of all its components, which is not possible, a more realistic approach consist in generalizing the previous control as

$$u(k) = {}^{w}\hat{v}_{c}(k) = {}^{w}\hat{v}_{b}(k) - \lambda\, {}^{c}R_{w}^{T}\left[{}^{c}p_{b}^{*} - {}^{c}\hat{p}_{b}(k)\right] \tag{66}$$

Where, the estimated variables are represented by the carets. A fundamental aspect in the performing of the visual controller is the adjustment of λ, therefore λ will be calculated in the less number of sample time periods and will consider the system limitations. This algorithm is based in the future positions of the camera and the ball; this lets to the robot reaching the control objective ($\left[{}^{c}p_{b}\ (k) = {}^{c}p_{b}^{*}\right]$). By supposing "n" small, the future position (in the $k+n$ instant) of the ball and the camera in the word coordinate system are:

$${}^{w}\hat{p}_{b}(k+n) = {}^{w}\hat{p}_{b} + {}^{w}\hat{v}_{b}(k)\, n\, T \tag{67}$$

$${}^{w}p_{c}(k+n) = {}^{w}p_{c} + {}^{w}v_{c}(k)\, n\, T \tag{68}$$

Where T is the visual sample time period ($8.333\ ms$). As was mentioned, the control objective is to reach the target position in the shorter time as be possible. By taking into account eq. (61), the estimated value ${}^{w}\hat{p}_{b}$ and by considering that the error is cero $e = 0$ in the instant $k+n$, we have:

$${}^{c}p_{b}^{*} - {}^{c}R_{w}\left[{}^{w}\hat{p}_{b}(k+n) - {}^{w}p_{c}(k+n)\right] = 0 \tag{69}$$

Substituting (67) and (68) into (69), we obtain (70).

$$ {}^{c}p_{b}^{*}={}^{c}R_{w}\left[{}^{w}\hat{p}_{b}(k)+{}^{w}\hat{v}_{b}(k)nT-{}^{w}p_{c}(k)-{}^{w}v_{c}(k)nT\right] \tag{70} $$

Taking into account that the estimate of the velocity of the ball is:

$$ {}^{c}\hat{p}_{b}(k)={}^{c}R_{w}\left({}^{w}\hat{p}_{b}(k)-{}^{w}p_{c}(k)\right) \tag{71} $$

Then the control law can be expressed as:

$$ u(k)={}^{w}\hat{v}_{c}(k)={}^{w}\hat{v}_{b}(k)-\frac{1}{nT}{}^{c}R_{w}^{T}\left[{}^{c}p_{b}^{*}-{}^{c}\hat{p}_{b}(k)\right] \tag{72} $$

If (66) and (72) are compared, we can obtain the λ parameter as:

$$ \lambda=\frac{1}{nT} \tag{73} $$

The equation (73) gives a criterion for adjust λ as a function of the number of samples required (n) for reaching the control target. The visual control architecture proposed above does not consider the physical limitations of the system such as delays and the maximum operation of the components. If we consider that the visual information (${}^{c}\hat{p}_{b}\ (k)$) has a delay of 2 sampling times $(r=2)$ with respect to the joint information, then at an instant $k+n$, the future position of the ball can be:

$$ {}^{w}\hat{p}_{b}(k+n)={}^{w}\hat{p}_{b}(k-r)+{}^{w}\hat{v}_{b}(k-r)T\left[n+r\right] \tag{74} $$

The future position of the camera in the word coordinate system is given by (68). Using the (74) is possible to adjust the λ for the control law by considering the following aspect:
-The wished velocity of the end effector is represented by (72). In physical systems the maximal velocity is necessary to be limited. In our system the maximal velocity of each joint is taken into account to calculate λ. Value of λ depends of the instant position of the end effector. Therefore through the robot jacobian is possible to know the velocity that requires moving each joint and the value of λ is adjusted to me more constrained joint (maximal velocity of the joint).

4.2 Dynamic case

Static case is useful when the distance between the ball and the camera must be fixed but in future tasks it is desirable that this distance change in real time. In this section, in order to carry out above task a dynamic visual controller is designed. This controller receives two parameters as are the target position and the target velocity. By means of above parameters the robot can be able to carry out several tasks as are: catching, touching or hitting objects

that are static or while are moving. In this article the principal objective is the robot hits the ball in a specific point and with a specific velocity. In this section ${}^{c}p_{b}^{*}$ is no constant and ${}^{c}p_{b}^{*}(k)$ is considered instead, ${}^{c}v_{b}^{*}(k)$ is the relative target velocity between the ball and the camera and the error between the target and measured position is expressed as:

$$e(k)={}^{c}p_{b}^{*}(k)-{}^{c}p_{b}(k) \tag{75}$$

Substituting (60) in (75) and supposing that only ${}^{c}R_{w}$ is constant, we obtain its derivate as:

$$\dot{e}(k)={}^{c}v_{b}^{*}(k)-{}^{c}R_{w}\left({}^{w}v_{b}(k)-{}^{w}v_{c}(k)\right) \tag{76}$$

Where ${}^{c}v_{b}^{*}(k)$ is considered as the target velocity to carry out the task. By following a similar analysis that in the static case, our control law would be:

$$u(k)={}^{w}\hat{v}_{c}(k)={}^{w}\hat{v}_{b}(k)-{}^{c}R_{w}^{T}\left[\lambda\left({}^{c}p_{b}^{*}(k)-{}^{c}\hat{p}_{b}(k)\right)+{}^{c}v_{b}^{*}(k)\right] \tag{77}$$

Where ${}^{c}\hat{p}_{b}(k)$ and ${}^{w}\hat{v}_{b}(k)$ are estimated and are the position and the velocity of the ball. Just as to the static case, from the eq. (61) λ is calculated if the error is cero in $k+n$.

$$0={}^{c}p_{b}^{*}(k+n)-{}^{c}R_{w}\left({}^{w}\hat{p}_{b}(k+n)-{}^{w}p_{c}(k+n)\right) \tag{78}$$

Substituting (67) and (68) in (78) and taking into account the approximation:

$${}^{c}p_{b}^{*}(k+n)\cong{}^{c}p_{b}^{*}(k)+nT\,{}^{c}v_{b}^{*}(k) \tag{79}$$

Is possible to obtain:

$$0={}^{c}p_{b}^{*}(k)+n\,T\,{}^{c}v_{b}^{*}(k)-{}^{c}R_{w}\left({}^{w}\hat{p}_{b}(k)+nT\,{}^{w}\hat{v}_{b}(k)-{}^{w}p_{c}(k)-n\,T\,{}^{w}v_{c}(k)\right) \tag{80}$$

Taking into account the eq. (71), the control law can be obtained as:

$$u(k)={}^{w}\hat{v}_{c}(k)={}^{w}\hat{v}_{b}(k)-{}^{c}R_{w}^{T}\left(\frac{1}{n\,T}\left({}^{c}p_{b}^{*}(k)-{}^{c}\hat{p}_{b}(k)\right)+{}^{c}v_{b}^{*}(k)\right) \tag{81}$$

From eq. (77) it can be observed that λ can be $\lambda=\frac{1}{nT}$ where "n" is "small enough".

4.3 Stability and errors influence

By means of Lyapunov analysis is possible to probe the system stability; it can be demonstrated that the error converges to zero if ideal conditions are considered; otherwise it

can be probed that the error will be bounded under the influence of the estimation errors and non modelled dynamics. We choose a Lyapunov function as:

$$V=\frac{1}{2}e^T(k)e(k) \tag{82}$$

$$\dot{V}=e^T(k)\dot{e}(k) \tag{83}$$

If the error behavior is described by the eq. (62) then

$$\dot{V}=-e^T(k)\lambda\, e(k)<0 \tag{84}$$

The eq. (84) implies $e(k)\to 0$ when $k\to\infty$ and this is only true if $u(k)={}^w v_c(k)$. Note that above is not true due to estimations (${}^w\hat{v}_b(k)$, ${}^c\hat{p}_b(k)$) and dynamics that are not modelled. Above errors are expressed in $\rho(k)$ and is more realistic to consider ($u(k)={}^w\hat{v}_c(k)$):

$$u(k)={}^w\hat{v}_c(k)={}^w v_c(k)+\rho(k) \tag{85}$$

By considering the estimated velocity of the ball (${}^w\hat{v}_b$) in eq. (76) and substituting the eq. (85) is possible to obtain:

$$\dot{e}(k)={}^c v_b^*(k)-{}^c R_w\left[{}^w v_b(k)-{}^w v_c(k)-\rho(k)\right] \tag{86}$$

Note that estimate errors are already included in ρ. Consequently the value of ${}^w v_c(k)$ is:

$${}^w v_c(k)={}^w v_b(k)-{}^c R_w^T\left[\lambda\left({}^c p_b^*(k)-{}^c p_b(k)\right)+{}^c v_b^*(k)\right] \tag{87}$$

Substituting eq. (87) in (86):

$$\dot{e}(k)={}^c v_b^*(k)-{}^c R_w\left[{}^w v_b(k)-{}^w v_b(k)+{}^c R_w^T\left[\lambda\left({}^c p_b^*(k)-{}^c p_b(k)\right)+{}^c v_b^*(k)\right]-\rho(k)\right] \tag{88}$$

Operating in order to simplify:

$$\dot{e}(k)={}^c v_b^*(k)-{}^c v_b^*(k)-\lambda\left({}^c p_b^*(k)-{}^c p_b(k)\right)+{}^c R_w\rho(k)={}^c R_w\rho(k)-\lambda e(k) \tag{89}$$

Taking into account the Lyapunov function in eq. (82):

$$\dot{V}=e^T(k)\dot{e}(k)=-e^T(k)\lambda e(k)+e^T(k){}^c R_w\rho(k) \tag{90}$$

Thus, by considering $\dot{V}<0$ we have that the following condition has to be satisfied:

$$\|e\| > \frac{\|\rho\|}{\lambda} \tag{91}$$

Above means that if the error is bigger that $\frac{\|\rho\|}{\lambda}$ then the error will decrease but it will not tend to cero, finally the error is bounded by.

$$\|e\| < \frac{\|\rho\|}{\lambda} \tag{92}$$

By considering that errors from the estimation of the position and velocity are bigger that errors from the system dynamics, then $\rho(k)$ can be obtained if we replace (77) and (87) in (85)

$$\rho(k) = {}^{w}\hat{v}_b(k) - {}^{w}v_b(k) + {}^{c}R_w^T \lambda \left({}^{c}\hat{p}_b(k) - {}^{c}p_b(k) \right) \tag{93}$$

5. Conclusions and future works

In this work the full architecture of the Robotenis system and a novel structure of visual control were shown in detail. In this article no results are shown but the more important elements to control and simulate the robot and visual controller were described. Two kinematic models were described in order to obtain two different jacobians were each jacobian is used in different tasks: the System simulator and the real time controller. By means of the condition index of the robot jacobians some singularities of the robot are obtained. In real time tasks the above solution and the condition index of the second jacobian are utilized to bound the work space and avoid singularities, in this work if some point forms part or is near of a singularity then the robot stop the end effector movement and waits to the next target point.

Inverse dynamics of the robot is obtained by means of the Lagrange multipliers. The inverse dynamics is used in a non linear feed forward in order to improve the PD joint controller. Although improvement of the behaviour of the robot in notorious, in future works is important to measure how the behaviour is modified when the dynamics fed forward is added and when is not.

The trajectory planner is added with two principal objectives: the trajectory planner assures that the robot capabilities are not exceeded and assures that the robot moves softly. The trajectory planner takes into account the movements of the end effector, this consideration has drawbacks: the principal is that the maximum end-effector capabilities are not necessarily the maximum joint capabilities, depends on the end effector position. Above drawbacks suggest redesigning the trajectory planner in order to apply to the joint space, this as another future work.

Above elements are used in the visual controller and the robot controller has to satisfied the visual controller requirements. Thanks to the joint controller the robot is supposed stable and its response is considered faster than the visual system. Two cases are presented in this paper: the static case that is exposed in other works and some results and videos are shown, the another controller is called the dynamic case. An objective of the system is to play ping pong by itself and the controller of the dynamic case was specially designed in order to

reach this objective. The objective of the dynamic visual controller is to reach some point with a desired velocity, this allows to the robot hit the ball with a desired speed and direction. In order to hit the ball a special and partially spherical paddled is being designed in order to give the desired effect to the ball. Finally the stability of visual controllers is demonstrated by means of Lyapunov theory and the errors in the estimations are bounded. As a future works, efforts of the vision group will be concentrated in the design of visual controllers in order to improve the robot positioning and tracking.

6. References

Angel, L., J.M. Sebastian, R. Saltaren, R. Aracil, and J. Sanpedro. "Robotenis: optimal design of a parallel robot with high performance." *IEEE/RSJ International Conference on, (IROS 2005).* IEEE Intelligent Robots and Systems, 2005. 2134- 2139.

Bonev, Ilian A., and Clément Gosselin. *Fundamentals of Parallel Robots.* Edited by Springer. 2009.

Chaumette, F., and S. Hutchinson. "Visual servo control. I. Basic approaches." (Robotics & Automation Magazine, IEEE) 13, no. 4 (December 2006): 82-90.

Chaumette, F., and S. Hutchinson. "Visual servo control. II Advanced approaches." (Robotics & Automation Magazine, IEEE) 14, no. 1 (March 2007): 109 – 118.

Clavel, Reymond. "DELTA: a fast robot with parallel geometry." Sidney: 18th International Symposium on Industrial Robot., 1988. 91–100.

Corke, Peter I. "Visual Control Of Robot Manipulators -- A Review." In *Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback (Series in Robotics and Automated Systems)*, edited by Hashimoto Kagami, 1-31, 300. World Scientific Publishing Co Pte Ltd, 1993.

Corrochano, Eduardo Bayro, and Detlef Kähler. "Motor Algebra Approach for Computing the Kinematics of Robot Manipulators." *Journal of Robotic Systems* (Wiley Periodicals), 2000: 495 - 516.

Davidson, J. K., and J. K. Hunt Davidson. *Robots and Screw Theory: Applications of Kinematics and Statics to Robotics* . 1. Publisher: Oxford University Press, USA, 2004.

Kaneko, Makoto, Mitsuru Higashimori, Akio Namiki, and Masatoshi Ishikawa. "The 100G Capturing Robot - Too Fast to See." Edited by P. Dario and R. Chatila. Robotics Research, 2005. 517–526.

Kragic, Danica, and Christensen Henrik I. "Advances in robot vision." Vol. 52. Edited by Elsevier. Robotics and Autonomous Systems, Science Direct, May 2005. 1-3.

Merlet, J.P. *Parallel Robots (Solid Mechanics and Its Applications).* Edited by Springer. 2006.

Morikawa, Sho, Taku Senoo, Akio Namiki, and Masatoshi Ishikawa. "Realtime collision avoidance using a robot manipulator with light-weight small high-speed vision systems." Roma: Robotics and Automation IEEE International Conference on, April 2007. 794-797.

Oda, Naoki, Masahide Ito, and Masaaki Shibata. "Vision-based motion control for robotic systems." Vol. 4. no. 2. Edited by Hoboken. John Wiley. February 2009.

Santibañez, Victor, and Rafael. Kelly. "PD control with feedforward compensation for robot manipulators: analysis and experimentation." *Robotica* (Cambridge University Press) 19, no. 1 (2001): 11-19.

Sebastián, J.M., A. Traslosheros, L. Angel, F. Roberti, and R. Carelli. "Parallel robot high speed objec tracking." Chap. 3, by Image Analysis and recognition, edited by Aurélio Campilho Mohamed Kamel, 295-306. Springer, 2007.

Senoo, T., A. Namiki, and M. Ishikawa. "High-speed batting using a multi-jointed manipulator." Vol. 2. Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2004. 1191- 1196 .

Stamper, Richard Eugene, and Lung Wen Tsai. "A three Degree of freedom parallel manipulator with only translational degrees of freedom." PhD Thesis, Department of mechanical engineering and institute for systems research, University of Maryland, 1997, 211.

Stramigioli, Stefano, and Herman Bruyninckx. *Geometry and Screw Theory for Robotics (Tutorial).* Tutorial, IEEE ICRA 2001, 2001.

Tsai, Lung Wen. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators.* 1. Edited by Wiley-Interscience. 1999.

Yoshikawa, Tsuneo. "Manipulability and Redundancy Ccontrol of Robotic Mechanisms." Vol. 2. Robotics and Automation. Proceedings. 1985 IEEE International Conference on, March 1985. 1004- 1009.

9

Nonlinear Adaptive Model Following Control for a 3-DOF Model Helicopter

Mitsuaki Ishitobi and Masatoshi Nishi
Department of Mechanical Systems Engineering
Kumamoto University
Japan

1. Introduction

Interest in designing feedback controllers for helicopters has increased over the last ten years or so due to the important potential applications of this area of research. The main difficulties in designing stable feedback controllers for helicopters arise from the nonlinearities and couplings of the dynamics of these aircraft. To date, various efforts have been directed to the development of effective nonlinear control strategies for helicopters (Sira-Ramirez *et al.*, 1994; Kaloust *et al.*, 1997; Kutay *et al.*, 2005; Avila *et al.*, 2003). Sira-Ramirez *et al.* applied dynamical sliding mode control to the altitude stabilization of a nonlinear helicopter model in vertical flight. Kaloust *et al.* developed a Lyapunov-based nonlinear robust control scheme for application to helicopters in vertical flight mode. Avila *et al.* derived a nonlinear 3-DOF (degree-of-freedom) model as a reduced-order model for a 7-DOF helicopter, and implemented a linearizing controller in an experimental system. Most of the existing results have concerned flight regulation.

This study considers the two-input, two-output nonlinear model following control of a 3-DOF model helicopter. Since the decoupling matrix is singular, a nonlinear structure algorithm (Shima *et al.*, 1997; Isurugi, 1990) is used to design the controller. Furthermore, since the model dynamics are described linearly by unknown system parameters, a parameter identification scheme is introduced in the closed-loop system.

Two parameter identification methods are discussed: The first method is based on the differential equation model. In experiments, it is found that this model has difficulties in obtaining a good tracking control performance, due to the inaccuracy of the estimated velocity and acceleration signals. The second parameter identification method is designed on the basis of a dynamics model derived by applying integral operators to the differential equations expressing the system dynamics. Hence this identification algorithm requires neither velocity nor acceleration signals. The experimental results for this second method show that it achieves better tracking objectives, although the results still suffer from tracking errors. Finally, we introduce additional terms into the equations of motion that express model uncertainties and external disturbances. The resultant experimental data show that the method constructed with the inclusion of these additional terms produces the best control performance.

2. System Description

Consider the tandem rotor model helicopter of Quanser Consulting, Inc. shown in Figs. 1 and 2. The helicopter body is mounted at the end of an arm and is free to move about the elevation, pitch and horizontal travel axes. Thus the helicopter has 3-DOF: the elevation ε, pitch θ and travel ϕ angles, all of which are measured via optical encoders. Two DC motors attached to propellers generate a driving force proportional to the voltage output of a controller.

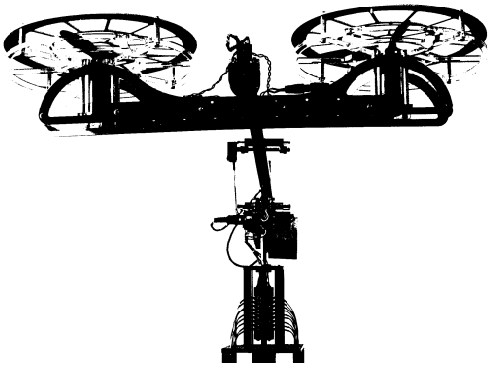


Fig. 1. Overview of the present model helicopter.

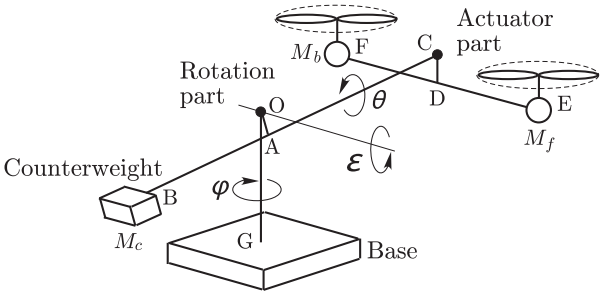


Fig. 2. Notation.

The equations of motion about axes ε, θ and ϕ are expressed as

$$\begin{aligned} J_\varepsilon \ddot{\varepsilon} &= -\left(M_f + M_b\right) g \frac{L_a}{\cos\delta_a} \cos\left(\varepsilon - \delta_a\right) + M_c g \frac{L_c}{\cos\delta_c} \cos\left(\varepsilon + \delta_c\right) - \eta_\varepsilon \dot{\varepsilon} \\ &\quad + K_m L_a \left(V_f + V_b\right) \cos\theta \end{aligned} \tag{1}$$

$$J_\theta \ddot{\theta} = -M_f g \frac{L_h}{\cos\delta_h} \cos\left(\theta - \delta_h\right) + M_b g \frac{L_h}{\cos\delta_h} \cos\left(\theta + \delta_h\right) - \eta_\theta \dot{\theta} + K_m L_h \left(V_f - V_b\right) \tag{2}$$

$$J_\phi \ddot{\phi} = -\eta_\phi \dot{\phi} - K_m L_a \left(V_f + V_b\right) \sin\theta. \tag{3}$$

A complete derivation of this model is presented in (Apkarian, 1998). The system dynamics are expressed by the following highly nonlinear and coupled state variable equations

$$\dot{\boldsymbol{x}}_p = f(\boldsymbol{x}_p) + [g_1(\boldsymbol{x}_p),\ g_2(\boldsymbol{x}_p)]\boldsymbol{u}_p \tag{4}$$

where

$$\begin{aligned}
\boldsymbol{x}_p &= [x_{p1},\ x_{p2},\ x_{p3},\ x_{p4},\ x_{p5},\ x_{p6}]^T \\
&= [\varepsilon,\ \dot{\varepsilon},\ \theta,\ \dot{\theta},\ \phi,\ \dot{\phi}]^T \\
\boldsymbol{u}_p &= [u_{p1},\ u_{p2}]^T \\
u_{p1} &= V_f + V_b \\
u_{p2} &= V_f - V_b \\
f(\boldsymbol{x}_p) &= \begin{bmatrix} \dot{\varepsilon} \\ p_1 \cos\varepsilon + p_2 \sin\varepsilon + p_3\dot{\varepsilon} \\ \dot{\theta} \\ p_5 \cos\theta + p_6 \sin\theta + p_7\dot{\theta} \\ \dot{\phi} \\ p_9\dot{\phi} \end{bmatrix} \\
g_1(\boldsymbol{x}_p) &= [0,\ p_4 \cos\theta,\ 0,\ 0,\ 0,\ p_{10} \sin\theta]^T \\
g_2(\boldsymbol{x}_p) &= [0,\ 0,\ 0,\ p_8,\ 0,\ 0]^T \\
p_1 &= \Big[-(M_f + M_b)gL_a + M_c g L_c\Big] \Big/ J_\varepsilon \\
p_2 &= -\Big[(M_f + M_b)gL_a \tan\delta_a + M_c g L_c \tan\delta_c\Big] \Big/ J_\varepsilon \\
p_3 &= -\eta_\varepsilon / J_\varepsilon \\
p_4 &= K_m L_a / J_\varepsilon \\
p_5 &= (-M_f + M_b)gL_h \Big/ J_\theta \\
p_6 &= -(M_f + M_b)gL_h \tan\delta_h \Big/ J_\theta \\
p_7 &= -\eta_\theta / J_\theta \\
p_8 &= K_m L_h / J_\theta \\
p_9 &= -\eta_\phi / J_\phi \\
p_{10} &= -K_m L_a / J_\phi \\
\delta_a &= \tan^{-1}\{(L_d + L_e)/L_a\} \\
\delta_c &= \tan^{-1}(L_d / L_c) \\
\delta_h &= \tan^{-1}(L_e / L_h)
\end{aligned}$$

The notation employed above is defined as follows: V_f, V_b [V]: Voltage applied to the front motor, voltage applied to the rear motor,
M_f, M_b [kg]: Mass of the front section of the helicopter, mass of the rear section,
M_c [kg]: Mass of the counterbalance,
L_d, L_c, L_a, L_e, L_h [m]: Distances OA, AB, AC, CD, DE=DF,
g [m/s^2]: gravitational acceleration,

J_ε, J_θ, J_ϕ [kg·m^2]: Moment of inertia about the elevation, pitch and travel axes,
η_ε, η_θ, η_ϕ [kg·m^2/s]: Coefficient of viscous friction about the elevation, pitch and travel axes.
The forces of the front and rear rotors are assumed to be F_f=$K_m V_f$ and F_b=$K_m V_b$ [N], respectively, where K_m [N/V] is a force constant. It may be noted that all the parameters p_i $(i = 1 \ldots 10)$ are constants. For the problem of the control of the position of the model helicopter, two angles, the elevation ε and the travel ϕ angles, are selected as the outputs from the three detected signals of the three angles. Hence, we have

$$\boldsymbol{y}_p = [\varepsilon,\ \phi]^T \tag{5}$$

3. Nonlinear Model Following Control

3.1 Control system design

In this section, a nonlinear model following control system is designed for the 3-DOF model helicopter described in the previous section.
First, the reference model is given as

$$\begin{cases} \dot{\boldsymbol{x}}_M = A_M \boldsymbol{x}_M + B_M \boldsymbol{u}_M \\ \boldsymbol{y}_M = C_M \boldsymbol{x}_M \end{cases} \tag{6}$$

where

$$\begin{aligned}
\boldsymbol{x}_M &= [x_{M1},\ x_{M2},\ x_{M3},\ x_{M4},\ x_{M5},\ x_{M6},\ x_{M7},\ x_{M8}]^T \\
\boldsymbol{y}_M &= [\varepsilon_M,\ \phi_M]^T \\
\boldsymbol{u}_M &= [u_{M1},\ u_{M2}]^T \\
A_M &= \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} \\
K_i &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k_{i1} & k_{i2} & k_{i3} & k_{i4} \end{bmatrix},\ i = 1, 2 \\
B_M &= \begin{bmatrix} \boldsymbol{i}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{i}_1 \end{bmatrix} \\
C_M &= \begin{bmatrix} \boldsymbol{i}_2{}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{i}_2{}^T \end{bmatrix} \\
\boldsymbol{i}_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},\ \boldsymbol{i}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}$$

From (4) and (6), the augmented state equation is defined as follows.

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u} \tag{7}$$

where

$$\begin{aligned} \boldsymbol{x} &= [\boldsymbol{x}_p^T,\ \boldsymbol{x}_M^T]^T \\ \boldsymbol{u} &= [\boldsymbol{u}_p^T,\ \boldsymbol{u}_M^T]^T \\ f(\boldsymbol{x}) &= \begin{bmatrix} f(\boldsymbol{x}_p) \\ A_M \boldsymbol{x}_M \end{bmatrix} \\ G(\boldsymbol{x}) &= \begin{bmatrix} g_1(\boldsymbol{x}_p) & g_2(\boldsymbol{x}_p) & O \\ \mathbf{0} & \mathbf{0} & B_M \end{bmatrix} \end{aligned}$$

Here, we apply a nonlinear structure algorithm to design a model following controller (Shima *et al*., 1997; Isurugi, 1990). New variables and parameters in the following algorithm are defined below the input (19).

- Step 1

The tracking error vector is given by

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x_{M1} - x_{p1} \\ x_{M5} - x_{p5} \end{bmatrix} \tag{8}$$

Differentiating the tracking error (8) yields

$$\begin{aligned} \dot{\boldsymbol{e}} &= \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}} \{f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u}\} \\ &= \begin{bmatrix} -x_{p2} + x_{M2} \\ -x_{p6} + x_{M6} \end{bmatrix} \end{aligned} \tag{9}$$

Since the inputs do not appear in (9), we proceed to step 2.

- Step 2

Differentiating (9) leads to

$$\ddot{\boldsymbol{e}} = \frac{\partial \dot{\boldsymbol{e}}}{\partial \boldsymbol{x}} \{f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u}\} \tag{10}$$

$$= \begin{bmatrix} r_1(\boldsymbol{x}) \\ -p_9 x_{p6} + x_{M7} \end{bmatrix} + [B_u(\boldsymbol{x}),\ B_r(\boldsymbol{x})]\,\boldsymbol{u} \tag{11}$$

where

$$B_u(\boldsymbol{x}) = \begin{bmatrix} -p_4 \cos x_{p3} & 0 \\ -p_{10} \sin x_{p3} & 0 \end{bmatrix},\ B_r(\boldsymbol{x}) = O$$

From (11), the decoupling matrix $B_u(\boldsymbol{x})$ is obviously singular. Hence, this system is not decouplable by static state feedback. The equation (11) can be re-expressed as

$$\ddot{e}_1 = r_1(\boldsymbol{x}) - p_4 \cos x_{p3} u_{p1} \tag{12}$$

$$\ddot{e}_2 = -p_9 x_{p6} + x_{M7} - p_{10} \sin x_{p3} u_{p1} \tag{13}$$

then, by eliminating u_{p1} from (13) using (12) under the assumption of $u_{p1} \neq 0$, we obtain

$$\ddot{e}_2 = -p_9 x_{p6} + x_{M7} + \frac{p_{10}}{p_4} \tan x_{p3} (\ddot{e}_1 - r_1(\boldsymbol{x})) \tag{14}$$

- Step 3

Further differentiating (14) gives rise to

$$\begin{aligned} e_2^{(3)} &= \frac{\partial \ddot{e}_2}{\partial \boldsymbol{x}} \{f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u}\} + \frac{\partial \ddot{e}_2}{\partial \ddot{e}_1} e_1^{(3)} \\ &= \frac{p_{10}}{p_4} \tan x_{p3} \left\{ -x_{p2} \left(p_1 \sin x_{p1} - p_2 \cos x_{p1} \right) + p_3 (x_{M3} - r_1(\boldsymbol{x})) - x_{M4} + e_1^{(3)} \right\} \\ &\quad - \frac{p_{10}}{p_4 \cos x_{p3}} x_{p4} \left(\ddot{e}_1 - r_1(\boldsymbol{x}) \right) - p_9^2 x_{p6} + x_{M8} \\ &\quad + \left[p_{10} \sin x_{p3} (p_3 - p_9),\ 0,\ 0,\ 0 \right] \boldsymbol{u} \end{aligned} \tag{15}$$

As well as step 2, we eliminate u_{p1} from (15) using (12), and it is obtained that

$$\begin{aligned} e_2^{(3)} &= \frac{p_{10}}{p_4} \tan x_{p3} \{ p_3 x_{M3} - x_{p2} \left(p_1 \sin x_{p1} - p_2 \cos x_{p1} \right) - p_3 r_1(\boldsymbol{x}) - x_{M4} + e_1^{(3)} \\ &\quad - (p_3 - p_9) \left(\ddot{e}_1 - r_1(\boldsymbol{x}) \right) \} + x_{M8} - p_9^2 x_{p6} - \frac{p_{10}}{p_4 \cos x_{p3}} x_{p4} \left(\ddot{e}_1 - r_1(\boldsymbol{x}) \right) \end{aligned} \tag{16}$$

- Step 4

It follows from the same operation as step 3 that

$$\begin{aligned} e_2^{(4)} &= \frac{\partial e_2^{(3)}}{\partial \boldsymbol{x}} \{f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u}_x\} + \frac{\partial e_2^{(3)}}{\partial \ddot{e}_1} e_1^{(3)} + \frac{\partial e_2^{(3)}}{\partial e_1^{(3)}} e_1^{(4)} \\ &= r_2(\boldsymbol{x}) + [d_1(\boldsymbol{x}),\ d_2(\boldsymbol{x}),\ d_3(\boldsymbol{x}),\ 1]\, \boldsymbol{u} \end{aligned} \tag{17}$$

From (12) and (17), we obtain

$$\begin{bmatrix} e_1^{(2)} \\ e_2^{(4)} \end{bmatrix} = \begin{bmatrix} r_1(\boldsymbol{x}) \\ r_2(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} -p_4 \cos x_{p3} & 0 & 0 & 0 \\ d_1(\boldsymbol{x}) & d_2(\boldsymbol{x}) & d_3(\boldsymbol{x}) & 1 \end{bmatrix} \boldsymbol{u}_M \tag{18}$$

The system is input-output linearizable and the model following input vector is determined by

$$\boldsymbol{u}_p = R(\boldsymbol{x}) + S(\boldsymbol{x})\, \boldsymbol{u}_M \tag{19}$$

$$\begin{aligned} R(\boldsymbol{x}) &= \frac{1}{d_2(\boldsymbol{x}) p_4 \cos x_{p3}} \begin{bmatrix} -d_2(\boldsymbol{x}) & 0 \\ d_1(\boldsymbol{x}) & p_4 \cos x_{p3} \end{bmatrix} \begin{bmatrix} \bar{e}_1 - r_1(\boldsymbol{x}) \\ \bar{e}_2 - r_2(\boldsymbol{x}) \end{bmatrix} \\ S(\boldsymbol{x}) &= \frac{-1}{d_2(\boldsymbol{x}) p_4 \cos x_{p3}} \begin{bmatrix} -d_2(\boldsymbol{x}) & 0 \\ d_1(\boldsymbol{x}) & p_4 \cos x_{p3} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ d_3(\boldsymbol{x}) & 1 \end{bmatrix} \end{aligned}$$

where

$$
\begin{aligned}
\bar{e}_1 &= -\sigma_{12}\dot{e}_1 - \sigma_{11}e_1 \\
\bar{e}_2 &= -\sigma_{24}e_2^{(3)} - \sigma_{23}\ddot{e}_2 - \sigma_{22}\dot{e}_2 - \sigma_{21}e_2 \\
r_1(\boldsymbol{x}) &= -p_1\cos x_{p1} - p_2\sin x_{p1} - p_3 x_{p2} + x_{M3} \\
r_2(\boldsymbol{x}) &= \left\{ -\left(p_1\sin x_{p1} - p_2\cos x_{p1}\right)\left(\frac{p_9 p_{10}}{p_4}\tan x_{p3} + \frac{p_{10}}{p_4\cos x_{p3}}x_{p4}\right)\right. \\
&\quad \left. -\frac{p_{10}}{p_4}x_{p2}\tan x_{p3}\left(p_1\cos x_{p1} + p_2\sin x_{p1}\right)\right\} x_{p2} \\
&\quad + \left\{ \frac{p_3 p_{10}}{p_4\cos x_{p3}}x_{p4} + \frac{p_{10}}{p_4}\tan x_{p3}\left(p_3 p_9 - p_1\sin x_{p1} + p_2\cos x_{p1}\right)\right\}\left(x_{M3} - r_1(\boldsymbol{x})\right) \\
&\quad + \left\{ p_3\left(x_{M3} - r_1(\boldsymbol{x})\right) + \left(2x_{p4}\tan x_{p3} - p_3 + p_9\right)\left(\ddot{e}_1 - r_1(\boldsymbol{x})\right)\right. \\
&\quad \left. -x_{M4} + e_1^{(3)} - x_{p2}\left(p_1\sin x_{p1} - p_2\cos x_{p1}\right)\right\}\frac{p_{10}}{p_4\cos x_{p3}}x_{p4} \\
&\quad + \frac{p_{10}}{p_4\cos x_{p3}}\left(\ddot{e}_1 - r_1(\boldsymbol{x})\right)\left(p_5\cos x_{p3} + p_6\sin x_{p3} + p_7 x_{p4}\right) \\
&\quad + \left\{ \frac{p_{10}}{p_4\cos x_{p3}}x_{p4} - \frac{p_{10}}{p_4}\left(p_3 - p_9\right)\tan x_{p3}\right\} e_1^{(3)} \\
&\quad + \frac{p_{10}}{p_4}\tan x_{p3}\left\{\left(p_3 - p_9\right)x_{M4} - k_1 x_{M1} - k_2 x_{M2} - k_3 x_{M3} - k_4 x_{M4}\right\} \\
&\quad - \frac{p_{10}}{p_4\cos x_{p3}}x_{p4}x_{M4} + k_5 x_{M5} + k_6 x_{M6} + k_7 x_{M7} + k_8 x_{M8} + \frac{p_{10}}{p_4}e_1^{(4)}\tan x_{p3} - p_9^3 x_{p6} \\
d_1(\boldsymbol{x}) &= \left(p_3 p_9 - p_1\sin x_{p1} + p_2\cos x_{p1} - p_9^2\right)p_{10}\sin x_{p3} + \frac{p_3 p_{10}}{\cos x_{p3}} \\
d_2(\boldsymbol{x}) &= \frac{p_8 p_{10}}{p_4\cos x_{p3}}\left(\ddot{e}_1 - r_1(\boldsymbol{x})\right) \\
d_3(\boldsymbol{x}) &= -\frac{p_{10}}{p_4}\tan x_{p3} \\
e_1 &= x_{M1} - x_{p1} \\
\dot{e}_1 &= x_{M2} - x_{p2} \\
\ddot{e}_1 &= -\sigma_{12}\dot{e}_1 - \sigma_{11}e_1 \\
e_1^{(3)} &= (\sigma_{12}^2 - \sigma_{11})\dot{e}_1 + \sigma_{12}\sigma_{11}e_1 \\
e_1^{(4)} &= (-\sigma_{12}^3 + 2\sigma_{12}\sigma_{11})\dot{e}_1 - \sigma_{11}(\sigma_{12}^2 - \sigma_{11})e_1 \\
e_2 &= x_{M5} - x_{p5} \\
\dot{e}_2 &= x_{M6} - x_{p6} \\
\ddot{e}_2 &= \frac{p_{10}}{p_4}\tan x_{p3}\left(\ddot{e}_1 - r_1(\boldsymbol{x})\right) - p_9 x_{p6} + x_{M7} \\
e_2^{(3)} &= \frac{p_{10}}{p_4}\tan x_{p3}\left\{p_3\left(x_{M3} - r_1(\boldsymbol{x})\right) - x_{p2}\left(p_1\sin x_{p1} - p_2\cos x_{p1}\right)\right. \\
&\quad \left. + e_1^{(3)} + \left(p_3 - p_9\right)\left(r_1(\boldsymbol{x}) - \ddot{e}_1\right) - x_{M4}\right\} + x_{M8} + \frac{p_{10}}{p_4\cos x_{p3}}x_{p4}\left(\ddot{e}_1 - r_1(\boldsymbol{x})\right) - p_9^2 x_{p_6}
\end{aligned}
$$

The input vector is always available since the term $d_2(\boldsymbol{x})\cos x_{p3}$ does not vanish for $-\pi/2 < \theta < \pi/2$. The design parameters σ_{ij} $(i = 1,2, j = 1,\cdots,4)$ are selected so that the following characteristic equations are stable.

$$\lambda^2 + \sigma_{12}\lambda + \sigma_{11} = 0 \tag{20}$$

$$\lambda^4 + \sigma_{24}\lambda^3 + \sigma_{23}\lambda^2 + \sigma_{22}\lambda + \sigma_{21} = 0 \tag{21}$$

Then, the closed-loop system has the following error equations

$$\ddot{e}_1 + \sigma_{12}\dot{e}_1 + \sigma_{11}e_1 = 0 \tag{22}$$

$$e_2^{(4)} + \sigma_{24}e_2^{(3)} + \sigma_{23}\ddot{e}_2 + \sigma_{22}\dot{e}_2 + \sigma_{21}e_2 = 0 \tag{23}$$

and the plant outputs converge to the reference outputs. From (11) and (17), u_{p1} and u_{p2} appear first in $\ddot{e}_1$ and $e_2^{(4)}$, respectively. Thus, there are no zero dynamics and the system is minimum phase since the order of (4) is six. Further, we can see that the order of the reference model should be eight so that the inputs (19) do not include the derivatives of the reference inputs $\boldsymbol{u}_M$.

Since the controller requires the angular velocity signals $\dot{\varepsilon}$, $\dot{\theta}$ and $\dot{\phi}$, in the experiment these signals are calculated numerically from the measured angular positions by a discretized differentiator with the first-order filter

$$H_l(z) = \frac{\alpha\left(1 - z^{-1}\right)}{1 - z^{-1} + \alpha T_s} \tag{24}$$

which is derived by substituting

$$s = \frac{(1 - z^{-1})}{T_s} \tag{25}$$

into the differentiator

$$G_l(s) = \frac{\alpha s}{s + \alpha} \tag{26}$$

where z^{-1} is a one-step delay operator, T_s is the sampling period and the design parameter α is a positive constant. Hence, for example, we have

$$\dot{\varepsilon}(k) \approx \frac{1}{\alpha T_s + 1}\left[\dot{\varepsilon}(k-1) + \alpha\left\{\varepsilon(k) - \varepsilon(k-1)\right\}\right]$$

$$\ddot{\varepsilon}(k) \approx \frac{1}{\alpha T_s + 1}\left[\ddot{\varepsilon}(k-1) + \alpha\left\{\dot{\varepsilon}(k) - \varepsilon(k-1)\right\}\right]$$

$$\dot{\theta}(k) \approx \frac{1}{\alpha T_s + 1}\left[\dot{\theta}(k-1) + \alpha\left\{\theta(k) - \theta(k-1)\right\}\right]$$

$$\ddot{\theta}(k) \approx \frac{1}{\alpha T_s + 1}\left[\ddot{\theta}(k-1) + \alpha\left\{\dot{\theta}(k) - \theta(k-1)\right\}\right]$$

$$\dot{\phi}(k) \approx \frac{1}{\alpha T_s + 1}\left[\dot{\phi}(k-1) + \alpha\left\{\phi(k) - \phi(k-1)\right\}\right]$$

$$\ddot{\phi}(k) \approx \frac{1}{\alpha T_s + 1}\left[\ddot{\phi}(k-1) + \alpha\left\{\dot{\phi}(k) - \phi(k-1)\right\}\right]$$

3.2 Experimental studies

The control algorithm described above was applied to the experimental system shown in Section 2. The nominal values of the physical constants are as follows: J_ε=0.86 [kg·m^2], J_θ=0.044 [kg·m^2], J_ϕ=0.82 [kg·m^2], L_a=0.62 [m], L_c=0.44 [m], L_d=0.05 [m], L_e=0.02 [m], L_h=0.177 [m], M_f=0.69 [kg], M_b=0.69 [kg], M_c=1.67 [kg], K_m=0.5 [N/V], g=9.81 [m/s^2], η_ε=0.001 [kg·m^2/s], η_θ=0.001 [kg·m^2/s], η_ϕ=0.005 [kg·m^2/s].

The design parameters are given as follows: The sampling period of the inputs and the outputs is set as $T_s = 2$ [ms]. The inputs u_{M1} and u_{M2} of the reference model are given by

$$u_{M1} = \begin{cases} 0.3, & 45k - 30 \le t < 45k - 7.5 \\ -0.1, & 45k - 7.5 \le t < 45k + 15 \end{cases}$$
$$u_{M2} = \begin{cases} 0, & 0 \le t < 7.5 \\ 0.4, & 45k - 37.5 \le t < 45k - 22.5 \\ -0.4, & 45k - 22.5 \le t < 45k \end{cases} \quad (27)$$
$$k = 0, 1, 2, \cdots$$

All the eigenvalues of the matrices K_1 and K_2 are -1, and the characteristic roots of the error equations (22) and (23) are specified as $(-2.0,\ -3.0)$ and $(-2.0,\ -2.2,\ -2.4,\ -2.6)$, respectively. The origin of the elevation angle ε is set as a nearly horizontal level, so the initial angle is $\varepsilon = -0.336$ when the voltages of two motors are zero, i.e., $V_f = V_b = 0$.

The outputs of the experimental results are shown in Figs. 3 and 4. The tracking is incomplete since there are parameter uncertainties in the model dynamics.

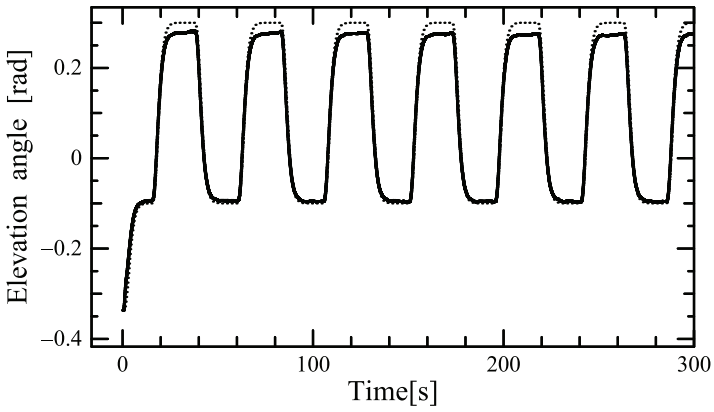


Fig. 3. Time evolution of angle ε (—) and reference output ε_M ($\cdots$).

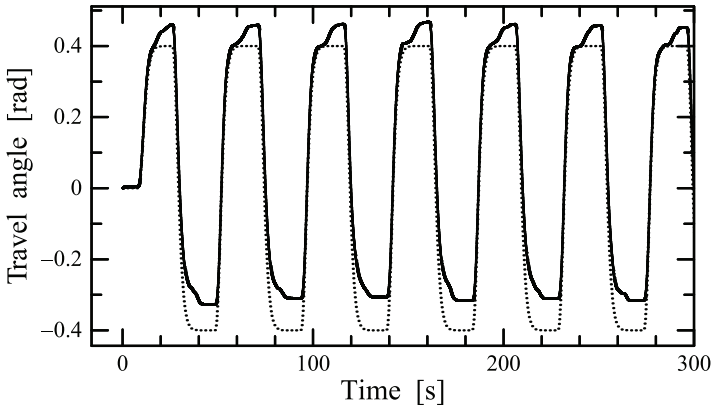


Fig. 4. Time evolution of angle ϕ (—) and reference output ϕ_M ($\cdots$).

4. Parameter Identification Based on the Differential Equations

4.1 Parameter identification algorithm

It is difficult to obtain the desired control performance by applying the algorithm in the previous section directly to the experimental system, since there are parameter uncertainties in the model dynamics. However, it is straightforward to see that the system dynamics (4) are linear with respect to unknown parameters, even though the equations are nonlinear. It is therefore possible to introduce a parameter identification scheme in the feedback control loop. In the present study, the parameter identification scheme is designed in discrete-time form using measured discrete-time signals. Hence, the estimated parameters are calculated recursively at every instant kT, where T is the updating period of the parameters and k is a nonnegative integer. Henceforth we omit T for simplicity. Then, the dynamics of the model helicopter given by equation (4) can be re-expressed as

$$\begin{aligned} w_1(k) &\equiv \ddot{\varepsilon}(k) \\ &= \boldsymbol{\zeta}_1^T \boldsymbol{v}_1(k) \end{aligned} \tag{28}$$

$$\begin{aligned} w_2(k) &\equiv \ddot{\theta}(k) \\ &= \boldsymbol{\zeta}_2^T \boldsymbol{v}_2(k) \end{aligned} \tag{29}$$

$$\begin{aligned} w_3(k) &\equiv \ddot{\phi}(k) \\ &= \boldsymbol{\zeta}_3^T \boldsymbol{v}_3(k) \end{aligned} \tag{30}$$

where

$$\begin{aligned}
\boldsymbol{\zeta}_1 &= [p_1,\ p_2,\ p_3,\ p_4]^T \\
\boldsymbol{\zeta}_2 &= [p_5,\ p_6,\ p_7,\ p_8]^T \\
\boldsymbol{\zeta}_3 &= [p_9,\ p_{10}]^T \\
\boldsymbol{v}_1(k) &= [v_{11}(k),\ v_{12}(k),\ v_{13}(k),\ v_{14}(k)]^T \\
\boldsymbol{v}_2(k) &= [v_{21}(k),\ v_{22}(k),\ v_{23}(k),\ v_{24}(k)]^T \\
\boldsymbol{v}_3(k) &= [v_{31}(k),\ v_{32}(k)]^T \\
v_{11}(k) &= \cos\varepsilon(k),\ v_{12}(k) = \sin\varepsilon(k) \\
v_{13}(k) &= \dot{\varepsilon}(k),\ v_{14}(k) = u_{p1}\cos\theta(k) \\
v_{21}(k) &= \cos\theta(k),\ v_{22}(k) = \sin\theta(k) \\
v_{23}(k) &= \dot{\theta}(k),\ v_{24}(k) = u_{p2}(k) \\
v_{31}(k) &= \dot{\phi}(k),\ v_{32}(k) = u_{p1}\sin\theta(k)
\end{aligned}$$

Defining the estimated parameter vectors corresponding to the vectors $\boldsymbol{\zeta}_1,\ \boldsymbol{\zeta}_2,\ \boldsymbol{\zeta}_3$ as $\widehat{\boldsymbol{\zeta}}_1(k)$, $\widehat{\boldsymbol{\zeta}}_2(k)$, $\widehat{\boldsymbol{\zeta}}_3(k)$, the estimated values of $w_1(k),\ w_2(k),\ w_3(k)$ are obtained as

$$\widehat{w}_1(k) = \widehat{\boldsymbol{\zeta}}_1^T(k)\boldsymbol{v}_1(k) \tag{31}$$

$$\widehat{w}_2(k) = \widehat{\boldsymbol{\zeta}}_2^T(k)\boldsymbol{v}_2(k) \tag{32}$$

$$\widehat{w}_3(k) = \widehat{\boldsymbol{\zeta}}_3^T(k)\boldsymbol{v}_3(k) \tag{33}$$

respectively.
Along with the angular velocities, the angular accelerations $w_1(k) = \ddot{\varepsilon}(k)$, $w_2(k) = \ddot{\theta}(k)$, $w_3(k) = \ddot{\phi}(k)$ are also obtained by numerical calculation using a discretized differentiator.

The parameters are estimated using a recursive least squares algorithm as follows.

$$\begin{aligned}
\widehat{\boldsymbol{\zeta}}_i(k) &= \widehat{\boldsymbol{\zeta}}_i(k-1) + \frac{P_i(k-1)\boldsymbol{v}_i(k-1)\left[w_i(k-1) - \widehat{w}_i(k-1)\right]}{\lambda_i + \boldsymbol{v}_i^T(k-1)P_i(k-1)\boldsymbol{v}_i(k-1)} \\
P_i^{-1}(k) &= \lambda_i P_i^{-1}(k-1) + \boldsymbol{v}_i(k-1)\boldsymbol{v}_i^T(k-1) \\
&\qquad P_i^{-1}(0) > 0\,, \quad 0 < \lambda_i \leq 1,\ i = 1,\ 2,\ 3
\end{aligned} \tag{34}$$

Then, the tracking of the two outputs is achieved under the persistent excitation of the signals $\boldsymbol{v}_i, i = 1, 2, 3$.

4.2 Experimental studies

The estimation and control algorithm described above was applied to the experimental system shown in Section 2.
The design parameters are given as follows: The sampling period of the inputs and the outputs is set as $T_s = 2$ [ms] and the updating period of the parameters, T, takes the same value,

$T = 2$ [ms]. Further, the filter parameter, α, for the estimation of velocities and accelerations is $\alpha = 100$. The variation ranges of the identified parameters are restricted as

$$\begin{aligned} -1.8 \le \widehat{p}_1 \le -0.8, \quad & -2.2 \le \widehat{p}_2 \le -1.2 \\ -0.3 \le \widehat{p}_3 \le 0.0, \quad & 0.1 \le \widehat{p}_4 \le 0.6 \\ -0.5 \le \widehat{p}_5 \le 0.5, \quad & -7.0 \le \widehat{p}_6 \le -5.2 \\ -0.6 \le \widehat{p}_7 \le 0.0, \quad & 1.5 \le \widehat{p}_8 \le 2.2 \\ -0.5 \le \widehat{p}_9 \le 0.0, \quad & -0.5 \le \widehat{p}_{10} \le -0.1 \end{aligned} \tag{35}$$

The design parameters of the identification algorithm are fixed at the values $\lambda_1 = 0.999$, $\lambda_2 = 0.9999$, $\lambda_3 = 0.999$ and ${P_1}^{-1}(0) = {P_2}^{-1}(0) = 10^4 I_4$, ${P_3}^{-1}(0) = 10^4 I_2$. The other design parameters are the same as those of the previous section. The values of the design parameters above are chosen by mainly trial and error. The selection of the sampling period is most important. The achievable minimum sampling period is 2 [ms] due to the calculation ability of the computer. The longer it is, the worse the tracking control performance is.
The outputs of the experimental results are shown in Figs. 5 and 6. The tracking is incomplete because the neither of the output errors of ε or ϕ converge. Figures 7, 8 and 9 display the estimated parameters. All of the estimated parameters move to the limiting values of the variation range.

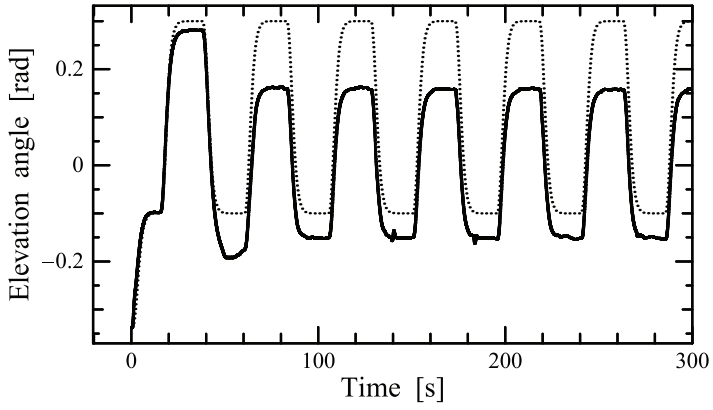


Fig. 5. Time evolution of angle ε (—) and reference output ε_M ($\cdots$).

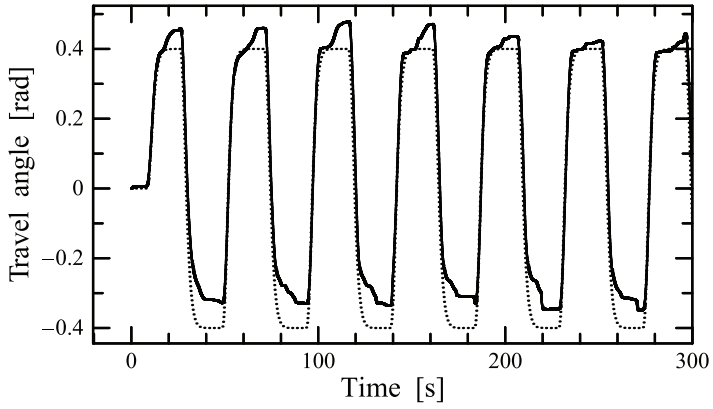


Fig. 6. Time evolution of angle ϕ (—) and reference output ϕ_M ($\cdots$).

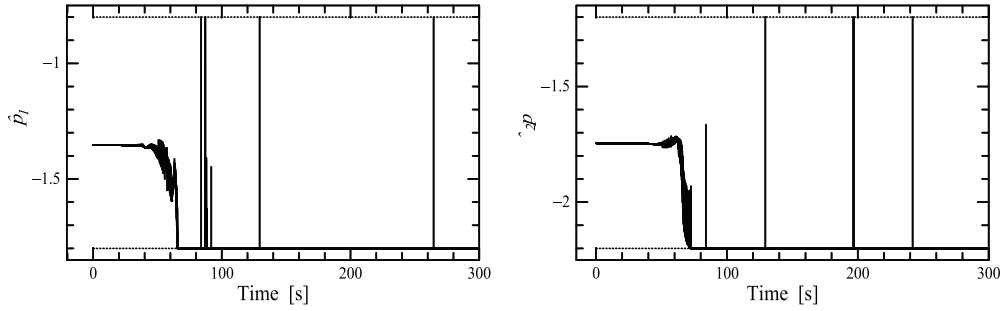
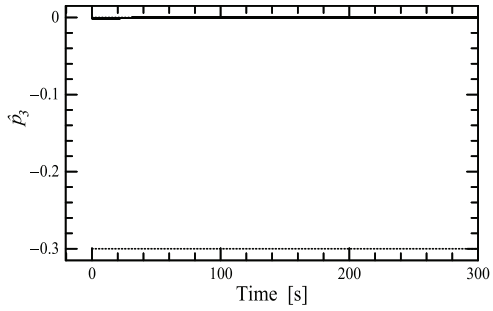
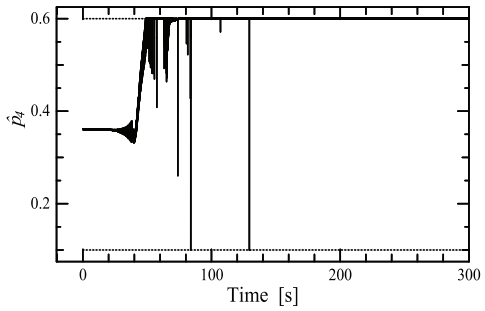
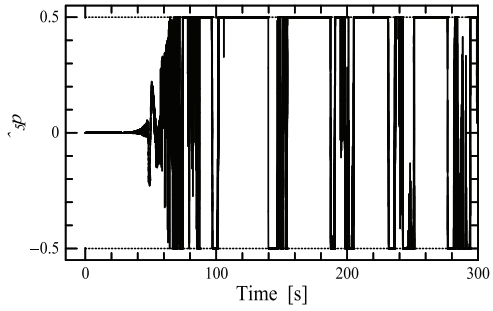
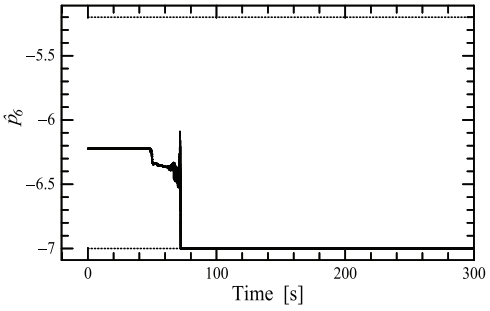


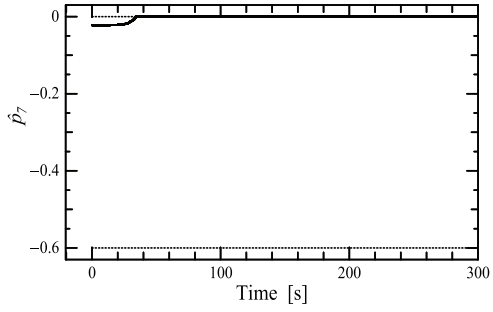
Fig. 7. Time evolution of the estimated parameters $\hat{p}_1$ and $\hat{p}_2$. The dotted lines represent the limited values of variation.

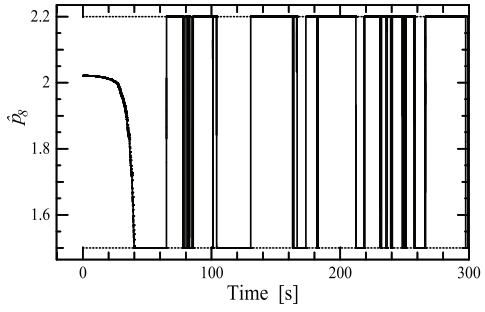

$\hat{p}_3$
0
−0.1
−0.2
−0.3
0
100
200
300
Time [s]


$\hat{p}_4$
0.6
0.4
0.2
0
100
200
300
Time [s]


$\hat{p}_5$
0.5
0
−0.5
0
100
200
300
Time [s]


$\hat{p}_6$
−5.5
−6
−6.5
−7
0
100
200
300
Time [s]


$\hat{p}_7$
0
−0.2
−0.4
−0.6
0
100
200
300
Time [s]


$\hat{p}_8$
2.2
2
1.8
1.6
0
100
200
300
Time [s]

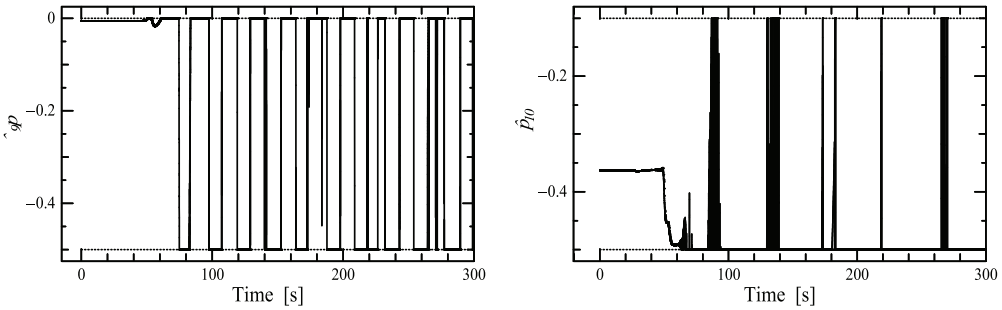


Fig. 8. Time evolution of the estimated parameters from $\hat{p}_3$ to $\hat{p}_{10}$. The dotted lines represent the limited values of variation.

5. Parameter Identification Based on the Integral Form of the Model Equations

5.1 The model equations without model uncertainties and external disturbances

5.1.1 Parameter identification algorithm

The main reason why the experimental results exhibit the poor tracking performance described in the previous subsection 4.2 lies in the fact that the parameter identification is unsatisfactory due to the inaccuracy of the estimation of the velocity and the acceleration signals. To overcome this problem, a parameter estimation scheme is designed for modified dynamics equations obtained by applying integral operators to the differential equations expressing the system dynamics (28)-(30) in this subsection. Neither velocities nor accelerations appear in these modified equations. Define $z_1(k)$ by the following double integral

$$z_1(k) \equiv \int_{kT-nT}^{kT} \int_{\tau-nT}^{\tau} \ddot{\varepsilon}(\sigma) d\sigma d\tau \tag{36}$$

Then, the direct calculation of the right-hand side of equation (36) leads to

$$\begin{aligned} \int_{kT-nT}^{kT} \int_{\tau-nT}^{\tau} \ddot{\varepsilon}(\sigma) d\sigma d\tau &= \int_{kT-nT}^{kT} \left(\dot{\varepsilon}(\tau) - \dot{\varepsilon}(\tau - nT)\right) d\tau \\ &= \varepsilon(kT) - 2\varepsilon(kT - nT) + \varepsilon(kT - 2nT) \end{aligned} \tag{37}$$

Next, discretizing the double integral of the right-hand side of equation (28) yields

$$\begin{aligned} & p_1 \int_{kT-nT}^{kT} \int_{\tau-nT}^{\tau} \cos\varepsilon(\sigma) d\sigma d\tau + \cdots + p_3 \int_{kT-nT}^{kT} \{\varepsilon(\tau) - \varepsilon(\tau - nT)\} d\tau + \cdots \\ & \approx p_1 T^2 \sum_{l=k-(n-1)}^{k} \sum_{i=l-(n-1)}^{l} \cos\varepsilon(i) + \cdots + p_3 T \sum_{l=k-(n-1)}^{k} \{\varepsilon(l) - \varepsilon(l-(n-1))\} + \cdots \end{aligned} \tag{38}$$

As a result, the integral form of the dynamics is obtained as

$$z_i(k) = \boldsymbol{\zeta}_i^T \bar{\boldsymbol{v}}_i(k),\ i = 1,\ 2,\ 3 \tag{39}$$

where

$$z_1(k) \equiv \varepsilon(k) - 2\varepsilon(k-n) + \varepsilon(k-2n) \tag{40}$$
$$z_2(k) \equiv \theta(k) - 2\theta(k-n) + \theta(k-2n) \tag{41}$$
$$z_3(k) \equiv \phi(k) - 2\phi(k-n) + \phi(k-2n) \tag{42}$$
$$\bar{\boldsymbol{v}}_1(k) = [\bar{v}_{11}(k),\ \bar{v}_{12}(k),\ \bar{v}_{13}(k),\ \bar{v}_{14}(k)]^T$$
$$\bar{\boldsymbol{v}}_2(k) = [\bar{v}_{21}(k),\ \bar{v}_{22}(k),\ \bar{v}_{23}(k),\ \bar{v}_{24}(k)]^T$$
$$\bar{\boldsymbol{v}}_3(k) = [\bar{v}_{31}(k),\ \bar{v}_{32}(k)]^T$$
$$\bar{v}_{ij}(k) = T \sum_{l=k-(n-1)}^{k} \tilde{v}_{ij}(l),\ \text{for } (i,j) = \{(1,3),\ (2,3),\ (3,1)\}$$
$$\bar{v}_{ij}(k) = T^2 \sum_{l=k-(n-1),\, m=l-(n-1)}^{k} \sum^{l} v_{ij}(m),\ \text{for other } (i,j)$$
$$\tilde{v}_{13}(l) \equiv \varepsilon(l) - \varepsilon(l-(n-1))$$
$$\tilde{v}_{23}(l) \equiv \theta(l) - \theta(l-(n-1))$$
$$\tilde{v}_{31}(l) \equiv \phi(l) - \phi(l-(n-1))$$

Hence, the estimate model for (39) is given by

$$\hat{z}_i(k) = \hat{\boldsymbol{\zeta}}_i^T(k)\bar{\boldsymbol{v}}_i(k),\ i = 1,\ 2,\ 3 \tag{43}$$

and the system parameters $\hat{\boldsymbol{\zeta}}_i(k)$ can be identified from expression (43) without use of the velocities or accelerations of ε, θ and ϕ.
Finally, the following recursive least squares algorithm is applied to the estimate model (43).

$$\hat{\boldsymbol{\zeta}}_i(k) = \hat{\boldsymbol{\zeta}}_i(k-1) + \frac{P_i(k-1)\bar{\boldsymbol{v}}_i(k-1)\left[z_i(k-1) - \hat{z}_i(k-1)\right]}{\bar{\lambda}_i + \bar{\boldsymbol{v}}_i^T(k-1)P_i(k-1)\bar{\boldsymbol{v}}_i(k-1)} \tag{44}$$
$$P_i^{-1}(k) = \bar{\lambda}_i P_i^{-1}(k-1) + \bar{\boldsymbol{v}}_i(k-1)\bar{\boldsymbol{v}}_i^T(k-1)$$
$$P_i^{-1}(0) > 0\,, \quad 0 < \bar{\lambda}_i \le 1,\ i = 1,\ 2,\ 3$$

Note here that the estimated velocity and acceleration signals are still used in the control input (19).

5.1.2 Experimental studies

The design parameters for the integral form of the identification algorithm are given by $n = 100$, $\bar{\lambda}_1 = \bar{\lambda}_2 = \bar{\lambda}_3 = 0.9999$ and ${P_1}^{-1}(0) = {P_2}^{-1}(0) = 10^3 I_4$, ${P_3}^{-1}(0) = 10^3 I_2$. The reference inputs u_{M1} and u_{M2} are given by

$$u_{M1} = \begin{cases} 0.3, & 45k - 30 \le t < 45k - 7.5 \\ -0.1, & 45k - 7.5 \le t < 45k + 15 \end{cases}$$
$$u_{M2} = \begin{cases} 0, & 0 \le t < 7.5 \\ -0.8, & 45k - 37.5 \le t < 45k - 22.5 \\ 0.8, & 45k - 22.5 \le t < 45k \end{cases} \tag{45}$$
$$k = 0, 1, 2, \cdots$$

The other parameters are the same as those of the previous section.
The outputs are shown in Figs. 9 and 10. The tracking performance of both the outputs ε and ϕ is improved in comparison with the previous section. However, there remains a tracking error. The estimated parameters are plotted in Figs. 11 and 12. All of the parameters change slowly, and the variation of the estimated parameters in Figs. 11 and 12 is smaller than that of the corresponding value shown in Figs. 7 and 8.

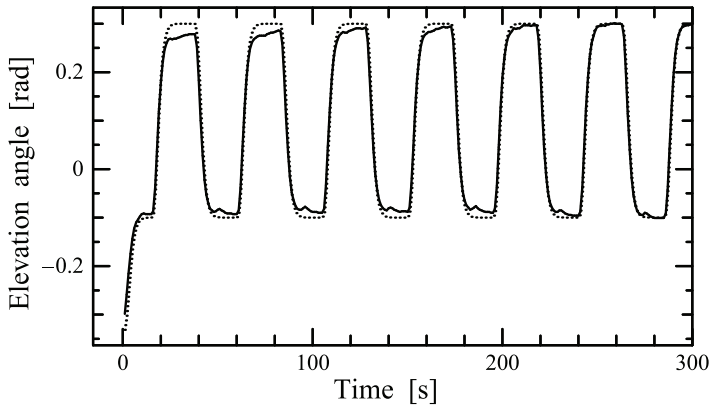


Fig. 9. Time evolution of angle ε (—) and reference output ε_M ($\cdots$).

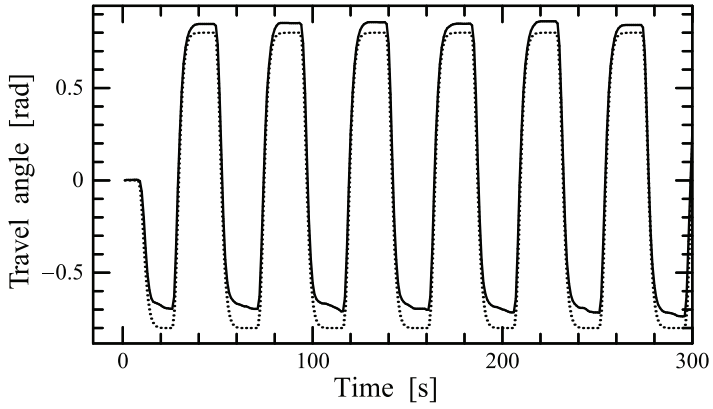


Fig. 10. Time evolution of angle ϕ (—) and reference output ϕ_M ($\cdots$).

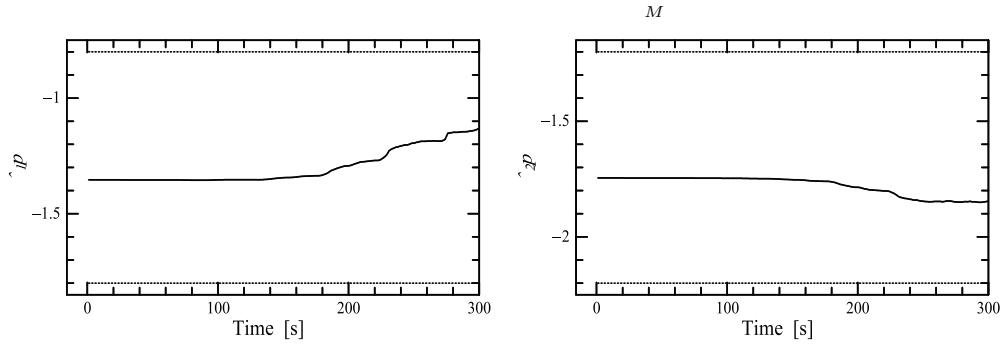
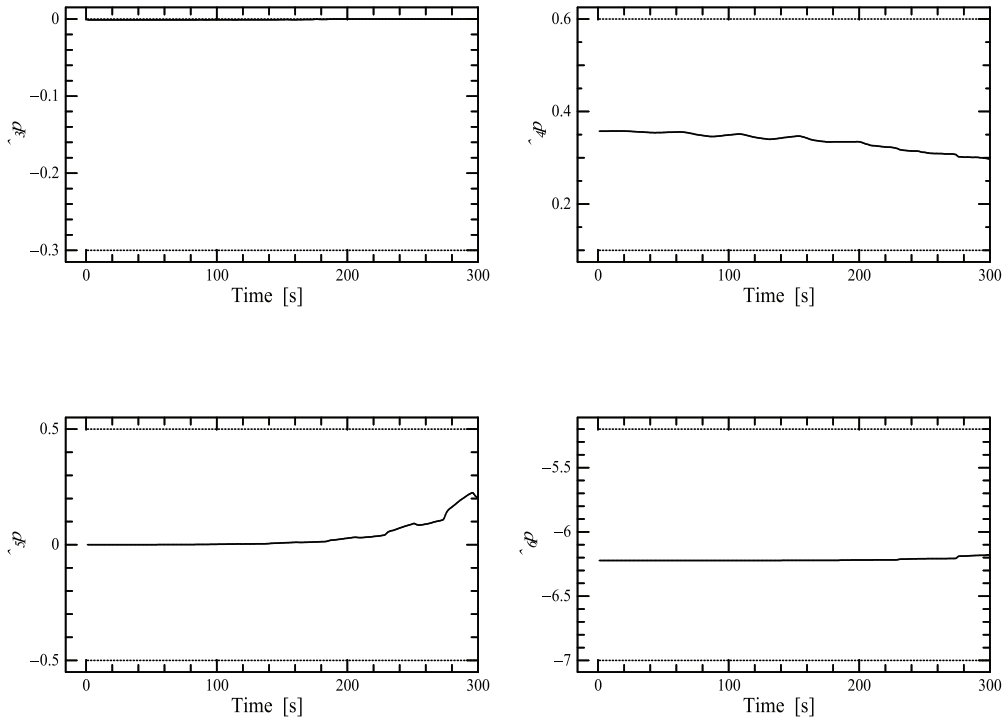


Fig. 11. Time evolution of the estimated parameters $\hat{p}_1$ and $\hat{p}_2$. The dotted lines represent the limited values of variation.



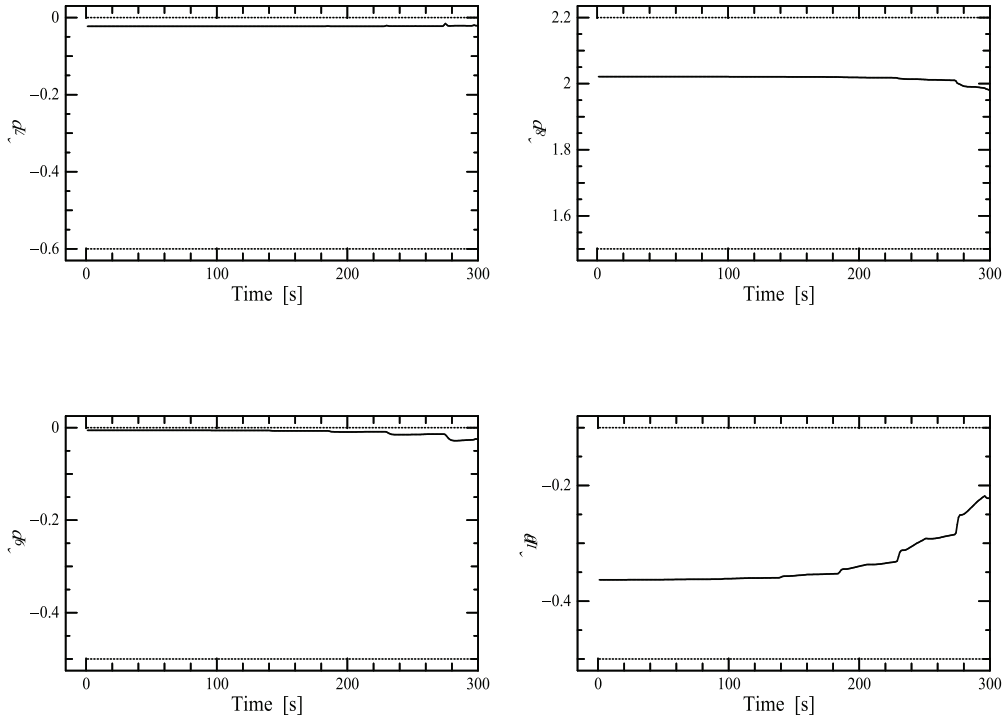


Fig. 12. Time evolution of the estimated parameters from $\hat{p}_3$ to $\hat{p}_{10}$. The dotted lines represent the limited values of variation.

5.2 The model equations with model uncertainties and external disturbances

5.2.1 Parameter identification algorithm

Although the use of the integral form of the dynamics has improved the tracking performance of both the outputs ε and θ, tracking errors still remain. On the basis that these errors are caused by model uncertainties and external disturbances, for example, motor dynamics or friction (other than viscous friction), we add the additional terms f_ε, f_θ and f_ϕ into equation (4) to represent model uncertainties and external disturbances. Generally, the additional terms should be given as, for instance,

$$f_\varepsilon(t) = \sum c_i f_i(t) \tag{46}$$

where c_i is a constant and f_i is a known function of time. For simplicity, however, here we assume these additional terms are constant because tracking errors in the experimental results approximately remain constant in Fig. 10. Then, the system dynamics are expressed as

$$\begin{aligned} w_1(k) \equiv \ddot{\varepsilon}(k) &= \boldsymbol{\xi}_1^T \boldsymbol{q}_1(k) \\ w_2(k) \equiv \ddot{\theta}(k) &= \boldsymbol{\xi}_2^T \boldsymbol{q}_2(k) \qquad (47) \\ w_3(k) \equiv \ddot{\phi}(k) &= \boldsymbol{\xi}_3^T \boldsymbol{q}_3(k) \qquad (48) \end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\xi}_1 &= \left[\boldsymbol{\zeta}_1^T,\ p_{11}\right]^T \\
\boldsymbol{\xi}_2 &= \left[\boldsymbol{\zeta}_2^T,\ p_{12}\right]^T \\
\boldsymbol{\xi}_3 &= \left[\boldsymbol{\zeta}_3^T,\ p_{13}\right]^T \\
\boldsymbol{q}_1(k) &= \left[\boldsymbol{v}_1^T(k),\ 1\right]^T \\
\boldsymbol{q}_2(k) &= \left[\boldsymbol{v}_2^T(k),\ 1\right]^T \\
\boldsymbol{q}_3(k) &= \left[\boldsymbol{v}_3^T(k),\ 1\right]^T \\
p_{11} &= f_\varepsilon / J_\varepsilon \\
p_{12} &= f_\theta / J_\theta \\
p_{13} &= f_\phi / J_\phi
\end{aligned}$$

It is worth noting that all the parameters p_i $(i = 1, \ldots, 13)$ of the equations are constant. Then, the integral form of the dynamics is obtained as well as the previous subsection as

$$z_i(k) = \boldsymbol{\xi}_i^T \bar{\boldsymbol{q}}_i(k),\ i = 1,\ 2,\ 3 \tag{49}$$

where

$$\begin{aligned}
\bar{\boldsymbol{q}}_1(k) &= \left[\bar{\boldsymbol{v}}_1^T(k),\ \bar{q}_{15}(k)\right]^T \\
\bar{\boldsymbol{q}}_2(k) &= \left[\bar{\boldsymbol{v}}_2^T(k),\ \bar{q}_{25}(k)\right]^T \\
\bar{\boldsymbol{q}}_3(k) &= \left[\bar{\boldsymbol{v}}_3^T(k),\ \bar{q}_{33}(k)\right]^T \\
\bar{q}_{15}(k) &= T^2 \sum_{l=k-(n-1)}^{k} \sum_{i=l-(n-1)}^{l} 1 \\
&= T^2 (n-1)^2 \\
\bar{q}_{25}(k) &= T^2 (n-1)^2 \\
\bar{q}_{33}(k) &= T^2 (n-1)^2
\end{aligned}$$

Hence, defining the estimated parameter vectors corresponding to the vectors $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, $\boldsymbol{\xi}_3$ as $\widehat{\boldsymbol{\xi}}_1(k)$, $\widehat{\boldsymbol{\xi}}_2(k)$, $\widehat{\boldsymbol{\xi}}_3(k)$, the estimate model for expression (49) is given by

$$\widehat{z}_i(k) = \widehat{\boldsymbol{\xi}}_i^T(k) \bar{\boldsymbol{q}}_i(k), \quad i = 1,\ 2,\ 3 \tag{50}$$

and the system parameters $\widehat{\boldsymbol{\xi}}_i(k)$ can be identified by the following recursive least squares algorithm.

$$\begin{aligned}
\widehat{\boldsymbol{\xi}}_i(k) &= \widehat{\boldsymbol{\xi}}_i(k-1) + \frac{P_i(k-1)\bar{\boldsymbol{q}}_i(k-1)\left[z_i(k-1) - \widehat{z}_i(k-1)\right]}{\bar{\lambda}_i + \bar{\boldsymbol{q}}_i^T(k-1) P_i(k-1) \bar{\boldsymbol{q}}_i(k-1)} \\
P_i^{-1}(k) &= \bar{\lambda}_i P_i^{-1}(k-1) + \bar{\boldsymbol{q}}_i(k-1)\bar{\boldsymbol{q}}_i^T(k-1) \\
& \qquad P_i^{-1}(0) > 0\,, \quad 0 < \bar{\lambda}_i \leq 1,\ i = 1,\ 2,\ 3
\end{aligned} \tag{51}$$

5.2.2 Experimental studies

The weighting factor of the least squares algorithm is given by

$$\widetilde{\lambda} = 0.9995 + 0.0005 \exp\left(-5\sqrt{e_1^2 + e_2^2}\right)$$

The updating period of the parameters, T, is $T = 10$ [ms], while other design parameters are the same as those of the previous section. The value of T stated above led to the best experimental result unlike the previous experimental studies.

The outputs are depicted in Figs. 13 and 14, while the estimated parameters are shown in Figs. 15, 16 and 17. The tracking performance of both of the outputs ε and ϕ has been further improved by the inclusion of the uncertainties.

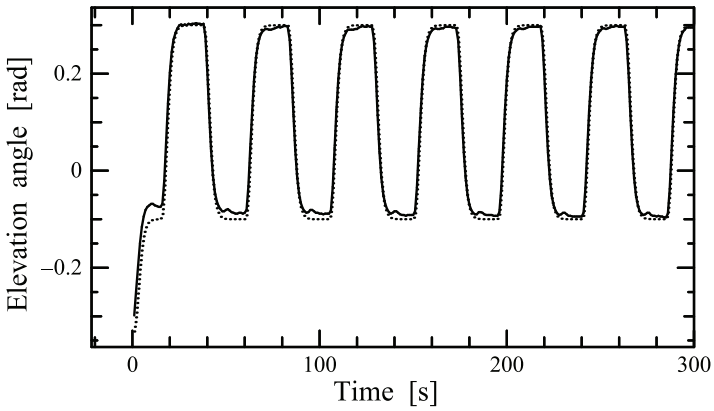


Fig. 13. Time evolution of angle ε (—) and reference output ε_M ($\cdots$).

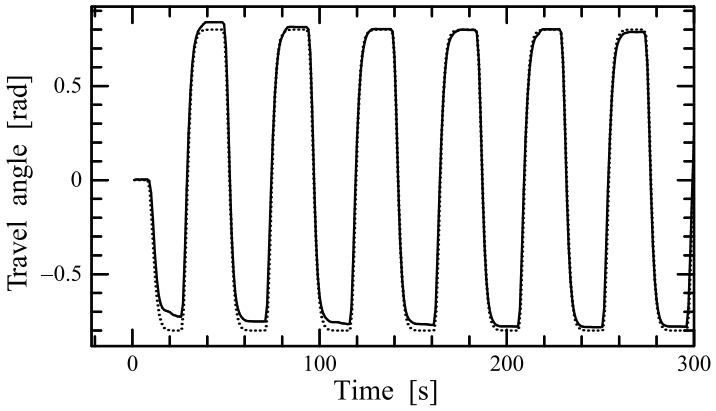


Fig. 14. Time evolution of angle ϕ (—) and reference output ϕ_M ($\cdots$).

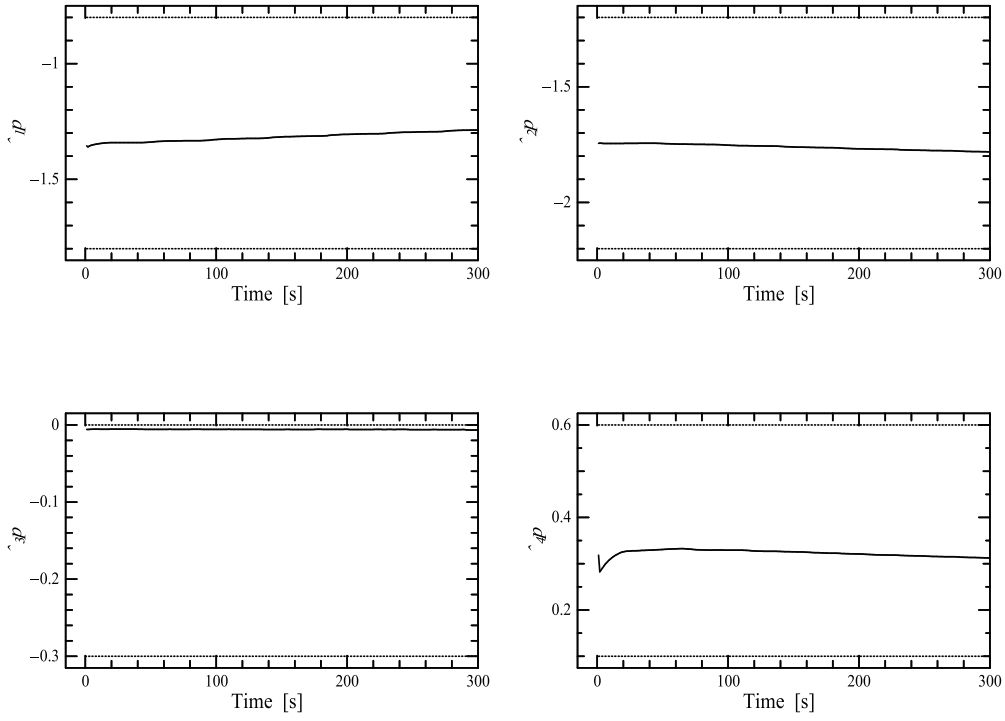
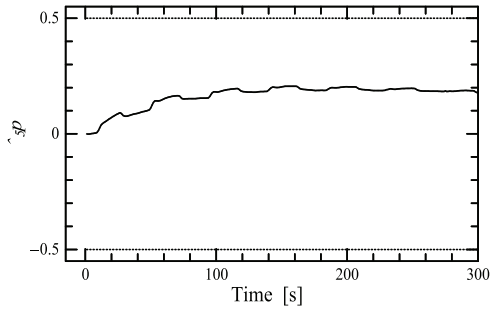
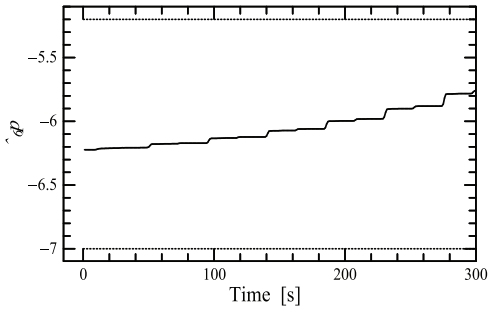
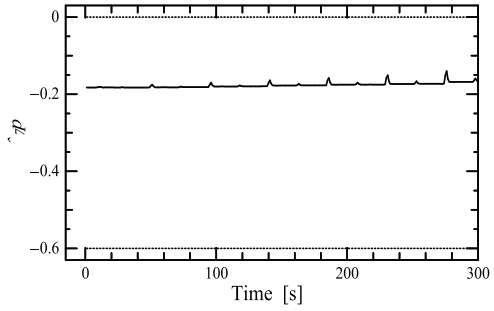
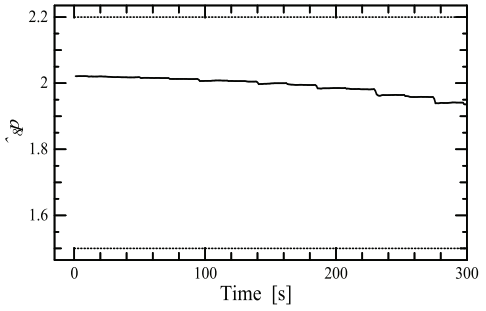


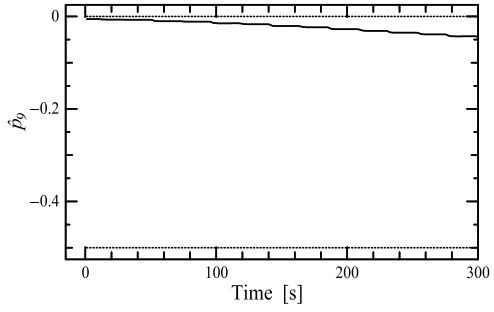
Fig. 15. Time evolution of the estimated parameters from $\hat{p}_1$ to $\hat{p}_4$. The dotted lines represent the limited values of variation.

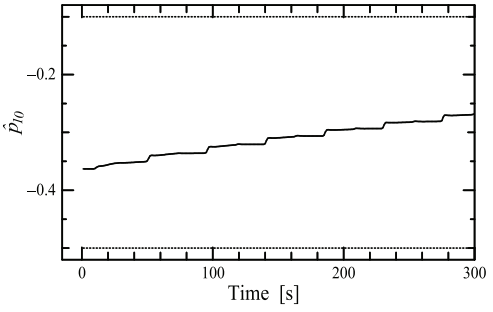

0.5
0
−0.5
0
100
200
300
Time [s]


−5.5
−6
−6.5
−7
0
100
200
300
Time [s]


0
−0.2
−0.4
−0.6
0
100
200
300
Time [s]


2.2
2
1.8
1.6
0
100
200
300
Time [s]


0
−0.2
−0.4
0
100
200
300
Time [s]


−0.2
−0.4
0
100
200
300
Time [s]

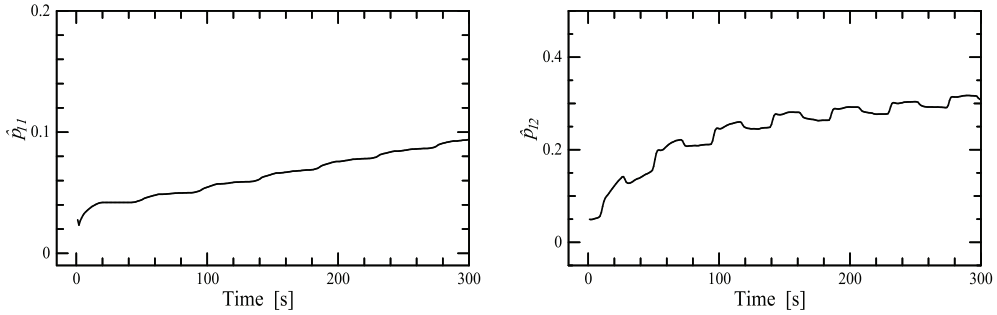
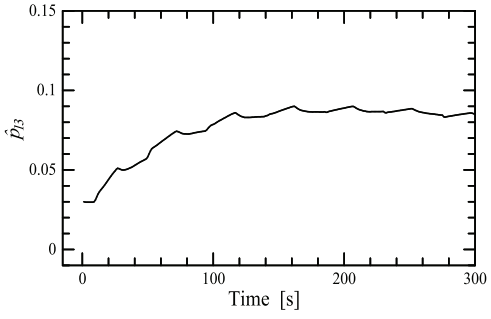


Fig. 16. Time evolution of the estimated parameters from $\hat{p}_5$ to $\hat{p}_{12}$. The dotted lines represent the limited values of variation.



Q

Fig. 17. Time evolution of the estimated parameter $\hat{p}_{13}$. The dotted lines represent the limited values of variation.

6. Conclusions

This paper considers the nonlinear adaptive model following control of a 3-DOF model helicopter. The system model here is not decouplable by static state feedback, and the nonlinear structure algorithm is applied. When a simple model following controller is designed, it is not easy to obtain a good control performance mainly due to the parameter uncertainties. Then, two parameter identification schemes are discussed: The first scheme is based on the differential equation model. This scheme is unable to obtain a good tracking control performance because of the inaccuracy of the estimated velocity and acceleration signals. The second scheme is designed for a dynamics model derived by applying integral operators to the differential equations expressing the system dynamics. Hence, this identification algorithm requires neither velocity nor acceleration signals. The experimental results show that the second method yields a better tracking result, although tracking errors still remain. Finally, we introduce additional terms into the equations of motion to express model uncertainties and external disturbances. With reference to experimental results, this modification is shown to further improve the tracking control performance.

7. References

[1] Apkarian, J. (1998). 3D Helicopter experiment manual, Quanser Consulting, Inc
[2] Avila, J, C.; Brogliato, B.; Dzul, A. & Lozano, R. (2003). Nonlinear modelling and control of
helicopters, *Automatica*, Vol. 39, No. 9, pp. 1583-1596
[3] Isurugi, Y. (1990). Model following control for nonlinear systems, INRIA Reports, No.1332
[4] Kaloust, J.; Ham, C. & Qu, Z. (1997). Nonlinear autopilot control design for a 2-DOF helicopter
model, *IEE Proc. Control Theory and Appl.*, Vol. 144, No. 6, pp. 612-616
[5] Kutay, A, T.; Calise, A, J; Idan, M. & Hovakimyan, N. (2005). Experimental results on adaptive
output feedback control using a laboratory model helicopter, *IEEE Trans. Control Systems Technology*, Vol. 13, No. 2, pp. 196-202
[6] Sira-Ramirez, H.; Zribi, M. & Ahmad, S. (1994). Dynamical sliding mode control approach for
vertical flight regulation in helicopters, *IEE Proc. Control Theory and Appl.*, Vol. 141, No. 1, pp. 19-24
[7] Shima, M.; Isurugi, Y.; Yamashita, Y.; Watanabe, A.; Kawamura, T. & Yokomichi M. (1997).
Control Theory of Nonlinear Systems (in Japanese), Corona Publishing Co., Ltd.

10

Application of Higher Order Derivatives to Helicopter Model Control

Roman Czyba, Michal Serafin
Silesian University of Technology, Poland

1. Introduction

Control of a helicopter model is a problem of both theoretical and practical interest. With the proliferation of autonomous unmanned aerial vehicles (UAVs) (Castillo et al., 2005; Valavanis, 2007) autopilot modes have become very important. Dynamic properties of a controlled helicopter depend on both its structure and aerodynamic qualities as well as on the control law applied. The problem of output regulation has received much attention and especially during the last decade, its nonlinear version has been intensively developed (Isidori & Byrnes, 1990), (Slotine & Li, 1991). The well known approach to decoupling problem solution based on the Non-linear Inverse Dynamics (NID) method (Balas et al., 1995) may be used if the parameters of the plant model and external disturbances are exactly known. Usually, incomplete information about systems in real practical tasks takes place. In this case adaptive control methods (Astrom & Wittenmark, 1994) or control systems with sliding mode (Utkin, 1992) may be used for solving this control problem. A crucial feature of the sliding mode techniques is that in the sliding phase the motion of the system is insensitive to parameter variation and disturbances in the system. A way of the algorithmic solution of this problem under condition of incomplete information about varying parameters of the plant and unknown external disturbances is the application of the Localization Method (LM) (Vostrikov & Yurkevich, 1993), which allows to provide the desired transients for nonlinear time-varying systems. A development of LM is applied in the present paper, and proposed in (Błachuta et al., 1999; Czyba & Błachuta, 2003; Yurkevich, 2004), method which based on two ideas. The first – the use of high gain in feedback to suppress the disturbances and varying parameters; the second – the use of higher order output derivatives in the feedback loop. The high gain and "dynamics" of the controller are separated by means of the summing junction with set point signal placed between them. This structure is the implementation of the model reference control with the reference model transfer function which is equal to the inverse of the controller "dynamics". It becomes that the proposed structure and method is insensitive to plant parameters changes and external disturbances, and works well both lineal, nonlinear, stationary and nonstationary objects. In the present paper, the proposed method is applied to control of the helicopter model, which is treated as a multivariable system.

In general, the goal of the design of a helicopter model control system is to provide decoupling, i.e. each output should be independently controlled by a single input, and to provide desired output transients under assumption of incomplete information about varying parameters of the plant and unknown external disturbances. In addition, we require that transient processes have desired dynamic properties and are mutually independent.
The paper is part of a continuing effort of analytical and experimental studies on aircraft control (Czyba & Błachuta, 2003), and BLDC motor control (Szafrański & Czyba, 2008). The main aim of this research effort is to examine the effectiveness of a designed control system for real physical plant – laboratory model of the helicopter. The paper is organized as follows. First, a mathematical description of the helicopter model is introduced. Section 3 includes a background of the discussed method and the method itself are summarized. The next section contains the design of the controller, and finally the results of experiments are shown. The conclusions are briefly discussed in the last section.

2. Helicopter model

The CE150 helicopter model was designed by Humusoft for the theoretical study and practical investigation of basic and advanced control engineering principles. The helicopter model (Fig.1) consists of a body, carrying two propellers driven by DC motors, and massive support. The body has two degrees of freedom. The axes of the body rotation are perpendicular as well as the axes of the motors. Both body position angles, i.e. azimuth angle in horizontal and elevation angle in vertical plane are influenced by the rotating propellers simultaneously. The DC motors for driving propellers are controlled proportionally to the output signals of the computer. The helicopter model is a multivariable dynamical system with two manipulated inputs and two measured outputs. The system is essentially nonlinear, naturally unstable with significant crosscouplings.

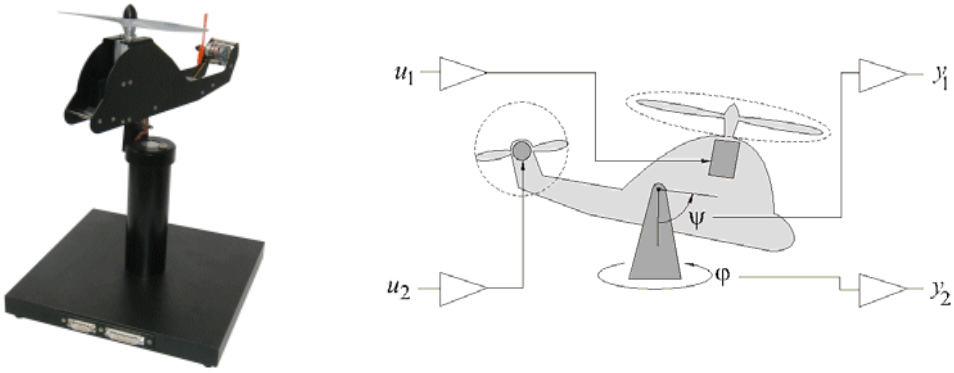


Fig. 1. CE150 Helicopter model (Horacek, 1993)

In this section a mathematical model by considering the force balances is presented (Horacek, 1993). Assuming that the helicopter model is a rigid body with two degrees of freedom, the following output and control vectors are adopted:

$$\bar{Y} = \left[\psi, \varphi\right]^T \tag{1}$$

$$\bar{u} = \left[u_1, u_2\right]^T \tag{2}$$

where: ψ - elevation angle (pitch angle); φ - azimuth angle (yaw angle); u_1 - voltage of main motor; u_2 - voltage of tail motor.

2.1 Elevation dynamics

Let us consider the forces in the vertical plane acting on the vertical helicopter body, whose dynamics are given by the following nonlinear equation:

$$I_\psi \psi^{(2)} = \tau_1 + \tau_{\varphi^{(1)}} - \tau_{f1} - \tau_m + \tau_G \tag{3}$$

with

$$\tau_1 = k_{\omega_1} \omega_1^2 \tag{4}$$

$$\tau_{\varphi^{(1)}} = \frac{1}{2} ml \left(\varphi^{(1)}\right)^2 \sin 2\psi \tag{5}$$

$$\tau_{f1} = C_\psi sign\psi^{(1)} + B_\psi \psi^{(1)} \tag{6}$$

$$\tau_m = mgl \sin\psi \tag{7}$$

$$\tau_G = K_G \omega_1 \varphi^{(1)} \cos\psi \tag{8}$$

where:

I_ψ - moment of inertia around horizontal axis
τ_1 - elevation driving torque
$\tau_{\varphi^{(1)}}$ - centrifugal torque
τ_{f1} - friction torque (Coulomb and viscous)
τ_m - gravitational torque
τ_G - gyroscopic torque
ω_1 - angular velocity of the main propeller
m - mass
g - gravity
l - distance from z-axis to main rotor
k_{ω_1} - constant for the main rotor
K_G - gyroscopic coefficient
B_ψ - viscous friction coefficient (around y-axis)
C_ψ - Coulomb friction coefficient (around y-axis)

2.2 Azimuth dynamics

Let us consider the forces in the horizontal plane, taking into account the main forces acting on the helicopter body in the direction of φ angle, whose dynamics are given by the following nonlinear equation:

$$I_{\varphi}\varphi^{(2)} = \tau_2 - \tau_{f2} - \tau_r \tag{9}$$

with

$$I_{\varphi} = I_{\psi}\sin\psi \tag{10}$$

$$\tau_2 = k_{\omega_2}\omega_2^2 \tag{11}$$

$$\tau_{f2} = C_{\varphi} sign\varphi^{(1)} + B_{\varphi}\varphi^{(1)} \tag{12}$$

where:

I_{φ} - moment of inertia around vertical axis
τ_2 - stabilizing motor driving torque
τ_{f2} - friction torque (Coulomb and viscous)
τ_r - main rotor reaction torque
k_{ω_2} - constant for the tail rotor
ω_2 - angular velocity of the tail rotor
B_{φ} - viscous friction coefficient (around z-axis)
C_{φ} - Coulomb friction coefficient (around z-axis)

2.3 DC motor and propeller dynamics modeling

The propulsion system consists two independently working DC electrical engines. The model of a DC motor dynamics is achieved based on the following assumptions:

Assumption1: The armature inductance is very low.

Assumption2: Coulomb friction and resistive torque generated by rotating propeller in the air are significant.

Assumption3: The resistive torque generated by rotating propeller depends on ω in low and ω^2 in high rpm.

Taking this into account, the equations are following:

$$I_j\omega_j^{(1)} = \tau_j - \tau_{cj} - B_j\omega_j - \tau_{pj} \tag{13}$$

with

$$\tau_j = K_{ij}i_j \tag{14}$$

$$i_j = \frac{1}{R_j}\left(u_j - K_{bj}\omega_j\right) \tag{15}$$

$$\tau_{cj} = C_j sign\left(\omega_j\right) \tag{16}$$

$$\tau_{pj} = B_{pj}\omega_j + D_{pj}\omega_j^2 \tag{17}$$

where:

$j = 1, 2$ - motor number (1- main, 2- tail)

I_j - rotor and propeller moment of inertia

τ_j - motor torque

τ_{cj} - Coulomb friction load torque

τ_{pj} - air resistance load torque

B_j - viscous-friction coefficient

K_{ij} - torque constant

i_j - armature current

R_j - armature resistance

u_j - control input voltage

K_{bj} - back-emf constant

C_j - Coulomb friction coefficient

B_{pj} - air resistance coefficient (laminar flow)

D_{pj} - air resistance coefficient (turbulent flow)

Block diagram of nonlinear dynamics of a complete system is to be assembled from the above derivations and the result is in Fig.2.

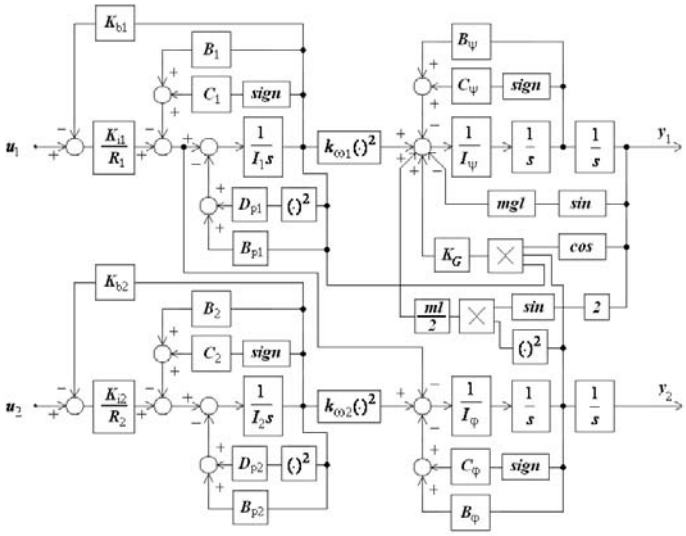


Fig. 2. Block diagram of a complete system dynamics

3. Control scheme

Let us consider a nonlinear time-varying system in the following form:

$$x^{(1)}(t) = h\left(x(t), u(t), t\right), \quad x(0) = x_0 \tag{18}$$

$$y(t) = g\left(t, x(t)\right) \tag{19}$$

where $x(t)$ is n–dimensional state vector, $y(t)$ is p–dimensional output vector and $u(t)$ is p-dimensional control vector. The elements of the $f\left(t, x(t)\right)$, $B\left(t, x(t)\right)$ and $g\left(t, x(t)\right)$ are differentiable functions.

Each output $y_i(t)$ can be differentiated m_i times until the control input appears. Which results in the following equation:

$$y^{(m)}(t) = f\left(t, x(t)\right) + B\left(t, x(t)\right)u(t) \tag{20}$$

where: $y^{(m)}(t) = \left[y_1^{(m_1)}, y_2^{(m_2)}, ..., y_p^{(m_p)} \right]$,

$\left| f_i(t, x) \right| \le f_i^{\max}, \qquad i = 1,\ 2, ..., p$,

$\det\left(B\left(t, x(t)\right) \right) \neq 0$.

The value m_i is a relative order of the system (18), (19) with respect to the output $y_i(t)$ (or so called the order of a relative higher derivative). In this case the value $y_i^{(m_i)}$ depends explicitly on the input $u(t)$.

The significant feature of the approach discussed here is that the control problem is stated as a problem of determining the root of an equation by introducing reference differential equation whose structure is in accordance with the structure of the plant model equations. So the control problem can be solved if behaviour of the $y_i^{(m_i)}$ fulfills the reference model which is given in the form of the following stable differential equation:

$$y_{i\,M}^{(m_i)}(t) = F_{i\,M}\left(\overline{y}_{i\,M}(t), r_i(t) \right) \tag{21}$$

where: $F_{i\,M}$ is called the desired dynamics of $y_i(t)$, $\overline{y}_{i\,M}(t) = \left[y_{i\,M}, y_{i\,M}^{(1)}, ..., y_M^{(m_i-1)} \right]^T$, $r_i(t)$ is the reference value and the condition $y_i = r_i$ takes place for an equilibrium point.

Denote the tracking error as follows:

$$\Delta(t) = r(t) - y(t). \tag{22}$$

The task of a control system is stated so as to provide that

$$\lim_{t\to\infty} \Delta(t) = 0 . \tag{23}$$

Moreover, transients $y_i(t)$ should have the desired behavior defined in (21) which does not depend either on the external disturbances or on the possibly varying parameters of system in equations (18), (19). Let us denote

$$\Delta^F = F_M\left(\overline{y}(t), r(t)\right) - y^{(m)}(t) \tag{24}$$

where: Δ^F is the error of the desired dynamics realization, $F_M = \left[F_{1M}, F_{2M}, ..., F_{pM}\right]^T$ is a vector of desired dynamics.

As a result of (20), (21), (24) the desired behaviour of $y_i(t)$ will be provided if the following condition is fulfilled:

$$\Delta^F\left(x(t), \overline{y}(t), r(t), u(t), t\right) = 0 . \tag{25}$$

So the control action $u(t)$ which provides the control problem solution is the root of equation (25). Above expression is the insensitivity condition of the output transient performance indices with respect to disturbances and varying parameters of the system in (18), (19).

The solution of the control problem (25) bases on the application of the higher order output derivatives jointly with high gain in the controller. The control law in the form of a stable differential equation is constructed such that its stable equilibrium is the solution of equation (25). Such equation can be presented in the following form (Yurkevich, 2004)

$$\begin{aligned} &\mu_i^{q_i} \nu_i^{(q_i)} + \sum_{j=0}^{q_i-1} \mu_i^j d_{i,j} \nu_i^{(j)} = k\Delta_i^F \\ &\nu_i(0) = \nu_{i,0} \end{aligned} \tag{26}$$

where:

$i = 1, ..., p$,

$\nu_i(t) = \left[\nu_i, \nu_i^{(1)}, ..., \nu_i^{(q_i-1)}\right]^T$ - new output of the controller,

μ_i - small positive parameter $\mu_i > 0$,

k - gain,

$d_{i,0}, ..., d_{i,q_i-1}$ - diagonal matrices.

To decoupling of control channel during the fast motions let us use the following output controller equation:

$$u(t) = K_0 K_1 \nu(t) \tag{27}$$

where:

$K_1 = diag\left(k_1, k_2, ..., k_p\right)$ is a matrix of gains,

K_0 is a nonsingular matching matrix (such that BK_0 is positive definite).

Let us assume that there is a sufficient time-scale separation, represented by a small parameter μ_i, between the fast and slow modes in the closed loop system. Methods of singularly perturbed equations can then be used to analyze the closed loop system and, as a result, slow and fast motion subsystems can be analyzed separately. The fast motions refer to the processes in the controller, whereas the slow motions refer to the controlled object.

Remark 1: It is assumed that the relative order of the system (18), (19), determined in (20), and reference model (21) is the same m_i.

Remark 2: Assuming that $q_i \geq m_i$ (where $i = 1, 2, ..., p$), then the control law (26) is proper and it can be realized without any differentiation.

Remark 3: The asymptotically stability and desired transients of $\nu_i(t)$ are provided by choosing $\mu_i, k, d_{i,0}, d_{i,1}, ..., d_{i,q_i-1}$.

Remark 4: Assuming that $d_{i,0} = 0$ in equation (26), then the controller includes the integration and it provides that the closed-loop system is type I with respect to reference signal.

Remark 5: If the order of reference model (21) is $m_i - 1$, such that the relative order of the open loop system is equal one, then we obtain sliding mode control.

4. Helicopter controller design

The helicopter model described by equations (1)–(17), will be used to design the control system that achieves the tracking of a reference signal. The control task is stated as a tracking problem for the following variables:

$$\lim_{t\to\infty}\left[\psi_0(t) - \psi(t)\right] = 0 \tag{28}$$

$$\lim_{t\to\infty}\left[\varphi_0(t) - \varphi(t)\right] = 0 \tag{29}$$

where $\psi_0(t)$, $\varphi_0(t)$ are the desired values of the considered variables.

In addition, we require that transient processes have desired dynamic properties, are mutually independent and are independent of helicopter parameters and disturbances.

The inverse dynamics of (18), (19) are constructed by differentiating the individual elements of y sufficient number of times until a term containing u appears in (20). From equations of helicopter motion (3)–(17) it follows that:

$$\psi^{(3)} = f_1 + \frac{K_{i1}\left(2k_{\omega_1}\omega_1 + K_G\varphi^{(1)}\cos\psi\right)}{I_1 R_1 I_\psi}u_1 \tag{30}$$

$$\varphi^{(3)} = f_2 + \frac{K_{i1}\left(K_{b1} + B_1\right)}{I_1 R_1 I_\psi \sin\psi}u_1 + \frac{2K_{i2}k_{\omega_2}\omega_2}{I_2 R_2 I_\psi \sin\psi}u_2 \tag{31}$$

Following (20), the above relationship becomes:

$$\begin{bmatrix}\psi^{(3)}\\ \varphi^{(3)}\end{bmatrix} = \begin{bmatrix}f_1\\ f_2\end{bmatrix} + B\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{32}$$

where values of f_1, f_2 are bounded, and the matrix B is given in the following form

$$B = \begin{bmatrix}b_{11} & 0\\ b_{21} & b_{22}\end{bmatrix}. \tag{33}$$

In normal flight conditions we have $\det\left(B\left(t, x(t)\right)\right) \neq 0$. This is a sufficient condition for the existence of an inverse system model to (18), (19).

Let us assume that the desired dynamics are determined by a set of mutually independent differential equations:

$$\tau_\psi^3\psi^{(3)} = -3\tau_\psi^2\alpha_\psi\psi^{(2)} - 3\tau_\psi\alpha_\psi^2\psi^{(1)} - \psi + \psi_0 \tag{34}$$

$$\tau_\varphi^3\varphi^{(3)} = -3\tau_\varphi^2\alpha_\varphi\varphi^{(2)} - 3\tau_\varphi\alpha_\varphi^2\varphi^{(1)} - \varphi + \varphi_0 \tag{35}$$

Parameters τ_i and α_i $(i = \psi, \varphi)$ have very well known physical meaning and their particular values have to be specified by the designer.

The output controller equation from (27) is as follows:

$$\begin{pmatrix}u_1\\ u_2\end{pmatrix} = K_0K_1\begin{pmatrix}\nu_\psi\\ \nu_\varphi\end{pmatrix} \tag{36}$$

where $K_1 = diag\left(k_\psi, k_\varphi\right)$ and assume that $K_0 = \left(B\right)^{-1}$ because matrix BK_0 must be positive definite. Moreover $BK_0 \cong I$ assures decoupling of fast mode channels, which makes controller's tuning simpler.

The dynamic part of the control law from (26) has the following form:

$$\mu_\psi^3 \nu_\psi^{(3)} + 3\mu_\psi^2 d_{\psi,2} \nu_\psi^{(2)} + 3\mu_\psi d_{\psi,1} \nu_\psi^{(1)} + d_{\psi,0} \nu_\psi = \\ = k\left(-\tau_\psi^3 \psi^{(3)} - 3\tau_\psi^2 \alpha_\psi \psi^{(2)} - 3\tau_\psi \alpha_\psi^2 \psi^{(1)} - \psi + \psi_0\right) \tag{37}$$

$$\mu_\varphi^3 \nu_\varphi^{(3)} + 3\mu_\varphi^2 d_{\varphi,2} \nu_\varphi^{(2)} + 3\mu_\varphi d_{\varphi,1} \nu_\varphi^{(1)} + d_{\varphi,0} \nu_\varphi = \\ = k\left(-\tau_\varphi^3 \varphi^{(3)} - 3\tau_\varphi^2 \alpha_\varphi \varphi^{(2)} - 3\tau_\varphi \alpha_\varphi^2 \varphi^{(1)} - \varphi + \varphi_0\right) \tag{38}$$

The entire closed loop system is presented in Fig.3.

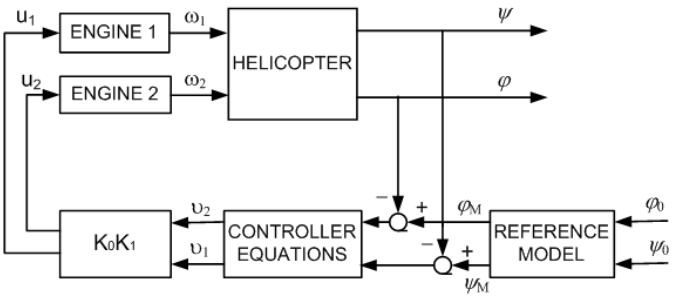


Fig. 3. Closed-loop system

5. Results of control experiments

In this section, we present the results of experiment which was conducted on the helicopter model HUMUSOFT CE150, to evaluate the performance of a designed control system. As the user communicates with the system via Matlab Real Time Toolbox interface, all input/output signals are scaled into the interval <-1,+1>, where value "1" is called Machine Unit and such a signal has no physical dimension. This will be referred in the following text as MU.

The presented maneuver (experiment 1) consisted in transition with predefined dynamics from one steady-state angular position to another. Hereby, the control system accomplished a tracking task of reference signal. The second experiment was chosen to expose a robustness of the controller under transient and steady-state conditions. During the experiment, the entire control system was subjected to external disturbances in the form of a wind gust. Practically this perturbation was realized mechanically by pushing the helicopter body in required direction with suitable force. The helicopter was disturbed twice during the test: $t_1 = 130\ [s]$, $t_2 = 170\ [s]$.

5.1 Experiment 1 – tracking of a reference trajectory

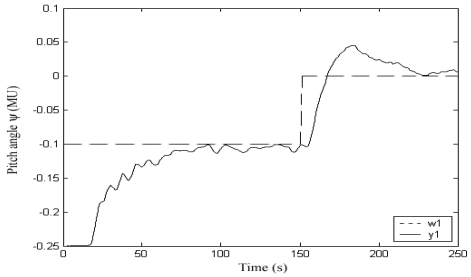


Fig. 4. Time history of pitch angle ψ

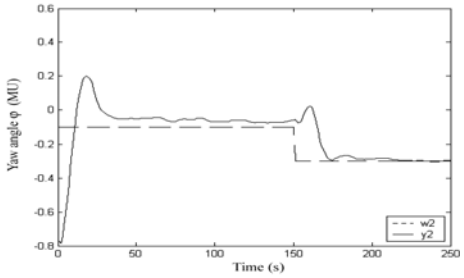


Fig. 5. Time history of yaw angle φ

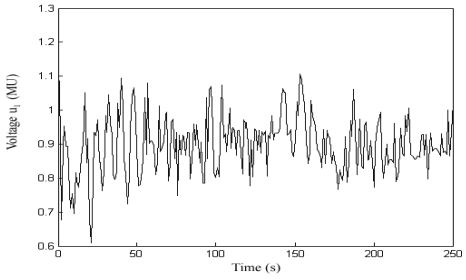


Fig. 6. Time history of main motor voltage u_1

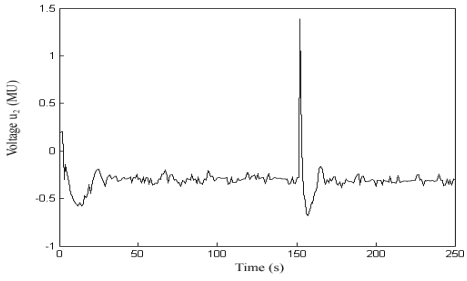


Fig. 7. Time history of tail motor voltage u_2

5.2 Experiment 2 – influence of a wind gust in vertical plane

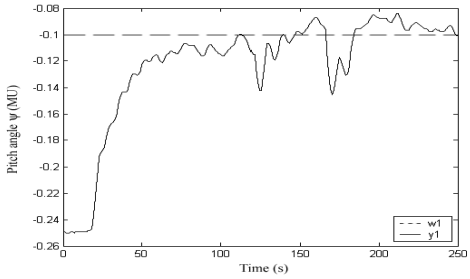


Fig. 8. Time history of pitch angle ψ

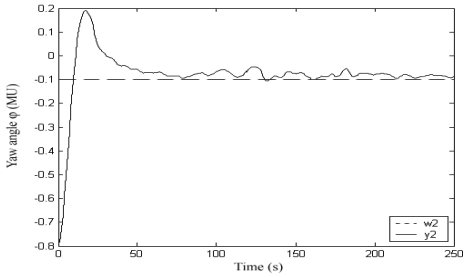


Fig. 9. Time history of yaw angle φ

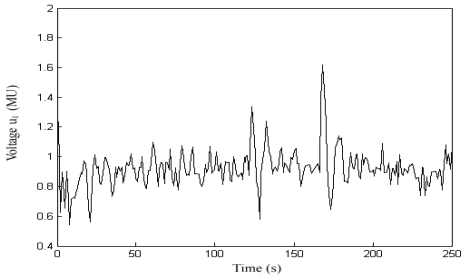


Fig. 10. Time history of main motor voltage u_1

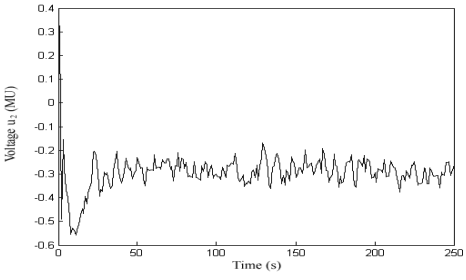


Fig. 11. Time history of tail motor voltage u_2

6. Conclusion

The applied method allows to create the expected outputs for multi-input multi-output nonlinear time-varying physical object, like an exemplary laboratory model of helicopter, and provides independent desired dynamics in control channels. The peculiarity of the propose approach is the application of the higher order derivatives jointly with high gain in the control law. This approach and structure of the control system is the implementation of the model reference control. The resulting controller is a combination of a low-order linear dynamical system and a matrix whose entries depend non-linearly on some known process variables. It becomes that the proposed structure and method is insensitive to external disturbances and also plant parameter changes, and hereby possess a robustness aspects. The results suggest that the approach we were concerned with can be applied in some region of automation, for example in power electronics.

7. Acknowledgements

This work has been granted by the Polish Ministry of Science and Higher Education from funds for years 2008-2011.

8. References

Astrom, K. J. & Wittenmark, B. (1994). *Adaptive control.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Balas, G.; Garrard, W. & Reiner, J. (1995). Robust dynamic inversion for control of highly maneuverable aircraft, *J. of Guidance Control & Dynamics*, Vol. 18, No. 1, pp. 18-24.

Błachuta, M.; Yurkevich, V. D. & Wojciechowski, K. (1999). Robust quasi NID aircraft 3D flight control under sensor noise, *Kybernetika*, Vol. 35, No.5, pp. 637-650.

Castillo, P.; Lozano, R. & Dzul, A. E. (2005). *Modelling and Control of Mini-flying Machines.* Springer-Verlag.

Czyba, R. & Błachuta, M. (2003). Dynamic contraction method approach to robust longitudinal flight control under aircraft parameters variations, *Proceedings of the AIAA Conference*, AIAA 2003-5554, Austin, USA.

Horacek P. (1993). *Helicopter Model CE 150 – Educational Manual*, Czech Technical University in Prague.

Isidori, A. & Byrnes, C. I. (1990). Output regulation of nonlinear systems, *IEEE Trans. Automat. Control*, Vol. 35, pp. 131-140.

Slotine, J. J. & Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs.

Szafrański, G. & Czyba R. (2008). Fast prototyping of three-phase BLDC Motor Controller designed on the basis of Dynamic Contraction Method, *Proceedings of the IEEE 10th International Workshop on Variable Structure Systems,* pp. 100-105, Turkey.

Utkin, V. I. (1992). *Sliding modes in control and optimization*. Springer-Verlag.

Valavanis, K. P. (2007). *Advances in Unmanned Aerial Vehicles.* Springer-Verlag.

Vostrikov, A. S. & Yurkevich, V. D. (1993). Design of control systems by means of Localisation Method, *Preprints of 12-th IFAC World Congress*, Vol. 8, pp. 47-50.

Yurkevich, V. D. (2004). *Design of Nonlinear Control Systems with the Highest Derivative in Feedback.* World Scientific Publishing.

11

Laboratory Experimentation of Guidance and Control of Spacecraft During On-orbit Proximity Maneuvers

Jason S. Hall and Marcello Romano
Naval Postgraduate School
Monterey, CA, USA

1. Introduction

The traditional spacecraft system is a monolithic structure with a single mission focused design and lengthy production and qualification schedules coupled with enormous cost. Additionally, there rarely, if ever, is any designed preventive maintenance plan or re-fueling capability. There has been much research in recent years into alternative options. One alternative option involves autonomous on-orbit servicing of current or future monolithic spacecraft systems. The U.S. Department of Defense (DoD) embarked on a highly successful venture to prove out such a concept with the Defense Advanced Research Projects Agency's (DARPA's) Orbital Express program. Orbital Express demonstrated all of the enabling technologies required for autonomous on-orbit servicing to include refueling, component transfer, autonomous satellite grappling and berthing, rendezvous, inspection, proximity operations, docking and undocking, and autonomous fault recognition and anomaly handling (Kennedy, 2008). Another potential option involves a paradigm shift from the monolithic spacecraft system to one involving multiple interacting spacecraft that can autonomously assemble and reconfigure. Numerous benefits are associated with autonomous spacecraft assemblies, ranging from a removal of significant intra-modular reliance that provides for parallel design, fabrication, assembly and validation processes to the inherent smaller nature of fractionated systems which allows for each module to be placed into orbit separately on more affordable launch platforms (Mathieu, 2005).

With respect specifically to the validation process, the significantly reduced dimensions and mass of aggregated spacecraft when compared to the traditional monolithic spacecraft allow for not only component but even full-scale on-the-ground Hardware-In-the-Loop (HIL) experimentation. Likewise, much of the HIL experimentation required for on-orbit servicing of traditional spacecraft systems can also be accomplished in ground-based laboratories (Creamer, 2007). This type of HIL experimentation complements analytical methods and numerical simulations by providing a low-risk, relatively low-cost and potentially high-return method for validating the technology, navigation techniques and control approaches associated with spacecraft systems. Several approaches exist for the actual HIL testing in a laboratory environment with respect to spacecraft guidance, navigation and control. One

such method involves reproduction of the kinematics and vehicle dynamics for 3-DoF (two horizontal translational degrees and one rotational degree about the vertical axis) through the use of robotic spacecraft simulators that float via planar air bearings on a flat horizontal floor. This particular method is currently being employed by several research institutions and is the validation method of choice for our research into GNC algorithms for proximity operations at the Naval Postgraduate School (Machida et al., 1992; Ullman, 1993; Corrazzini & How, 1998; Marchesi et al., 2000; Ledebuhr et al., 2001; Nolet et al., 2005; LeMaster et al., 2006; Romano et al., 2007). With respect to spacecraft involved in proximity operations, the in-plane and cross-track dynamics are decoupled, as modeled by the Hill-Clohessy-Wiltshire (HCW) equations, thus the reduction to 3-Degree of Freedom (DoF) does not appear to be a critical limiter. One consideration involves the reduction of the vehicle dynamics to one of a double integrator. However, the orbital dynamics can be considered to be a disturbance that needs to be compensated for by the spacecraft navigation and control system during the proximity navigation and assembly phase of multiple systems. Thus the flat floor testbed can be used to capture many of the critical aspects of an actual autonomous proximity maneuver that can then be used for validation of numerical simulations. Portions of the here-in described testbed, combined with the first generation robotic spacecraft simulator of the Spacecraft Robotics Laboratory (SRL) at Naval Postgraduate School (NPS), have been employed to propose and experimentally validate control algorithms. The interested reader is referred to (Romano et al., 2007) for a full description of this robotic spacecraft simulator and the associated HIL experiments involving its demonstration of successful autonomous spacecraft approach and docking maneuvers to a collaborative target with a prototype docking interface of the Orbital Express program.

Given the requirement for spacecraft aggregates to rendezvous and dock during the final phases of assembly and a desire to maximize the useable surface area of the spacecraft for power generation, sensor packages, docking mechanisms and payloads while minimizing thruster impingement, control of such systems using the standard control actuator configuration of fixed thrusters on each face coupled with momentum exchange devices can be challenging if not impossible. For such systems, a new and unique configuration is proposed which may capitalize, for instance, on the recently developed carpal robotic joint invented by Dr. Steven Canfield with its hemispherical vector space (Canfield, 1998). It is here demonstrated through Lie algebra analytical methods and experimental results that two vectorable in-plane thrusters in an opposing configuration can yield a minimum set of actuators for a controllable system. It will also be shown that by coupling the proposed set of vectorable thrusters with a single degree of freedom Control Moment Gyroscope, an additional degree of redundancy can be gained. Experimental results are included using SRL's second generation reduced order (3 DoF) spacecraft simulator. A general overview of this spacecraft simulator is presented in this chapter (additional details on the simulators can be found in: Hall, 2006; Eikenberry, 2006; Price, W., 2006; Romano & Hall, 2006; Hall & Romano, 2007a; Hall & Romano, 2007b).

While presenting an overview of a robotic testbed for HIL experimentation of guidance and control algorithms for on-orbit proximity maneuvers, this chapter specifically focuses on exploring the feasibility, design and evaluation in a 3-DoF environment of a vectorable thruster configuration combined with optional miniature single gimbaled control moment gyro (MSGCMG) for an agile small spacecraft. Specifically, the main aims are to present and practically confirm the theoretical basis of small-time local controllability for this unique

actuator configuration through both analytical and numerical simulations performed in previous works (Romano & Hall, 2006; Hall & Romano, 2007a; Hall & Romano, 2007b) and to validate the viability of using this minimal control actuator configuration on a small spacecraft in a practical way. Furthermore, the experimental work is used to confirm the controllability of this configuration along a fully constrained trajectory through the employment of a smooth feedback controller based on state feedback linearization and linear quadratic regulator techniques and proper state estimation methods. The chapter is structured as follows: First the design of the experimental testbed including the floating surface and the second generation 3-DoF spacecraft simulator is introduced. Then the dynamics model for the spacecraft simulator with vectorable thrusters and momentum exchange device are formulated. The controllability concerns associated with this uniquely configured system are then addressed with a presentation of the minimum number of control inputs to ensure small time local controllability. Next, a formal development is presented for the state feedback linearized controller, state estimation methods, Schmitt trigger and Pulse Width Modulation scheme. Finally, experimental results are presented.

2. The NPS Robotic Spacecraft Simulator Testbed

Three generations of robotic spacecraft simulators have been developed at the NPS Spacecraft Robotics Laboratory, in order to provide for relatively low-cost HIL experimentation of GNC algorithms for spacecraft proximity maneuvers (see Fig.1). In particular, the second generation robotic spacecraft simulator testbed is used for the here-in presented research. The whole spacecraft simulator testbed consists of three components. The two components specifically dedicated to HIL experimentation in 3-DoF are a floating surface with an indoor pseudo-GPS (iGPS) measurement system and one 3-DoF autonomous spacecraft simulator. The third component of the spacecraft simulator testbed is a 6-DoF simulator stand-alone computer based spacecraft simulator and is separated from the HIL components. Additionally, an off-board desktop computer is used to support the 3-DoF spacecraft simulator by providing the capability to upload software, initiate experimental testing, receive logged data during testing and process the iGPS position coordinates. Fig. 2 depicts the robotic spacecraft simulator in the Proximity Operations Simulator Facility (POSF) at NPS with key components identified. The main testbed systems are briefly described in the next sections with further details given in (Hall, 2006; Price, 2006; Eikenberry, 2006; Romano & Hall, 2006; Hall & Romano, 2007a; Hall & Romano 2007b).

Fig. 1. Three generations of spacecraft simulator at the NPS Spacecraft Robotics Laboratory (first, second and third generations from left to right)

2.1 Floating Surface

A 4.9 m by 4.3 m epoxy floor surface provides the base for the floatation of the spacecraft simulator. The use of planar air bearings on the simulator reduces the friction to a negligible level and with an average residual slope angle of approximately 2.6×10^{-3} deg for the floating surface, the average residual acceleration due to gravity is approximately 1.8×10^{-3} ms^{-2}. This value of acceleration is 2 orders of magnitude lower than the nominal amplitude of the measured acceleration differences found during reduced gravity phases of parabolic flights (Romano et al, 2007).

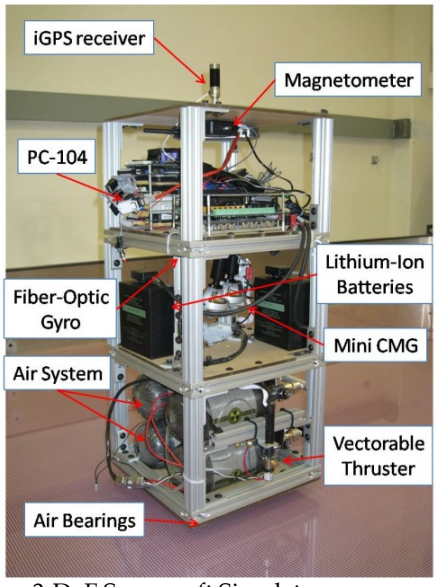


Fig. 2. SRL's 2nd Generation 3-DoF Spacecraft Simulator

2.2 3-DoF Robotic Spacecraft Simulator

SRL's second generation robotic spacecraft simulator is modularly constructed with three easily assembled sections dedicated to each primary subsystem. Prefabricated 6105-T5 Aluminum fractional t-slotted extrusions form the cage of the vehicle while one square foot, .25 inch thick static dissipative rigid plastic sheets provide the upper and lower decks of each module. The use of these materials for the basic structural requirements provides a high strength to weight ratio and enable rapid assembly and reconfiguration. Table 1 reports the key parameters of the 3-DoF spacecraft simulator.

2.2.1 Propulsion and Flotation Subsystems

The lowest module houses the flotation and propulsion subsystems. The flotation subsystem is composed of four planar air bearings, an air filter assembly, dual 4500 PSI (31.03 MPa) carbon-fiber spun air cylinders and a dual manifold pressure reducer to provide 75 PSI (.51 MPa). This pressure with a volume flow rate for each air bearing of 3.33 slfm (3.33×10^{-3} m^3/min) is sufficient to keep the simulator in a friction-free state for nearly 40 minutes of continuous experimentation time. The propulsion subsystem is composed of dual vectorable supersonic on-off cold-gas thrusters and a separate dual carbon-fiber spun air cylinder and pressure reducer package regulated at 60 PSI (.41 MPa) and has the capability of providing the system 31.1 m/s ΔV .

2.2.2 Electronic and Power Distribution Subsystems

The power distribution subsystem is composed of dual lithium-ion batteries wired in parallel to provide 28 volts for up to 12 Amp-Hours and is housed in the second deck of the simulator. A four port DC-DC converter distributes the requisite power for the system at 5, 12 or 24 volts DC. An attached cold plate provides heat transfer from the array to the power system mounting deck in the upper module. The current power requirements include a single PC-104 CPU stack, a wireless router, three motor controllers, three separate normally-closed solenoid valves for thruster and air bearing actuation, a fiber optic gyro, a magnetometer and a wireless server for transmission of the vehicle's position via the pseudo-GPS system.

Subsystem	Characteristic	Parameter
Structure	Length and width	.30 m
	Height	.69 m
	Mass (Overall)	26 kg
	J_z (Overall)	.40 kg-m^2
Propulsion	Propellant	Compressed Air
	Equiv. storage capacity	.05 m^3 @ 31.03 Mpa
	Operating pressure	.41 Mpa
	Thrust (x2)	.159 N
	ISP	34.3 s
	Total ΔV	31.1 m/s
Flotation	Propellant	Air
	Equiv. storage capacity	0.05 m^3 @ 31.03 Mpa
	Operating pressure	.51 Mpa
	Linear air bearing (x4)	32 mm diameter
	Continuous operation	~40 min
CMG Attitude Control	Max torque	.668 Nm
	Momentum storage	.098 Nms
Electrical & Electronic	Battery type	Lithium-Ion
	Storage capacity	12 Ah @ 28V
	Continuous Operation	~6 h
	Computer	1 PC104 Pentium III
Sensors	Fiber optic gyro	KVH Model DSP-3000
	Position sensor	Metris iGPS
	Magnetometer	MicroStrain 3DM-GX1

Table 1. Key Parameters of the 2nd generation 3-DoF Robotic Spacecraft Simulator

2.2.3 Translation and Attitude Control System Actuators

The 3-DoF robotic spacecraft simulator includes actuators to provide both translational control and attitude control. A full development of the controllability for this unique configuration of dual rotating thrusters and one-axis Miniature-Single Gimbaled Control Moment Gyro (MSGCMG) will be demonstrated in subsequent sections of this paper. The translational control is provided by two cold-gas on-off supersonic nozzle thrusters in a dual vectorable configuration. Each thruster is limited in a region $\pm\pi/2$ with respect to the face normal and, through experimental testing at the supplied pressure, has been demonstrated to have an ISP of 34.3 s and able to provide .159 N of thrust with less than 10 msec actuation time (Lugini, 2008). The MSGCMG is capable of providing .668 Nm of torque with a maximum angular momentum of .098 Nms.

2.3 6-DoF Computer-Based Numerical Spacecraft Simulator

A separate component of SRL's spacecraft simulator testbed at NPS is a 6-DoF computer-based spacecraft simulator. This simulator enables full 6-DoF numerical simulations to be conducted with realistic orbital perturbations including aerodynamic, solar pressure and

third-body effects, and earth oblateness up to J4. Similar to the 3-DoF robotic simulator, the numerical simulator is also modularly designed within a MATLAB®/Simulink® architecture to allow near seamless integration and testing of developed guidance and control algorithms. Additionally, by using the MATLAB®/Simulink® architecture with the added Real Time Workshop™ toolbox, the developed control algorithms can be readily transitioned into C-code for direct deployment onto the 3-DoF robotic simulator's onboard processor. A full discussion of the process by which this is accomplished and simplified for rapid real-time experimentation on the 3-DoF testbed for either the proprietary MATLAB® based XPCTarget™ operating system is given in (Hall, 2006; Price, 2006) or for an open-source Linux based operating system with the Real Time Application Interface (RTAI) is given in (Bevilacqua et al., 2009).

3. Dynamics of a 3-DoF Spacecraft Simulator with Vectorable Thrusters and Momentum Exchange Device

Two sets of coordinate frame are established for reference: the inertial coordinate system (ICS) designated by *XYZ* and body-fixed coordinate system (BCS) designated by *xyz*. These reference frames are depicted in **Fig. 3** along with the necessary external forces and parameters required to properly define the simulators motion. The origin of the body-fixed coordinate system is taken to be the center of mass *C* of the spacecraft simulator and this is assumed to be collocated with the simulator's geometric center. The body *z*-axis is aligned with the inertial *Z*-axis while the body *x*-axis is in line with the thrusters points of action. In the ICS, the position and velocity vectors of *C* are given by $\mathbf{X}$ and $\mathbf{V}$ so that $\mathbf{X}(X,Y)$ marks the position of the simulator with respect to the origin of the ICS as measured by the inertial measurement sensors and provides the vehicle's two degrees of translational freedom. The vehicle's rotational freedom is described by an angle of rotation ψ between the *x*-axis and the *X*-axis about the *z*-axis. The angular velocity is thus limited to one degree of freedom and is denoted by ω_z. The spacecraft simulator is assumed to be rigid and therefore a constant moment of inertia (J_z) exists about the *z*-axis. Furthermore, any changes to the mass of the simulator (m) due to thruster firing are neglected.

The forces imparted at a distance L from the center of mass by the vectorable on-off thrusters are denoted by F_1 and F_2 respectively. The direction of the thrust vector $\mathbf{F}_1$ is determined by α_1 which is the angle measured from the outward normal of face one in a clockwise direction (right-hand rotation) to where thruster one's nozzle is pointing. Likewise, the direction of the thrust vector $\mathbf{F}_2$ is determined by α_2 which is the angle measured from the outward normal of face two in a clockwise direction (right-hand rotation) to where thruster two's nozzle is pointing. The torque imparted on the vehicle by a momentum exchange device such as a control moment gyro is denoted by T_{MED} and can be constrained to exist only about the yaw axis as demonstrated in (Hall, 2006; Romano & Hall, 2006).

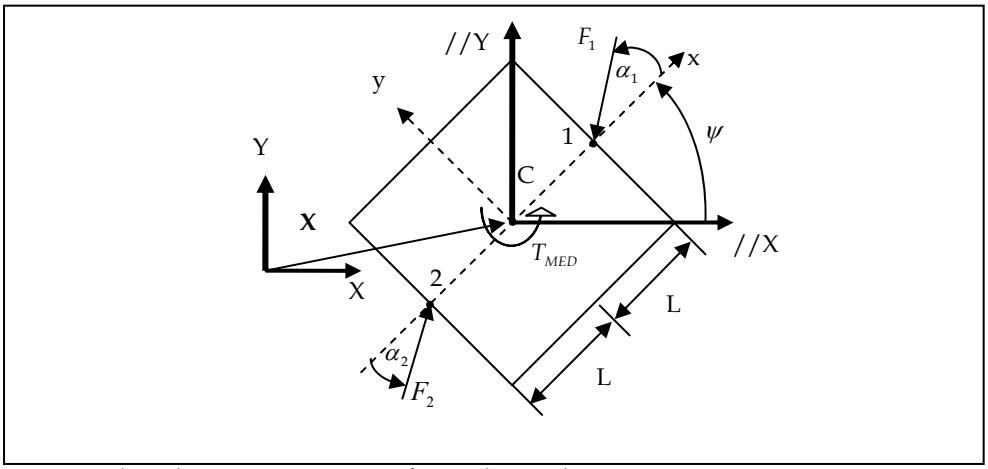


Fig. 3. SRL's 2nd Generation Spacecraft Simulator Schematic

The translation and attitude motion of the simulator are governed by the equations

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{V} \\ \dot{\mathbf{V}} &= m^{-1}\,{}^{I}\mathbf{R}_{B}(\psi)\,{}^{B}\mathbf{F} \\ \dot{\psi} &= \omega_z \\ \dot{\omega}_z &= J_z^{-1}\,{}^{B}T \end{aligned} \tag{1}$$

where ${}^{B}\mathbf{F} \in \mathbb{R}^2$ are the thruster inputs limited to the region $\pm\pi/2$ with respect to each face normal and ${}^{B}T \in \mathbb{R}$ is the attitude input. ${}^{I}\mathbf{R}_{B}(\psi)$, ${}^{B}\mathbf{F}$ and ${}^{B}T$ are given by

$${}^{I}\mathbf{R}_{B}(\psi) = \begin{bmatrix} \mathrm{c}\psi & -\mathrm{s}\psi \\ \mathrm{s}\psi & \mathrm{c}\psi \end{bmatrix} \tag{2}$$

$${}^{B}\mathbf{F}^{T} = {}^{B}\mathbf{F}_1^{T} + {}^{B}\mathbf{F}_2^{T} = \left[-F_1\,\mathrm{c}\alpha_1 + F_2\,\mathrm{c}\alpha_2, -F_1\,\mathrm{s}\alpha_1 + F_2\,\mathrm{s}\alpha_2\right]^{T} \tag{3}$$

$${}^{B}T = \left[T_{MED} + L\left(-F_1\,\mathrm{s}\alpha_1 - F_2\,\mathrm{s}\alpha_2\right)\right] \tag{4}$$

where $\mathrm{s}\bullet = \sin(\bullet)$, $\mathrm{c}\bullet = \cos(\bullet)$.

The internal dynamics of the vectorable thrusters are assumed to be linear according to the following equations

$$\dot{\alpha}_1 = \beta_1,\ \dot{\beta}_1 = J_1^{-1}T_1,\ \dot{\alpha}_2 = \beta_2,\ \dot{\beta}_2 = J_2^{-1}T_2 \tag{5}$$

where J_1 and J_2 represent the moments of inertia about each thruster rotational axis respectively and $T_1 \in \mathbb{R}$, $T_2 \in \mathbb{R}$ represent the corresponding thruster rotation control input.

The system's state equation given by Eq. (1) can be rewritten in control-affine system form as (LaValle, 2006)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^{N_u} u_i \mathbf{g}_i(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u},\ x \in \mathbb{R}^{N_x} \tag{6}$$

where N_u is the number of controls. With $\mathbb{R}^{N_x}$ representing a smooth N_x-dimensional manifold defined be the size of the state-vector and the control vector to be in $\mathbb{R}^{N_u}$. Defining the state vector $\mathbf{x} \in \mathbb{R}^{10}$ as $\mathbf{x}^T = [x_1, x_2, ..., x_{10}] = [X, Y, \psi, \alpha_1, \alpha_2, V_x, V_y, \omega_z, \beta_1, \beta_2]$ and the control vector $\mathbf{u} \in \mathbb{U}^5$ as $\mathbf{u}^T = [u_1, u_2, ..., u_5] = [F_1, F_2, T_{MED}, T_1, T_2]$, the system's state equation, becomes

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u} = [x_6, x_7, x_8, x_9, x_{10}, \mathbf{0}_{1x5}]^T + \begin{bmatrix} \mathbf{0}_{5x5} \\ G_1(\mathbf{x}) \end{bmatrix} \mathbf{u} \tag{7}$$

where the matrix $G_1(\mathbf{x})$ is obtained from Eq. (1) as

$$G_1(\mathbf{x}) = \begin{bmatrix} -m^{-1}[\mathrm{c}x_3\,\mathrm{c}x_4 - \mathrm{s}x_3\,\mathrm{s}x_4] & m^{-1}[\mathrm{c}x_3\,\mathrm{c}x_5 - \mathrm{s}x_3\,\mathrm{s}x_5] & 0 & 0 & 0 \\ -m^{-1}[\mathrm{c}x_3\,\mathrm{s}x_4 + \mathrm{s}x_3\,\mathrm{c}x_4] & m^{-1}[\mathrm{c}x_3\,\mathrm{s}x_5 + \mathrm{s}x_3\,\mathrm{c}x_5] & 0 & 0 & 0 \\ -J_z^{-1}\,\mathrm{s}x_4 L & -J_z^{-1}\,\mathrm{s}x_5 L & J_z^{-1} & 0 & 0 \\ 0 & 0 & 0 & J_2^{-1} & 0 \\ 0 & 0 & 0 & 0 & J_2^{-1} \end{bmatrix} \tag{8}$$

With the system in the form of Eq. (6) given the vector fields in Eqs. (7) and (8), and given that $\mathbf{f}(\mathbf{x})$ (the drift term) and $G(\mathbf{x})$ (the control matrix of control vector fields) are smooth functions, it is important to note that it is not necessarily possible to obtain zero velocity due to the influence of the drift term. This fact places the system in the unique subset of control-affine systems with drift and, as seen later, will call for an additional requirement for determining the controllability of the system. Furthermore, when studying controllability of systems, the literature to date restricts the consideration to cases where the control is proper. Having a proper control implies that the affine hull of the control space is equal to $\mathbb{R}^{N_u}$ or that the smallest subspace of $\mathbb{U}$ is equal to the number of control vectors and that it is closed (Sussman, 1987; Sussman, 1990; Bullo & Lewis, 2005; LaValle, 2006). With a system such as a spacecraft in general or the simplified model of the 3-DoF simulator in particular, the use of on-off cold-gas thrusters restrict the control space to only positive space with respect to both thrust vectors leading to an unclosed set and thus improper control space. In order to overcome this issue, a method which leverages the symmetry of the system is used by which the controllability of the system is studied by considering only one virtual rotating thruster that is positioned a distance L from the center of mass with the vectored thrust resolved into a y and x-component. In considering this system perspective, the thruster combination now spans $\mathbb{R}^2$ and therefore is proper and is analogous to the planar body with variable-direction force vector considered in (Lewis & Murray, 1997; Bullo & Lewis, 2005). Furthermore, under the assumption that the control bandwidth of the thrusters's rotation is much larger than the control bandwidth of the system dynamics, the internal dynamics of the vectorable thrusters can be decoupled from the state and control vectors for the system yielding a thrust vector dependent on simply a commanded angle. Thus the system's state vector, assuming that both thrusters and a momentum exchange device are available,

becomes $\mathbf{x}^T = [x_1, x_2, ..., x_6] = [X, Y, \psi, V_X, V_Y, \omega_z] \in \mathbb{R}^6$ and the control vector is $\mathbf{u}^T = [u_1, u_2, u_3] = [{}^B F_x, {}^B F_y, {}^B T_z] = \mathbb{U}^3$ so that the system's state equation becomes

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u} = [x_4, x_5, x_6, 0, 0, 0]^T + \begin{bmatrix} \mathbf{0}_{3x3} \\ G_1(\mathbf{x}) \end{bmatrix} \mathbf{u} \tag{9}$$

where the matrix $G_1(\mathbf{x})$ can be obtained by considering the relation of the desired control vector to the body centered reference system, in the two cases of positive force needed in the x direction (${}^B U_x > 0$) and negative force needed in the x direction (${}^B U_x < 0$). In this manner, the variables in Eq. (8) and Eq. (9) can be defined as

$$\begin{aligned} {}^B U_x < 0 \rightarrow & \begin{cases} \mathbf{u}^T = [{}^B F_x, {}^B F_y, {}^B T_z] = [-F_1 \mathrm{c}\, x_4, -F_1 \mathrm{s}\, x_4, T_{MED}] \\ d = -L, F_2 = 0 \end{cases} \\ {}^B U_x > 0 \rightarrow & \begin{cases} \mathbf{u}^T = [{}^B F_x, {}^B F_y, {}^B T_z] = [F_2 \mathrm{c}\, x_5, F_2 \mathrm{s}\, x_5, T_{MED}] \\ d = L, F_1 = 0 \end{cases} \end{aligned} \tag{10}$$

yielding the matrix in $G_1(\mathbf{x})$ through substitution into Eq. (8) as

$$G_1(\mathbf{x}) = \begin{bmatrix} m^{-1} \mathrm{c}\, x_3 & -m^{-1} \mathrm{s}\, x_3 & 0 \\ m^{-1} \mathrm{s}\, x_3 & m^{-1} \mathrm{c}\, x_3 & 0 \\ 0 & -d J_z^{-1} & J_z^{-1} \end{bmatrix} \tag{11}$$

When the desired control input to the system along the body *x*-axis is zero, both thrusters can be used to provide a control force along the *y*-axis, while a momentum exchange device provides any required torque. In this case, the control vector in (9) becomes $\mathbf{u}^T = [u_1, u_2] = [{}^B F_y, {}^B T_z] = \mathbb{U}^2$ such that the variables in Eq. (8) and (9) can be defined as

$${}^B U_x = 0 \rightarrow \begin{cases} \mathbf{u}^T = [{}^B F_y, {}^B T_z] = [F s\alpha, T_{MED}] \\ F = F_1 = F_2, \alpha = x_4 = -x_5 = -\dfrac{\pi}{2} sign({}^B U_y) \end{cases} \tag{12}$$

which yields the matrix $G_1(\mathbf{x})$ through substitution into Eq. (8) as

$$G_1(\mathbf{x}) = \begin{bmatrix} -2m^{-1} \mathrm{s}\, x_3 & 0 \\ 2m^{-1} \mathrm{c}\, x_3 & 0 \\ 0 & J_z^{-1} \end{bmatrix} \tag{13}$$

As will be demonstrated in later, the momentum exchange device is not necessary to ensure small time controllability for this system. In considering this situation, which also occurs when a control moment gyroscope is present but is near the singular conditions and therefore requires desaturation, the thruster not being used for translation control can be slewed to $\pm\pi/2$ depending on the required torque compensation and fired to affect the desired angular rate change. The desired control input to the system with respect to the body x-axis $\left({}^{B}U_{x}\right)$ can again be used to define the desired variables such that

$$\begin{aligned} {}^{B}U_{x} \le 0 \rightarrow & \begin{cases} \mathbf{u}^{T} = \left[{}^{B}F_{x}, {}^{B}F_{y}, {}^{B}T_{z}\right] = \left[-F_{1}\,\mathrm{c}x_{4}, -F_{1}\,\mathrm{s}x_{4}, F_{2}d\,\mathrm{s}x_{5}\right] \\ d = -L,\ x_{5} = \pm\dfrac{\pi}{2} \end{cases} \\ {}^{B}U_{x} > 0 \rightarrow & \begin{cases} \mathbf{u}^{T} = \left[{}^{B}F_{x}, {}^{B}F_{y}, {}^{B}T_{z}\right] = \left[F_{2}\,\mathrm{c}x_{5}, F_{2}\,\mathrm{s}x_{5}, -F_{1}d\,\mathrm{s}x_{4}\right] \\ d = L,\ x_{4} = \pm\dfrac{\pi}{2} \end{cases} \end{aligned} \tag{14}$$

which yields the matrix $G_{1}(\mathbf{x})$ through substitution into Eq. (8)as

$$G_{1}(\mathbf{x}) = \begin{bmatrix} m^{-1}\,\mathrm{c}x_{3} & -m^{-1}\,\mathrm{s}x_{3} & -(md)^{-1}\,\mathrm{s}x_{3} \\ m^{-1}\,\mathrm{s}x_{3} & m^{-1}\,\mathrm{c}x_{3} & (md)^{-1}\,\mathrm{c}x_{3} \\ 0 & -dJ_{z}^{-1} & J_{z}^{-1} \end{bmatrix} \tag{15}$$

In case of zero force requested along x with only thrusters acting, the system cannot in general provide the requested torque value.
A key design consideration with this type of control actuator configuration is that with only the use of an on/off rotating thruster to provide the necessary torque compensation, fine pointing can be difficult and more fuel is required to affect a desired maneuver involving both translation and rotation.

4. Small-Time Local Controllability

Before studying the controllability for a nonlinear control-affine system of the form in Eq. (6), it is important to review several definitions. First, the set of states reachable in time at most T is given by $R_{\Sigma}(\mathbf{x}_{0}, \le T)$ by solutions of the nonlinear control-affine system.

Definition 1 (Accessibility)
A system is *accessible* from $\mathbf{x}_{0}$ (the initial state) if there exists $T > 0$ such that the interior of $R_{\Sigma}(\mathbf{x}_{0}, \le t)$ is not an empty set for $t \in]0, T]$ (Bullo & Lewis, 2005).

Definition 2 (Proper Small Time Local Controllability)
A system is *small time locally controllable* (STLC) from $\mathbf{x}_{0}$ if there exists $T > 0$ such that $\mathbf{x}_{0}$ lies in the interior of $R_{\Sigma}(\mathbf{x}_{0}, \le t)$ for each $t \in]0, T]$ for every proper control set U (Bullo &

Lewis, 2005). Assuming that at $\mathbf{x}(0)=\mathbf{0}$ this can also be seen under time reversal as the equilibrium for the system x_0 can be reached from a neighborhood in small time (Sussman, 1987; Sussman, 1990).

Definition 3 (Proper Control Set) A control set $\mathbf{u}^T=[u_1,...,u_k]$ is termed to be *proper* if the set satisfies a constraint $\mathbf{u}\in K$ where K affinely spans $\mathbb{U}^k$. (Sussman, 1990; Bullo & Lewis, 2005; LaValle, 2006).

Definition 4 (Lie derivative) The *Lie derivative* of a smooth scalar function $g(\mathbf{x})\in\mathbb{R}$ with respect to a smooth vector field $\mathbf{f}(\mathbf{x})\in\mathbb{R}^{N_x}$ is a scalar function defined as (Slotine, 1991, pg. 229)

$$L_{\mathbf{f}}g=\nabla g\,\mathbf{f}=\left[\frac{\partial g}{\partial x_1}...\frac{\partial g}{\partial x_{N_x}}\right]\begin{bmatrix}f_1(\mathbf{x})\\ \vdots\\ f_{N_x}(\mathbf{x})\end{bmatrix}. \tag{16}$$

Definition 5 (Lie Bracket): The Lie bracket of two vector fields $\mathbf{f}(\mathbf{x})\in\mathbb{R}^{N_x}$ and $\mathbf{g}(\mathbf{x})\in\mathbb{R}^{N_x}$ is a third vector field $[\mathbf{f},\mathbf{g}]\in\mathbb{R}^{N_x}$ defined by $[\mathbf{f},\mathbf{g}]=\nabla\mathbf{g}\,\mathbf{f}-\nabla\mathbf{f}\,\mathbf{g}$, where the *i*-th component can be expressed as (Slotine, 1991)

$$[\mathbf{f},\mathbf{g}]_i=\sum_{j=1}^{N_x}\left(f_j\frac{\partial g_i}{\partial x_j}-g_j\frac{\partial f_i}{\partial x_j}\right). \tag{17}$$

Using Lie bracketing methods which produce motions in directions that do not seem to be allowed by the system distribution, sufficient conditions can be met to determine a system's STLC even in the presence of a drift vector as in the equations of motion developed above. These sufficient conditions involve the Lie Algebra Rank Condition (LARC).

Definition 6 (Associated Distribution $\Delta(\mathbf{x})$) Given a system as in Eq. (6), the associated distribution $\Delta(\mathbf{x})$ is defined as the vector space (subspace of $\mathbb{R}^{N_x}$) spanned by the system vector fields $\mathbf{f},\mathbf{g}_1,...\mathbf{g}_{N_u}$.

Definition 7 The Lie algebra of the associated distribution $\mathcal{L}(\Delta)$ is defined to be the distribution of all independent vector fields that can be obtained by applying subsequent Lie bracket operations to the system vector fields. Of note, no more than N_x vector fields can be produced (LaValle, 2006). With $\dim(\mathcal{L}(\Delta))\le N_x$,the computation of the elements of $\mathcal{L}(\Delta)$ ends either when N_x independent vector fields are obtained or when all subsequent Lie brackets are vector fields of zeros.

Definition 8 (Lie Algebra Rank Condition (LARC)) The *Lie Algebra Rank Condition* is satisfied at a state $\mathbf{x}$ if the rank of the matrix obtained by concatenating the vector fields of the Lie algebra distribution at $\mathbf{x}$ is equal to N_x (the number of state).

For a driftless control-affine system, following the Chow-Rashevskii Theorem, the system is STLC if the LARC is satisfied (Lewis & Murray, 1997; Bullo & Lewis, 2005; LaValle, 2006). However, given a system with drift, in order to determine the STLC, the satisfaction of the

LARC it is not sufficient: in addition to the LARC, it is necessary to examine the combinations of the vectors used to compose the Lie brackets of the Lie algebra. From Sussman's General Theorem on Controllability, if the LARC is satisfied and if there are no ill formed brackets in $\mathcal{L}(\Delta)$, then the system is STLC from its equilibrium point (Sussman, 1987). The Sussman's theorem, formally stated is reported here below.

Theorem 1 (Sussman's General Theorem on Controllability) Consider a system given by Eq. (6) and an equilibrium point $\mathbf{p} \in \mathbb{R}^{N_x}$ such that $\mathbf{f}(\mathbf{p}) = \mathbf{0}$. Assume $\mathcal{L}(\Delta)$ satisfies the LARC at $\mathbf{p}$. Furthermore, assume that whenever a potential Lie bracket consists of the drift vector $\mathbf{f}(\mathbf{x})$ appearing an odd number of times while $\mathbf{g}_1(\mathbf{x}),...,\mathbf{g}_{N_u}(\mathbf{x})$ all appear an even number of times to include zero times (indicating an ill formed Lie bracket), there are sufficient successive Lie brackets to overcome this ill formed Lie bracket to maintain LARC. Then the system is STLC from $\mathbf{p}$. (Sussman, 1987; Sussman, 1990).

As it is common in literature, an ill formed bracket is dubbed a "bad" bracket (Sussman, 1987; Sussman, 1990; Lewis & Murray, 1997, Bullo & Lewis, 2005; LaValle, 2006). Conversely, if a bracket is not "bad", it is termed "good". As an example, for a system with a drift vector and two control vectors, the bracket $\left[\mathbf{f},\left[\mathbf{g}_1,\mathbf{g}_1\right]\right]$ is bad, as the drift vector occurs only once while the first control vector appears twice and the second control vector appears zero times. Similarly, the bracket $\left[\mathbf{f},\left[\mathbf{f},\left[\mathbf{f},\mathbf{g}_1\right]\right]\right]$ is good as the first control vector appears only once. Therefore, it can be summarized that if the rank of the Lie algebra of a control-affine system with drift is equal to the number of states and there exist sufficient "good" brackets to overcome the "bad" brackets to reach the required LARC rank, then the system is small time locally controllable.

4.1 Small-Time Local Controllability Considerations for the 3-DoF Spacecraft Simulator

The concept of small time local controllability is better suitable than the one of accessibility for the problem of spacecraft rendezvous and docking, as a spacecraft is required to move in any directions in a small interval of time dependent on the control actuator capabilities (e.g. to avoid obstacles). The finite time T can be arbitrary if the control input is taken to be unbounded and proper (Sussman, 1990; Bullo & Lewis, 2005; LaValle, 2006).

While no theory yet exists for the study of the general controllability for a non-linear system, the STLC from an equilibrium condition can be studied by employing Sussman's theorem. For the case of spacecraft motion, in order to apply Sussman's theorem, we hypothesize that the spacecraft is moving from an initial condition with velocity close to zero (relative to the origin of an orbiting reference frame).

In applying Sussman's General Theorem on Controllability to the reduced system equations of motion presented in Eq. (9) with $G_1(\mathbf{x})$ given in Eq. (11), the Lie algebra evaluates to

$$\mathcal{L}(\Delta) = span\left\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \left[\mathbf{f},\mathbf{g}_1\right], \left[\mathbf{f},\mathbf{g}_2\right], \left[\mathbf{f},\mathbf{g}_3\right]\right\} \tag{18}$$

so that $\dim\left(\mathcal{L}(\Delta)\right) = N_x = 6$. In order to verify that this is the minimum number of actuators required to ensure STLC, the Lie algebra is reinvestigated for each possible combination of controls. The resulting analysis, as summarized in **Table 2**, demonstrates that the system is STLC from the systems equilibrium point at $\mathbf{x}_0 = \mathbf{0}$ given either two rotating thrusters in complementary semi-circle planes or fixed thrusters on opposing faces providing a normal force vector to the face in opposing directions and a momentum exchange device about the center of mass. For instance, in considering the case of control inputs ${}^{B}F_y, {}^{B}T_z = T_{MED}$, Eq. (9) becomes

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})u_1 + \mathbf{g}_2(\mathbf{x})u_2 \\ &= \left[x_4, x_5, x_6, 0, 0, 0\right]^T + \left[0, 0, 0, m^{-1}sx_3, m^{-1}cx_3, J_z^{-1}L\right]^T u_1 + \left[0, 0, 0, 0, 0, J_z^{-1}\right]^T u_2\end{aligned} \tag{19}$$

where $\mathbf{u} = \left[u_1, u_2\right] = \left[{}^{B}F_y, {}^{B}T_z\right] \in \mathbb{U}^2$. The equilibrium point $\mathbf{p}$ such that $\mathbf{f}(\mathbf{p}) = \mathbf{0}$ is $\mathbf{p} = \left[x_1, x_2, x_3, 0, 0, 0\right]^T$. The $\mathcal{L}(\Delta)$ is formed by considering the associated distribution $\Delta(\mathbf{x})$ and successive Lie brackets as

$\mathbf{f}$,	$\mathbf{g}_1$,	$\mathbf{g}_2$
$[\mathbf{f},\mathbf{g}_1]$,	$[\mathbf{g}_1,\mathbf{g}_2]$,	$[\mathbf{f},\mathbf{g}_2]$
$[\mathbf{f},[\mathbf{f},\mathbf{g}_1]]$,	$[\mathbf{f},[\mathbf{g}_1,\mathbf{g}_2]]$,	$[\mathbf{f},[\mathbf{f},\mathbf{g}_2]]$,
$[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]]$,	$[\mathbf{g}_1,[\mathbf{g}_1,\mathbf{g}_2]]$,	$[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_2]]$,
$[\mathbf{g}_2,[\mathbf{f},\mathbf{g}_1]]$,	$[\mathbf{g}_2,[\mathbf{g}_1,\mathbf{g}_2]]$,	$[\mathbf{g}_2,[\mathbf{f},\mathbf{g}_2]]$,
$[\mathbf{f},[\mathbf{f},[\mathbf{f},\mathbf{g}_1]]]$,	$[\mathbf{f},[\mathbf{f},[\mathbf{g}_1,\mathbf{g}_2]]]$,	$[\mathbf{f},[\mathbf{f},[\mathbf{f},\mathbf{g}_2]]]$, $[\mathbf{f},[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]]]$

The sequence can first be reduced by considering any "bad" brackets in which the drift vector appears an odd number of times and the control vector fields each appear an even number of times to include zero. In this manner the Lie brackets $[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]]$ and $[\mathbf{g}_2,[\mathbf{f},\mathbf{g}_2]]$ can be disregarded.

By evaluating each remaining Lie bracket at the equilibrium point $\mathbf{p}$, the linearly independent vector fields can be found as

$$\begin{aligned}
&\mathbf{g}_1 = \left[0, 0, 0, -m^{-1}sx_3, m^{-1}cx_3, -J_z^{-1}L\right]^T \\
&\mathbf{g}_2 = \left[0, 0, 0, 0, 0, J_z^{-1}\right]^T \\
&[\mathbf{f},\mathbf{g}_1] = \nabla\mathbf{g}_1 \cdot \mathbf{f} - \nabla\mathbf{f} \cdot \mathbf{g}_1 = \left[m^{-1}sx_3, -m^{-1}cx_3, -J_z^{-1}L, 0, 0, 0\right]^T \\
&[\mathbf{f},\mathbf{g}_2] = \nabla\mathbf{g}_2 \cdot \mathbf{f} - \nabla\mathbf{f} \cdot \mathbf{g}_2 = \left[0, 0, -J_z^{-1}, 0, 0, 0\right]^T \\
&[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_2]] = \nabla[\mathbf{f},\mathbf{g}_2] \cdot \mathbf{g}_1 - \nabla\mathbf{g}_1 \cdot [\mathbf{f},\mathbf{g}_2] = \left[0, 0, 0, -m^{-1}J_z^{-1}cx_3, -m^{-1}J_z^{-1}sx_3, 0\right]^T \\
&[\mathbf{f},[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]]] = \nabla[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]] \cdot \mathbf{f} - \nabla\mathbf{f} \cdot [\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]] = \left[2Lm^{-1}J_z^{-1}cx_3, 2Lm^{-1}J_z^{-1}sx_3, 0, 0, 0, 0\right]^T
\end{aligned} \tag{20}$$

Therefore, the Lie algebra comprised of these vector fields is

$$\mathcal{L}(\Delta) = span\left\{\mathbf{g}_1, \mathbf{g}_2, [\mathbf{f}, \mathbf{g}_1], [\mathbf{f}, \mathbf{g}_2], \left[\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_2]\right], \left[\mathbf{f}, \left[\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_1]\right]\right]\right\} \tag{21}$$

yielding $\dim\left(\mathcal{L}(\Delta)\right) = N_x = 6$, and therefore the system is small time locally controllable.

Control	Thruster Positions	$\dim\left(\mathcal{L}(\Delta)\right)$	Controllability
$\mathbf{u}^T = \left[{}^B F_x, 0, 0\right]$	$\alpha_1 = \alpha_2 = 0$	2	Inaccessible
$\mathbf{u}^T = \left[0, {}^B F_y, 0\right]$	$\alpha_1 = -\alpha_2 = \mp\pi/2$	2	Inaccessible
$\mathbf{u}^T = \left[0, 0, {}^B T_z\right]$	NA	2	Inaccessible
$\mathbf{u}^T = \left[0, {}^B F_y, {}^B T_z = F_j L s\alpha_j\right]$	$\alpha_i = \mp\pi/2, \alpha_j = \pm\pi/2$	5	Inaccessible
$\mathbf{u}^T = \left[{}^B F_x, {}^B F_y, 0\right]$	$\lvert\alpha_1\rvert \le \pi/2, \lvert\alpha_2\rvert \le \pi/2$	6	STLC
$\mathbf{u}^T = \left[{}^B F_x, 0, {}^B T_z\right]$	$\alpha_1 = \alpha_2 = 0$	6	STLC
$\mathbf{u}^T = \left[0, {}^B F_y, {}^B T_z = T_{MED}\right]$	$\alpha_1 = -\alpha_2 = \mp\pi/2$	6	STLC

Table 2. STLC Analysis for the 3-DoF Spacecraft Simulator

5. Navigation and Control of the 3-DoF Spacecraft Simulator

In the current research, the assumption is made that the spacecraft simulator is maneuvering in the proximity of an attitude stabilized target spacecraft and that this spacecraft follows a Keplarian orbit. Furthermore, the proximity navigation maneuvers are considered to be fast with respect to the orbital period. A pseudo-GPS inertial measurement system by Metris, Inc. (iGPS) is used to fix the ICS in the laboratory setting for the development of the state estimation algorithm and control commands. The *X*-axis is taken to be the vector between the two iGPS transmitters with the *Y* and *Z* axes forming a right triad through the origin of a reference system located at the closest corner of the epoxy floor to the first iGPS transmitter. Navigation is provided by fusing of the magnetometer data and fiber optic gyro through a discrete Kalman filter to provide attitude estimation and through the use of a linear quadratic estimator to estimate the translation velocities given inertial position measurements. Control is accomplished through the combination of a state feedback linearized based controller, a linear quadratic regulator, Schmitt trigger logic and Pulse Width Modulation using the minimal control actuator configuration of the 3-DoF spacecraft simulator. Fig. 4 reports a block diagram representation of the control system.

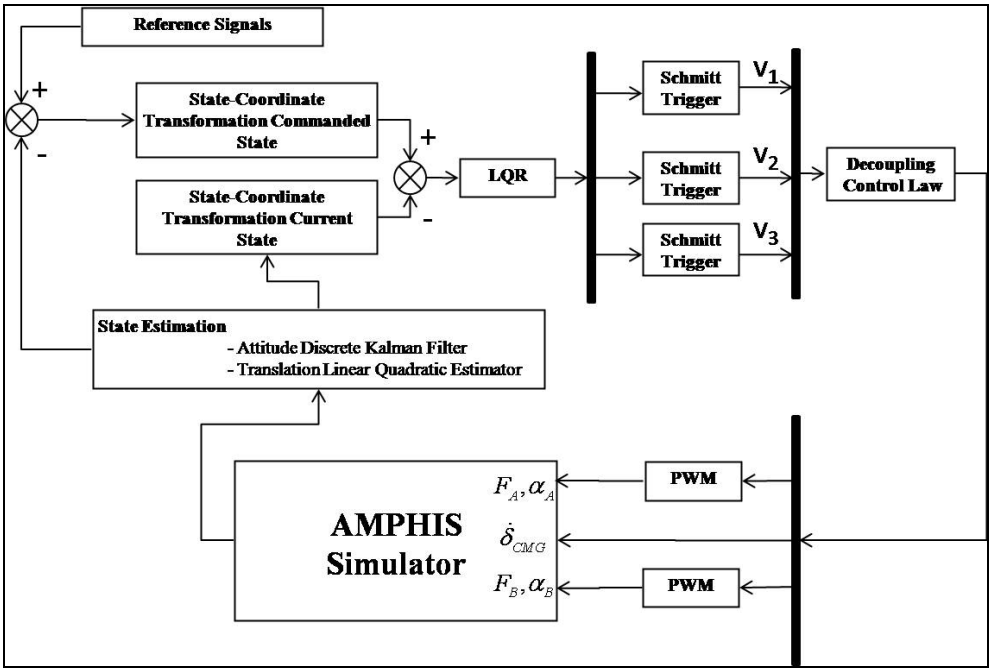


Fig. 4. Block Diagram of the Control System of the 3-DoF Spacecraft simulator

5.1 Navigation using Inertial Measurements with Kalman Filter and Linear Quadratic Estimator

In the presence of the high accuracy, low noise, high bandwidth iGPS sensor with position accuracy to within 5.4 mm with a standard deviation of 3.6 mm and asynchronous measurement availability with a nominal frequency of 40 Hz, a full-order linear quadratic estimator with respect to the translation states is implemented to demonstrate the capability to estimate the inertial velocities in the absence of accelerometers. Additionally, due to the affect of noise and drift rate in the fiber-optic gyro, a discrete-time linear Kalman filter is employed to fuse the data from the magnetometer and the gyro. Both the gyro and magnetometer are capable of providing new measurements asynchronously at 100 Hz.

5.1.1 Attitude Discrete-Time Kalman Filter

With the attitude rate being directly measured, the measurement process can be modeled in state-space equation form as:

$$\begin{bmatrix} \omega_z \\ \dot{\beta}_g \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \psi \\ \beta_g \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\mathbf{B}} \omega_g + \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \eta_{\omega g} \\ \eta_{\beta g} \end{bmatrix} \tag{22}$$

$$z = \psi_m = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \psi \\ \beta_g \end{bmatrix} + \eta_{\psi m} \tag{23}$$

where ω_g is the measured gyro rate, β_g is the gyro drift rate, $\eta_{\omega g}$ and $\eta_{\beta g}$ are the associated gyro output measurement noise and the drift rate noise respectively. ψ_m is the measured angle from the magnetometer, and $\eta_{\psi m}$ is the associated magnetometer output measurement noise. It is assumed that $\eta_{\omega g}$, $\eta_{\beta g}$ and $\eta_{\psi m}$ are zero-mean Gaussian white-noise processes with variances given by $\sigma_{\omega g}^2$, $\sigma_{\beta g}^2$ and $\sigma_{\psi m}^2$ respectively. Introducing the state variables $\mathbf{x}^T = \left[\psi, \beta_g\right]$, control variables $u = \omega_g$, and error variables $\mathbf{w}^T = \left[\eta_{\omega g}, \eta_{\beta g}\right]$ and $v = \eta_{\psi m}$, Eqs. (22) and (23) can be expressed compactly in matrix form as

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + G(t)\mathbf{w}(t) \tag{24}$$

$$\mathbf{z}(t) = H\mathbf{x}(t) + \mathbf{v}(t) \tag{25}$$

In assuming a constant sampling interval Δt in the gyro output, the system equation Eq. (24) and observation equations Eq. (25) can be discretized and rewritten as

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k \tag{26}$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k \tag{27}$$

where

$$\Phi_k = e^{\mathbf{A}\Delta t} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \tag{28}$$

and

$$\Gamma_k = \int_0^{\Delta t} e^{A\tau} B d\tau = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \tag{29}$$

The process noise covariance matrix used in the propagation of the estimation error covariance given by (Gelb, 1974; Crassidis & Junkins, 2004)

$$\left(\Upsilon_k Q_k \Upsilon_k\right)^T = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) G(\tau) E\left\{\mathbf{w}(\tau)\mathbf{w}^T(\alpha)\right\} G^T(\alpha) \Phi(t_{k+1}, \alpha) d\tau d\alpha \tag{30}$$

can be properly numerically estimated given a sufficiently small sampling interval by following the numerical solution by van Loan (Crassidis & Junkins, 2004). First, the following 2*n* x 2*n* matrix is formed:

$$\mathcal{A} = \begin{bmatrix} -A & GQG^T \\ 0 & A^T \end{bmatrix} \Delta t \tag{31}$$

where Δt is the constant sampling interval, *A* and *G* are the constant continuous-time state matrix and error distribution matrix given in Eq. (24), and *Q* is the constant continuous-time process noise covariance matrix

$$Q = E\left\{\mathbf{w}(t)\mathbf{w}^T(t)\right\} = \begin{bmatrix} \sigma_g^2 & 0 \\ 0 & \sigma_{\beta g}^2 \end{bmatrix} \tag{32}$$

The matrix exponential of Eq. (31) is then computed by

$$\mathcal{B} = e^{\mathcal{A}} \equiv \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \\ 0 & \mathcal{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_{11} & \Phi_k^{-1}\mathcal{Q}_k \\ 0 & \Phi_k^T \end{bmatrix} \tag{33}$$

where Φ_k is the state transition matrix from Eq. (28) and $\mathcal{Q}_k = \left(\Upsilon_k Q_k \Upsilon_k\right)^T$. Therefore, the discrete-time process noise covariance is

$$\mathcal{Q}_k = \left(\Upsilon_k Q_k \Upsilon_k\right)^T = \Phi_k \mathcal{B}_{12} = \begin{bmatrix} 1/3\sigma_{\beta g}^2 \Delta t^3 + \sigma_g^2 \Delta t & -1/2\sigma_{\beta g}^2 \Delta t^2 \\ -1/2\sigma_{\beta g}^2 \Delta t^2 & \sigma_{\beta g}^2 \Delta t \end{bmatrix} \tag{34}$$

The discrete-time measurement noise covariance is

$$r_k = E\left\{\mathbf{v}_k \mathbf{v}_k^T\right\} = \sigma_{\theta m}^2 \tag{35}$$

Given the filter model as expressed in Eqs. (22) and (23), the estimated states and error covariance are initialized where this initial error covariance is given by $P_0 = E\left\{\tilde{\mathbf{x}}(t_0)\tilde{\mathbf{x}}^T(t_0)\right\}$. If a measurement is given at the initial time, then the state and covariance are updated using the Kalman gain formula

$$K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + r_k\right]^{-1} \tag{36}$$

where P_k^- is the *a priori* error covariance matrix and is equal to P_0. The updated or *a posteriori* estimates are determined by

$$\begin{aligned} \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + K_k\left[z_k - H_k \hat{\mathbf{x}}_k^-\right] \\ P_k^+ &= \left[I_{2x2} - K_k H_k\right] P_k^- \end{aligned} \tag{37}$$

where again with a measurement given at the initial time, the *a priori* state $\hat{\mathbf{x}}_k^-$ is equal to $\hat{\mathbf{x}}_0$. The state estimate and covariance are propagated to the next time step using

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^- &= \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k u_k \\ P_{k+1}^- &= \Phi_k P_k^+ \Phi_k^T + \mathcal{Q}_k \end{aligned} \tag{38}$$

If a measurement isn't given at the initial time step or any time step during the process, the estimate and covariance are propagated to the next available measurement point using Eq. (38).

5.1.2 Translation Linear Quadratic Estimator

With the measured translation state from the iGPS sensor, being given by

$$\mathbf{z} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} X, Y, V_x, V_y \end{bmatrix}^T}_{\mathbf{x}} \tag{39}$$

the dynamics of a full-order state estimator is described by the equation

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} = \left[A\mathbf{x} + B\mathbf{u}\right] - \left[A\hat{\mathbf{x}} + B\mathbf{u} + L_{LQE}\left(\mathbf{z} - C\hat{\mathbf{x}}\right)\right] \\ &= A\left(\mathbf{x} - \hat{\mathbf{x}}\right) - L_{LQE}\left(C\mathbf{x} - C\hat{\mathbf{x}}\right) \\ &= \left(A - L_{LQE}C\right)\tilde{\mathbf{x}} \end{aligned} \tag{40}$$

where

$A\mathbf{x} + B\mathbf{u}$	:	linearized plant dynamics
$A\hat{\mathbf{x}} + B\mathbf{u}$	:	system model
L_{LQE}	:	linear quadratic estimator gain matrix
$C\hat{\mathbf{x}}$	:	measurement if $\mathbf{x}$ were $\hat{\mathbf{x}}$

The observer gain matrix L_{LQE} can be solved using standard linear quadratic estimator methods as (Bryson, 1993)

$$L_{LQE} = PC^T R_T^{-1} \tag{41}$$

where P is the solution to the algebraic Riccati equation

$$AP + PA^T - PC^T R_T^{-1} CP + Q_T = 0 \tag{42}$$

and Q_T and R_T are the associated weighting matrices with respect to the translational degree of freedom defined as

$$\begin{aligned} Q_T &= diag\left(1/\left\|\Delta X_{\max}\right\|^2, 1/\left\|\Delta Y_{\max}\right\|^2, 1/\left\|\Delta V_{y,\max}\right\|^2, 1/\left\|\Delta V_{x,\max}\right\|^2\right) \\ R_T &= diag\left(1/\left(F_{\max}\right)^2, 1/\left(F_{\max}\right)^2\right) \end{aligned} \tag{43}$$

where $\Delta X_{\max}, \Delta Y_{\max}, \Delta V_{x,\max}, \Delta V_{y,\max}$ are taken to be the maximum allowed errors between the current and estimated translational states and $F_{\max}$ is the maximum possible imparted force from the thrusters.

Table 3 lists the values of the attitude Kalman filter and translation state observer used for the experimental tests.

Δt	10^{-2} s
$\sigma_{\beta g}$	3.76 x 10^{-3} rad-s$^{-3/2}$
$\sigma_{\omega g}$	1.43 x 10^{-4} rad-s$^{-3/2}$
$\sigma_{\psi m}$	5.59 x 10^{-3} rad
P_0	$diag\left[10^{-15}, 10^{-8}\right]$
$\hat{\mathbf{x}}_0$	$\left[0,0\right]^T$
$\Delta X_{\max}, \Delta Y_{\max}$	10^{-2} m
$\Delta V_{X,\max}, \Delta V_{Y,\max}$	3 x 10^{-3} m-s^{-1}
F_{max}	.159 N
L_{LQE}	$\begin{bmatrix} 18.9423 & 0 \\ 0 & 18.9423 \\ 53 & 0 \\ 0 & 53 \end{bmatrix}$

Table 3. Kalman Filter Estimation Paramaters

5.2 Smooth Feedback Control via State Feedback Linearization and Linear Quadratic Regulation

Considering a Multi-Input Multi-Output (MIMO) nonlinear system in control-affine form, the state feedback linearization problem of nonlinear systems can be stated as follows: obtain a proper state transformation

$$\mathbf{z} = \Phi(\mathbf{x}) \text{ where } \mathbf{z} \in \mathbb{R}^{N_x} \tag{44}$$

and a static feedback control law

$$\mathbf{u} = \alpha(\mathbf{x}) + \beta(\mathbf{x})\mathbf{v} \text{ where } \mathbf{v} \in \mathbb{R}^{N_u} \tag{45}$$

such that the closed-loop system in the new coordinates and controls become

$$\dot{\mathbf{z}} = \left\{\frac{\partial \Phi}{\partial \mathbf{x}}\left[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\alpha(\mathbf{x})\right]\right\}\Big|_{\mathbf{x}=\Phi^{-1}(\mathbf{z})} + \left\{\frac{\partial \Phi}{\partial \mathbf{x}}\left[G(\mathbf{x})\boldsymbol{\beta}(\mathbf{x})\right]\right\}\Big|_{\mathbf{x}=\Phi^{-1}(\mathbf{z})} \mathbf{v} \tag{46}$$

is both linear and controllable. The necessary conditions for a MIMO system to be considered for input-output linearization are that the system must be square or $N_u = N_y$ where N_u is defined as above to be the number of control inputs and N_y is the number of outputs for a system of the expanded form (Isidori, 1989; Slotine, 1990)

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \sum_{i=1}^{N_y} h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \end{aligned} \tag{47}$$

The input-output linearization is determined by differentiating the outputs y_i in Eq. (47) until the inputs appear. Following the method outlined in (Slotine, 1990) by which the assumption is made that the partial relative degree r_i is the smallest integer such that at least one of the inputs appears in $y_i^{(r_i)}$, then

$$y_i^{(r_i)} = L_{\mathbf{f}}^{r_i} h_i + \sum_{j=1}^{N_y} L_{\mathbf{g}_j} L_{\mathbf{f}}^{r_i - 1} h_i(\mathbf{x}) u_j \tag{48}$$

with the restriction that $L_{\mathbf{g}_j} L_{\mathbf{f}}^{r_i - 1} h_i(\mathbf{x}) \neq 0$ for at least one j in a neighborhood of the equilibrium point $\mathbf{x}_0$. Letting

$$E(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_1 - 1} h_1(\mathbf{x}) & \cdots & L_{\mathbf{g}_{N_u}} L_{\mathbf{f}}^{r_1 - 1} h_1(\mathbf{x}) \\ L_{\mathbf{g}1} L_{\mathbf{f}}^{r_2 - 1} h_2(\mathbf{x}) & \cdots & L_{\mathbf{g}_{N_u}} L_{\mathbf{f}}^{r_2 - 1} h_2(\mathbf{x}) \\ \cdots & \cdots & \cdots \\ L_{\mathbf{g}_1} L_{\mathbf{f}}^{r_{N_y} - 1} h_{N_y}(\mathbf{x}) & \cdots & L_{\mathbf{g}_{N_u}} L_{\mathbf{f}}^{r_{N_y} - 1} h_{N_y}(\mathbf{x}) \end{bmatrix} \tag{49}$$

so that Eq. (49) is in the form

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \cdots \\ y_{N_y}^{(r_{N_y})} \end{bmatrix} = \begin{bmatrix} L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}) \\ L_{\mathbf{f}}^{r_2} h_2(\mathbf{x}) \\ \cdots \\ L_{\mathbf{f}}^{r_{N_y}} h_{N_y}(\mathbf{x}) \end{bmatrix} + E(\mathbf{x})\mathbf{u} \tag{50}$$

the decoupling control law can be found where the $N_y \times N_y$ matrix $E(\mathbf{x})$ is invertible over the finite neighborhood of the equilibrium point for the system as

$$\mathbf{u} = E(\mathbf{x})^{-1} \begin{bmatrix} v_1 - L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}) \\ v_2 - L_{\mathbf{f}}^{r_2} h_2(\mathbf{x}) \\ \cdots \\ v_{N_y} - L_{\mathbf{f}}^{r_{N_y}} h_{N_y}(\mathbf{x}) \end{bmatrix} \tag{51}$$

With the above stated equations for the simulator dynamics in Eq. (9) given $G_1(\mathbf{x})$ as defined in Eq. (11), if we choose

$$\mathbf{h}(\mathbf{x}) = [X, Y, \psi]^T \tag{52}$$

the state transformation can be chosen as

$$\mathbf{z}^T = [h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x}), L_{\mathbf{f}} h_1(\mathbf{x}), L_{\mathbf{f}} h_2(\mathbf{x}), L_{\mathbf{f}} h_3(\mathbf{x})] = [X, Y, \psi, V_x, V_y, \omega_z] \tag{53}$$

where $\mathbf{z}^T = \left[z_1, z_2, ..., z_6\right] \in \mathbb{R}^6$ are new state variables, and the system in Eq. (9) is transformed into

$$\dot{\mathbf{z}}^T = \left[z_4, z_5, z_6, m^{-1}\left(\mathrm{c}z_3\, {}^B F_x - \mathrm{s}z_3\, {}^B F_y\right), m^{-1}\left(\mathrm{s}z_3\, {}^B F_x + \mathrm{c}z_3\, {}^B F_y\right), J_z^{-1}\, {}^B T \right] \tag{54}$$

The dynamics given by Eq. (9) considering the switching logic described in Eqs. (10), (12) and (14) can now be transformed using Eq. (54) and the state feedback control law

$$\left[{}^B\mathbf{F}, {}^B T \right] = E(\mathbf{x})^{-1}\left(\mathbf{v} - \mathbf{b}\right) \tag{55}$$

into a linear system

$$\dot{\mathbf{z}} = \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix} \mathbf{z} + \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{I}_{3x3} \end{bmatrix} \mathbf{v} \tag{56}$$

where

$$\mathbf{b} = \left[L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}), L_{\mathbf{f}}^{r_2} h_2(\mathbf{x}), L_{\mathbf{f}}^{r_3} h_3(\mathbf{x}) \right]^T \tag{57}$$

and $E(\mathbf{x})$ given by Eq. (49) with equivalent inputs $\mathbf{v} = \left[v_1, v_2, v_3\right]^T$ and relative degree of the system at the equilibrium point $\mathbf{x}_0$ is $\left(r_1, r_2, r_3\right) = (2,2,2)$. Therefore the total relative degree of the system at the equilibrium point, which is defined as the sum of the relative degree of the system, is six. Given that the total relative degree of the system is equal to the number of states, the nonlinear system can be exactly linearized by state feedback and with the equivalent inputs v_i, both stabilization and tracking can be achieved for the system without concern for the stability of the internal dynamics (Slotine, 1990).

One of the noted limitations of a feedback linearized based control system is the reliance on a fully measured state vector (Slotine, 1990). This limitation can be overcome through the employment of proper state estimation. HIL experimentation on SRL's second generation robotic spacecraft simulator using these navigation algorithms combined with the state feedback linearized controller as described above coupled with a linear quadratic regulator to ensure the poles of Eq. (56) lie in the open left half plane demonstrate satisfactory results as reported in the following section.

5.2.1 Feedback Linearized Control Law with MSGCMG Rotational Control and Thruster Translational Control

By applying Eq. (55) to the dynamics in Eq. (9) given $G_1(\mathbf{x})$ as defined in Eq. (11) where the system is taken to be observable in the state vector $\mathbf{y} = \left[X, Y, \psi\right]^T = \left[x_1, x_2, x_3\right]^T$ and by using thruster two for translational control (i.e. for the case ${}^B U_x > 0$ where ${}^B U_x = v_1 \mathrm{c}\psi + v_2 \mathrm{s}\psi$ and ${}^B U_y = -v_1 \mathrm{s}\psi + v_2 \mathrm{c}\psi$), the feedback linearized control law is

$$\mathbf{u}^T = \left[{}^B F_x, {}^B F_y, {}^B T_z \right] = \left[m\, {}^B U_x, m\, {}^B U_x, mL\, {}^B U_y + J_z v_3 \right] \tag{58}$$

which is valid for all $\mathbf{x}$ in a neighborhood of the equilibrium point $\mathbf{x}_0$. Similarly, the feedback linearized control law when ${}^{B}U_x < 0$ (thruster one is providing translation control)

$$\mathbf{u}^T = \left[{}^{B}F_x, {}^{B}F_y, {}^{B}T_z \right] = \left[m\,{}^{B}U_x, m\,{}^{B}U_y, -mL\,{}^{B}U_y + J_z v_3 \right] \tag{59}$$

Finally, when ${}^{B}U_x = 0$ (both thrusters used for translational control) given $G_1(\mathbf{x})$ as defined in Eq. (13) is

$$\mathbf{u}^T = \left[{}^{B}F_y, {}^{B}T_z \right] = \left[m\,{}^{B}U_y / 2, J_z v_3 \right] \tag{60}$$

5.2.2 Feedback Linearized Control Law for Thruster Roto-Translational Control

As mentioned previously, by considering a momentum exchange device for rotational control, momentum storage must be managed. For a control moment gyroscope based moment exchange device, desaturation is necessary near gimbal angles of $\pi/2$. In this region, due to the mathematical singularity that exists, very little torque can be exchanged with the vehicle and thus it is essentially ineffective as an actuator. To accommodate these regions of desaturation, logic can be easily employed to define controller modes as follows: If the MSGCMG is being used as a control input and if the gimbal angle of the MSGCMG is greater than 75 degrees, the controller mode is switched from normal operation mode to desaturation mode and the gimbal angle rate is directly commanded to bring the gimbal angle to a zero degree nominal position while the thruster not being directly used for translational control is slewed as appropriate to provide torque compensation. In these situations, the feedback linearizing control law for the system dynamics in Eq. (9) given $G_1(\mathbf{x})$ as defined in Eq. (15) where thruster two is providing translational control (${}^{B}U_x > 0$), and thruster one is providing the requisite torque is

$$\mathbf{u}^T = \left[{}^{B}F_x, {}^{B}F_y, {}^{B}T_z \right] = \left[m\,{}^{B}U_x, \left(mL\,{}^{B}U_y - J_z v_3 \right) / 2L, \left(mL\,{}^{B}U_y + J_z v_3 \right) / 2 \right] \tag{61}$$

Similarly, the feedback linearizing control law for the system assuming thruster one is providing translational control $\left({}^{B}U_x \le 0 \right)$ while thruster two provides the requisite torque is

$$\mathbf{u}^T = \left[{}^{B}F_x, {}^{B}F_y, {}^{B}T_z \right] = \left[m\,{}^{B}U_x, \left(mL\,{}^{B}U_y + J_z v_3 \right) / 2L, \left(mL\,{}^{B}U_y - J_z v_3 \right) / 2 \right] \tag{62}$$

5.2.3 Determination of the thruster angles, forces and MSGCMG gimbal rates

In either mode of operation, the pertinent decoupling control laws are used to determine the commanded angle for the thrusters and whether or not to open or close the solenoid for the thruster. For example, if ${}^{B}U_x > 0$, Eq. (58) or (61) can be used to determine the angle to command thruster two as

$$\alpha_2 = \tan^{-1}\left({}^{B}F_y / {}^{B}F_x \right) \tag{63}$$

and the requisite thrust as

$$F_2 = \sqrt{{}^{B}F_x^2 + {}^{B}F_y^2} \tag{64}$$

If the MSGCMG is being used, the requisite torque commanded to the CMG is taken directly from Eq. (58). In the normal operation mode, with the commanded angle for thruster one not pertinent, it can be commanded to zero without affecting control of the system. Similarly, if ${}^{B}U_X < 0$, Eq. (59) or (62) can be used to determine the angle to command thruster one and the requisite thrust analogous to Eqs. (63) and (64). The requisite torque commanded to the CMG is similarly taken directly from Eq. (59). The required CMG torques can be used to determine the gimbal rate $\left(\dot{\delta}_{CMG}\right)$ to command the MSGCMG by solving the equation (Hall, 2006; Romano & Hall, 2006)

$$\dot{\delta}_{CMG} = -T_{CMG} / \left(h_w \cos\delta_{CMG}\right) \tag{65}$$

where h_w is the constant angular momentum of the rotor wheel and δ_{CMG} is the current angular displacement of the wheel's rotational axis with respect to the horizontal.

If the momentum exchange device is no longer available and ${}^{B}U_x > 0$, the thruster angle commands and required thrust value for the opposing thruster can be determined by using Eq. (61) as

$$\alpha_1 = -\pi/2\, sign\left(mL\,{}^{B}U_y + J_z v_3\right) \tag{66}$$

and

$$F_1 = -sign(\alpha_1)\left(mL\,{}^{B}U_y + J_z v_3\right)/L \tag{67}$$

given ${}^{B}T_z = -F_1 \mathrm{s}\,\alpha_1 L$. Likewise, the thruster angle commands and required thrust value for the opposing thruster given ${}^{B}U_x < 0$ can be determined by using Eq. (62) as

$$\alpha_2 = \pi/2\, sign\left(mL\,{}^{B}U_y - J_z v_3\right) \tag{68}$$

and

$$F_2 = sign(\alpha_2)\left(mL\,{}^{B}U_y - J_z v_3\right)/L \tag{69}$$

given ${}^{B}T_z = -F_2 \sin\alpha_2 L$.

5.2.4 Linear Quadratic Regulator Design

In order to determine the linear feedback gains used to compute the requisite equivalent inputs v_i to regulate the three degrees of freedom so that

$$\begin{aligned}
&\lim_{t\to\infty} z_1(t) = X(t) = X_{ref}, \quad &&\lim_{t\to\infty} z_2(t) = Y(t) = Y_{ref}, \quad &&\lim_{t\to\infty} z_3(t) = \psi(t) = \psi_{ref} \\
&\lim_{t\to\infty} z_4(t) = V_X(t) = V_{X,ref}, \quad &&\lim_{t\to\infty} z_5(t) = V_Y(t) = V_{Y,ref}, \quad &&\lim_{t\to\infty} z_6(t) = \omega_z(t) = \omega_{z,ref}
\end{aligned} \tag{70}$$

a standard linear quadratic regulator is employed where the state-feedback law $\mathbf{v} = -K\mathbf{z}$ minimizes the quadratic cost function

$$J(\mathbf{v}) = \int_0^\infty \left(\mathbf{z}^T Q \mathbf{z} + \mathbf{v}^T R \mathbf{v}\right) dt \tag{71}$$

subject to the feedback linearized state-dynamics of the system given in Eq. (56) . Given the relation between the linearized state and true state of the system, the corresponding gain matrices R and Q in Eq. (71) are chosen to minimize the appropriate control and state errors as

$$\begin{aligned} Q &= diag\begin{pmatrix} 1/\|\Delta X_{\max}\|^2, , 1/\|\Delta Y_{\max}\|^2, 1/\|\Delta \psi_{\max}\|^2, \\ 1/\|\Delta V_{y,\max}\|^2, 1/\|\Delta V_{x,\max}\|^2, 1/\|\Delta \omega_{z,\max}\|^2 \end{pmatrix} \\ R &= diag\left(1/(F_{\max})^2, 1/(F_{\max})^2, 1/(T_{CMG,\max})^2\right) \end{aligned} \tag{72}$$

where $\Delta X_{\max}, \Delta Y_{\max}, \Delta V_{x,\max}, \Delta V_{y,\max}, \Delta \psi_{\max}, \Delta \omega_{z,\max}$ are taken to be the maximum errors allowed between the current states and reference states while $F_{\max}$ and $T_{CMG,\max}$ are taken to be the maximum possible imparted force and torques from the thrusters and MSGCMG respectively.

Given the use of discrete cold-gas thrusters in the system for translational control throughout a commanded maneuver and rotational control when the continuously acting momentum exchange device is unavailable, Schmitt trigger switching logic is imposed. Schmitt triggers have the unique advantage of reducing undesirable chattering and subsequent propellant waste nearby the reference state through an output-versus-input logic that imposes a dead zone and hysteresis to the phase space as shown in **Fig. 5**.

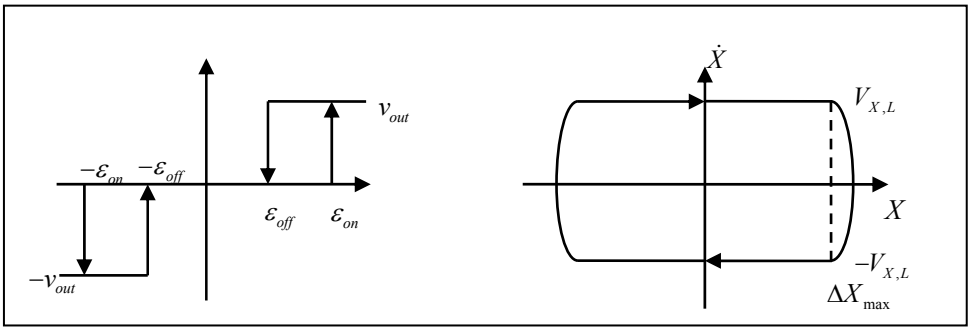


Fig. 5. Schmitt Trigger Characteristics with Design Parameters Considering X Coordinate Control Logic

Three separate Schmitt triggers are used with the design parameters of the Schmitt trigger shown in **Fig. 5** (as demonstrated for the X coordinate control logic). In the case of the two translational DoF Schmitt triggers, the parameters are chosen such that

$$\begin{aligned}\varepsilon_{on} &= K_X X_{db} + K_{\dot{X}} V_{X,L} \\ \varepsilon_{off} &= K_X X_{db} - K_{\dot{X}} V_{X,L}\end{aligned} \tag{73}$$

where $V_{X,L} = V_{Y,L} = \dot{X}_L = F_{\max}\Delta t/2m$. X_{db}, Y_{db} are free parameters that are constrained by mission requirements. v_{out} is chosen such that the maximum control command from the decoupling control law yields a value less than or equal to $F_{\max}$ for the translational thruster. In the case of the rotational DoF Schmitt trigger when the momentum exchange device is unavailable, the parameters are chosen such that

$$\begin{aligned}\varepsilon_{on} &= K_\psi \psi_{db} + K_{\omega_z} \omega_{z,L} \\ \varepsilon_{off} &= K_\psi \psi_{db} - K_{\omega_z} \omega_{z,L}\end{aligned} \tag{74}$$

where $\omega_{z,L} = F_{\max} L\Delta t/2J_z$. ψ_{db} is a free parameter that is again constrained by mission requirements. For both modes of operation (i.e. with or without a momentum exchange device), Eqs. (58) through (60) can be used to determine that

$$v_{1,\max} = v_{2,\max} = F_{\max}/\sqrt{2}m \tag{75}$$

and when the thrusters are used for rotational control

$$v_{3,\max} = F_{\max} L/J_z \tag{76}$$

When the momentum exchange device is available, the desired torque as determined by the LQR control law as described above is passed directly through the Schmitt trigger to the decoupling control law to determine the required gimbal rate command to the MSGCMG.
The three Schmitt trigger blocks output the requested control inputs along the ICS frame. The appropriate feedback linearizing control law is then used to transform these control inputs into requested thrust, thruster angle and MSGCMG gimbal rate along the BCS frame. From these, a vector of specific actuator commands are formed such that

$$\mathbf{u}_c^T = \left[F_1, \alpha_1, F_2, \alpha_2, \dot{\delta}_{CMG}\right] \tag{77}$$

Each thruster command is normalized with respect to $F_{\max}$ and then fed with its corresponding commanded angle into separate Pulse Width Modulation (PWM) blocks. Each PWM block is then used to obtain an approximately linear duty cycle from on-off actuators by modulating the opening time of the solenoid valves (Wie, 1998). Additionally, due to the linkage between the thruster command and the thruster angle, the thruster firing sequence is held until the actual thruster angle is within a tolerance of the commanded thruster angle. Furthermore, in order to reduce over-controlling the system, the LQR, Schmitt trigger logic and decoupling control algorithm are run at the PWM bandwidth of 8.33 Hz. From each PWM, digital outputs (either zero or one) command the two thrusters while the corresponding angle is sent via RS-232 to the appropriate thruster gimbal motor.

Δt	10^{-2} s
$\Delta X_{\max}, \Delta Y_{\max}$	10^{-2} m
$\Delta V_{X,\max}, \Delta V_{Y,\max}$	3 x 10^{-3} m-s^{-1}
$\Delta \psi_{\max}$	1.8 x 10^{-2} rad
$\Delta \omega_{z,\max}$	1.8 x 10^{-2} rad-s^{-1}
F_{max}	.159 N
$T_{CMG,\max}$	.668 Nm
$K_X = K_Y = K_{LQR}(1,1) = K_{LQR}(2,2)$	15.9
$K_{\dot{X}} = K_{\dot{Y}} = K_{LQR}(1,4) = K_{LQR}(2,5)$	84.54 s
$K_{\psi} = K_{LQR}(3,3)$	1.39
$K_{\omega_z} = K_{LQR}(3,6)$	1.75 s
$X_{db} = Y_{db}$	10^{-2} m
$V_{X,L} = V_{Y,L}$	3.05 x 10^{-5} m-s^{-1}
ψ_{db}	1.8 x 10^{-2} rad
$\omega_{z,L}$	1.8 x 10^{-2} rad-s^{-1}
$\varepsilon_{on}(X) = \varepsilon_{on}(Y)$	1.61 x 10^{-1} m
$\varepsilon_{off}(X) = \varepsilon_{off}(Y)$	1.56 x 10^{-1} m
$\varepsilon_{on}(\psi)$	2.47 x 10^{-2} rad
$\varepsilon_{off}(\psi)$	2.37 x 10^{-2} rad
PWM min pulse width	10^{-2} s
PWM sample time	1.2 x 10^{-1} s

Table 4. Values of the Control Parameters

Table 4 lists the values of the control parameters used for the experimental tests reported in the following section. In particular, $\Delta V_{X,\max}, \Delta V_{Y,\max}$ are chosen based typical maximum relative velocities during rendezvous scenarios while $\Delta \psi_{\max}$ is taken to be 1 degree and $\Delta \omega_{z,\max}$ is chosen to be 1 degrees/sec which correspond to typical slew rate requirements for small satellites (Roser & Schedoni, 1997;Lappas et al., 2002). The minimum opening time of the PWM was based on experimental results for the installed solenoid valves reported in (Lugini & Romano, 2009).

6. Experimental Results

The navigation and control algorithms introduced above were coded in MATLAB®-Simulink® and run in real time using MATLAB XPC Target™ embedded on the SRL's second generation spacecraft simulator's on-board PC-104. Two experimental tests are

presented to demonstrate the effectiveness of the designed control system. The scenario presented represents a potential real-world autonomous proximity operation mission where a small spacecraft is tasked with performing a full 360 degree circle around another spacecraft for the purpose of inspection or pre-docking. These experimental tests validate the navigation and control approach and furthermore demonstrate the capability of the robotic spacecraft simulator testbed.

6.1 Autonomous Proximity Maneuver using Vectorable Thrusters and MSGCMG along a Closed Circular Path

Fig. 6, **Fig. 7**, and Fig. 8 report the results of an autonomous proximity maneuver along a closed circular trajectory of NPS SRL's second generation robotic spacecraft simulator using its vectorable thrusters and MSGCMG. The reference path for the center of mass of the simulator consists of 200 waypoints, taken at angular intervals of 1.8 deg along a circle of diameter 1m with a center at the point [2.0 m, 2.0 m] in the ICS, which can be assumed, for instance, to be the center of mass of the target. The reference attitude is taken to be zero throughout the maneuver. The entire maneuver lasts 147 s. During the first 10 s, the simulator is maintained fixed in order to allow the attitude Kalman filter time to converge to a solution. At 10 s into the experiment, the solenoid valve regulating the air flow to the linear air bearings is opened and the simulator begins to float over the epoxy floor. At this point, the simulator begins to follow the closed path through autonomous control of the two thrusters and the MSGCMG.

As evidenced in Fig. 6a through Fig. 6d, the components of the center of mass of the simulator as estimated by the translation linear quadratic estimator are kept close to the reference signals by the action of the vectorable thrusters. Specifically, the mean of the absolute value of the tracking error is 1.3 cm for ΔX, with a standard deviation of 9.1 mm, 1.4 cm mean for ΔY with a standard deviation of 8.6 mm, 2.4 mm/s mean for ΔV_X with a standard deviation of 1.8 mm/s and 3.0 mm/s mean for ΔV_Y with a standard deviation of 2.7 mm/s. Furthermore, the mean of the absolute value of the estimated error in X is 2 mm with a standard deviation of 2 mm and 4 mm in Y with a standard deviation of 3 mm. Likewise, Fig. 6e and Fig. 6f demonstrate the accuracy of the attitude tracking control through a comparison of the commanded and actual attitude and attitude rate. Specifically, the mean of the absolute value of tracking error for $\Delta\psi$ is 0.14 deg with a standard deviation of 0.11 deg and 0.14 deg/s for $\Delta\omega_z$ with a standard deviation of 0.15 deg/s. These control accuracies are in good agreement with the set parameters of the Schmitt triggers and the LQR design.

Fig. 7a through Fig. 7d report the command signals to the simulator's thrusters along with their angular positions. The commands to the thrusters demonstrate that the Schmitt trigger logic successfully avoids chattering behavior and the feedback linearized controller is able to determine the requisite thruster angles. Fig. 7e and Fig. 7f show the gimbal position of the miniature single-gimbaled control moment gyro and the delivered torque. Of note, the control system is able to autonomously maneuver the simulator without saturating the MSGCMG.

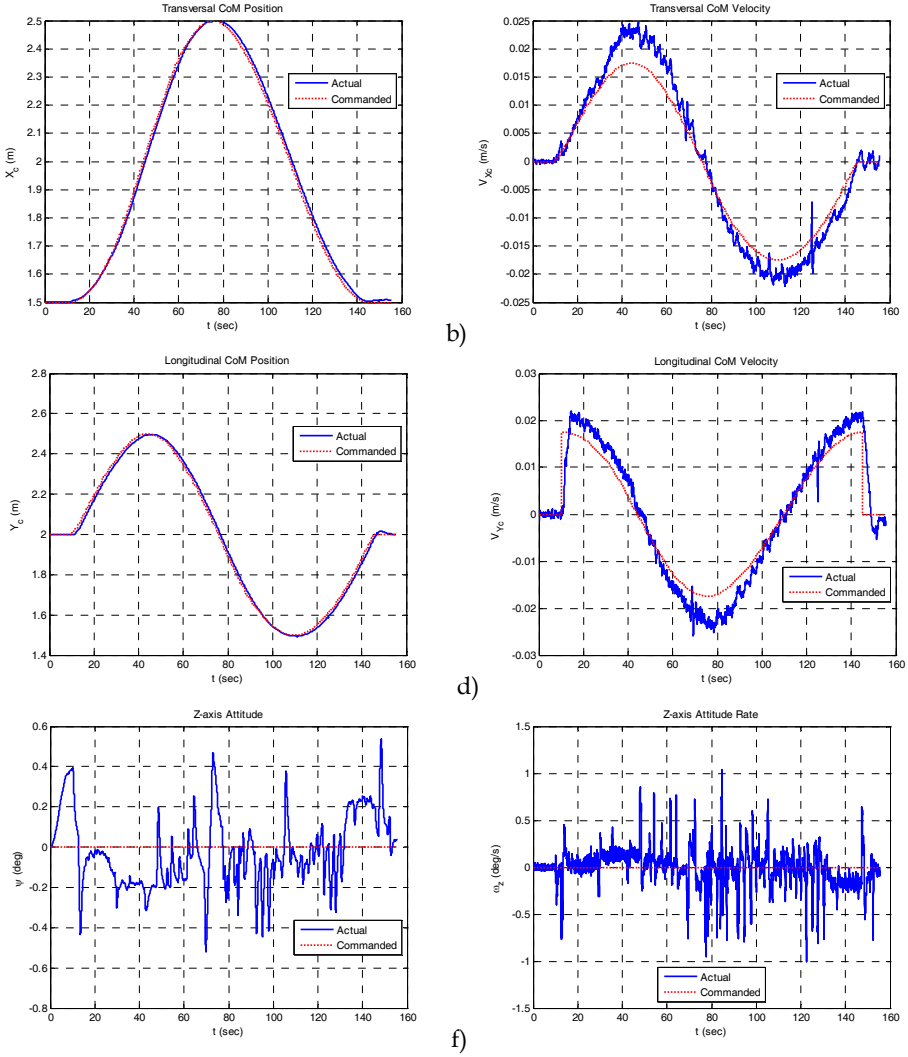


Fig. 6. Logged data versus time of an autonomous proximity maneuver of NPS SRL's 3-DoF spacecraft simulator along a closed path using vectorable thrusters and MSGCMG. The simulator begins floating over the epoxy floor at t = 10 s. a) Transversal position of the center of mass of the simulator in ICS; b) Transversal velocity of the center of mass of the simulator in ICS; c) Longitudinal position of the center of mass of the simulator; d) Longitudinal velocity of the center of mass of the simulator; e) Attitude; f) Attitude rate

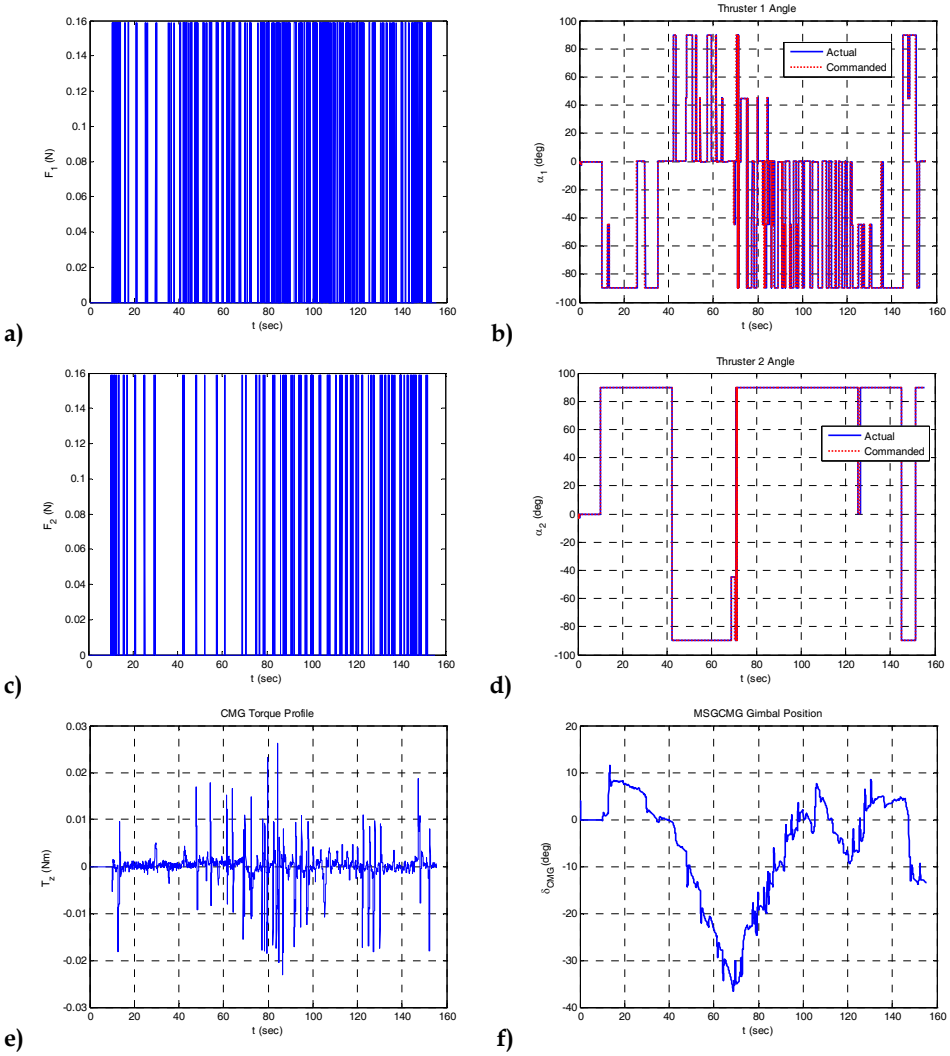


Fig. 7. Control actuator actions during autonomous proximity manuever of NPS SRL's 3-DoF spacecraft simulator along a closed path using vectorable thrusters and MSGCMG. a) Thruster 1 firing profile; b) Thruster 1 position; c) Thruster 2 firing profile; d) Thruster 2 position; f) MSGCMG torque profile; e) MSGCMG gimbal position

Fig. 8 depicts a bird's-eye view of the spacecraft simulator motion. Of particular note, the good control accuracy can be evaluated by the closeness of the actual ground-track line to the commanded circular trajectory and of the initial configuration of the simulator to the final one. The total ΔV required during this experimental test was .294 m/s which correspond to a total impulse of 7.65 Ns.

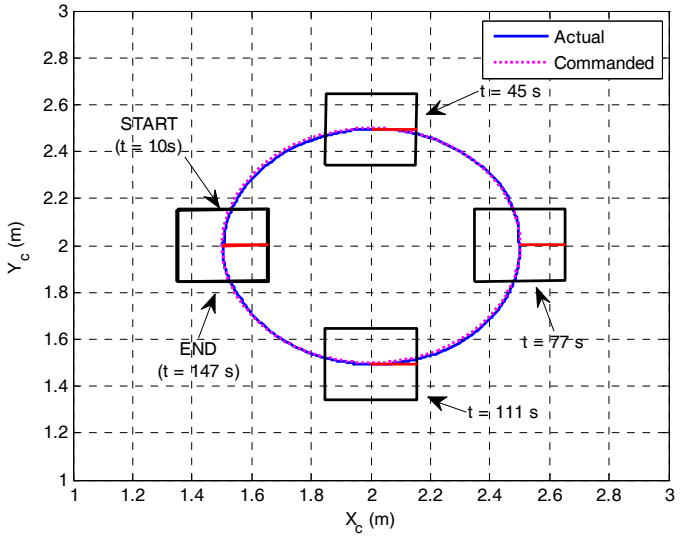


Fig. 8. Bird's-eye view of autonomous proximity manuever of NPS SRL's 3-DoF spacecraft simulator along a closed path using vectorable thrusters and MSGCMG

6.2 Autonomous Proximity Maneuver using only Vectorable Thrusters along a Closed Circular Path

Fig. 9, **Fig. 10**, and Fig. 11 report the results of maneuvering the spacecraft simulator along the same reference maneuver as in Section 6.1 but by using only the vectorable thrusters. This maneuver is presented to demonstrate the experimental validation of the STLC analytical results. As before, during the first 10 s, the simulator is not floating and kept stationary while the attitude Kalman filter converges.

The tracking and estimation errors for this maneuver are as follows with the logged positions, attitudes and velocities shown in **Fig. 9**. The mean of the absolute value of the tracking error is 1.4 cm for ΔX, with a standard deviation of 8.5 mm, 1.4 cm mean for ΔY with a standard deviation of 8.6 mm, 2.5 mm/s mean for ΔV_X with a standard deviation of 1.9 mm/s and 3.1 mm/s mean for ΔV_Y with a standard deviation of 2.8 mm/s. The mean of the absolute value of the estimated error in X is 3 mm with a standard deviation of 3 mm and 4 mm in Y with a standard deviation of 5 mm. The mean of the absolute value of tracking error for $\Delta\psi$ is 0.52 deg with a standard deviation of 0.31 deg and 0.24 deg/s for $\Delta\omega_z$ with a standard deviation of 0.20 deg/s. These control accuracies are in good agreement with the set parameters of the Schmitt triggers and LQR design.

Fig. 10 reports the command signals to the simulator's thrusters with the commands to the thrusters again demonstrating that the feedback linearized controller is able to determine the requisite thruster angles to take advantage of this fully minimized actuation system. Fig. 11 depicts a bird's-eye view of the motion of the simulator during this maneuver. The total ΔV required during this experimental test was .327 m/s which correspond to a total impulse of 8.55 Ns.

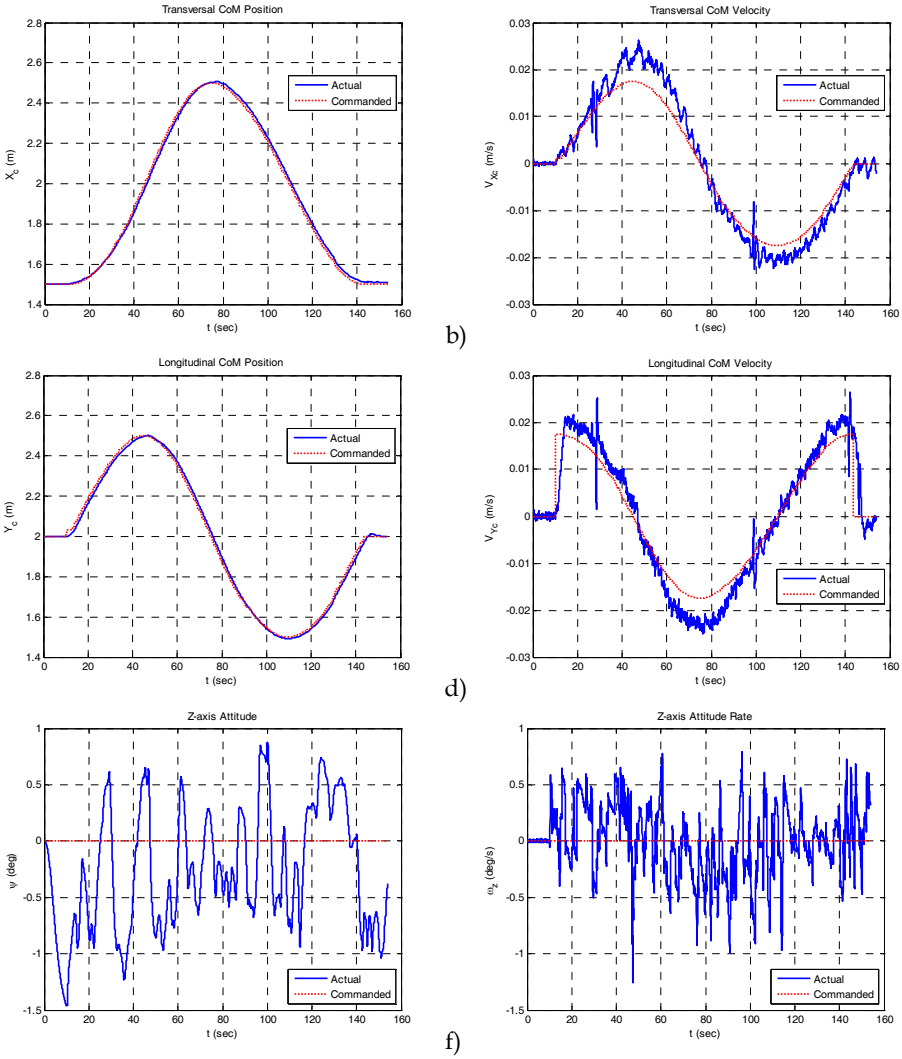


Fig. 9. Logged data versus time of an autonomous proximity maneuver of NPS SRL's 3-DoF spacecraft simulator along a closed path using only vectorable thrusters. The simulator begins floating over the epoxy floor at t = 10 s. a) Transversal position of the center of mass of the simulator in ICS; b) Transversal velocity of the center of mass of the simulator in ICS; c) Longitudinal position of the center of mass of the simulator; d) Longitudinal velocity of the center of mass of the simulator; e) Attitude; f) Attitude rate

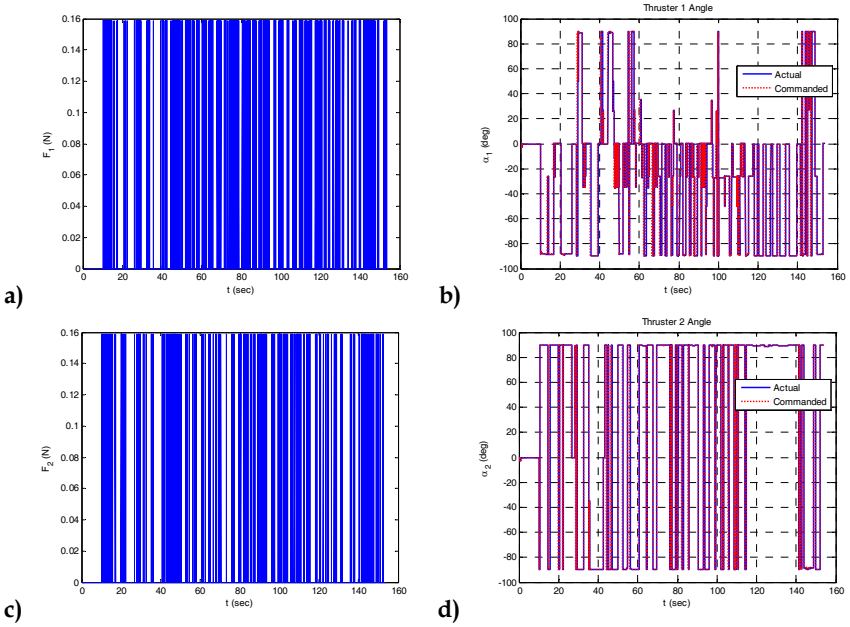


Fig. 10. Control actuator actions during autonomous proximity manuever of NPS SRL's 3-DoF spacecraft simulator along a closed path using only vectorable thrusters. a) Thruster 1 firing profile; b) Thruster 1 position; c) Thruster 2 firing profile; d) Thruster 2 position

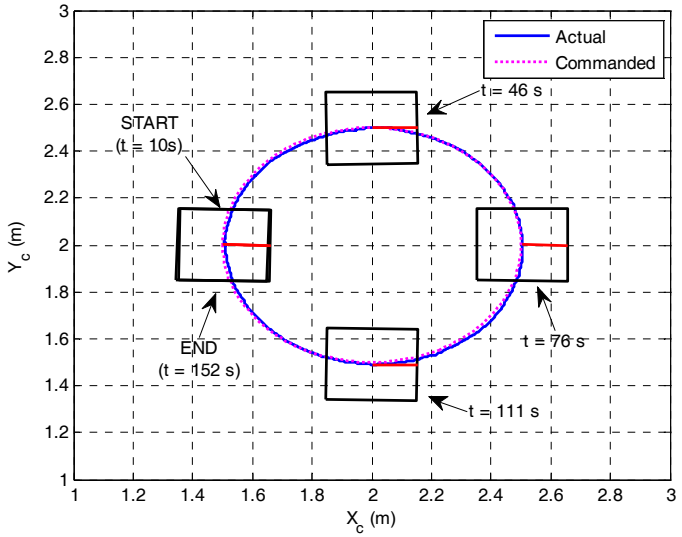


Fig. 11. Autonomous proximity maneuver of NPS SRL's 3-DoF spacecraft simulator along a closed path using only thrusters

7. Conclusion

A planar laboratory testbed was introduced for the simulation of autonomous proximity maneuvers of a uniquely control actuator configured spacecraft. The testbed consists of a floating robotic simulator equipped with dual vectorable cold-gas thrusters and a miniature control moment gyro floating via planar air bearings on a flat floor. Inertial position and attitude measurements are obtained with a discrete Kalman filter and linear quadratic estimator for navigation; feedback linearized control coupled with a linear quadratic regulator is used to command the control moment gyro and while the same feedback linearized controller is used coupled with Schmitt triggers and Pulse Width Modulation to command the vectorable thrusters.

The presented experimental tests of autonomous closed path proximity maneuvers of the spacecraft simulator offer significant sample cases. The experimental results, which show good repeatability and robustness against disturbance and sensor noise, validate the proposed estimation and control approaches and demonstrate in particular, the small time local controllability of the system, confirming the analytical results. The achieved accuracy in following the reference trajectory (respectively ~ 1 cm for translation and ~ .5 deg for rotation given only the vectorable thrusters as control inputs) demonstrates both a feasible and promising actuator configuration for small spacecraft.

NPS SRL's robotic spacecraft simulator testbed, despite its reduction to only 3-DoF, allows experiments to be conducted in a low-risk and relatively low-cost environment where intermediate validation can occur between analytical/numerical simulations and full flight proximity navigation missions. Furthermore, the controllability analysis and the algorithms proposed for the state estimation and control can be in principle extended to full-fidelity 6-DoF spacecraft applications. The next step in this ongoing research will focus on the expansion of the presented analytical methods for non-linear control-affine systems with drift to numerical simulations on a full 6-DoF spacecraft model as well as work to develop further controllers that can take advantage of the minimum number of control actuator configuration of only two thrusters and no momentum exchange devices.

8. References

Bullo, F. & Lewis, A.D. (2005). *Geometric Control of Mechanical Systems*, Springer Science+Business Media, Inc., ISBN:0-387-22195-6, New York, NY, USA

Bryson, A.E. (1994). *Control of Spacecraft and Aircraft*, Princeton University Press, ISBN: 0-691-08782-2, Princeton, NJ, USA

Bevilacqua, R., Hall, J.S., Horning, J. & Romano, M. (2009). Ad Hoc Networking and Shared Computation Based Upon Linux for Autonomous Multi-Robot Systems, *Journal of Aerospace Computing, Information, and Communication*. To Appear.

Canfield, S.L. & Reinholtz, C.F. (1998). Development of the Carpal Robotic Wrist, *Lecture Notes in Control and Information Sciences*, Vol. 232, pp. 423-434, ISBN: 978-3-540-76218-8, Springer Berlin

Corrazzini, T. & How, J.P. (1998) Onboard GPS Signal Augmentation for Spacecraft Formation Flying, *Proceedings of the 11th International Technical Meeting of the Satellite Division of the Institute of Naviagation (ION) GPS 1998*, pp. 1937-1946, Nashville, TN, September 1998, ION, Manassas, VA, USA

Crassidis, J.L. & Junkins, J.L. (2004). *Optimal Estimation of Dynamic Systems*, CRC Press, LLC, ISBN: 1-58488-391-X, Boca Raton, FL, USA

Creamer, G. (2007). The SUMO/FREND Project: Technology Development for Autonomous Grapple of Geosynchronous Satellites, *Advances in the Astronautical Sciences*, Vol. 128, pp. 895-910, ISBN: 978-0-87703-542-8, San Diego, CA, USA

Eikenberry, B.D. (2006). *Guidance and Navigation Software Architecture Design for the Autonomous Multi-Agent Physically Interacting Spacecraft (AMPHIS) Testbed*, M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA

Gelb, A. (1974). *Applied Optimal Estimation*, The MIT Press, ISBN: 0-262-57048-3, Cambridge, MA, USA

Hall, J.S. (2006). *Design and Interaction of a Three Degrees-of-Freedom Robotic Vehicle with Control Moment Gyro for the Autonomous Multi-Agent Physically Interacting Spacecraft (AMPHIS) Testbed*, M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA

Hall, J.S. & Romano, M. (2007). Autonomous Proximity Operations of Small Satellites with Minimum Numbers of Actuators, *Proceedings of the 21st AIAA/USU Small Satellite Conference,* Logan, UT, USA, August 2007, AIAA/USU

Hall, J.S. & Romano, M. (2007). Novel Robotic Spacecraft Simulator with Mini-Control Moment Gyroscopes and Rotating Thrusters, *Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1-6, ISBN: 878-1-4244-1264-8, Zurich, Switzerland, September 2007, IEEE

Isidori, A. (1989). *Nonlinear Control Systems: An Introduction*, Springer-Verlag New York, Inc., ISBN: 0-387-50601-2, New York, NY, USA

Kennedy, F. (2008). Orbital Express: Accomplishments and Lessons Learned, *Advances in the Astronautical Sciences*, Vol. 131, pp. 575-586, ISBN: 878-0-87703-545-9, San Diego, CA, USA

Lappas, V.J.; Steyn, W.H. & Underwood, C.I. (2002). Practical Results on the Development of a Control Moment Gyro Based Attitude Control System for Agile Small Satellites, *Proceedings of the 16th Annual AIAA/USU Small Satellite Conference*, Logan, UT, USA, August 2002, AIAA/USU

LaValle, S.M. (2006). *Planning Algorithms*, Cambridge University Press, ISBN: 0-521-86205-1, New York, NY, USA

Ledebuhr, A.G.; Ng, L.C.; Jones, M.S.; Wilson, B.A.; Gaughan, R.J.; Breitfeller, E.F.; Taylor, W.G.; Robinson, J.A.; Antelman, D.R. & Nielsen, D.P. (2001). Micro-Satellite Ground Test Vehicle for Proximity and Docking Operations Development, *Proceedings of the 2001 Aerospace Conference*, Vol. 5, pp. 2493-2504, ISBN: 0-7803-6599-2, Big Sky, MT, USA, March 2001, IEEE

LeMaster, E.A; Schaechter, D.B & Carrington, C.K. (2006). Experimental Demonstration of Technologies for Autonomous On-Orbit Robotic Assembly, *Space 2006,* pp. 1-13, San Jose, CA, USA, September 2006, AIAA

Lewis, A.D. & Murray, R.M. (1997). Configuration Controllability of Simple Mechanical Control Systems, *SIAM Journal on Control and Optimization,* Vol. 35, No. 3, pp. 766-790, SIAM

Lugini, C. & Romano, M. (2009). A ballistic-pendulum test stand to characterize small cold-gas thruster nozzles, *Acta Astronautica*, Vol. 64, No. 5-6, pp. 615-625, Elsevier LTD, DOI: 10.1016/j.actaastro.2008.11.001

Machida, K.; Toda, Y. & Iwata, T. (1992). Maneuvering and Manipulation of Flying Space Telerobotics System, *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 3-10, ISBN: 0-7803-0737-2, Raleigh, NC, USA, July 1992, IEEE

Marchesi, M.; Angrilli, F. & Venezia, R. (2000). Coordinated Control for Free-flyer Space Robots, *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 5, pp. 3550-3555, ISBN: 0-7803-6583-6, Nashville, TN, USA, October 2000, IEEE

Mathieu, C. & Weigel, A. L. (2005). Assessing the Flexibility Provided by Fractionated Spacecraft, *Space 2005,* pp. 1-12, Long Beach, CA, USA, August 2005, AIAA

Nolet, S.; Kong, E. & Miller, D.W. (2005). Design of an Algorithm for Autonomous Docking with a Freely Tumbling Target, *Proceedings of Modeling, Simulation and Verification of Space-based Systems II,* Vol. 5799, No. 123, pp. 123-134, Orlando, FL, USA, March 2005, SPIE

Price, W. (2006). *Control System of a Three DOF Spacecraft Simulator by Vectorable Thrusters and Control Moment Gyros*, M.S. Thesis, Naval Postgraduate School, Monterey, CA, USA

Romano, M.; Friedman, A. & Shay, T.J. (2007). Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target, *Journal of Spacecraft and Rockets*, Vol. 44, No. 1, pp. 164-173, DOI: 10.2514/1.22092, AIAA

Romano, M. & Hall, J.S (2006). A Testbed for Proximity Navigation and Control of Spacecraft for On-orbit Assembly and Reconfiguration, *Space 2006,* pp. 1-11, San Jose, CA, USA, September 2006, AIAA

Roser, X. & Sghedoni, M. (1997). Control Moment Gyroscopes (CMG's) and their Applications in Future Scientific Missions, *Proceedings of the 3rd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, pp. 523-528, ESTEC Noordwijk, the Netherlands, November 1996, European Space Agency

Sussman, H.J. (1987). A General Theorem on Local Controllability, *SIAM Journal on Control and Optimization*, Vol. 25, No. 1, pp. 158-194, SIAM

Sussman, H.J. (1990). *Nonlinear Controllability and Optimal Control*, Marcel Dekker, Inc., ISBN: 0-8247-8258-5, New York, NY, USA

Slotine, J.E. & Weiping, L. (1991). *Applied Nonlinear Control*, Prentice-Hall, Inc., ISBN: 0-13-040890-5, Upper Saddle River, NJ, USA

Ullman, M.A. (1993). *Experimentation in Autonomous Navigation and Control of Multi-Manipulator Free-Flying Space Robots*, Ph.D. Dissertation, Stanford University, Stanford, CA, USA

Wie, B. (1998). *Space Vehicle Dynamics and Control*, American Institute of Aeronautics and A stronautics, Inc., ISBN: 1-56347-261-9, Reston, VA, USA

12

Integrated Environment of Simulation and Real-Time Control Experiment for Control system

Kentaro Yano and Masanobu Koga
Kyushu Institute of Technology
Japan

1. Introduction

A design process of a control system is generally executed in order of modelling, design of controller, simulation, and control experiment. If a control plant is a robot or an inverted pendulum etc, Real-Time control is required and control experiment programs should be Real-Time programs (RT programs). An RT program is the program which assures the time limit of the process beginning and the process completion (Funaki & Ra, 1999). And, the control experiment program which is an RT program is called an RT control program.
An RT control program is often written by using a library provided by a Real-Time OS (RTOS) like RT-Linux (RTLinuxFree; Funaki & Ra, 1999). At this time, it is necessary to find the parts which should be altered by the change of the control plant from whole of the program. Since, the target-depend parts are scattering at large range of the program. Also, there is a high possibility that the miss which forget the partial change etc. get mixed in with the program.
A simulation is run to confirm the performance of the controller, and a simulation program is often written in a numerical computation language which makes it easy to write a mathematical formula (The MathWorks Matlab; Koga, 2000). After the affirmation of the results of the simulation, an RT control program is newly written. Therefore, it is impossible to execute the design process of control system efficiently, because an individually creation of a simulation program and an RT control program is needed and smoothly change from simulation phase to control experiment phase is impossible.
To solve this issue, methods which create RT control programs using the information written at simulation programs are proposed. For example, by using RtMaTX (Koga et al., 1998), it is able to create RT control programs by edit the function written in MaTX (Koga, 2000) which is a numerical computation language. And, Real-Time Workshop (RTW) (The MathWorks Real-Time Workshop) generates RT control programs written in C language from block diagrams of Simulink (The MathWorks Simulink). A method which improves RTW to industrial applications and generates iFix (GE Fanuc Automation) etc. from Matlab codes is also proposed (Grega & olek, 2002).

A simulation is run on a common OS like Windows, but an RT control is often run on an RTOS. And, the platform (hardware or OS) for a simulation and the platform for an experiment are often different. Then, it is necessary to deploy the RT control program which is automatically generated from a simulation program on the machine which executes a control experiment. Since a simulation and a control experiment are executed repeatedly, it is necessary to repeatedly change a simulation program, create an RT control program, and deploy it on an experiment machine.
To reduce a workload of a deployment of an RT control program, a method which uses a Web browser is proposed (Basso & Bangi, 2004). In this method, users can execute an automatic generation of an RT control program and control experiment by only selecting the simulation program written in Simulink on a Web browser. However, the simulation is executed by Simulink, and the RT control is run by the Web browser, so because of the difference of the interface, users are easy to be confused by the operations. And, HTML client which is the application that runs on a Web browser and uses UI parts provided by HTML, has problems like it is impossible to provide rich functions because the UI is poorer than an UI of a common program.
The purpose of this paper is to propose methods to solve the issues and make it possible to efficiently execute the iteration of the simulation and the Real-Time control at the design process of control system. And, this paper describes about the integrated environment for simulation and Real-Time control which is the implementation of the proposed methods.
In Section 2, we propose the methods for the issues. In Section 3, the developed integrated environment to actualize proposed methods is described. In Section 4, an illustrative example which is a stabilization control of inverted pendulum is explained. Finally, in section 5, conclusions are described.

2. Integrate methods of simulation and Real-Time control

2.1 RT Control Framework

In this section, we propose an RT control framework which provides a frame of the creation of RT control programs for target-dependent parts are scattering at the large range of an RT program.

2.1.1 Framework

In software engineering, a framework is used commonly in the sense of the technique which provides a frame of the whole software (Nakano, 2002). Based on a framework, it is possible to complete a final product efficiently by the creation where a developer corresponds to individual purposes. So, a framework is provided as a semi finished product. The parts which are changed or added by developers are called hotspots, and other parts are called frozen spots.
As shown in Fig. 1, a program written by users using a library only calls and uses codes in a library. On the other hand, in a framework, it calls and uses codes developed by users based on a specification of a framework.

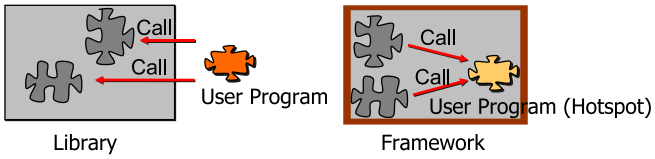


Fig. 1. Framework and Library

2.1.2 Architecture

The architecture of the framework which we propose is shown in Fig. 2. The RT control framework provides the frame of the creation of RT control programs. Target-dependent parts become apparent and it is able to control centrally by using the RT control framework. It is also possible to create an RT control program by only creating the target-dependent parts. Therefore, it makes writing RT control programs easy, and it is possible to raise the efficiency. It is possible to write an RT control program only creating pieces of Real-Time task, experimental data viewer, and commands interpretation. The details of the architecture are shown below.

- Real-Time Task module

Handle Real-Time processing, and compute the control law etc.

- Real-Time Task Management module

Generate/management the Real-Time Task

- Experimental Data Management module

Manage the experimental data

- Experimental Data Viewer module

Display the experimental data

- User Interface module

Handle commands (start/end of control etc.) from the user

- Commands Interpretation module

Interpret commands from the user, and handle them.

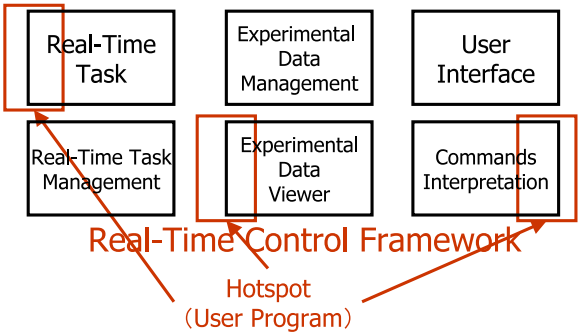


Fig. 2. Architecture of framework

2.2 Transformation of simulation program using object model

To solve the issues about simulation programs depend a platform, and individually creation of a simulation program and an RT control program is needed, this paper proposes the

transformation of a simulation program using the object model and Design Pattern (Gamma et al., 1995). First, this paper proposes a method which transforms a simulation program written in a numerical computation language to a simulation program written in a platform independent language. And, this paper also proposes a method which generates a Real-Time processing code（hotspots of the RT control framework）using Factory Method pattern (Gamma et al., 1995). It is able to raise the efficiency of the design process of control system by generating an RT control program automatically from a simulation program written in a numerical computation language. It is able to restrict the generating program to target-dependent parts by generating the hot spots of the RT control framework. And, it is able to execute a simulation and a control experiment on the same platform by transforming a simulation program to a platform independent language.

2.2.1 Object Model

An object model is the data structure which is the object oriented model of the data used by a program, and it is the interface of given operations for that. An object model of a programming language is a set of classes to create the objects which have same information of the source code that was written in that language. Figure 3 shows the schematic of the object modeling of a source code as an object tree which has the same information of the source code. Let a function which generates another programming language is defined at each node. Then, it is possible to transform the original source code to another programming language by pursuing the object tree and calling the defined functions written at each node.

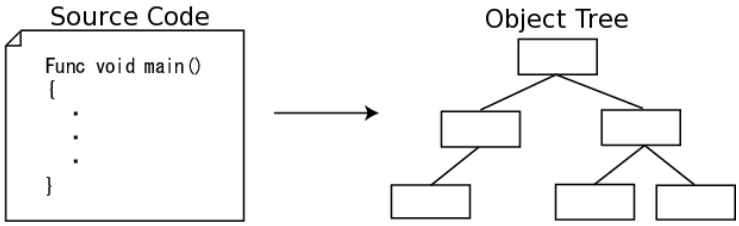


Fig. 3. Object Model

2.2.2 Transformation of program using object model

Figure 4 shows the schematic of the transformation of a simulation program using the object model. First, Parser receives a source code written in a numerical computation language. Parser construes the source code, and creates the object tree which has the information of the source code written in the numerical computation language based on the syntax. Objects which define a function to generate a platform independent code are used to build the tree. Then, the simulation program written in the platform independent language is generated from the generated object tree.

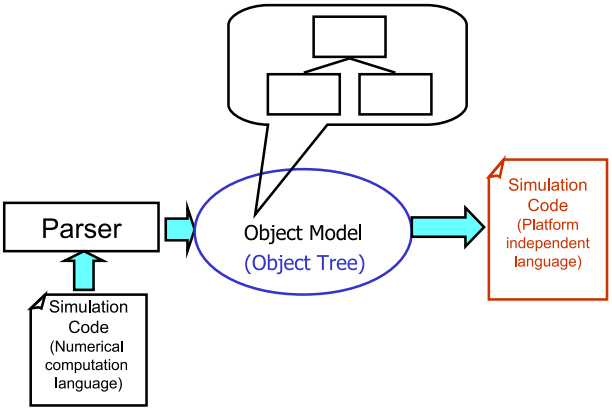


Fig. 4. Transformation of simulation program using Object Model

2.2.3 Generation of RT program using Factory Method pattern

Factory Method pattern is a design pattern which decides how to create an instance of a class defined at a super class, and the concrete creation of an instance is handled by each subclass (Gamma et al., 1995). The Factory Method Pattern defines that Creator creates Product. And ConcreteProduct which is the real instance is created by ConcreteCreator which is the subclass of Creator. Then, the class which provides the frame of the instantiation and classes for the generation of the real instance are separated.

It is possible to switch objects which are embedded in an object tree for any purpose by using Factory Method pattern when creating an object tree.

Figure 5 shows the schematic of the transformation of a simulation program using Factory Method pattern. Factory Method pattern are applied where Parser creates the object model. And, it defines that Creator creates the object model (Product). Objects (ConcreteProduct) which is used to generate a simulation code is created by SimCreator (ConcreteCreator) which is an implementation class of Creator.

Figure 6 shows the schematic of the automatic generation of an RT program using Factory Method pattern. Objects（ConcreteProduct）which generate an RT program automatically is created by RTCreator(ConcreteCreator）which is a subclass of SimCreator. Then, it is able to transform a platform independent simulation program and generate a Real-Time processing code automatically by using the object model which has the same structure. If a subclass of Creator and classes which are embedded in the object tree are created, it is possible to add another kind of code.

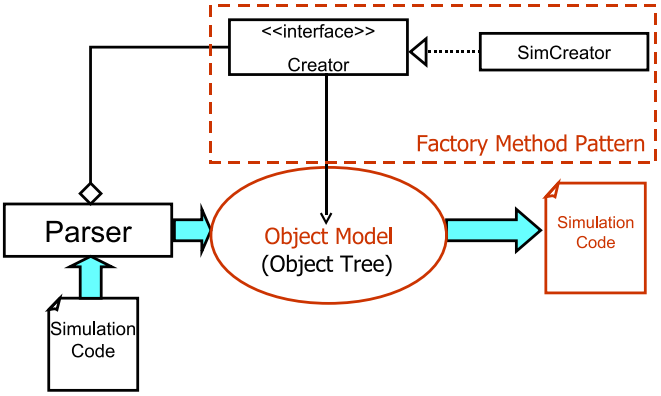


Fig. 5. Transformation of simulation program using Factory Method Pattern

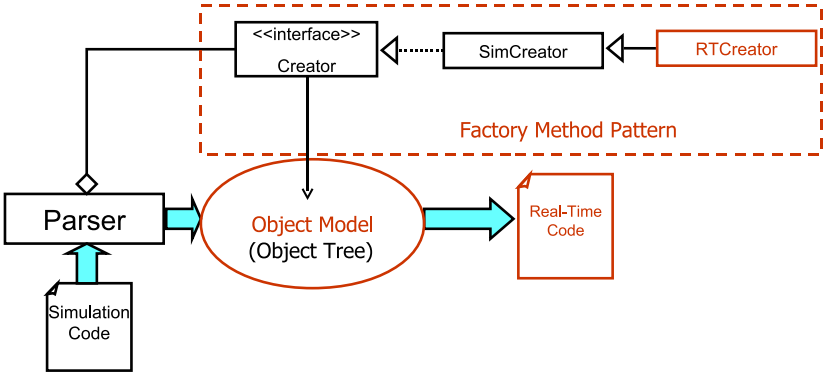


Fig. 6. Automatic generation of Real-Time control program using Factory Method Pattern

2.3 Separation of platform dependent parts

To solve the issue of RT control programs depend on a platform, this paper proposes a method which separates platform dependent parts and independent parts. And, this paper proposes that separates platform dependent parts of the RT control framework.

2.3.1 Separation of platform dependent parts of RT control program

In general, an RT control program consists two parts. One is Real-Time part which needs a Real-Time processing, and another part is non Real-Time part which doesn't need a Real-Time processing. If Real-Time part is written in a platform correspond language, and non Real-Time part is written in a platform independent language, non Real-Time part can be independent from a platform. The outline about the separation of platform dependent parts is shown by Fig. 7.

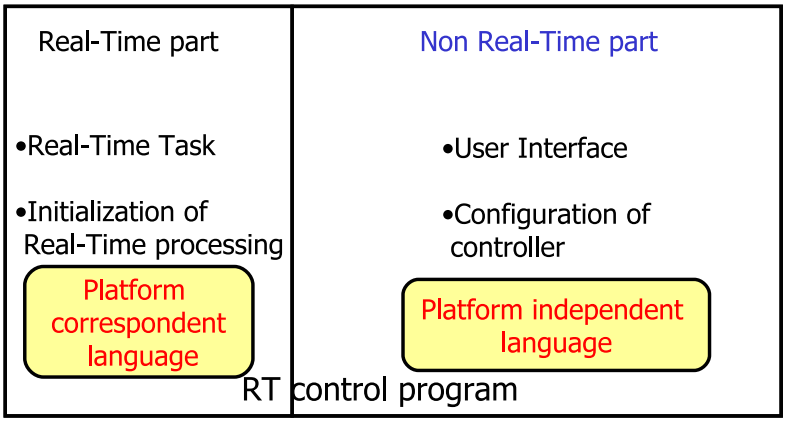


Fig. 7. Separation of platform dependent parts

2.3.2 Server and client system

This section describes an improvement of a portability of programs by combining of the method mentioned the above section and server and client system which use network.

Figure 8 shows the schematic of server and client system. A plant is connected to a server machine. The server machine is installed an RTOS, and it can execute RT control programs. The client machine is connected to the server machine via network.

In this system, Real-Time processing is executed by only the server machine, and the client machine runs only non Real-Time processing. And, non Real-Time information is sent between the server machine and the client machine. So, the client machine doesn't need the special environment for Real-Time programs. And, the client machine doesn't depend on the particular OS, since non Real-Time processing parts are platform independent. So, the combining of the separation of platform independent parts and server and client system improve the portability of the program.

And, if a mechanism which automatically send an RT control program from the client machine to the server machine is implemented, the issue about deploy of an RT program is solved.

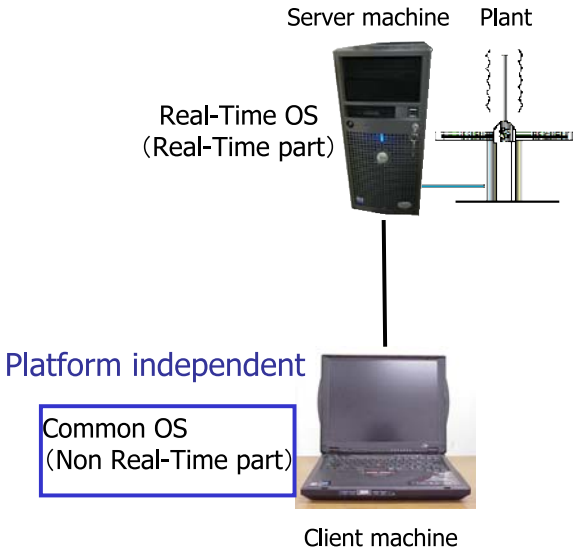


Fig. 8. Server and client system

3. Implementation of proposed method

We implemented the proposed methods, and developed the integrated environment for simulation and Real-Time control experiment using C language and Java language.
Figure 9 shows the schematic of the integrated environment.
An RT control program using the RT control framework is generated automatically from a simulation program by the integrated environment. The integrated environment can raise the efficiency of the design process of control system, since users can run the integrated simulation and Real-Time control experiment.

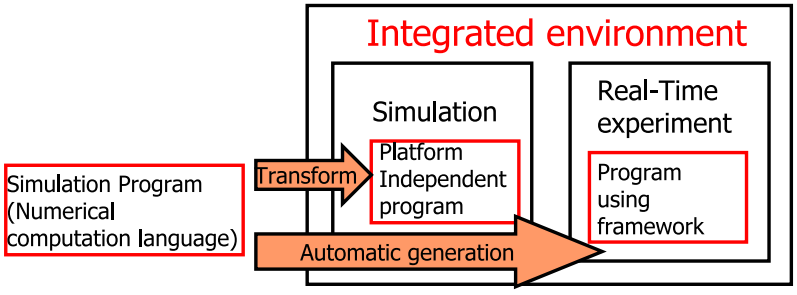


Fig. 9. Schematic of Integrated Environment

Figure 10 shows the architecture of the developed integrated environment. The RT control framework runs on an RTOS and the Java VM. The framework has two hotspots. Hotspot 1use the function of an RTOS, and hotspots 2use the function of the Java VM. It is able to execute the simulation and the control experiment by using the GUI which runs on the

framework. And the automatic code generation is used for the simulation and the control experiment.

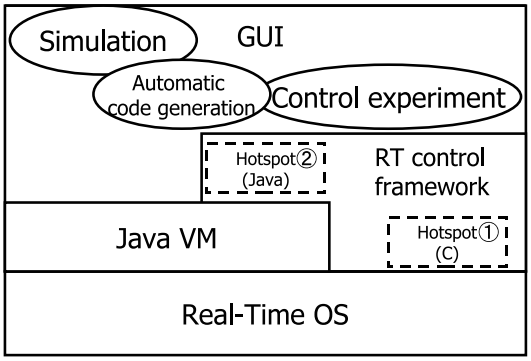


Fig. 10. Architecture of Integrated Environment

3.1 RT control framework

We developed ReTiCoF(Real Time Control Framework) (Yano & Koga, 2006) which is the implementation of the Real-Time control framework proposed by this paper. ReTiCoF runs on RT-Linux (RTLinuxFree) which is one of the RTOS, and it is implemented by C language and Java language.

The architecture of MK-Task which is the execution parts of Real-Time processing in ReTiCoF is shown in Fig. 11. MK-Task is implemented by C language, and it provides mechanisms for easy use of functions of RT-Linux.

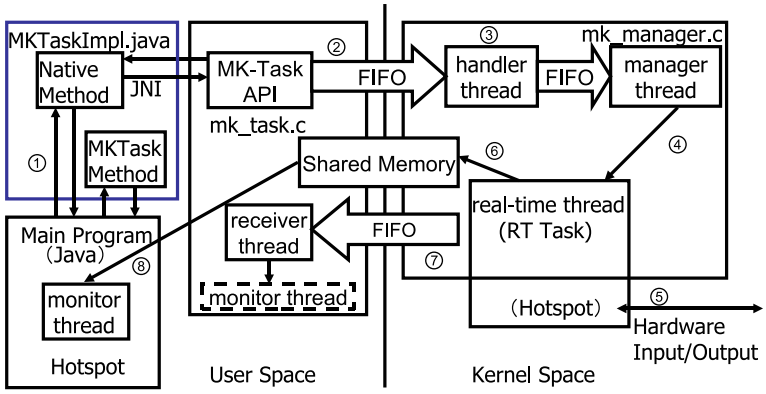


Fig. 11. Architecture of MK-Task

MK-Task consists of mk_task.c which runs on User space, and mk_manager.c which runs on Kernel space. mk_task.c has the APIs of MK-Task, and codes of receiver thread which receives data from Kernel space. mk_manager.c has the handler thread which runs when a call from User space is sent, manager thread which manages a Real-Time task, and Real-Time task as thread which has the Real-Time constraint. And MKTaskImpl.java has native

methods which call the APIs of MK-Task using JNI, and methods which are needed when MK-Task is used from Java. JNI(Java Native Interface) (Gordon, 1998) is the technique to use a native code like C language from Java.
The process flow of the Real-Time control using MK-Task is shown below.

1. Main program(Java) calls the function of MK-Task as a native method of MKTaskImpl class
2. A request form user space using the function of MK-Task is sent to Kernel space via RT-FIFO
3. Handler thread starts
4. The request received by handler thread is passed to manager thread, and manager thread manages the RT task as required
5. The RT task inputs or outputs to a hardware, and execute the Real-Time control
6. Experimental data of the Real-Time control are written into the shared memory of user space and kernel space
7. Receiver thread receives total count of data from RT task via RT-FIFO
8. Monitor thread receives the data from the shared memory, and display the data

3.2 Transformation of program using object model

The developed integrated environment employed MaTX as the numerical computation language, and Java as the platform independent language. The integrated environment creates an object model of MaTX from a source code of MaTX using matj(Koga et al., 2005) which is an interpreter of MaTX implemented by Java, and generates a source code of Java by pursing the object model.
Factory Method pattern is applied to the Parser of matj, and a Real-Time processing code (hotspots of the RT control framework) is generated automatically from a simulation program written in MaTX. In particular, we created classes which generates Real-Time code(matrix calculations, trigonometric functions etc.), and a class which embeds those classes to an object tree.
The integrated environment is also able to generate an RT control program from a block diagram created by Jamox (Koga et al., 2006). The block diagram which is supported by the integrated environment has a controller created by a block of MaTX or cascade connections of linear systems.
Jamox can run the modeling and the simulation, so it is possible to execute all parts of the design process of control system by using Jamox and the integrated environment.
Jamox(Java Agile MOdeling Tool for Control System) is a modeling and simulation tool for control system using an adjacency matrix, and it supports a linear system and a nonlinear system modeled by continuous time, discrete time, and sampled data system(which is a mix of continuous time and discrete time).
An adjacency matrix is one of an expressive form of a graph used for graph theory. It is easy to simulate a control system represented by a block diagram drawn by Jamox. Jamox not only can simulate time response of system, but also can compute a state space representation of all system and frequency response (Bode diagram) easily, if that system is a linear.
Figure 12 shows an execution screen of Jamox. It is able to allocate blocks (provided by the tool) by drag and drop from the library to the canvas. Each block is able to be connected by mouse action, and it is easy to create a block diagram on the screen displays.

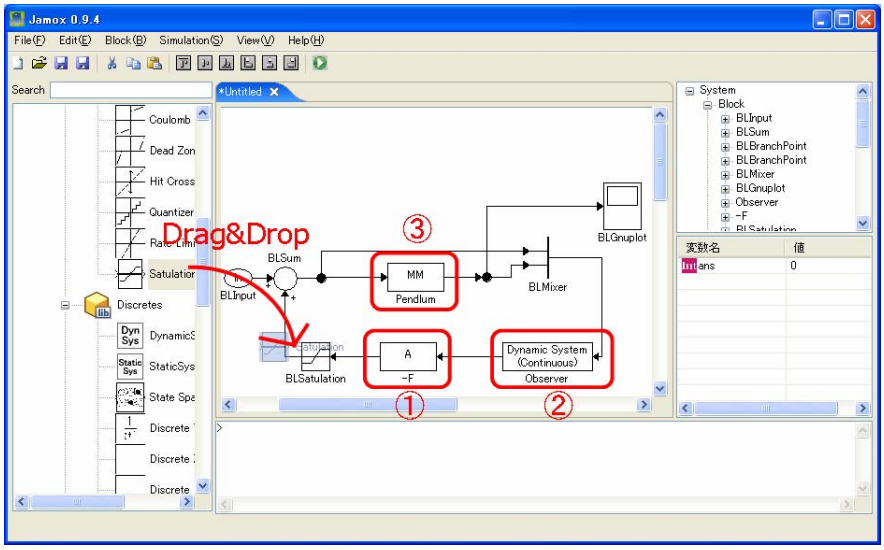


Fig. 12. Modeling and Simulation tool Jamox

The block 1is a basic block registered by the library. Kinds or the structure of the models are defined, and only the parameter is customizable. The block 2is a customer-defined block which represents any dynamic systems and static systems. And it is related with the Java class which extends *DynamicSystem* class or *StaticSystem* class. It is able to create the high reusable blocks efficiently based on the grouping of the control system and use of the inheritance of class.And, it is easy to create the fast computation code using NFC (Koga & Matsuki, 2003) which is a Java numerical computation package. The block 3is a block which treats a system written in MaTX syntax, and it is possible to reuse existing MaTX codes.
In the simulation calculation, the solution of a differential equation is solved by *RungeKutta* class etc. which implements *OdeSolver* interface provided by Ode package of NFC. If an algebraic loop exists, a solution of a nonlinear simultaneous equation is solved by using Newton-Raphson method**.**

3.3 Separation of platform dependent parts

We use C language for the platform correspond language and Java language for the platform independent language. We use JNI to use C language and Java language together.
In this integrated environment, server and client system is implemented by using RMI(Remote Method Invocation) (Trail: RMI) which is the implementation of distributed objects in Java. The distributed object is an object which is able to call methods through network. And, we implement the mechanism which sends an RT control program from a client machine to a server machine using RMI.

3.4 Local and remote compatible GUI

Figure 13 shows the GUI provided by the integrated environment. It has the menu bar and the toolbar which are related to the simulation, the automatic generation, and the control

experiment. It also has the display part of the experimental data, the property of the Real-Time task, and messages for user.

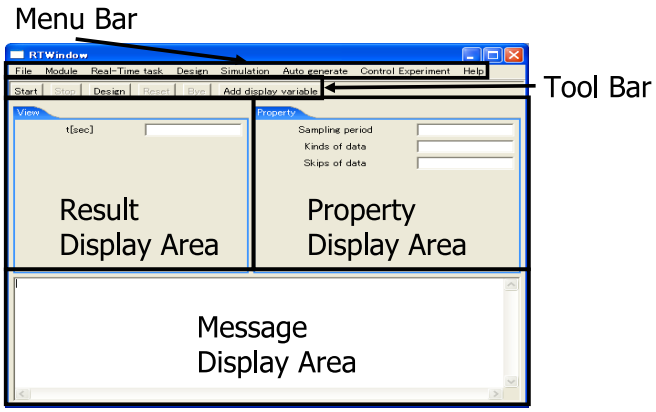


Fig. 13. GUI of integrated environment

In this implemented integrated environment, it is possible to use the same GUI between the local control using only the server machine, and the remote control using the server machine and the client machine. Proxy pattern (Gamma et al., 1995) is used for the mechanism of the local and remote compatible GUI. Proxy pattern is the design pattern which makes Proxy class as the proxy of RealSubject class which is the real actor, and it handles processes of RealSubject class as much as possible.
Figure 14 shows the class diagram of the integrated environment using Proxy pattern. The GUI which is shown by Fig. 13 is provided by RTWindow class. RTWindow class has the classes which provide the GUI parts like ViewPanel class, and the implementation class of MKTask interface which executes the Real-Time control. MKTaskImpl class is used for the local control, RemoteMKTaskProxy(Proxy) class is used for the remote control instead of RemoteMKTaskImpl(RealSubject).

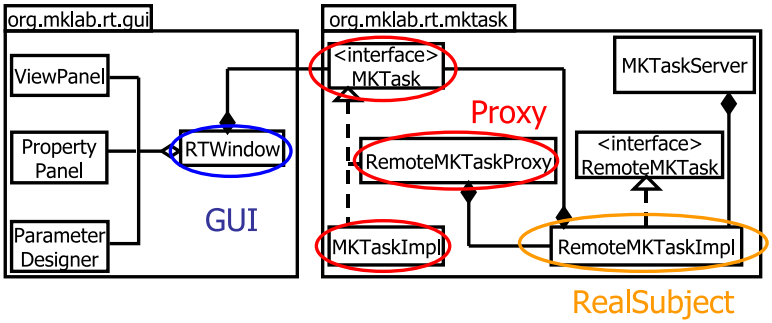


Fig. 14. Class Diagram of Real-Time control system

4. Experiment of stabilization control of an inverted pendulum

To verify the effectivity of the method proposed by this paper, we execute the experiment of the stabilization control of an inverted pendulum using the developed integrated environment.
First, do modeling a pendulum using Jamox and design a controller using MaTX. Figure 15 shows the result of the modeling and the design of controller.

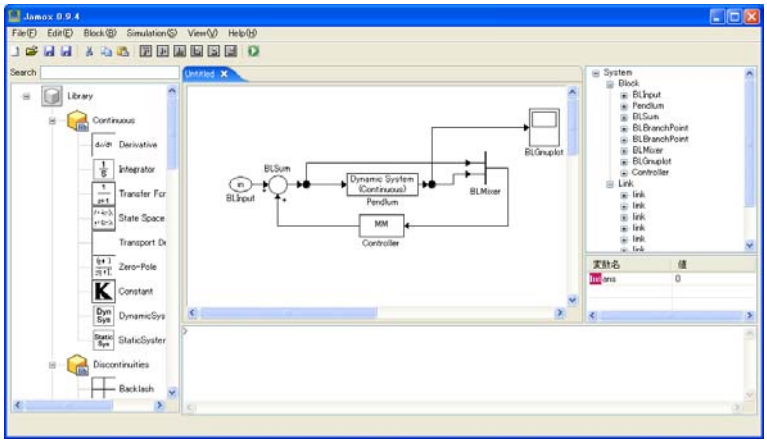


Fig. 15. Modeling of pendulum using Jamox

Simulation is executed by using Jamox. At this time, a user selects a solver of the differential equation, and input the start/end time of the simulation, and the initial state. Figure 16 shows the results of the simulation.

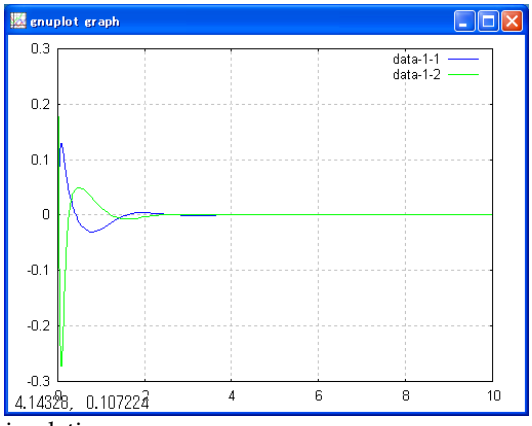


Fig. 16. Execution of simulation

In the automatic generation of Real-Time program, user selects the simulation program(created by using Jamox), the function name to extract information, the name of the

output variable, and input the name/unit of display variables for the GUI. Figure 17 shows the dialog provided by the GUI of the integrated environment, and it is used to input the name/unit of display variables for the GUI.
After the above information is inputted, the Real-Time program is generated automatically. And the variable displayed on the GUI is added, the display of the GUI is changed for the pendulum.

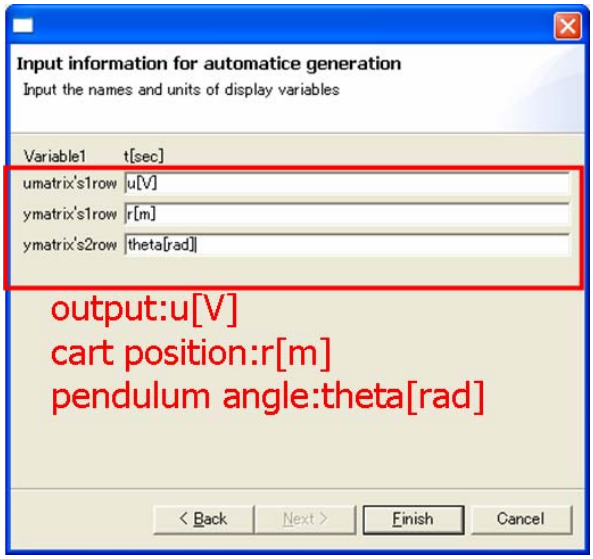


Fig. 17. Display variable name・unit for GUI

The experiment is executed, after select the generated RT control program, the module of the experimental apparatus, and input the parameter of the controller. Figure 18 shows the screen of the RTWindow which is executing the control experiment.
It is shown that the GUI has changed for the inverted pendulum by using the information input by Fig. 17 by comparison Fig. 13 and Fig. 18.
Figure 19 shows the experiment results, and abscissa axis is time[sec], ordinate axis is angle of the pendulum[rad]. And, the sampling time of the experiment is 5[ms].
As shown in Fig. 19, the angle of the pendulum is close to the 0[rad], the experiment of the stabilization control of the inverted pendulum become successful.

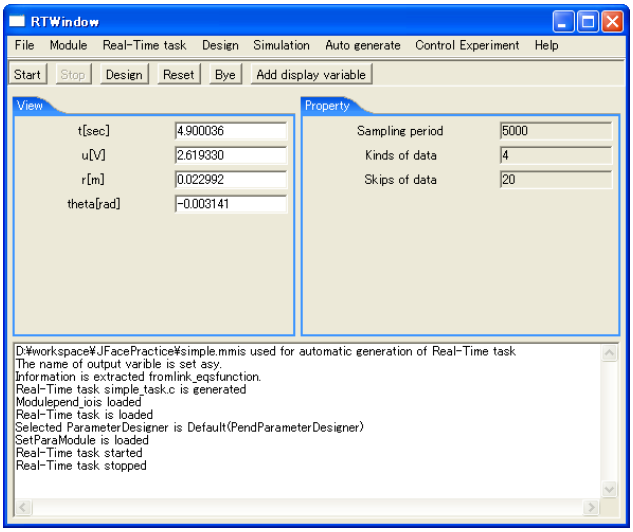


Fig. 18. Execution of control experiment

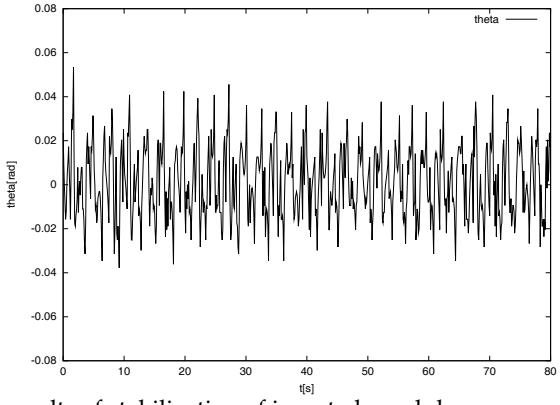


Fig. 19. Experiment results of stabilization of inverted pendulum

5. Conclusions

This paper proposed the methods which make the execution of the iterative design process of control system efficiently. In particular, this paper proposed the method which is based on the RT control framework, transformation of a program using the object model, and separation of platform dependent parts. And, we developed the integrated environment for the simulation and the Real-Time control experiment which is the implementation of proposed methods. The effectivity of the proposed methods was shown by the stabilization of an inverted pendulum.

In the implementation example, we used RT-Linux as the RTOS, but a method which runs RT control program on Linux Kernel 2.6(Kishida & Koga 2005) is proposed. So, we would like to make the integrated environment is corresponded to Linux Kernel 2.6 in future works.

6. References

Basso, M. & Bangi, G. (2004). ARTIST:A Real-Time Interactive Simulinkbased Telelab, *proceedings of the 2004 IEEE Conference on Computer Aided Control Systems Design*, pp.196-201

Funaki, M. & Ra, S. (1999). *Guide book for Real-Time sensing and control by Linux*, Shuwa system, 97804879668493

Gamma, E.; Helm, R. Johnson, R. & Vlissides, J. (1995). *Design Patterns:Elements of Reusable Object-Oriented Software*, Addison-Wesley Pub, 978-0201633610

GE Fanuc Automation: Proficy HMI/SCADA - iFIX, http://www.gefanuc.com/en/ProductServices/AutomationSoftware/Hmi Scada/iFIX/

Gordon, R. (1998). *Essential JNI: Java Native Interface*, Prentice Hall Ptr, 978-0136798958

GREGA, W. & KOLEK, K. (2002). Simulation and Real-Time Control:from Simulink to Industrial Applications, *2002 IEEE International Symposium on Computer Aided Control System Design Proceedings*, pp.104-109

Kishida, K.; Koga, M. (2005). Development of Real-Time Control Package with Linux Kernel 2.6, 49th Annual Conference of the Institute of Systems, Control and Information Engineers

Koga, M.; Tsutsui, Y. & Yabuuchi, J. (2006). Java Simulation Platform for Control System based on Block Diagram, *IEEE 2006 CCA/CACSD*, pp.2304–2308

Koga, M.; Matsuki, T & Sada, H. (2005). Development of Environment of Numerical Computation in Java based on Object Model of Programming Language, 49th Annual Conference of the Institute of Systems, Control and Information Engineers

Koga, M. & Matsuki, T. (2003). Development of OS-Neutral Numerical Foundation Class Library and its Application to Control System, *SICE 6th Annual Conference on Control Systems*

Koga, M. (2000). *MaTX for control/numerical analysis*, Tokyo Denki University Press, 978-4501531003

Koga, M. Toriumi, H. & Sampei, M. (1998). An integrated software environment for the design and real-time implementation of control systems, *Control Engineering Practice*, Vol. 6, pp. 1287–1293

Nakano, T. (2002). Understanding the framework, *Java WORLD*, Vol. 6, No. 62, 54–67

RTLinuxFree, http://www.rtlinuxfree.com/

The MathWorks Matlab, http://www.mathworks.com/products/matlab/

The MathWorks Real-Time Workshop, http://www.mathworks.com/products/rtw/

The MathWorks Simulink, http://www.mathworks.com/products/simulink/

Trail: RMI http://java.sun.com/docs/books/tutorial/rmi/index.html

Yano, K. & Koga, M. (2006). Platform Independent Integrated Environment for Simulation and Real-Time Control Experiment, *SICE-ICASE International Joint Conference 2006*

Reliability Analysis Methods for an Embedded Open Source Software

Yoshinobu Tamura † and Shigeru Yamada ‡
Graduate School of Science and Engineering, Yamaguchi University†
Graduate School of Engineering, Tottori University‡

1. Introduction

Many software systems have been produced under host-concentrated development environment. In such host-concentrated one, the progress of software development tools has caused several issues. For instance, one of them is that all of software development management has to be suspended when the host computer is down. Since the late 1980s, personal computers have been spread on our daily life instead of conventional mainframe machines, because the price and performance of personal computers have been extremely improved. Hence, computer systems which aid the software development have been also changing into UNIX workstations or personal computers to reduce the development cost. A Client/Server System (CSS) which is a new development method have come into existence as a result of the progress of networking technology by UNIX systems. Such CSS's have been used more and more in the period of network computing. The CSS's are horizontally distributed systems which consist of a server and client computers. The CSS's differ from conventional host/terminal computer systems from the point of view that the CSS's have the property that each computer on network can be a server or client as well. Thus, the CSS's have expanded with the technique of internet. At present, the software development environment has been changing into distributed one because of such progress of network computing technologies. For instance, basic CSS's which consists of 2-layers structure have been expanded to N-layers one, because such CSS's can be easily and rapidly introduced for the purpose of software development with low cost. The recent progress of network technologies in social systems is remarkable. As a result of the progress, software development environment has been changing into new development paradigm in such CSS's and distributed development by using network computing technologies (Takahashi, 1998; Umar, 1993; Vaughn, 1994).
The methodology of the object-oriented design and analysis is a feature of such distributed development environment and greatly successful in the field of programming-language, simulation, GUI (graphical user interface), and constructing on database in the software development. A general idea of object-oriented design and analysis is developed as a technique which can easily construct and maintain the complex systems. Therefore, the distributed development paradigm based on such an object-oriented methodology will rapidly grow in the future, because this technique is expected as a very effective approach to

improve software quality and productivity. Software composition by object-oriented technologies is expected as a very effective approach to improve software quality and productivity. Considering the software composition, it is expected that even the host-concentrated development environment can yield the quality of software system to some extent regardless of the content of applications, because the software system is structured on a single hardware environment. On the other hand, it is known that software systems under distributed development environment are difficult to be developed, since the architecture of such systems can have different development styles.

As mentioned above, software development environment has been changing into new development paradigms such as concurrent distributed development environment and the so-called open source project by using network computing technologies. Especially, such Open Source Software (OSS) systems which serve as key components of critical infrastructures in the society are still ever-expanding now (E-Soft Inc.).

Software reliability growth models (SRGM's) (Misra, 1983; Musa et al. 1987; Yamada & Osaki, 1989; Yamada, 1991; Yamada 1994) have been applied to assess the reliability for quality management and testing-progress control of software development. On the other hand, the effective method of testing management for the new distributed development paradigm as typified by the open source project has only a few presented (Kuk, 2006; Li et al. 2004; MacCormack et al. 2006; Zhoum & Davis, 2005). In case of considering the effect of the debugging process on an entire system in the development of a method of reliability assessment for the OSS, it is necessary to grasp the deeply-intertwined factors, such as programming paths, size of each component, skill of fault-reporters, and so on.

In this chapter, we discuss a useful reliability assessment method of an embedded OSS developed under open source project. In order to consider the effect of each software component on the reliability of an entire system under such open source project, we apply a neural network (Karunanithi & Malaiya, 1996; Lippmann, 1987). Also, we propose a software reliability growth model based on stochastic differential equations in order to consider the active state of the open source project. Especially, we apply the intensity of inherent software failures which means the software failure-occurrence rate or the fault detection rate for the i -th component importance level to the interaction among components by introducing an acceleration parameters. Also, we assume that the software failure intensity depends on the time, and the software fault-reporting phenomena on the bug tracking system keep an irregular state in terms of the number of detected faults. Moreover, in order to consider the effect of each software component on the reliability of an entire system under such open source software, we propose a method of reliability assessment based on the Bayesian network (BN) for OSS. Furthermore, we analyze actual software fault-detection count data to show numerical examples of software reliability assessment considering the component importance levels for the open source project.

2. Reliability Assessment Method

2.1 Weight parameter for each component

In case of considering the effect of debugging process on an entire system on software reliability assessment for open source development paradigm, it is necessary to grasp the deeply-intertwined factors, such as programming paths, size of each component, skill of fault-reporters, and so on.

In this chapter, we propose a method of reliability assessment based on the neural network in terms of estimating the effect of each component on the entire system in a complicated situation. Especially, we consider that our method based on neural network is useful to assess the software reliability by using only data sets in bug tracking system on the website. Also, we can apply the importance level of faults detected during the testing of each component, the size of component, the skill of fault-reporters and so on, to the input data of neural network.

We assume that $w_{ij}^{1}(i=1,2,\cdots,I;j=1,2,\cdots,J)$ are the connection weights from i-th unit on the sensory layer to j-th unit on the association layer, and $w_{jk}^{2}(j=1,2,\cdots,J;k=1,2,\cdots,K)$ denote the connection weights from j-th unit on the association layer to k-th unit on the response layer. Moreover, $x_i(i=1,2,\cdots,I)$ represent the normalized input values of i-th unit on the sensory layer, and $y_k(k=1,2,\cdots,K)$ are the output values. We apply the normalized values of fault level, operating system, fault repairer, fault reporter to the input values $x_i(i=1,2,\cdots,I)$. Then, the input-output rules of each unit on each layer are given by

$$h_j = f\left(\sum_{i=1}^{I} w_{ij}^{1} x_i\right), \tag{1}$$

$$y_k = f\left(\sum_{j=1}^{J} w_{jk}^{2} h_j\right), \tag{2}$$

where a logistic activation function $f(\cdot)$ which is widely-known as a sigmoid function given by the following equation:

$$f(x) = \frac{1}{1+e^{-\theta x}}, \tag{3}$$

where θ is the gain of sigmoid function. We apply the multi-layered neural networks by back-propagation in order to learn the interaction among software components (Karunanithi & Malaiya, 1996; Lippmann, 1987). We define the error function by the following equation:

$$E = \frac{1}{2}\sum_{k=1}^{K}(y_k - d_k)^2, \tag{4}$$

where $d_k(k=1,2,\cdots,K)$ are the target input values for the output values. We apply the normalized values of the total number of detected faults for each component to the target input values $d_k(k=1,2,\cdots,K)$ for the output values, i.e., we consider the estimation and prediction model so that the property of the interaction among software components accumulates on the connection weights of neural networks.

By using the output values, $y_k\ (k=1,2,\cdots,K)$, derived from above mentioned method, we can obtain the total weight parameter p_k which represents the level of importance for each component by using the following equation:

$$p_k = \frac{y_k}{\sum_{k=1}^{K} y_k} \quad (k = 1, 2, ..., K). \tag{5}$$

2.2 Reliability assessment for entire system

Let $S(t)$ be the cumulative number of detected faults in the OSS system by operational time $t\,(t \geq 0)$. Suppose that $S(t)$ takes on continuous real values. Since the latent faults in the OSS system are detected and eliminated during the operational phase, $S(t)$ gradually increases as the operational procedures go on. Thus, under common assumptions for software reliability growth modeling, we consider the following linear differential equation:

$$\frac{dS(t)}{dt} = \lambda(t) S(t), \tag{6}$$

where $\lambda(t)$ is the intensity of inherent software failures at operational time t, and a non-negative function. In most cases, the faults of OSS are not reported to the bug tracking system at the same time as fault-detection but rather reported to the bug tracking system with the time lag of fault-detection and reporting. As for the fault-reporting to the bug tracking system, we consider that the software fault-reporting phenomena on the bug tracking system keep an irregular state. Moreover, the addition and deletion of software components is repeated under the development of OSS, i.e., we consider that the software failure intensity depends on the time (Tamura & Yamada, 2007). Therefore, we suppose that $\lambda(t)$ in Eq.(6) has the irregular fluctuation. That is, we extend Eq.(6) to the following stochastic differential equation (Arnold, 1974):

$$\frac{dS(t)}{dt} = \{\lambda(t) + \sigma\gamma(t)\} S(t), \tag{7}$$

where σ is a positive constant representing a magnitude of the irregular fluctuation and $\gamma(t)$ a standardized Gaussian white noise. We extend Eq.(7) to the following stochastic differential equation of an Itô type:

$$dS(t) = \left\{ \lambda(t) + \frac{1}{2}\sigma^2 \right\} S(t)dt + \sigma S(t) dW(t), \tag{8}$$

where $W(t)$ is a one-dimensional Wiener process which is formally defined as an integration of the white noise $\gamma(t)$ with respect to time t. The Wiener process is a Gaussian process and has the following properties:

$$\Pr[W(0) = 0] = 1, \tag{9}$$

$$\mathrm{E}[W(t)] = 0, \tag{10}$$

$$\mathrm{E}[W(t)W(t')] = \mathrm{Min}[t,t'], \tag{11}$$

where $\Pr[\mathrm{A}]$ means the probability of event A and $\mathrm{E}[\mathrm{B}]$ represents the expected value of B in the time interval $(0,t]$.
By using Itô's formula (Arnold, 1974), we can obtain the solution of Eq.(7) under the initial condition $S(0) = v$ as follows (Yamada et al. 1994):

$$S(t) = v \cdot \exp\left(\int_0^t \lambda(s)ds + \sigma W(t)\right), \tag{12}$$

where v is the total number of faults detected for the previous software version. Using solution process $S(t)$ in Eq.(12), we can derive several software reliability measures.
Moreover, we define the intensity of inherent software failures, $\lambda(t)$, as follows:

$$\int_0^t \lambda(s)ds = \sum_{i=1}^{K} p_i\left(1-\exp[-\alpha_i t]\right) \quad \left(\sum_{i=1}^{K} p_i = 1\right), \tag{13}$$

where $\alpha_i (i = 1,2,...,K)$ is an acceleration parameter of the intensity of inherent software failures for the i-th component importance level, $p_i\left(\sum_{i=1}^{K} p_i = 1\right)$ the weight parameter for the i-th component importance level, and K the number of the applied component. Similarly, we can apply the following S-shaped growth curve to Eq. (12) depending on the trend of fault importance level:

$$\int_0^t \lambda(s)ds = \sum_{i=1}^{K} p_i\{1-(1+\alpha_i t)\exp[-\alpha_i t]\}. \tag{14}$$

2.3 Reliability assessment measures

2.3.1 Expected Number of Detected Faults and Their Variances

We consider the mean number of faults detected up to operational time t. The density function of $W(t)$ is given by

$$f(W(t)) = \frac{1}{\sqrt{2\pi t}}\exp\left\{-\frac{W(t)^2}{2t}\right\}, \tag{15}$$

Data collection on the current total number of detected faults is important to estimate the situation of the progress on the software operational procedures. Since it is a random variable in our model, its expected value and variance can be useful measures. We can calculate them from Eq. (12) as follows (Yamada et al. 1994):

$$\mathrm{E}\left[S(t)\right] = v \cdot \exp\left(\int_0^t \lambda(s) ds + \frac{\sigma^2}{2} t \right), \tag{16}$$

$$\begin{aligned} \mathrm{Var}\left[S(t)\right] &\equiv \mathrm{E}\left[\left\{ S(t) - \mathrm{E}\left[S(t)\right] \right\}^2 \right] \\ &= v^2 \cdot \exp\left(2 \int_0^t \lambda(s) ds + \sigma^2 t \right) \left\{ \exp\left(\sigma^2 t\right) - 1 \right\}, \end{aligned} \tag{17}$$

where $\mathrm{Var}\left[S(t)\right]$ is the variance of the number of faults detected up to time t.

2.3.2 Mean Time between Software Failures

The instantaneous mean time between software failures (which is denoted by MTBF_I) is useful to measure the property of the frequency of software failure-occurrence.
First, the instantaneous MTBF is approximately given by

$$MTBF_I(t) = \frac{1}{\mathrm{E}\left[\frac{dS(t)}{dt} \right]}. \tag{18}$$

Therefore, we have the following instantaneous MTBF:

$$MTBF_I(t) = \frac{1}{v\left(\lambda(t) + \frac{1}{2}\sigma^2 \right) \cdot \exp\left(\int_0^t \lambda(s) ds + \frac{\sigma^2}{2} t \right)}. \tag{19}$$

Also, the cumulative MTBF is approximately given by

$$MTBF_C(t) = \frac{t}{\mathrm{E}\left[S(t)\right]}. \tag{20}$$

Therefore, we have the following cumulative MTBF:

$$MTBF_C(t) = \frac{t}{v \cdot \exp\left(\int_0^t \lambda(s) ds + \frac{\sigma^2}{2} t \right)}. \tag{21}$$

2.3.3 Mean Time between Software Failures

Since a one-dimensional Wiener process is a Gaussian process, $\log S(t)$ is a Gaussian process. We can derive its expected value and variance as follows:

$$\mathrm{E}\left[\log S(t)\right] = \log v + \int_0^t \lambda(s)ds, \tag{22}$$

$$\mathrm{Var}\left[\log S(t)\right] = \sigma^2 t. \tag{23}$$

Therefore, we have the following probability for the event $\{\log S(t) \geq x\}$:

$$\Pr\left[\log S(t) \leq x\right] = \Phi\left(\frac{x - \log v - \int_0^t \lambda(s)ds}{\sigma\sqrt{t}}\right), \tag{24}$$

where $\Pr[\mathrm{A}]$ means the probability of event A and $\Phi(\cdot)$ of the standard normal distribution function can defined as follows:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right)dz. \tag{25}$$

Therefore, the transitional probability of $S(t)$ is given by the following equation:

$$\Pr\left[\log S(t) \leq y | S(0) = v\right] = \Phi\left(\frac{\log v + \log y + \int_0^t \lambda(s)ds}{\sigma\sqrt{t}}\right). \tag{26}$$

3. Software Reliability Assessment Procedures

The procedures of reliability assessment in our method for OSS are shown as follows:

1. We processes the data file in terms of the data in bug-tracking system of the specified OSS for reliability assessment.
2. Using the fault-detection count data obtained from bug-tracking system, we process the input data for neural network.
3. We estimate the weight parameters $p_i \ (i = 1, 2, \cdots, K)$ for each component by using the neural network.
4. Also, the unknown parameters σ and $\alpha_i \ (i = 1, 2, \cdots, K)$ included in our model are estimated by using the least-square method of Marquardt-Levenberg.
5. We show the expected total number of detected faults, the instantaneous fault-detection rate, and the cumulative MTBF as software reliability assessment measures, and the predicted relative error.

4. Portability Assessment

4.1 Prior information for BN

Applying SRGM's for prior information in case of using BN, we analyze software fault-detection count data based on an NHPP model. Considering stochastic characteristics associated with the fault-detection procedures in the testing phase, we treat $\{N(t),\ t \geq 0\}$ as a nonnegative counting process where a random variable $N(t)$ means the cumulative number of faults detected up to testing-time t. The fault-detection process $\{N(t),\ t \geq 0\}$ are described as follows:

$$\Pr\left[N(t)=n\right]=\frac{\left\{H(t)\right\}^{n}}{n!}\exp\left[-H(t)\right] \quad (n=0,1,2,\ldots) \tag{27}$$

In Eq. (27), $\Pr[\mathrm{A}]$ means the probability of event A, and $H(t)$ is called a mean value function which represents the expected cumulative number of faults detected in the testing-time interval $(0,\ t]$, where $h(t)$ is called an intensity function which represents the fault-detection rate per one fault. From Eq. (27), the fault-detection rate per one remaining fault, which characterizes the software reliability growth in the fault-detection phenomenon, is defined as

$$d(t) \equiv \frac{h(t)}{a-H(t)}, \tag{28}$$

where a is the expected number of the initial inherent faults. We can analyze software fault-detection count data by using an SRGM based on the NHPP, because the NHPP models have been discussed in many literatures since they can be easily applied in the software reliability assessment. The SRGM based on an NHPP is based on the following assumptions :

- A software fault-detection phenomenon can be described by an NHPP.
- Software faults detected during the testing-phase are corrected certainly and completely, i.e., no new faults are introduced into the software system during the debugging.

It is empirically known that the cumulative number of detected faults shows an exponential growth curve when a software system consisting of several software components are tested in the testing-phase. On the other hand, the cumulative number of faults describes an S-shaped growth curve when a newly developed software system is tested. Thus, the former case is described by the exponential SRGM based on an NHPP, and the latter case is described by the delayed S-shaped SRGM which is also based on an NHPP. We describe the structure of the mean value function defined in the following, because an NHPP model is characterized by its mean value function. The mean value function of the exponential SRGM, $H_e(t)$, is characterized by the following function:

$$H_e(t)=a\left(1-e^{-bt}\right) \qquad (a>0,\ b>0), \tag{29}$$

where $H_e(t)$ represents the expected cumulative number of faults detected up to the module testing time t $(t \geq 0)$. In Eq. (29), a is the expected number of initial inherent faults, and b the software failure rate per inherent fault. In addition, the intensity function of the exponential SRGM is given as follows:

$$h_e(t) = abe^{-bt}, \tag{30}$$

where $h_e(t)$ represents the instantaneous fault-detection rate at the module testing time t $(t \geq 0)$.

Similarly, the mean value function of the delayed S-shaped SRGM, $H_s(t)$, is represented as :

$$H_s(t) = a\left[1-(1+bt)e^{-bt}\right] \qquad (a>0,\ b>0), \tag{31}$$

where $H_s(t)$ represents the expected cumulative number of faults detected up to the module testing time t $(t \geq 0)$. In Eq. (6), a is the expected number of initial inherent faults, and b the software failure rate per inherent fault. In addition, the intensity function of the delayed S-shaped SRGM is given as follows :

$$h_s(t) = ab^2te^{-bt} \tag{32}$$

where $h_s(t)$ represents the instantaneous fault-detection rate at the module testing time t $(t \geq 0)$.

In this chapter, we use Mean Squared Errors (MSE) discussed in 5.3 in order to select a better SRGM, Exponential SRGM or Delayed S-shaped SRGM for each software component.

4.2 Reliability assessment for OSS porting

We can estimate the probability of the phenomenon Z based on the following phenomena X and Y:

X: The fault is detected at the component X.
Y: The fault is detected at the component Y.
Z: The fault is detected at the Kernel component.

BN for above mentioned phenomena is shown in Fig. 1.

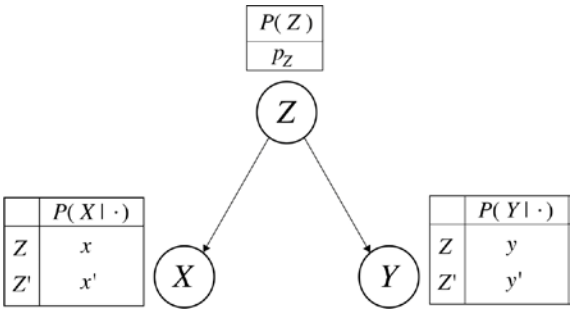


Fig. 1. The failure-occurrence probability model based on BN.

Fig. 1 means that the fault is detected at the component X or component Y as a result of the occurrence of phenomenon Z. Therefore, the failure probability of Kernel component is given by the following equation based on the Bayesian theory:

$$p_Z = \frac{xyp'_Z b_x b_y}{xyp'_Z + x'y'\overline{p}'_Z} + \frac{\overline{x}yp'_Z \overline{b}_x b_y}{\overline{x}yp'_Z + \overline{x}'y'\overline{p}'_Z} + \frac{x\overline{y}p'_Z b_x \overline{b}_y}{x\overline{y}p'_Z + x'\overline{y}'\overline{p}'_Z} + \frac{\overline{x}\overline{y}p'_Z \overline{b}_x \overline{b}_y}{\overline{x}\overline{y}p'_Z + \overline{x}'\overline{y}'\overline{p}'_Z}, \tag{33}$$

where x is the probability of X under the occurrence of phenomenon Z. Similarly, y is the probability of Y under the occurrence of phenomenon Z. The probability of X and Y are given by the b_x and b_y previously. Also, $\overline{p}$ means the $1-p$. p'_Z is the prior probability of Z, p'_Z is updated by using $p'_Z = p_Z$. The prior information b_x and b_y are given by the intensity function of NHPP model and SDE model applied for each component of OSS. We can estimate the portability of embedded software based on OSS by using Eq. (33).

5. Numerical Examples

5.1 Embedded OSS

We focus on the BusyBox (Erik Andersen) which is one of the embedded open source software developed under an open source project. The BusyBox combines tiny versions of many common UNIX utilities into a single small executable. It provides replacements for most of the utilities you usually find in GNU fileutils, shellutils, etc.

The fault-detection count data used in this chapter are collected in the bug tracking system on the website of BUSYBOX in August 2008.

5.2 Reliability assessment considering component level

Estimating the weight parameters based on the number of source lines of code for each component, we analyze the actual data for the case of $v = 14,\ p_1 = 0.28252,\ p_2 = 0.07213,\ \hat{\alpha}_3 = 0.64531$.

In case of the exponential intensity function of inherent software failures in Eq. (30), the following model parameters have been estimated:

$$\hat{\alpha}_1 = 0.324667,\ \hat{\alpha}_2 = 0.116271,\ \hat{\alpha}_3 = 0.121832,\ \hat{\sigma} = 0.153831.$$

In case of the S-shaped intensity function of inherent software failures in Eq. (32), the following model parameters have been estimated:

$$\hat{\alpha}_1 = 0.141554,\ \hat{\alpha}_2 = 0.168274,\ \hat{\alpha}_3 = 0.567777,\ \hat{\sigma} = 0.153845.$$

The estimated expected number of detected faults, $\hat{\mathrm{E}}[S(t)]$, in case of the exponential intensity function of inherent software failures, is shown in Fig. 2. Also, Fig. 3 shows the

estimated variance of the number of faults. From Fig. 3, we find that the variance of the number of detected faults grows as the time elapses after the release.

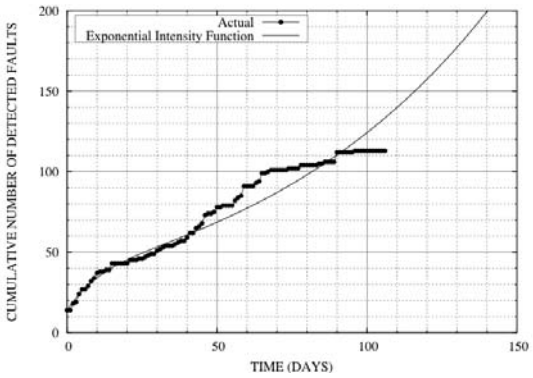


Fig. 2. The estimated expected cumulative number of detected faults.

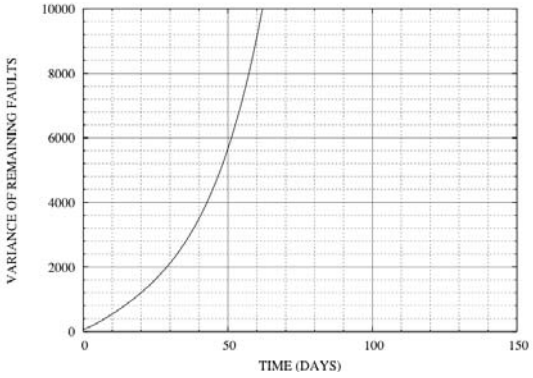


Fig. 3. The estimated variance of the number of detected faults.

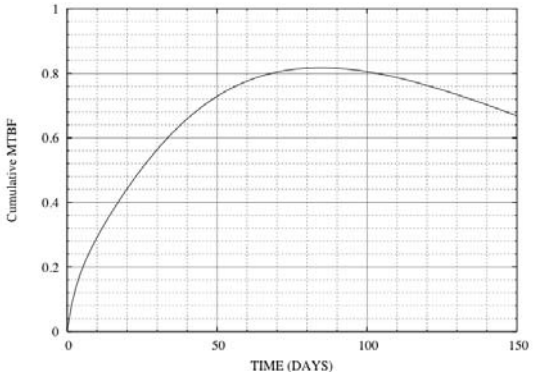


Fig. 4. The estimated $MTBF_C$.

Moreover, the estimated $MTBF_C$ is also plotted in Fig. 4. Fig. 4 shows that the MTBF increase as the operational procedures go on.
Furthermore, Fig. 5 shows the estimated transitional probability of Eq. (26) in case of the exponential intensity function of inherent software failures.

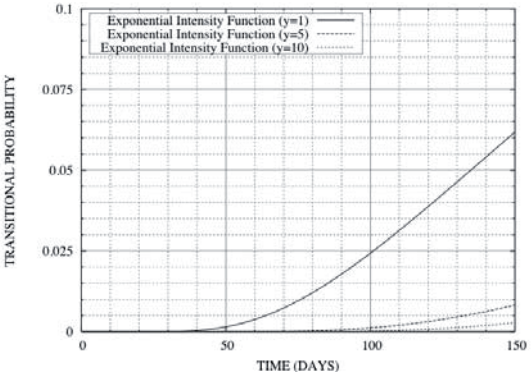


Fig. 5. The estimated transitional probability of $S(t)$.

5.3 Assessment Measures for OSS porting

We show numerical example in terms of OSS porting by using the fault-detection count data registered in the bug tracking system in actual open source project. In this chapter, we apply the fault data of BusyBox that is developed and used as Embedded OSS.
We focus on the buildroot and the uClibc which is one of the components in the Busybox. We apply the mean value functions of exponential SRGM and delayed S-shaped SRGM for each components. We show the result of parameter estimation and Mean Square Error (MSE) in Tables 1-3. Moreover, we show the estimation of expected values of the cumulative number of detected faults in Figs. 6-8. We define the error function in Eq. (34) by the following equation :

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \hat{y}_n\right)^2, \tag{34}$$

where y_k are the actual measurement values, and $\hat{y}_k$ are the predictive values. The MSE indicates that the selected model fits better to the observed data as MSE becomes small.

	a	b	MSE
Exponential SRGM	155.82	0.0026	5.3916
Delayed S-shaped SRGM	43.966	0.0305	2.2497

Table 1. The results of estimated unknown parameter and value of MSE for each SRGM in BusyBox.

	a	b	MSE
Exponential SRGM	159.19	0.0039	4.3498
Delayed S-shaped SRGM	60.7672	0.0318	5.7175

Table 2. The results of estimated unknown parameter and value of MSE for each SRGM in buildroot.

	a	b	MSE
Exponential SRGM	28.270	0.0056	0.8490
Delayed S-shaped SRGM	13.758	0.0351	0.3729

Table 3. The results of estimated unknown parameter and value of MSE for each SRGM in uClibc.

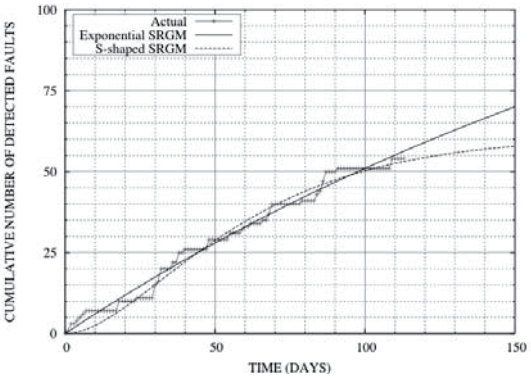


Fig. 6. The estimated expected cumulative number of detected faults for buildroot.

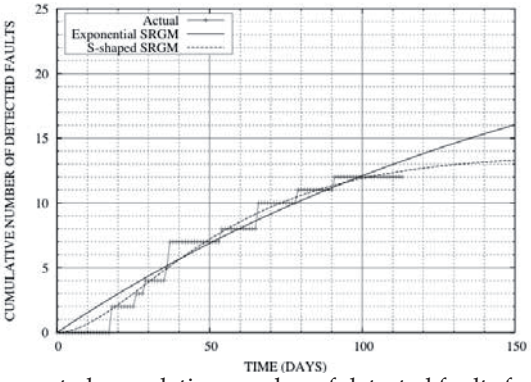


Fig. 7. The estimated expected cumulative number of detected faults for uClibc.

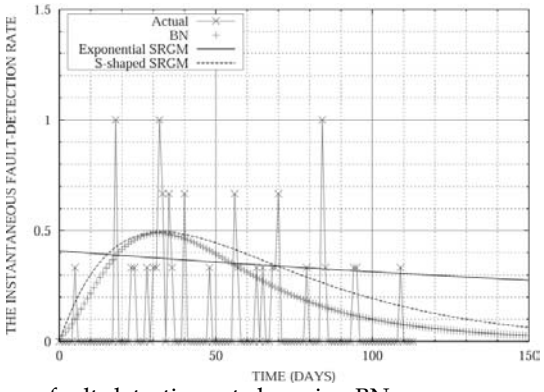


Fig. 8. The instantaneous fault-detection rate by using BN.

We can apply the exponential SRGM for buildroot from Fig. 6 and Table 2. On the other hand, we can apply the delayed S-shaped SRGM for uClibc from Fig. 7 and Table 3. Fig. 8 is shown the instantaneous fault-detection rate by using BN. From Fig. 8, we find that the fault-detection rate can be used as the portability assessment measure for the embedded OSS by using BN.

6. Conclusion

In this chapter, we have discussed the methods of reliability analysis for an embedded OSS developed under on an open source project. Especially, we have proposed a stochastic differential equation model in order to consider the active state of the open source project, where we have assumed that the software failure intensity depends on the time, and the software fault-reporting phenomena on the bug tracking system keep an irregular state. Moreover, we have derived several assessment measures in terms of imperfect debugging of an entire system under such open source development paradigm. Especially, we have applied the exponential and S-shaped intensity functions of inherent software failures for the i-th component importance level to the interaction among components by using an acceleration parameters. Furthermore, we have shown the transitional probability of $S(t)$ in Eq. (26).

In case of considering the effect of debugging process on an entire system on software reliability assessment for open source projects, it is necessary to grasp the deeply-intertwined factors. In this chapter, we have shown that our method can describe such deeply-intertwined factors. Especially, we have applied BN technique in order to consider the effect of each software component on the reliability of an entire system under such open source development paradigm. By using the BN, we have proposed the method of reliability assessment incorporating the interaction among software components. Moreover, we have proposed the fault-detection rate based on BN used as the portability assessment measure for the embedded OSS.

Finally, we have focused on an embedded OSS developed under open source projects. New distributed development paradigm typified by such open source project will evolve at a

rapid pace in the future. Our method is useful as the method of reliability assessment incorporating the importance of each component for an entire system.

7. Acknowledgments

This work was supported in part by the Grant-in-Aid for Young Scientists (B), Grant No. 21700044 from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

8. References

Arnold, L. (1974) *Stochastic Differential Equations-Theory and Applications*, John Wiley & Sons, New York

Erik Andersen, BUSYBOX. [Online]. Available: http://www.busybox.net/E-Soft Inc., Internet Research Reports. [Online].
Available: http://www.securityspace.com/s_survey/data/

Kuk, G. (2006) Strategic interaction and knowledge sharing in the KDE developer mailing list, *Informs J. Management Science*, Vol. 52, No. 7, pp. 1031-1042

Karunanithi, N. & Malaiya, Y. K. (1996) Neural networks for software reliability engineering, *Handbook of Software Reliability Engineering* M. R. Lyu (ed.), pp. 699-728, McGraw-Hill, New York

Li, P.; Shaw, M.; Herbsleb, J.; Ray, B. & Santhanam, P. (2004). Empirical evaluation of defect projection models for widely-deployed production software systems, *Proceedings of the 12th International Symposium on the Foundations of Software Engineering (FSE-12)*, pp. 263-272

Lippmann, R. P. (1987) An introduction to computing with neural networks, *IEEE Trans. ASSP*, Vol. 4, No. 2, pp. 4-22

MacCormack, A.; Rusnak, J. & Baldwin, C.Y. (2006) Exploring the structure of complex software designs: an empirical study of open source and proprietary code, *Informs J. Management Science*, Vol. 52, No. 7, pp. 1015-1030

Misra, P. N. (1983) Software reliability analysis, *IBM Systems J.*, Vol. 22, No. 3, pp. 262-270

Musa, J. D.; Iannino, A. & Okumoto, K. (1987) *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York

Takahashi, M. (1998) *The Method of Effort Estimation under Client/Server System Development: Models and Applications (in Japanese)*, Soft Research Center, Tokyo

Tamura, Y. & Yamada, S. (2007) Software reliability assessment and optimal version-upgrade problem for open source software, *Proceedings of the 2007 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1333-1338 Montreal, Canada

Tamura, Y. & Yamada, S. (2008) A method of reliability assessment based on deterministic chaos theory for an open source software, *Proceedings of the Second IEEE International Conference on Secure System Integration and Reliability Improvement*, pp. 60-66, Yokohama, Japan

Tamura, Y. & Yamada, S. (2008) Comparison of software reliability assessment methods based on deterministic chaos theory for an open source software, *Proceedings of the 2008 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3606-3611, Suntec, Singapore

Umar, A. (1993) *Distributed Computing and Client-Server Systems*, Prentice Hall, Englewood Cliffs, New Jersey

Vaughn, L. T. (1994) *Client/Server System Design and Implementation*, McGraw-Hill, New York

Yamada, S. & Osaki, S. (1985) Software reliability growth modeling: Models and applications, *IEEE Trans. Software Engineering*, Vol. SE-11, No. 12, pp. 1431-1437

Yamada, S. (1991) Software quality/reliability measurement and assessment: Software reliability growth models and data analysis, *J. information Processing*, Vol. 14, No. 3, pp. 254-266

Yamada, S. (1994). *Software Reliability Models: Fundamentals and Applications (in Japanese)*, JUSE Press, Tokyo

Yamada, S.; Kimura, M.; Tanaka, H. & Osaki, S. (1994) Software reliability measurement and assessment with stochastic differential equations, *IEICE Trans. Fundamentals*, Vol. E77-A, No. 1, pp. 109-116

Zhoum, Y. & Davis, J. (2005) Open source software reliability model: an empirical approach, *Proceedings of the Workshop on Open Source Software Engineering (WOSSE)*, pp.67-72, Vol. 30, No. 4

Architecture and Design Methodology of Self-Optimizing Mechatronic Systems

Prof. Dr.-Ing. Jürgen Gausemeier
Dipl.-Wirt.-Ing. Sascha Kahl
Heinz Nixdorf Institute
Fürstenallee 11, D-33102 Paderborn

Abstract
The conceivable development of information and communication technology will enable mechatronic systems with inherent partial intelligence. We refer to this by using the term "self-optimization". Self-Optimizing systems react autonomously and flexibly on changing operation conditions. They are able to learn and optimize their behavior at runtime. The development of mechatronic and especially self-optimizing systems is still a challenge. A significant milestone within the development is the principle solution. It determines the basic structure as well as the operation mode of the system and is the result of the conceptual design. Additionally it is the basis for the concretization of the system which involves experts from several domains, such as mechanics, electrical engineering/electronics, control engineering and software engineering. This contribution presents a new specification technique for the conceptual design of mechatronic and self-optimizing systems. It also uses the railway technology as a complex example, to demonstrate how to use this specification technique and in which way it profits for the development of future mechanical engineering systems.
Keywords
Design Methodology, Mechatronics, Self-Optimization, Principle Solution, Conceptual Design, Domain-Spanning Specification

1. Introduction

The products of mechanical engineering and related industrial sectors, such as the automobile industry, are increasingly based on the close interaction of mechanics, electronics and software engineering, which is aptly expressed by the term mechatronics. The conceivable development of communication and information technology opens up more and more fascinating perspectives, which move far beyond current standards of mechatronics: mechatronic systems having an inherent partial intelligence. We call these systems "self-optimizing systems". Self-Optimization of a technical system is the endogenous adaptation of the systems' objectives as a reaction to changing influences and the resulting autonomous adjustment of parameters or structure and consequently of the systems' behavior [ADG+08].

According to this self-optimizing systems have the ability to react autonomously and flexibly on changing operation conditions. Thereby self-optimization goes far beyond conventional control and adaptation strategies.
To develop mechatronic and especially self-optimizing systems, still is a challenge. The established design methodologies of the conventional engineering domains are no longer adequate. This particularly applies to the early design phase "conceptual design" which results in the so-called "principle solution". The principle solution represents a significant milestone because it determines the basic structure and the operation mode of the systems and, subsequently, it is the basis for further concretization. This need for action was the starting point for the collaborative research centre (CRC) 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" at the University of Paderborn funded by the German Research Foundation (DFG).
This contribution presents the essential results of the Collaborative Research Center 614. It first explains the paradigm of self-optimization and the key aspects of such systems. Afterwards it describes in detail the three actions of self-optimization, the so called Self-Optimization Process. For the realization of complex, mechatronic systems with inherent partial intelligence an adequate concept of structure as well as architecture for the information processing is needed. Hence the new concept of the Operator-Controller-Module (OCM) has been developed. This concept is also presented in detail. A new and powerful paradigm, such as self-optimization, naturally calls for new development methods as well as development tools. Therefore a new design methodology for self-optimizing and thus for mechatronic systems is introduced. It divides the development process into two main phases – the "conceptual design" and the "concretization". The main emphasis is on a holistic integrative specification of the principle solution. Therefore a new domain-spanning specification technique is presented. Within the "conceptual design" the specification of the principle solution forms the basis for all the experts' communication and cooperation. It will be described in which way the development activities of the subsequent "concretization", that take place in parallel, are going to be structured, coordinated and how the consistency of these activities is ensured on the basis of the principle solution.
All the works by the CRC 614 use the "Neue Bahntechnik Paderborn/RailCab" as a demonstrator. All examples throughout this contribution refer to that project. RailCab is an innovative railway system which is realized on a test track at a scale of 1:2.5. Autonomous vehicles (RailCabs) that supply transport for both passengers and cargo, establish the core of the system (figure 1). They drive on demand and not by schedule. The RailCabs act in a pro-active way, e.g. in order to reduce the required energy by forming convoys. The actuation is realized by a contact-free dual-feed electromagnetic linear drive [ZS05], [ZBS+05]. The stator of the linear drive is situated between the track and the rotor within the shuttle. The three-phase winding in the stator forms a magnetic field which moves asynchronous along the tracks and propels the vehicle with it. By the magnetic dynamic effects between the stator and rotor magnetic fields´, the vehicle is accelerated and also slowed down. The dual feed allows variable adjustment of the vehicle's magnetic field. Consequently, several RailCabs can be operated on the same stator section with different velocities. The linear drive allows power to be supplied to the vehicle without overhead lines or contact rails. The supporting and guiding of the shuttle take place by using wheel/track contact that allows the usage of already existing railway tracks. With an active tracking module, based on an independent axle chassis with loose wheels, the choice of direction by passing over a switch can now take

place vehicle-sided. In that case, the switches work in a passive way, in contrast to the conventional rail. An active spring technology with an additional tilt technology results in a high travelling comfort. The RailCab's basic technology is placed in the plain-built undercarriage on which the chassis for passengers or cargo will be set upon.

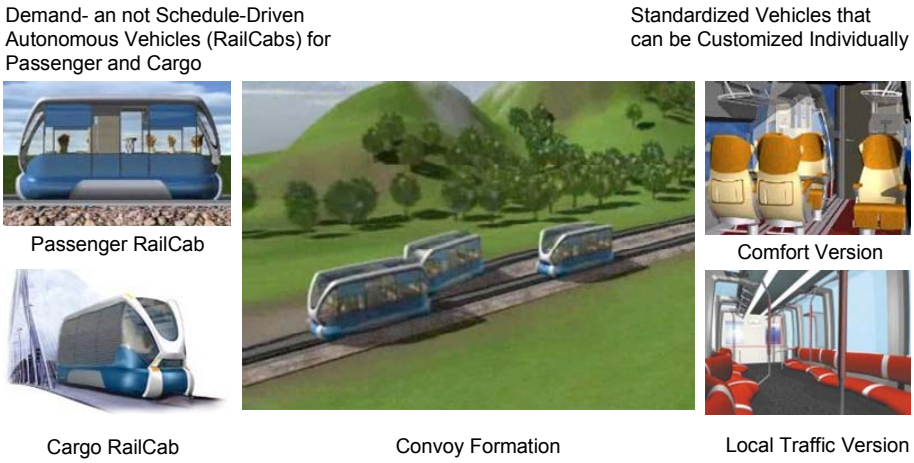


Fig. 1. RailCabs of the project „Neue Bahntechnik Paderborn/RailCab"

2. Self-Optimization

All future intelligent systems of mechanical engineering, regarded in this contribution, rely on mechatronics. The CRC 614 took up the hierarchical structure of complex mechatronic systems suggested by LÜCKEL and added the aspect of self-optimization (figure 2) [LHL01]. The basis consists of so-called mechatronic function modules (MFM) that comprise a mechanical basic structure, sensors, actors and a local information processing, which contains the controller. MFMs that are connected by information technology and/or mechanical elements result in autonomous mechatronic systems (AMS). They also feature information processing. Within this information processing, superior tasks are being realized, such as monitoring, fault diagnosis and maintenance decisions. Additionally targets for the local information processing of the MFM are generated. AMS form the so-called networked mechatronic systems (NMS). NMS are produced just by connecting the AMS parts via information processing. Similar to the AMS, the information processing of the NMS is realizing superior tasks. If the terms are to be transferred in the vehicle engineering, the spring and tilt module would be considered to be a MFM, the RailCab itself with an active chassis an AMS and a convoy a NMS.

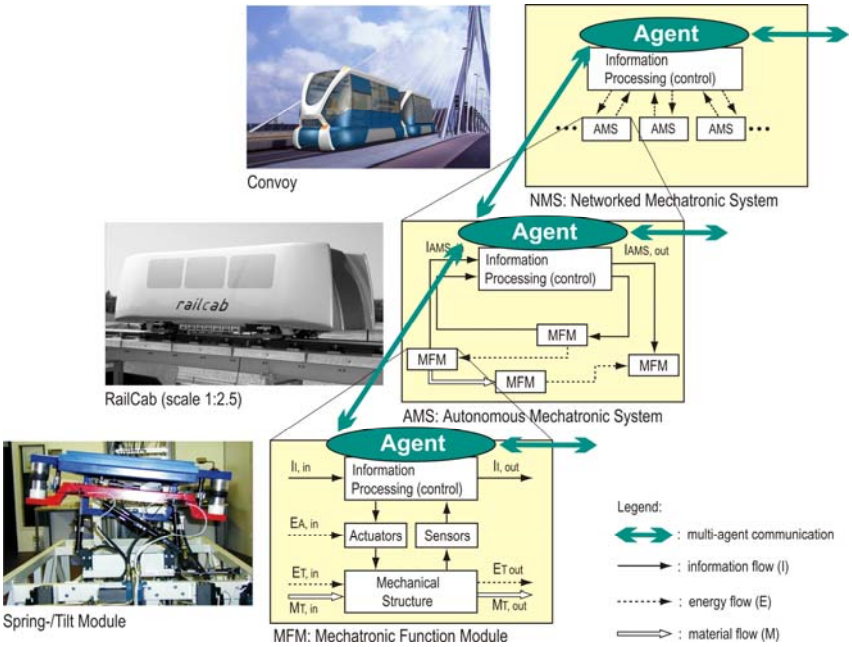


Fig. 2. Structure of a complex mechatronic system with inherent partial intelligence

On every level of this structure it is possible, to add the functionality of self-optimization to the controller. By this, the regarded system's elements (MFM, AMS, NMS) gain inherent partial intelligence. The behavior of the whole system is formed by the communication and cooperation of the intelligent system's elements. From an information processing point of view we consider these distributed systems to be multi-agent-systems.

Against this backdrop, we understand a system's self-optimization as endogenous adaptation on changing operating conditions, as well as a resulting objective-oriented adaptation of the parameters and, if necessary, of the structure and therefore the behavior of the system [ADG+08]. Thus self-optimization enables systems that have inherent "intelligence". They have the ability to react autonomously and flexibly on changing operating conditions.

The key aspects and the operation mode of a self-optimizing system are depicted in figure 3. Using the influences as a basis, the self-optimizing system determines the internal objectives that have to be pursued actively. These internal objectives are based on external ones, whereas those are set from the outside, e.g. by the user or other systems, and also on inherent objectives that reflect the design purpose of the system. Inherent objectives of a driving module can be for example: saving of the driving functions and a high efficiency. If we below talk about objectives, we refer to the internal ones, because those are part of the optimization. Low energy demand, high travelling comfort and low noise emission belong to internal objectives of a RailCab. The adaptation of objectives means, for instance, that the relative weighting of the objectives is modified, new objectives are added or existing objectives are discarded and no longer pursued.

Thus, the adaptation of the objectives leads to an adaptation of the system's behavior. The behavior's adaptation is achieved by an adaptation of the parameters and, if necessary, of

the structure. An adaptation of the parameters means an adaptation of the system's parameters, e.g. the adaptation of a controller parameter.
Adapting the structure, concerns the arrangement and relations of the system's elements. We differentiate between reconfiguration and compositional adaptation. Reconfiguration is the change of the relations of a fixed quantity of elements. Compositional adaptation means the integration of new elements in the already existing structure or the subtraction of elements from the structure. Self-Optimization takes place as a process that consists of the three following actions, called the **Self-Optimization Process**:
1. Analyzing the current situation: The regarded current situation includes the current state of the system as well as all observations of the environment that have been carried out. Observations can also be made indirectly by communication with other systems. Furthermore, a system's state contains possible previous observations that are saved. One basic aspect of this first step is the analysis of the fulfillment of the objectives.
2. Determining the system's objectives: The system's objectives can be extracted from choice, adjustment and generation. By choice we understand the selection of one alternative out of predetermined, discrete, finite quantity of possible objectives; whereas the adjustment of objectives means the gradual modification of existing objectives respectively of their relative weighting. We talk about generation, if new objectives are being created that are independent from the existing ones.

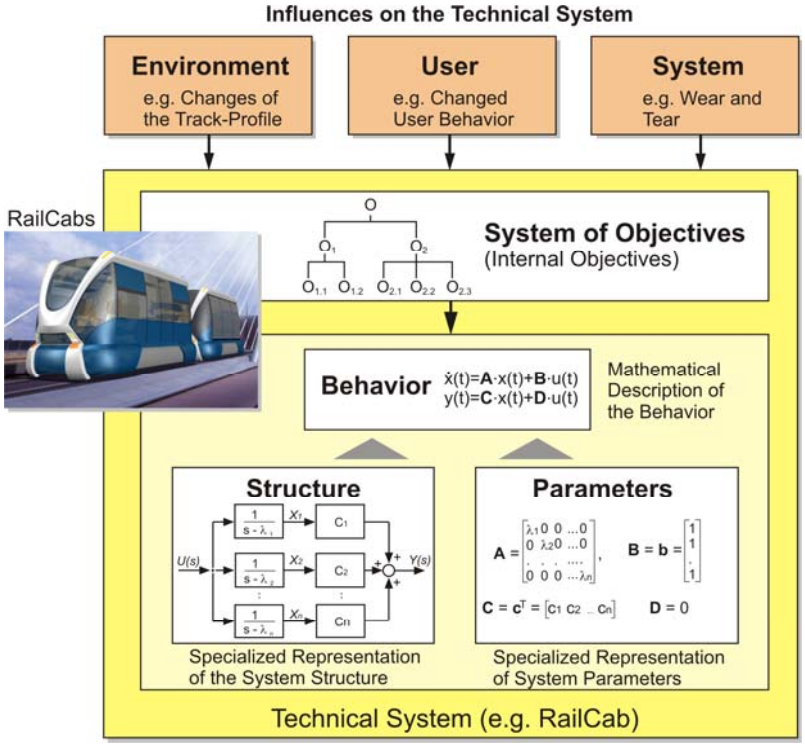


Fig. 3. Aspects of a self-optimizing system – influences on the system result in an adaptation of the objectives and an according adaptation of the system's behaviour

3. Adapting the system's behavior: The changed system of objectives demands an adaptation of the behavior of the system. As mentioned before this can be realized by adapting the parameters and, if required, by adapting the structure of the system. This action finally closes the loop of the self-optimization by adapting the system's behavior.
The self-optimizing process leads, according to changing influences, to a new state. Thus a state transition takes place. The Self-Optimizing Process describes the system's adaptation behavior. This can occur on every hierarchy level of a self-optimizing system shown in figure 2. The realization of mechatronic systems with inherent partial intelligence requires an adequate concept of structure as well as an architecture for the information processing. To make this possible, the new concept of the **Operator-Controller-Module (OCM)** has been developed: [ADG+08]. From an information processing point of view, it corresponds to an agent. Figure 4 shows its architecture. As a result, an OCM can be structured into three levels (Controller, Reflective Operator and Cognitive Operator) which will be examined in detail below.

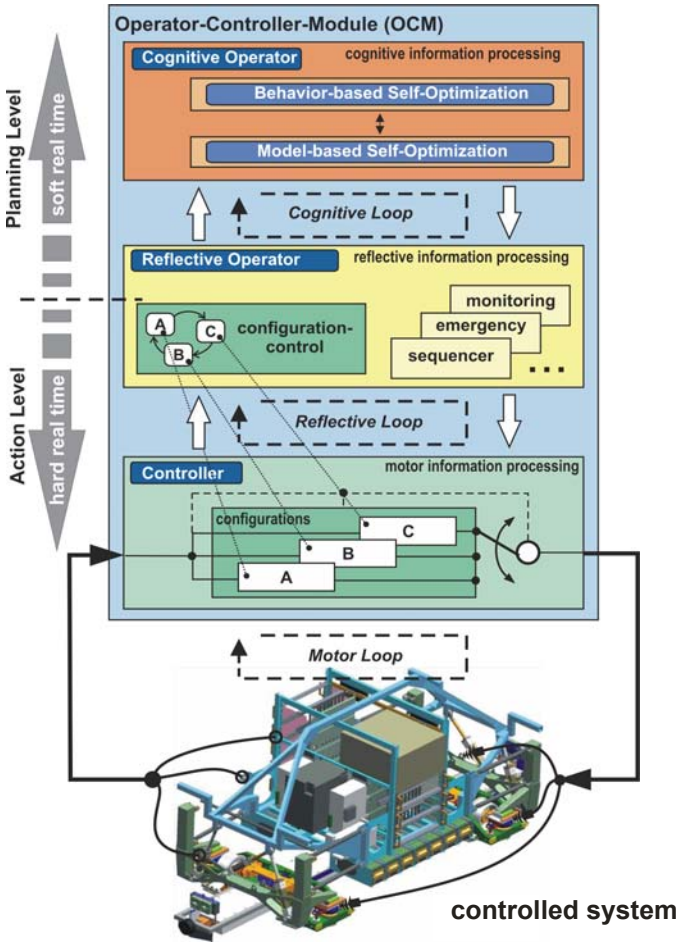


Fig. 4. Architecture of the Operator-Controller-Module (OCM)

- The **Controller** level stands for the control loop with direct access to the technical system. The software at this level operates continuously under hard real-time conditions. The controller itself can be made up of a number of controller configurations with the possibility of switching between them. The changeover takes one step; necessary cross-fading mechanisms and the like are summarized, in turn, into one controlling element.
- The **Reflective Operator** supervises and regulates the controller. It does not access directly the actuators of the system but it modifies the controller by initiating parameter changes or changes of the structure. If changes of the structure do appear (e.g. as in reconfigurations), not just the controllers will be replaced but also corresponding control and signal flows will be switched within the controller itself. Combinations that consist of controllers, circuit elements and corresponding control or signal flows are described as controller-configurations. Figure 4 shows possible controller-configurations, exemplified by the blocks A, B and C. The controlling of the configurations, realized by a state machine, defines which state of the system uses which kind of configuration. It also determines under which circumstances it is necessary to switch between the configurations. The reflecting operator mostly works event-oriented. The close connection with the controller requires a mode of operation in so-called hard real-time. The reflecting operator offers an interface (working as a conjunctional element to the cognitive level of the OCM) between the elements operating not in real-time or soft real-time and the controller. It filters the arriving signals and takes them to the levels underneath. Moreover, the reflecting operator is responsible for the real-time communication between the OCM, which together constitute a composed self-optimizing system.
- The **Cognitive Operator** is the highest level of the OCM-architecture. Here the system uses knowledge on itself as well as on its environment in order to improve its own behavior by using varied methods (such as learning methods and model-based optimization). The main emphasis is put on the cognitive abilities for carrying out of the self-optimizing process. Model-based processes allow a predictable optimization that is, to a large extent, decoupled from the underlying levels while the system is in operation.

Based on the OCM-architecture, the actions of the Self-Optimizing Process (1. analyzing the current situation, 2. determining the system's objectives and 3. adapting the system's behavior) can be realized in various ways. When the self-optimizing adaptation needs to fulfill real-time requirements all three actions are carried out in the reflective operator. Systems that do not have to run the self-optimization in real time can use more elaborate methods, which are settled within the cognitive operator. In that case, the behavior's adaptation is carried out indirectly, relayed by the reflective operator, which needs to synchronize the instructions of the behavior's adaptation with the controller's real-time course. In addition, there might occur mixtures within one single OCM. There are also hybrid forms that occur within a single OCM, when the two described forms of self-optimization take place simultaneously and asynchronously.

3. Challenges during the development of self-optimizing systems

A new and powerful paradigm, such as self-optimization, naturally calls for new development methods as well as development tools. Because of the high complexity and the participation of a multitude of different engineering domains the development of self-optimizing systems is still a challenge.

By the activities of the Heinz Nixdorf Institute of the University of Paderborn a practical guideline for the development of mechatronic systems was established – the VDI-guideline 2206 „Design methodology for mechatronic systems“ (figure 5). The guideline is based on the so called V model of the domain software engineering.

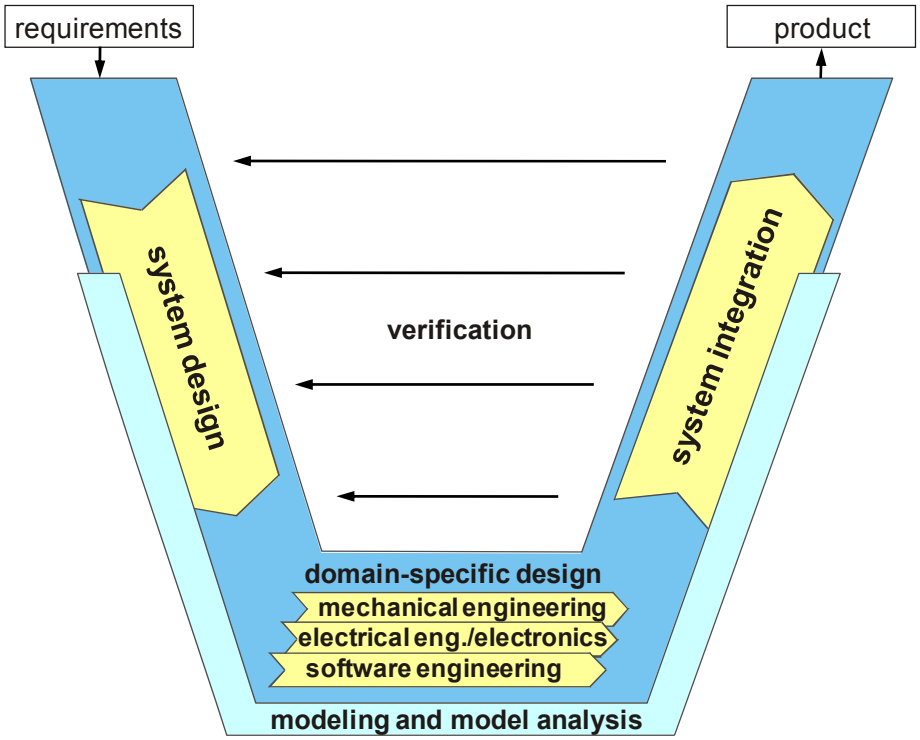


Fig. 5. V model of the VDI-guideline 2206 „Design methodology for mechatronic systems” [VDI04]

Starting point for the system design are the requirements. During the system design the basic structure and the main physical and logical operating characteristics of the system are described in a domain-spanning product concept (synonymous: principle solution). The principle solution forms the basis for the subsequent domain-specific concretization. The term “concretization” describes thereby the domain-specific design of a technical system. During the concretization the participating domains specify the system by using domain-specific procedure models, methodologies and tools. Afterwards the integration of the results from the individual domains into an overall system allows to investigate the characteristics of the system. During the different development steps the reliability and safety of the characteristics is analyzed by computer models. The result of one cycle of the V model is a product of a specific stage of maturity (e.g. functional model, prototype, pre-production model and repetition part). For the development of a product of a high stage of maturity a number of cycles are required. The presented model is a first step on the way to a holistic design methodology for mechatronic systems.

The holistic domain-spanning system design – thus the left bough of the V model – still bears the major challenge in the development of mechatronic systems. There is a gap between the normally used list of requirements, which is more or less a rough specification of the total system and, hence, leaves much space for interpretation, and well-established specification techniques of the involved domains used within the domain-specific concretization. This gap is the reason, why the engineers run in deep trouble when they have to integrate their results in the system integration in the right bough of the V model. This applies to mechatronic as well as to self-optimizing systems. It is the question, how established design methodologies can be extended to face these challenge. For the system design we became aware, that the basic structure of the conceptual design in classical mechanical engineering design methodology (formulation of the requirements, definition of the functions and searching for active principles for the realization of the tasks [PBF+07]) is also valid for mechatronic and self-optimizing systems. By looking into this more deeply, it became clear that an extension of the design methodology was urgently necessary. This appears, for instance, in the use of solution patterns as well as in the necessity of modeling the environment, application scenarios and the system of objectives. Therewith a holistic approach for the conceptual design of self-optimizing systems, comprising of an integrative specification of the description of the principle solution and a holistic procedure model, is the main need for action on the way to a design methodology for self-optimizing systems (compare figure 6). Both elements are presented in the following chapters.

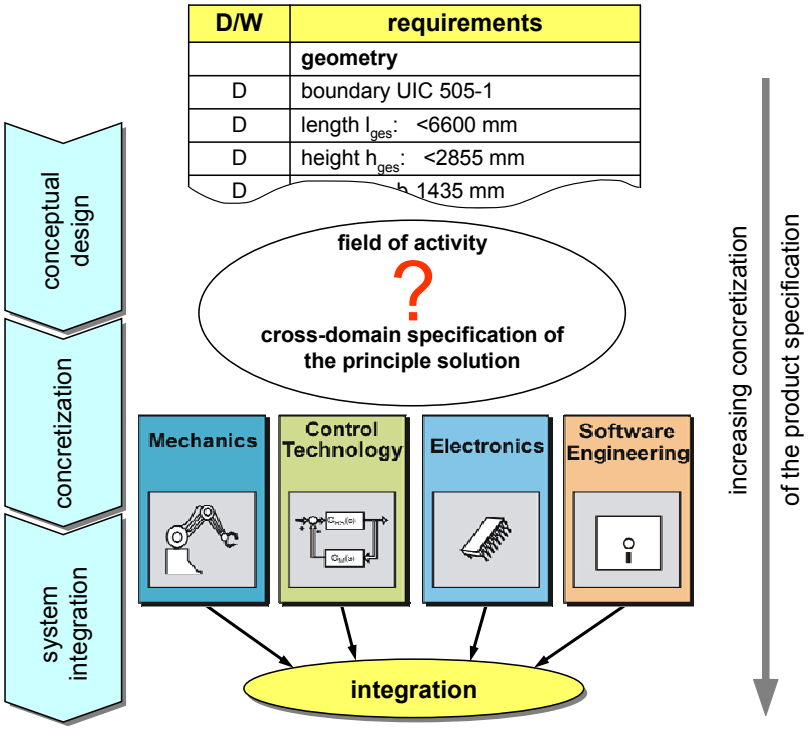


Fig. 6. Central challenge: a new specification technique for the description of the principle solution of a mechatronic respectively self-optimizing system

4. Specification of the principle solution

In the following, we present a specification technique for the description of the principle solution of a self-optimizing system. It is also applicable for mechatronic systems. The specification technique is based on the research of FRANK, GAUSEMEIER and KALLMEYER [GFD+08], [GEK01]. It became clear that a comprehensive description of the principle solution of a highly complex system needs to be divided into aspects. Those aspects are, according to figure 7: requirements, environment, system of objectives, application scenarios, functions, active structure, shape and behavior. The behavior consists of a whole group because there are different kinds of behavior, e.g. the logic behavior, the dynamic behavior of multi-body systems, the cooperative behavior of system components etc. The mentioned aspects are represented by partial models. The principle solution consists of a coherent system of partial models because the aspects are in relationship with each other and ought to form a coherent system. It is necessary to work alternately on the aspects and the according partial models. Nevertheles there is a certain order.

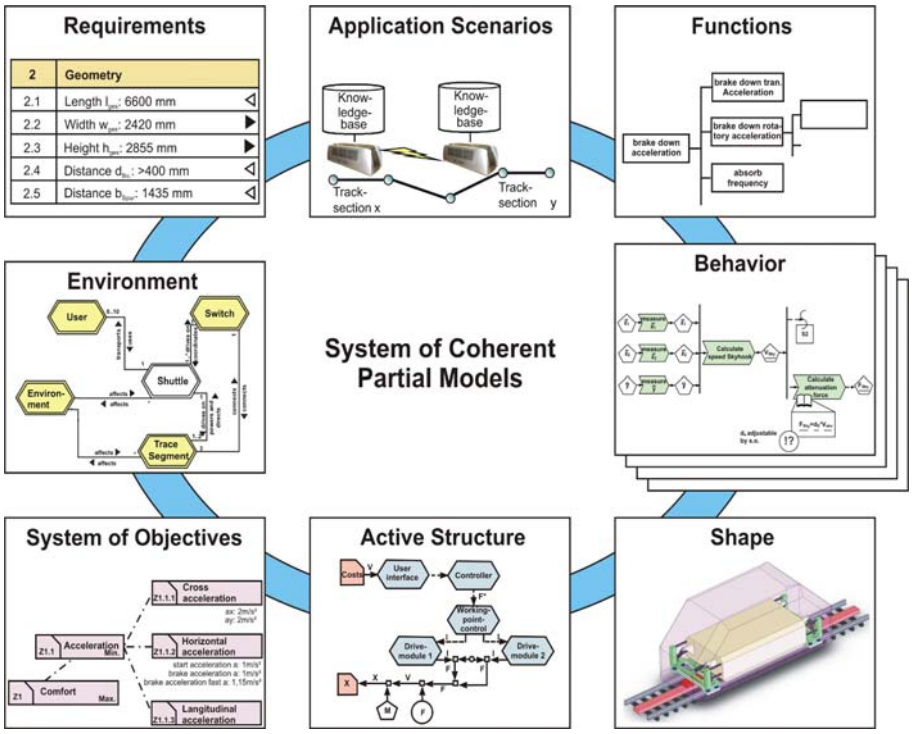


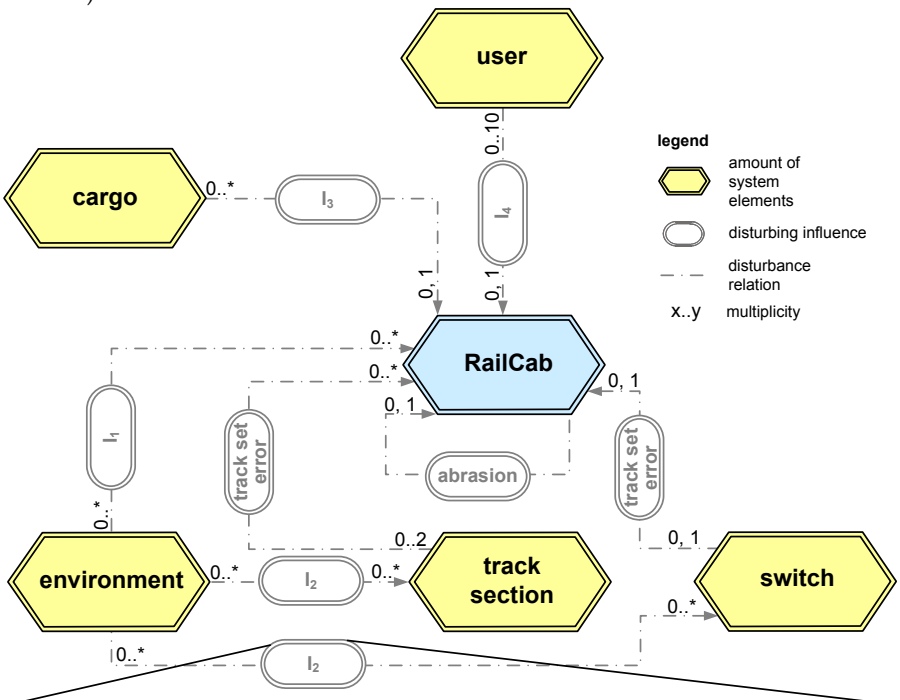
Fig. 7. Partial models for the domain-spanning description of the principle solution of self-optimizing systems

The description of the environment, the application scenarios and requirement serve as the starting point. They are usually followed by the system of objectives, the function hierarchy and the active structure. The active structure represents the core of the principle solution in

conventional mechanical engineering. The modeling of states and the state transitions as well as the impacts on the active structure play a decisive role in the specification of a self-optimizing system. This kind of modeling takes place within the group of behavior models. An example is given in subchapter 3.3. The following subchapters explain the partial models, the relationships between the partial models and the specific characteristic of the specification of self-optimizing systems.

4.1 Partial models

Environment: This model describes the environment of the system that has to be developed and its embedding into the environment. Relevant spheres of influence (such as weather, mechanical load, superior systems) and influences (such as thermal radiation, wind energy, information) will be identified.



No.	influence (denotation and short description)	attribute	characteristic	range of tolerance	frequency of appearance	kind of influence: supporting, disturbing, neutral
1	terrain	inclination	> 5 %	0,30 %	medium	disturbing
2	wind	velocity	> 50 km/h	2 km/h	often	disturbing
3	rain	liters per minute	1 l/min	0,1 l/min	medium	disturbing
4	snow	depth of snow	20 cm	4 cm	rare	disturbing
5	ice	adhesion	<< 1	3	rare	disturbing
6	flood	level above track	> 10 cm	2 cm	rare	disturbing
7	leaves	thickness on rail	> 8 mm	2 mm	rare	disturbing

Fig. 8. Modeling of the RailCab's environment (cut-out)

Disturbing influences on the system's purpose will be marked as disturbance variables. Furthermore, the interplays between the influences will be examined. We consider a situation to be one consistent amount of collectively occurring influences, in which the system has to work properly. We mark influences that cause a state transition of the system as events. Catalogues, that imply spheres of influences and influences, support the creation of environment models. Figure 8 shows, in detail, the specification of the RailCab's environment. The users and the cargo affect the driving behavior of a RailCab, e.g. by their weight. Influences of the environment, such as wind, snow, ice and leaves, affect both, the RailCab's driving behavior as well as the state of the track sections and switches. Track set errors of track sections also influence the RailCab's driving behavior as well as the abrasion of the RailCab itself. The example concretizes the amount of influences I_2 in the influence table (figure 8).

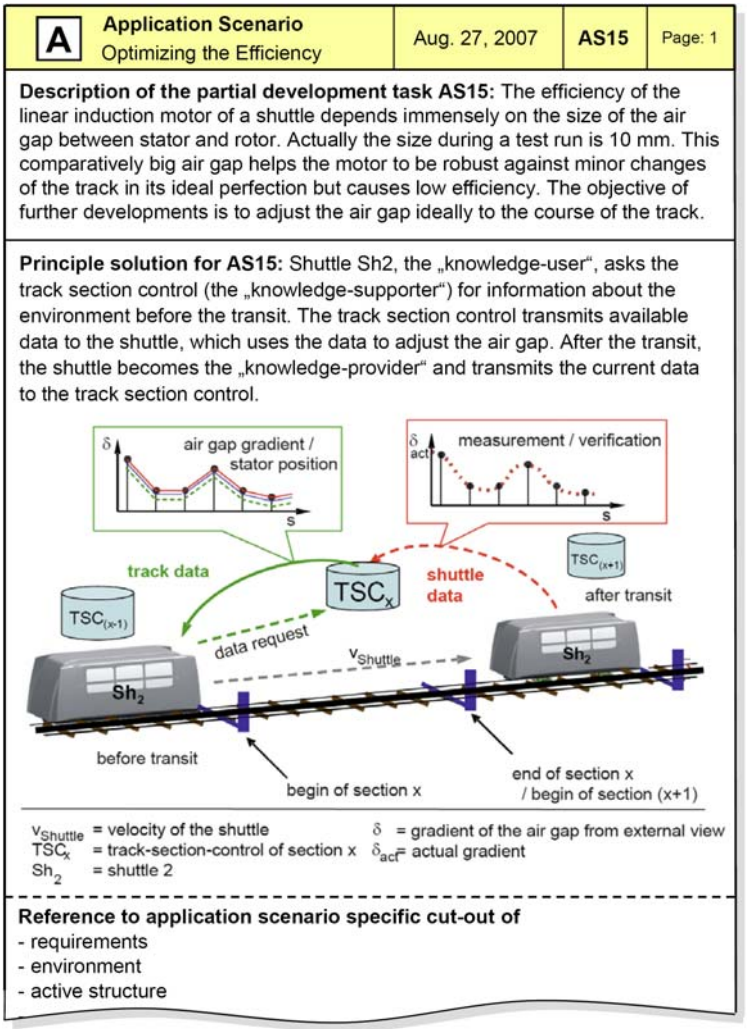


Fig. 9. Application scenario for the example "optimizing the efficiency" [GFP+08]

Application scenario: Application scenarios form first concretizations of the system. They concretize the system's behavior in a special state and a special situation and furthermore, what kinds of events initiate a state transitions. Application scenarios characterize a problem, which needs to be solved in special cases, and also roughly describe the possible solution. One application scenario of the RailCab "optimizing the efficiency" is shown in figure 9.

Requirements: This aspect considers the computer-intern representation of the requirements. The list of requirements sets up its basis. It presents an organized collection of requirements that need to be fulfilled during the product development (such as overall size, performance data). Requirements are separated into demands and wishes [PBF+07]. Every requirement is verbally described and, if possible, concretized by attributes and their characteristics. Several checklists assist the setting up of requirements; see for example [PBF+07], [Rot00], [Ehr03].

System of objectives: This aspect includes the representation of external, inherent and internal objectives and their connections (see chapter 2). A cut-out of the system of objectives of the RailCab is shown in figure 10. The external and inherent objectives are represented as a hierarchical tree. The hierarchy relations are specified by logical relations with declarations of the hierarchy criterion "is part-objective of". The potential internal objectives derive from the external and inherent objectives. The impact between the objectives will be expressed by an assisting influence matrix. The matrix shows if the objectives work in mutual support, if they influence each other negatively or if they are in a neutral relationship. The system is able to follow such systems simultaneously and without any problems, which work mutually and also those systems that have neutral relations. But if the systems influence each other in a negative way, this is an indication for the need of optimization. Instead of an influence matrix, graphs that model objectives and their interplays can be used.

Functions: This aspect concerns the hierarchical subdivision of the functionality. A function is the general and required coherence between input and output parameters, aiming at fulfilling a task. For the setting up of function hierarchies, there is a catalogue with functions which is based on BIRKHOFER [Bir80] and LANGLOTZ [Lan00]. This catalogue has been extended by functions, which especially describe self-optimizing functionality. Functions are realized by solution patterns and their concretizations. A subdivision into sub functions is taking place until useful solution patterns have been found for the functions.

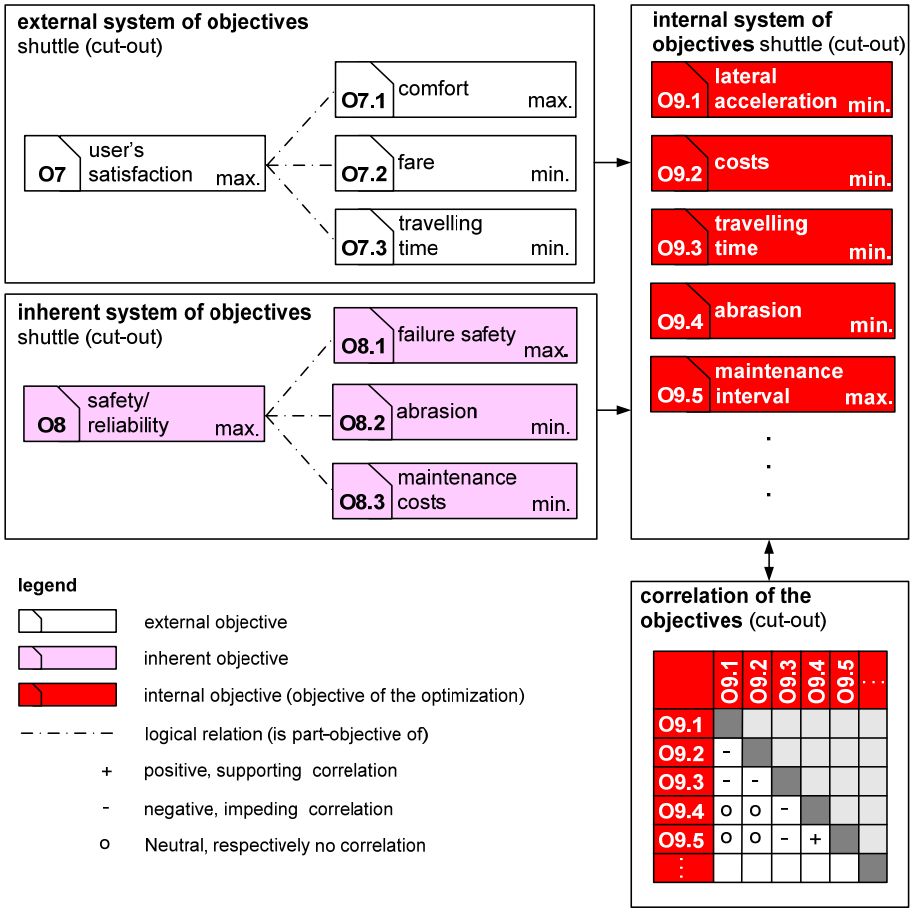


Fig. 10. The RailCab's system of objectives (cut-out)

Active structure: The active structure describes the system elements, their attributes as well as the relation of the system elements. It is the target to define the basic structure of a self-optimizing system, including all system configurations which can be thought anticipated. Figure 11 visualizes a cut-out of a shuttle's active structure. The active structure consists of system elements, such as drive and break modules, air gap adjustment, operating point control, tracking modules, spring- and tilt modules, energy management and their relations. Furthermore, incoming parameters are described, such as *comfort, costs* and *time*, which are external objectives of the user. The system is structured by logic groups in order to improve the necessary clearness. In this case, for example, the system elements of a driving module are combined. The system elements, which deal with the determination of system objectives of the self-optimization process, are marked by a slanting arrow (here: operating point control and air gap adjustment).

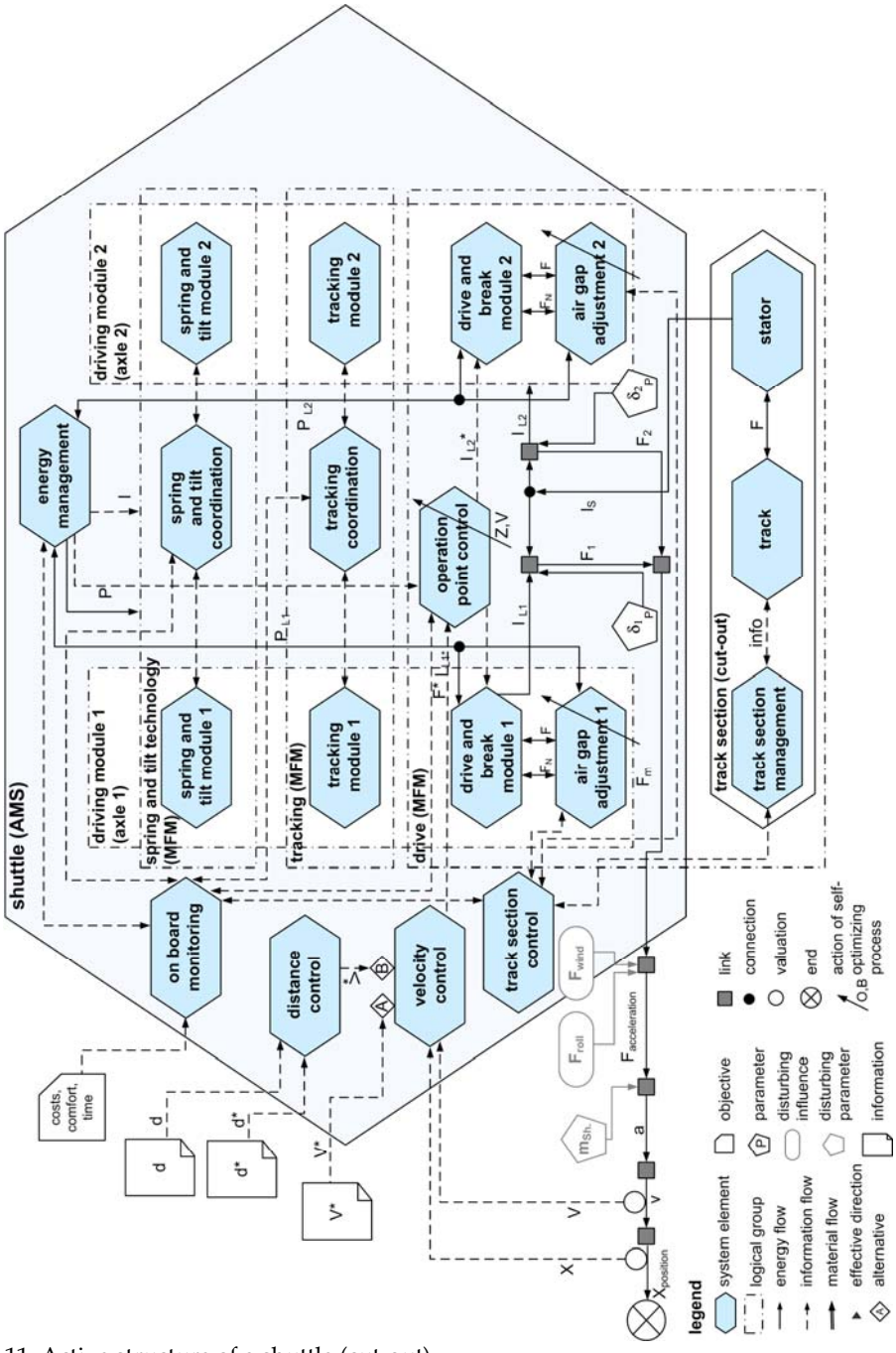


Fig. 11. Active structure of a shuttle (cut-out)

Shape: This aspect needs to be modeled because first definitions of the system's shape have to be carried out already in the phase of the conceptual design. This especially concerns working surfaces, working places, surfaces and frames. The computer-aided modeling takes place by using 3D CAD systems.
Behavior: This group of partial models comprises several kinds of behavior. Basically, what is needed to be modeled are the system's **states** with the connected **operation processes** and the **state transitions** with the **adaptation processes**. The adaptation processes represent the definite realization of the self-optimizing process. If there are several systems taking part in the self-optimizing process, the interplay of these systems needs to be described. Depending on the development task, more kinds of behavior, such as kinematics, dynamics or electromagnetic compatibility of the system's components need to be specified.

- The partial model **Behavior – States** defines the states and state transitions of a system. All of the system's states and state transitions, which can be thought ahead and thus, need to be considered, as well as the events initiating a state transition need to be described. Events can be characteristic influences on the system or already finished activities.
- The partial model **Behavior – Activities** describes the mentioned operation processes, that take place in a system's state, and the adaptation processes, which have the typical features of self-optimization. All in all, the processes are modeled by activities. "Determine fulfillment of current objectives" or "select adequate parameters and configuration" etc. are examples of such activities. Figure 12 introduces a cut-out of the self-optimizing process of the application scenario "optimizing the efficiency". The self-optimizing process is structured by logic groups in the analysis of the situation, the determination of objectives and the behavior adaptation. Within the analysis of the situation, the track data, the RailCab inputs (such as force, efficiency and safety) and information of the energy management module are being interrogated. Furthermore a continuous determination of the degree of fulfillment of the current objectives takes place. The analysis of the situation happens in soft real time. Based on certain information, the air gap adjustment forms a new system of objectives within the determination of the objectives themselves. Right at the beginning of the behavior's adaptation, the identification of the air gap's course concerning the track section takes place. Out of that air gap's course, useful parameters and accordingly configurations (for the self-optimizing air gap adjustment) are chosen in connection with the new system of objectives. Those parameters do not just find their use as an optimization result in the RailCab. Because they can be sent back to the RailCab's according track section control, they will be reused for future RailCabs when they pass over a track section.

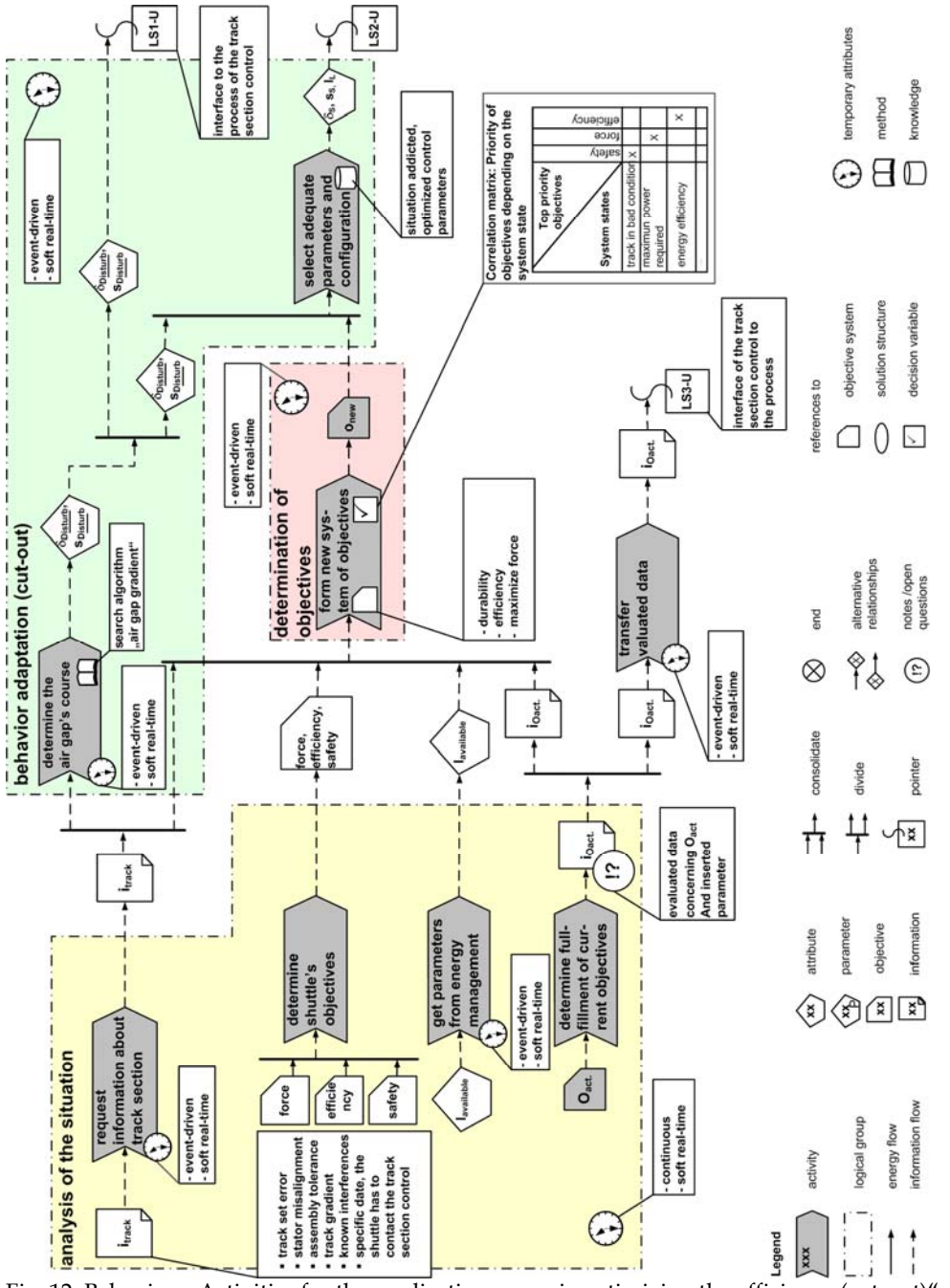


Fig. 12. Behavior – Activities for the application scenario „otimizing the efficiency (cut-out)"

- The partial model **Behavior – Sequence** describes the interaction of several system elements. The activities, being carried out during the interaction of the system elements, and the inter-changed information, are modeled in a chronological order.

4.2 Interrelations between the partial models

The partial models represent the different aspects of the principle solution of a self-optimizing system. The interrelations between the partial models which describe the coherence of the partial models are of high importance. Those interrelations are built up between the constructs of the relating partial models. There are, for example, functions (construct of the partial model *functions*) that are realized by system elements (construct of the partial model *active structure*). These system elements perform activities (construct of the partial model *behavior – activities*), whereas the activities might result out of the functions of the partial model *functions*. There could also be the achieving of a certain temperature (construct *influence* of the partial model *environment*) as an event (construct of the partial model *behavior – states*) that causes the activation of a new state (construct of the partial model *behavior – states*) and other activities. Table 1 shows a couple of interrelations between the partial models. The interrelations are shown directed to the right, i.e. in the table's left side there are the constructs which cause correlation, on the table's right side there are the constructs affecting the connections (an example in the 3rd line, table 1).

construct	partial model	kind of interrelation	construct	partial model
system element	active structure	realizes	function	functions
system element	active structure	performs	activity	behavior – activities
system element	active structure	takes	state	behavior – state
system element	active structure	persuades (opt.)	objective	system of objectives
system element	active structure	has (opt.)	volumes	shape
activity	behavior – activities	results from	function	functions
requirement	requirements	sets boundaries for	volumes	shape
requirement	requirements	decides	function	functions
function	functions	results from	requirement	requirements
influence/event	environment	activates	state	behavior – state
influence/event	environment	activates	activity	behavior – activities
•••				

Table 1. Interrelations between the partial models (cut-out)

A system element within the partial model *active structure* takes up a state in the partial model *behavior – states*. Optional interrelations are marked by *(opt.)*. Taking the information

in table 1 as a basis, a so-called integration model is created, which complements all the already described partial models.

4.3 Particularities within the specification of self-optimizing systems

Chapter 1 already pointed out that the self-optimizing process initiates a new state of the system. The system is transformed from one configuration into another. The partial model *behavior – states* displays all relevant states of the system. It also contains all the events initiating a state transition. The configuration of a system in a specific state is described by its active structure. That means, the active structure can be differently shaped in different states, for example, if different elements of the system (controllers, sensors) are used for the execution of the self-optimizing process. A system's behavior in a certain state is described by its operation process. Operation processes are for example the acquisition of information about the environment, the derivation of adequate control interactions, and the controlling itself. State transitions are realized by adaptation processes, i.e. by self-optimizing processes. The operation and adaptation processes are modeled in the partial model *behavior – activities*. In order to describe the self-optimizing process, all of the three partial models need to be considered simultaneously (figure 13). Every state of the partial models *behavior – states* is assigned to an operation process of the partial model *behavior – activities*, which is operating actively in that state. Moreover, every state is related to a configuration of the active structure, which also actively operates. One example: The state *S5*, the respective operation process and the configuration of the active structure are emphasized by light grey colored, logical groups. The operation process takes place in a periodic way.

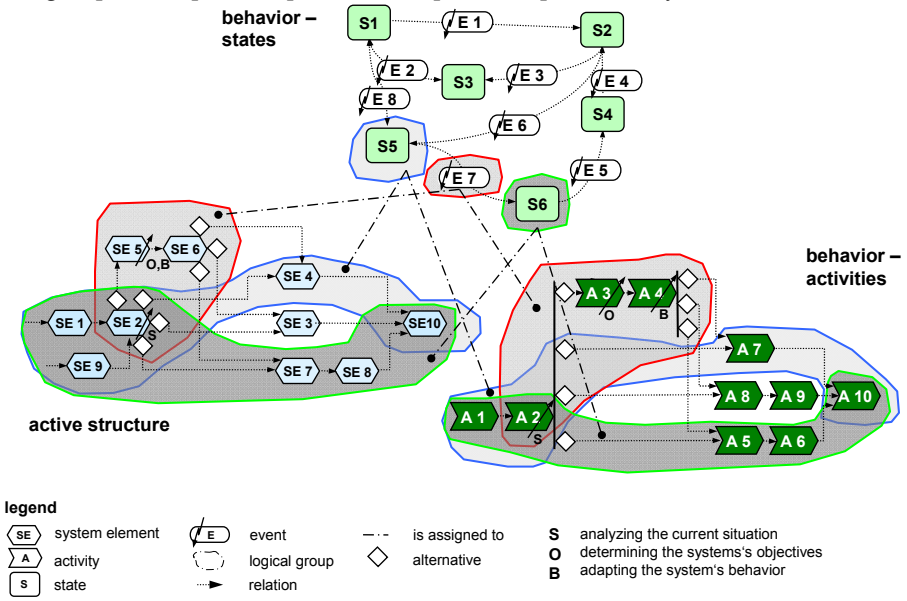


Fig. 13.Cooperation of the partial models active structure, behavior – states and behavior – activities in order to describe the self-optimization (simplified visualization of the principle)

Now – when event E7 appears, an adaptation process is triggered. Therefore, the necessary system elements are activated. Both, the adaptation process and the configuration of system elements, are assigned to the event E7 (see medium grey background in figure 13). After performing the adaptation process, the system takes over the new state S6. A new operation process and a new configuration of system elements are activated. They are colored in a dark grey within figure 13. The adaptation process and the used system elements are no longer activated.

5. Conceptual design of self-optimizing systems

As mentioned in chapter 2, the basic construction and the operation mode of the system are defined within the conceptual design phase. The basic procedure is divided into four sub-phases (figure 14), which are explained in detail below. [GFD+08]

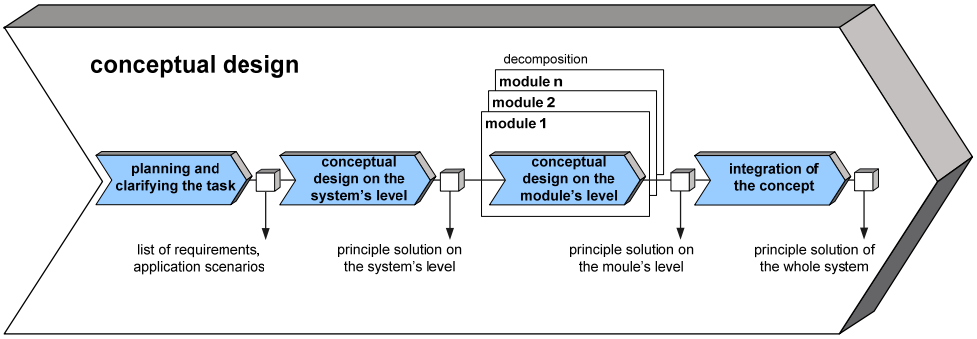


Fig. 14. Process of conceptual design of self-optimizing systems

Planning and clarifying the task

This sub-phase identifies the design task and the resulting requirements on the system is worked out in here (figure 15). At first the task is analyzed in detail. At this the predefined basic conditions for the product, the product program, and the product development are taken into account. This is followed by an analysis of the operational environment which investigates the most important boundary conditions and influences on the system. The external objectives emerge next to disturbances. Beyond that, consistent combinations of influences, so-called situations, are generated. By the combination of characteristic situations with a first discretion of the system's behavior, application scenarios occur. By using the structuring procedure by STEFFEN it is possible to identify a development-oriented product structure for the system and design rules, which guide the developers to realize this product structure type [Ste07]. The results of this sub-phase are the list of requirements, the environment model, the aspired product structure type and the assigned design rules as well as the application scenarios.

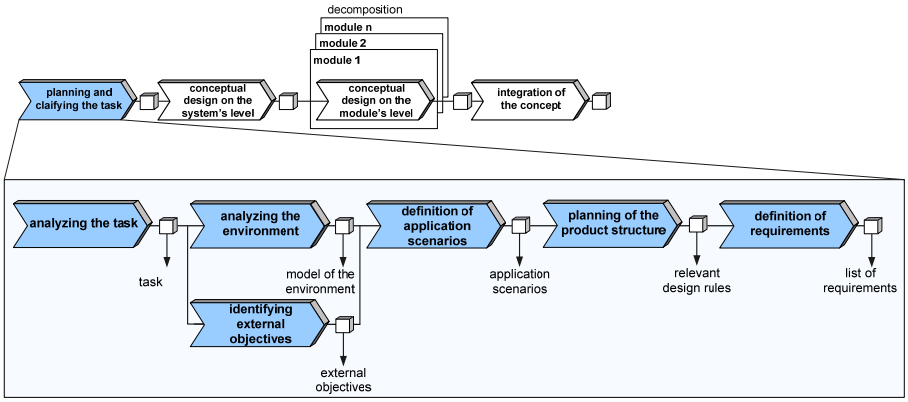


Fig. 15. Conceptual design phase "planning and clarifying the task"

Conceptual design on the system's level

Based on previously determined requirements of the system, solution variants are developed for each application scenario (figure 16). The main functions are derived from the requirements and set into a function hierarchy.

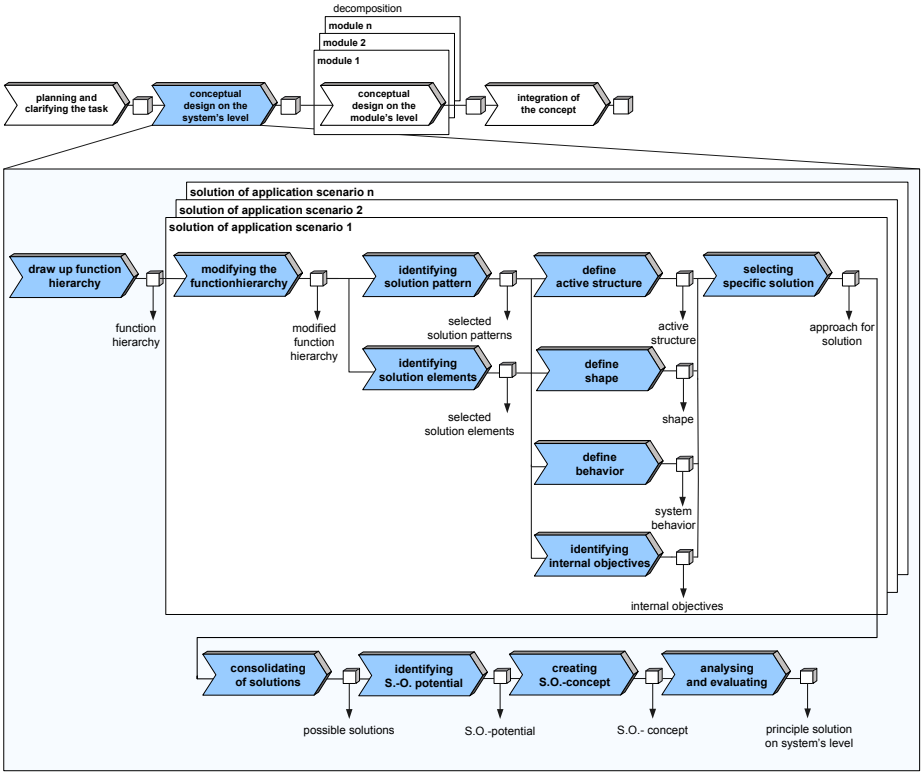


Fig. 16. Conceptual design phase "conceptual design on system's level"

The function hierarchy needs to be modified according to the specific application scenarios, e.g. irrelevant functions are removed and specific sub-functions are added. Then there is a search for "solution patterns" in order to realize the documented functions of the function hierarchy, which will be inserted into a morphologic box.
We use "solution pattern" as a general term. A pattern describes a reoccurring problem and also the solution's core of the problem [AIS+77]. Taking this as a starting point, it results in the classification shown in figure 17. We differentiate between solution patterns that rely on physical effects and between patterns exclusively serving the data processing. The design methodology of mechanical engineering describes the first group as active principles; they describe the principle solution for the realization of a function. The course of development concretizes active principles to material components and patterns of information processing to software components. The relations between active principles and components are of the type n:m; the characteristic depends on the basic method of embodiment design (differential construction method and integrated construction method). Within the integral construction, several active patterns are realized by one component; whereas in the differential construction several components fulfill one active pattern. This is exactly the same in the field of information processing. Basically, a definite modern mechanical engineering system consists of a construction structure that means an arrangement of shape-marked components within a space and their logic aggregation to assemblies and products, and a component structure that means the compound of software components.

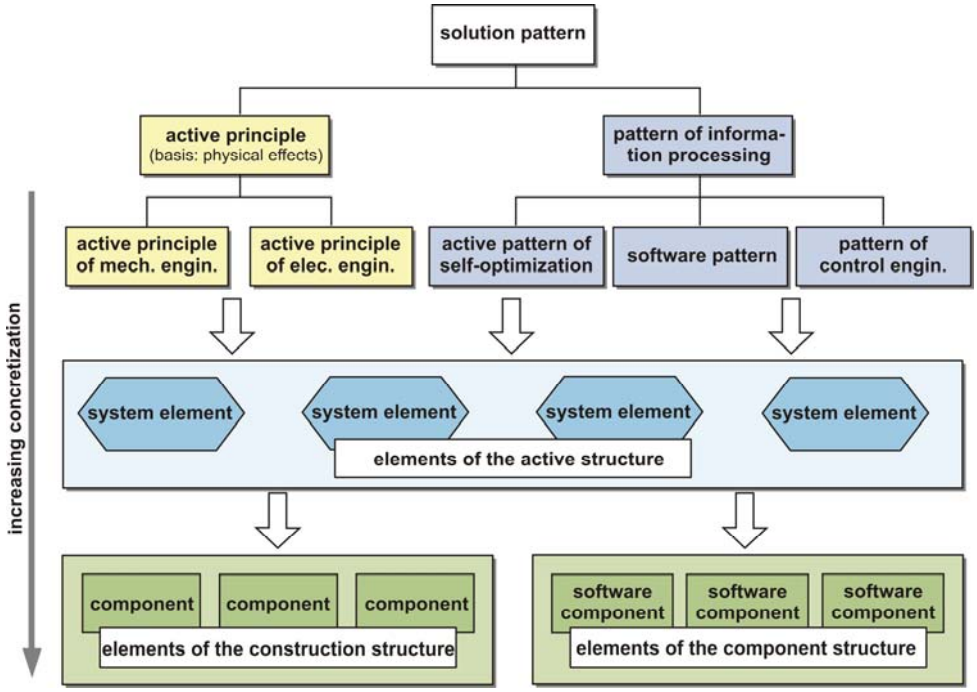


Fig. 17. classification of solution patterns

In some times, there are already existing, well-established solutions which we call "solution elements". If there are such solution elements, they will be chosen instead of the abstract solution patterns. The search for solution patterns is supported by a solution pattern catalogue. We use the consistency analysis in order to determine useful combinations of solution patterns of the morphologic box [Köc04]. As a result, there will be consistent bunches of solution patterns, with a solution pattern for each function.
The consistent bunches of solution patterns form the basis for the development of the active structure. In this step, the refinement of the solution patterns to system elements takes place as well. System elements form an intermediate step between solution patterns on one side and shape-marked components or rather software components on the other side. Based on the active structure, an initial construction structure can be developed because there are primal details on the shape within the system elements. In addition, the system's behavior is roughly modeled in this step. Basically, this concerns the activities, states and state transitions of the system as well as the communication and cooperation with other systems and subsystems. The analysis of the system's behavior produces an imagination of the optimizing processes, running within the system. The external, inherent and internal objectives can be defined.
The solutions for the application scenarios need to be combined. It is important that workable configurations are created which make a reconfiguration of the system possible. Keeping this information in mind, it is identified if there is a containing potential of self-optimization at all. There is a potential for self-optimization if the changing influences on the system require modifications of the pursued objectives and the system needs to adjust its behavior. If there is potential for self-optimization, the function hierarchy needs to be complemented by self-optimizing functions. In particular solution patterns of self-optimization are applied to enable self-optimizing behavior. The resulting changes and extensions of system structure and system behavior need to be included appropriately.
The best solution for each application scenario is chosen and these solutions are consolidated to a principle solution on the system's level. Afterwards, an analysis takes place which looks for contradictions within the principle solution of the system and which contradictions might be solved by self-optimization. Self-optimizing concepts for such contradictions are defined, which contain the three basic steps of self-optimization. The principle solution of a self-optimizing system on the system's level is the result of this phase.

Conceptual design on the module's level
The principle solution on the system's level describes the whole system. It is necessary to have a closer look at the solution, in order to give a statement on the technical and economical realization of the principle solution. For that purpose, the system is decomposed into modules by using the already mentioned structuring procedure by STEFFEN. The decomposition is based on the aspired product structure [Ste07], [GSD+09]. Afterwards a principle solution for each single module is developed. The development of a principle solution for each single module corresponds to the "conceptual design on the system's level", starting out with "planning and clarifying the task". This phase results in principle solutions on the module's level.

Integration of the concept
The module's principle solutions will be integrated into a detailed principle solution of the whole system. Again there is an analysis in order to find contradictions within the principle solutions of the modules and it is checked if these contradictions can be solved by self-optimization. Concluding, a technical-economical evaluation of the solution takes place. The result of this phase is a principle solution of the whole system that serves as a starting point for the subsequent concretization.

Integration of the concept: The module's principle solutions will be integrated into a detailed principle solution of the whole system. There is an analysis in order to find contradictions within the principle solutions of the modules. Again it will be checked if these contradictions can be solved by self-optimization. Concluding, a technical-economical evaluation of the solution is taking place. The result of that phase is a principle solution of the whole system that serves as a starting point for the subsequent concretization. This concretization is carried out parallel in the specific domains (mechanical engineering, electrical engineering, control engineering and software engineering). Chapter 7 gives further information on this.
On the basis of an example, the phases p*lanning and clarifying the task* as well as *conceptual design on the system's level* will be described into detail. There will not be any further consideration of the c*onceptual design on the module's level* because it operates by analogy with the c*onceptual design on the system's level*. The i*ntegration of the concept* has also been explained and is not being discussed anymore.

6. The role of the principle solution during the concretization

The communication and cooperation of the developers from the different domains throughout the whole development process is very important for a successful and efficient development of self-optimizing systems. The principle solution forms the basis for this communication and cooperation.
Within the conceptual design phase the domain-spanning development tasks are carried out in a cooperative way. Within the concretization the developers work on different modules and in different domains. Thus their specific development tasks in one domain of a module need to be synchronized with those of other domains respectively other modules. The development processes for the modules are synchronized by one superior process of the total system (figure 18). Within this process comprehensive aspects of the system like the shell or the dynamics of the whole system are developed in detail. [GRD+09]

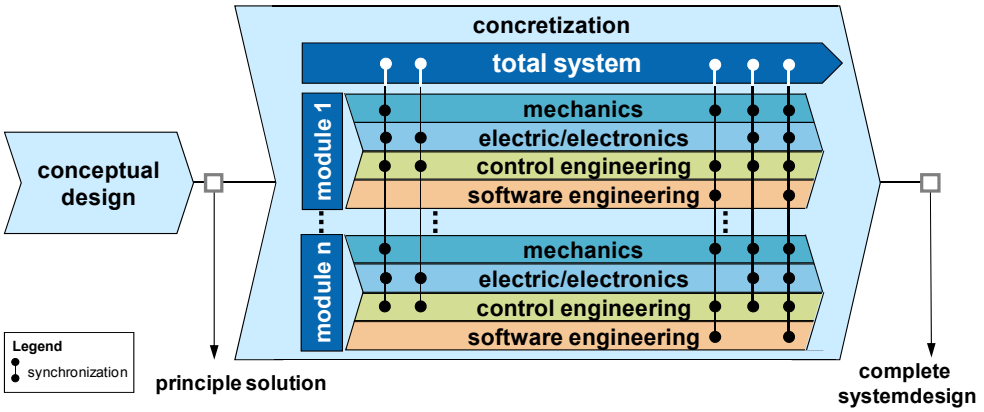


Fig. 18. Basic structure of the development process [GRD+09]

Furthermore, the information, based in the principle solution, serves as a fundament for deducing of domain-specific concretization tasks. In a first step, the system elements of a domain and their relations within the active structure will be identified. After that will be analyzed what kind of domain-specific functions are fulfilled by the system elements, which requirements they have to comply and which behavior is appropriate in certain situations. Following this, it will be checked if domain-specific requirements need to be added. In case of a software engineering, the necessary software components of the component structure, including the input- and output parameters, can be deduced by the system elements of the active structure (figure 18) [GSD+09].

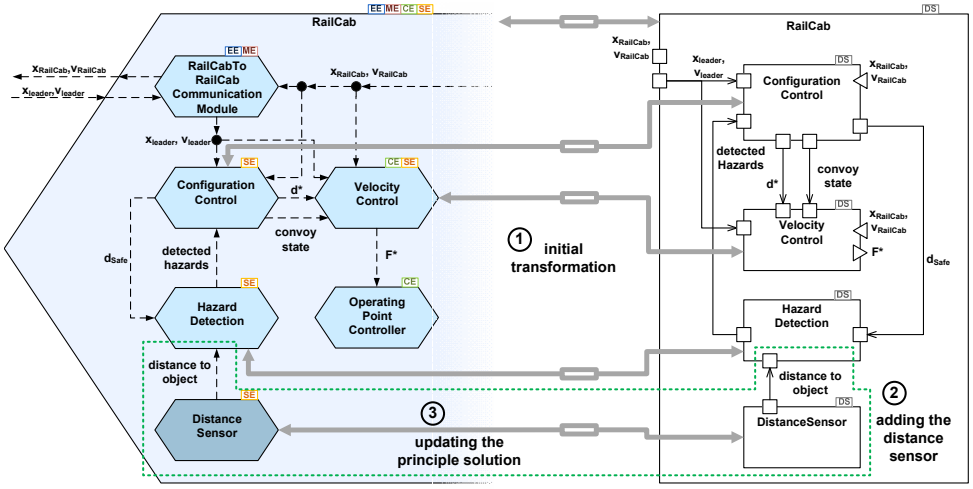


Fig. 19. The transformation from the active structure into a component diagram (software engineering) [GSD+09]

In case of changes occur during the domain-specific concretization, which affect other domains have to be transferred back into the principle solution. This happens for example if

there will be identified additional internal objectives during the course of concretization of a self-optimization process (in the frame of the determination of objectives). Thus the principle solution becomes a domain-spanning system model for the concretization. The aim is to keep this domain-spanning system model and the domain-specific models consistently. Figure 20 schematically shows the versions of the domain-spanning system specification and the different domain-specific models that are created in the course of the concretization. The shown change scenario can be realized by the use of automated model transformations [GSD+09].

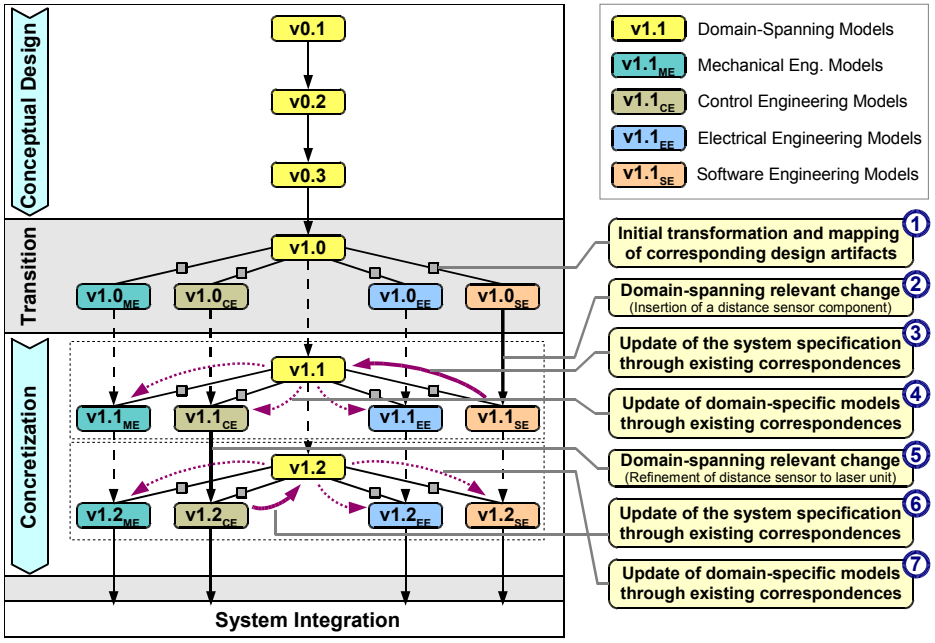


Fig. 20. Propagation of relevant changes between the domain-specific models and the domain-spanning system specification [GSD+09]

7. Conclusion

The paradigm of self-optimization will enable fascinating perspectives for the future development of mechanical engineering systems. These systems rely on the close interaction of mechanics, electrical engineering/electronics, control engineering and software engineering, which is aptly expressed by the term mechatronics. At present there is no established methodology for the conceptual design of mechatronic systems, let alone for self-optimizing systems. Concerning the conceptual design of such systems, the main challenge consists in the specification of a domain-spanning principle solution, which describes the basic construction as well as the mode of operation in a domain-spanning way. The presented specification technique offers the possibility to create a principle solution for advanced mechatronic systems, with regard to self-optimizing aspects, such as “application scenarios” and “system

of objectives". Simultaneously it outperforms classic specification techniques by appropriately encouraging the conceptual design process. It is fundamental to the communication and cooperation of the participating specialists and enables them to avoid design mistakes, which base on misunderstandings between them. It has been described in what way the according concretization, which takes place parallel to the participating domains, is going to be structured and coordinated on the basis of the principle solution. The practicability of the specification technique and the appropriate methodology was demonstrated by the example of a complex railway vehicle.

8. Acknowledgement

This contribution was developed and published in the course of the Collaborative Research Center 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" funded by the German Research Foundation (DFG) under grant number SFB 614.

9. References

[ADG+08] ADELT, P.; DONOTH, J.; GAUSEMEIER, J.; GEISLER, J.; HENKLER, S.; KAHL, S.; KLÖPPER, B.; KRUPP, A.; MÜNCH, E.; OBERTHÜR, S.; PAIZ, C.; PODLOGAR, H.; PORRMANN, M.; RADKOWSKI, R.; ROMAUS, C.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITKOWSKI, U.; WITTING, K.; ZNAMENSHCHYKOV, O.: Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte. HNI-Verlagsschriftenreihe, Band 234, Paderborn, 2008

[AIS+77] Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I.; Angel, A.: A Pattern Language. Oxford University Press, New York, 1977

[Bir80] Birkhofer, H.: Analyse und Synthese der Funktionen technischer Produkte. Dissertation, Technische Universität Braunschweig, 1980

[Ehr03] Ehrlenspiel, K.: Integrierte Produktentwicklung. Carl Hanser Verlag, München, 2003

[GEK01] Gausemeier, J.; Ebbesmeyer, P.; Kallmeyer, F.: Produktinnovation - Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag, München, 2001

[GFD+08] Gausemeier J., Frank U., Donoth J. and Kahl S. Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus – Teil 1/2. Konstruktion, Vol. 7/8 and 9, July/August and September 2008, pp. 59-66/ pp. 91-108 (Springer-VDI-Verlag, Düsseldorf).

[GRD+09] Geiger, C.; Reckter, H.; Dumitrescu, R.; Kahl, S.; Berssenbrügge, J.: A Zoomable User Interface for Presenting Hierarchical Diagrams on Large Screens. In: 13th International Conference on Human-Computer Interaction (HCI International 2009), July 19-24, 2009, San Diego, CA, USA, 2009

[GSD+09] Gausemeier, J.; Steffen, D.; Donoth, J.; Kahl, S.: Conceptual Design of Modularized Advanced Mechatronic Systems. In: 17th International Conference on Engineering Design (ICED`09), August 24-27, 2009, Stanford, CA, USA, 2009

[GSG+09] Gausemeier, J.; Schäfer, W.; Greenyer, J.; Kahl, S.; Pook, S.; Rieke, J.: Management of Cross-Domain Model Consistency during the Development of Advanced Mechatronic Systems. In: 17th International Conference on Engineering Design (ICED`09), August 24-27, 2009, Stanford, CA, USA, 2009

[Köc04] Köckerling, M.: Methodische Entwicklung und Optimierung der Wirkstruktur mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe Band 143, Paderborn, 2004

[Lan00] Langlotz, G.: Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte. Dissertation, Institut für Rechneranwendung in Planung und Konstruktion, Universität Karlsruhe, Shaker-Verlag, Band 2/2000, Aachen, 2000

[LHL01] Lückel, J.; Hestermeyer, T.; Liu-Henke, X.: Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2001), Villa Olmo; Como, Italy

[PBF+07]Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H.: Engineering Design – A Systematic Approach. ed. 3, 2007, Springer Verlag, London, 2007

[Rot00]Roth, K.-H.: Konstruieren mit Konstruktionskatalogen. Springer-Verlag, , Band 1 Konstruktionslehre, 3. Auflage, Berlin, 2000

[Ste07]Steffen, D.: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 207, 2007

[VDI04]Verein Deutscher Ingenieure (VDI): VDI-Richtlinie 2206 - Entwicklungsmethodik für mechatronische Systeme. Beuth-Verlag, Berlin, 2004

[ZBS+05] Zimmer, D.; Böcker, J.; Schmidt, A.; Schulz, B.: Elektromagnetische Direktantriebe im Vergleich. In: Antriebstechnik, no. 2/2005, Vereinigte Fachverlage GmbH, Mainz, 2005

[ZS05]Zimmer, D.; Schmidt, A.: Der Luftspalt bei Linearmotor-getriebenen Schienenfahrzeugen. In: Antriebstechnik, no. 2/2005, Vereinigte Fachverlage GmbH, Mainz, 2005

15

Contributions to the Multifunctional Integration for Micromechatronic Systems

M. Grossard Mathieu and M. Chaillet Nicolas
CEA, LIST, Service Robotique Interactive, 18 route du Panorama, BP6, FONTENAY AUX ROSES, F- 92265
France
FEMTO-ST Institute, Automatic Control and Micro-Mechatronic Systems Department
France

1. Introduction

Mechatronics is the interdisciplinary area related to the integration of mechanical, electronic and control engineering, as well as information technology to design the best solution to a given technological problem. It implies that mechatronics relates to the design of systems, devices and products aimed at achieving an optimal balance between basic mechanics and its overall control. Robotic systems design has certainly been the pioneer field of mechatronic applications.
Due to the increase in the difficulty to miniaturize these advanced (or intelligent) technological products, research in the microrobotic field is required to find novel solutions to design micromechatronic systems. When applying scale reduction to robotic systems usually encountered at the macroscopic scale, the miniaturization step necessarily implies functional integration of these systems. This general trend makes microsystems more and more functionally integrated, which makes them converging towards the adaptronic (or smart structures) concept.
In this coming mechatronic concept, all functional elements of a conventional closed-loop system are existent and at least one element is applied in a multifunctional way. The aim of such a system is to combine the greatest possible number of application-specific function in one single element. It aims at building up a microstructure that is marked by minor complexity and high functional density (Fig. 1).
The key idea followed in the micromechatronic design is that three of the four components (i.e. sensors, actuators and mechanical structure) in smart microrobotic structures are made of a single functional (or active) material, such as piezoelectric or shape memory alloys materials. They can perform actuation or/and sensing functions by interchanging energy forms (for example, electric energy, magnetic energy and mechanical energy).

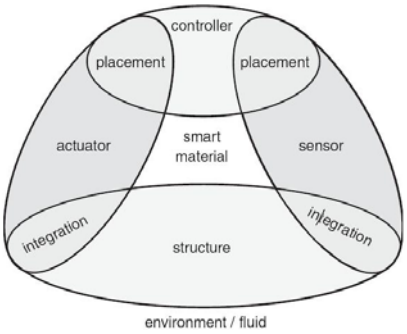


Fig. 1. Integrated smart structure (Hurlebaus, 2005).

Most often, these integrated microdevices are compliant mechanisms, i.e. single-bodies, elastic continua flexible structures that transmit a motion by undergoing elastic deformation, as opposed to jointed rigid body motions of conventional articulated mechanisms. Using compliant mechanisms for the design of small scale systems is of a great interest, because of simplified manufacturing, reduced assembly costs, reduced kinematic noise, no wear, no backlash, and ability to accommodate unconventional actuation schemes when they integrate active materials.

These micromechatronic devices consist of a dynamic system combining a flexible mechanical structure with integrated multifunctional materials. For the simulation and optimization of such microsystems, control and system theory together with proper modeling of the plant are to be applied. The finite element method is a widespread tool for numerical simulation and structural modeling that can include multiphysics due to the cross coupling effects of the active material. Afterwards, the efficiency and proper positioning of actuators and sensors in these systems can be analyzed using the concepts of controllability and observability. Then, the state-space representation is desirable to achieve model reduction and to perform control design methodologies.

A general overview of design specificities including mechanical and control considerations for micromechatronic structure is firstly presented in this chapter. Performance criteria including mechanical performances, spillover treatment, model reduction techniques and robust control are briefly presented afterwards.

Finally, an example of a new optimal synthesis method to design topology and associated robust control methodologies for monolithic compliant microstructures is presented. The method is based on the optimal arrangement of flexible building blocks thanks to a multi-criteria genetic algorithm. It exploits the piezoelectric effect, thus making realistic the adaptronic concept, i.e. integration of the actuation/sensing principle inside the mechanical structure.

2. Design and control specificities of active flexible micromechatronic systems

In the section, a particular attention is drawn on the approach used for modelling and optimizing these micromechanisms design.

2.1 Design and modelling

When compared to macroscale mechatronic systems, design of micromechatronic systems needs some particular attention. Indeed, this miniaturization step implies to rethink the main functionalities of the traditional systems in accordance to the specificities of the microscale:

- their microstrucure, as well as their fabrication and microassembly process ;

in many applications including Micro Electro Mechanical Systems (MEMS) (Lee 2003), (Chang 2006), (Kota 1994), surgical tools (Frecker 2005) (Houston 2007), etc, compliant mechanisms have already been used. They are single-body, elastic continua flexible structures, that deliver the desired motion by undergoing elastic deformation, as opposed to jointed rigid body motions of conventional mechanisms. There are many advantages of compliant mechanisms, among them: simplified manufacturing, reduced assembly costs, reduced kinematic noise, no wear, no backlash, high precision, and ability to accommodate unconventional actuation schemes.

- their actuators and sensors with high resolution and small size ;

new ways for producing actuation and sensing need to be studied in their physical principle, as well as their good adaptability for the achieving tasks at the microscale in term of displacement, force, controllability, observability, etc. The use of active functional material (also called multifunctional materials), which can convert energy from one form to the other, are thus widespread in the context of micro-actuator/sensor design.

- their control methodology and implementation.

The design of controllers for active flexible micromechanisms is a challenging problem because of nonlinearities in the structural system and actuators/sensors behavior, nonavailability of accurate mathematical models, a large number of resonant modes to accurately identify and control. Thus, robust control design methods need to be used.

Most often, modelling and simulating active flexible mechanisms can be made following several steps sketched on Fig. 2. Starting from the chosen active material (such as piezoelectric ceramics or magnetostrictive materials), coupled with some specific boundary conditions and system geometry inherent to the problem, the global equations for the system behavior are established using the equations of dynamic equilibrium and kinematics. Then, the finite element (FE) method is generally used for discretizing the spatial distribution of displacements within the flexible structure: it reduces the problem formulation to a discrete set of differential equations. In this manner, multiphysics problem can be treated when considering the electromechanical (in the case of piezoelectric materials) or magnetomechanical (in the case of magnetostrictive materials) couplings of the active materials. In the perspective of controlling these mechanisms, this dynamic input/output model is expressed using state-space formalism. Structural models obtained by using FE method exhibit a huge number of degrees of freedom. Thus, the resulting full order model has to be drastically reduced thanks to reduction techniques. Usual techniques of reduction consist in selecting the most influent modes that lie in the frequency spectrum of interest, i.e. those that are strongly controllable and observable with the actuator/sensor configuration.

Some examples of software tools related to the simulation (and, in some restrictive case, the optimization process as well) of smart structures can be found in (Janocha 2007).

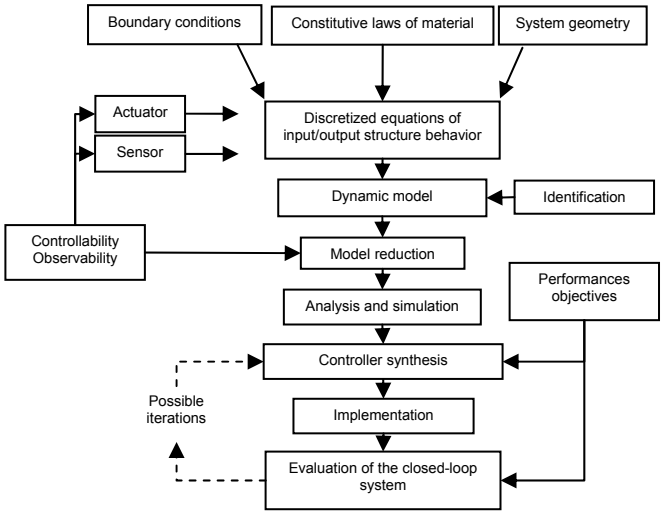


Fig. 2. General approach for modelling and testing active flexible micromechanisms

2.2 Design optimization

Modeling, simulating and controlling integrated flexible structures imply a parameterization of the considered system (geometry, material, etc). In link with the application task, parametric studies are generally led to determine the most adequate design for the structure, the actuators/sensors, the controller, etc. Thus, this design process can be formalized under an optimization problem to select the optimal solution(s).

A general strategy needs to be appropriate to deal with the coupling problem between the structure, the actuators and sensors, and the control of the system.

Generally, a decomposition approach is privileged, especially for complex problem. The optimization of some parts of the system is separately considered under several constraint hypothesis. For example, some papers deal solely with control systems for a specified structure. Other works deals with optimal actuator placement on a predetermined flexible structure, or with coupling flexible structures for single actuators, etc. A current work concerning design methodologies and application of formal optimization methods to the design of smart structures and actuators can be found in (Frecker 2000).

In the following, a particular attention is made on the use of piezoelectric ceramic as an active material for microrobotic tools. Indeed, one type of smart material-based actuator typically used to actuate compliant structures is piezoelectric ceramic PZT actuators: when compared to other conventional actuation principles at small scales, they have very appealing properties in the sense of micromechatronic design. When integrated inside a compliant mechanism, piezoelectric actuators can exert actuation forces to the host structure without any external support. They can also be manufactured into the desired shape, while making realistic the realization of piezoelectric monolithic compliant mechanisms, such as microgrippers (Breguet 1997). Piezoelectric actuation is mostly used for microrobot design in order to achieve nanometric resolutions, and has naturally become widespread in micromanipulation systems (Agnus 2005).

However, one limitation of piezoelectric actuators is that they are capable of producing only about 0.1% strain, resulting in a restricted range of motion. A number of papers only address the problem of optimally designing coupling structures to act as stroke amplifiers of the piezoelectric actuator (Kota 1999), (Lau 2000). Opposite to these methods, where the piezoelectric elements in the structure are predetermined, a large body of work related to optimization of active structures deals with the optimal location of actuators on a given structure (Barboni 2000). Another general approach to optimally design actuated structures is to simultaneously (Maddisetty 2002) or separately (Abdalla 2005) optimize the actuator size. Finally, few studies consider the topology optimization (shape) of monolithic PZT active structures (Nelli Silva 1999).

2.3 Dynamics of the flexible micromechanisms

There are a number of difficulties associated with the control of flexible structures (amongst them, variable resonance frequencies, highly resonant dynamics and time-varying states subjected to external disturbances).
For example, since the dynamic model of a flexible structure is characterized by a large number of resonant modes, accurate identification of all the dominant system dynamics often leads to very high order model. Thus, a model reduction is required by the designer. A number of approaches for model reduction have been developed, such as model reduction via balanced realization (Moore 1981). But, this reduction model step is quite delicate because of spillover effect, that is to say when unwanted interactions between the controlled system and neglected structural modes lead to instability.
Thus, an important condition for a controlled dynamic system is to guarantee its stability. Moreover, the stability of such controlled dynamic system has to be robust, that is to say it must stabilizes the real system in spite of modelling errors or parameters changes. Thus, traditional robust control system design techniques such as LQG, H_2 and H_∞ commonly appear in research works (Abreu 2003), (Halim 2002a), (Halim 2002b). The performances of such high authority controllers have to take into account model uncertainties and modelling errors introduced by model truncation.
For some specific class of flexible structures, which can be modelled as collocated resonant systems, active damping dissipative controllers (for example, Positive Position Feedback, Integral Force Feedback, Direct Velocity Feedback...) have proven to offer great robustness, performance, and ease of implementation relatively to traditional techniques. On the contrary of the advanced techniques, the direct use of dissipative collocated controllers can have the advantages to produce control systems of low order and good robustness, associated with high dynamic performance. These techniques are often focused on damping the dominant modes (Aphale 2007). The natural modes of the system must be controlled using proper actuators and sensors positions ('Control authorithy'): actuator and sensor positions are sought for influencing (controllability) and sensing (observability) the modal oscillations.

3. Example of an optimal synthesis tool for designing smart microrobotic structure

In this paragraph, a method developed for the optimal design of piezoactive compliant micromechanisms is presented. It is based on a flexible building block method, which uses

an evolutionary approach, to optimize a truss-like planar structure made of passive and active building blocks, made of piezoelectric material. An electromechanical approach, based on a mixed finite element formulation, is used to establish the model of the active piezoelectric blocks. From the first design step, in addition to conventional mechanical criteria, innovative control-based metrics can be considered in the optimization procedure to fit the open-loop frequency response of the synthesized mechanisms. In particular, these criteria have been drawn here to optimize modal controllability and observability of the system, which is particularly interesting when considering control of flexible structures.
More specific details on this method can be found in (Bernardoni 2004a), (Bernardoni 2004b), (Grossard 2007a), (Grossard 2007b).

3.1 Compliant building blocks

Two libraries of compliant elements in limited number are proposed in our method. These bases are composed respectively of 36 and 19 elements of passive and piezoactive blocks, made of beams assembly (Fig. 3). They are sufficient to build a high variety of topologies. In particular, the various topologies of piezoelectric active blocks allow them to furnish multiple coupled degrees of freedom, thus generating more complex movements with only one building block.

3.2 Principles of the method and design parameters

The specification of a planar compliant mechanism problem considers specific boundary conditions: fixed frame location, input (actuators), contacts and output (end-effector). In particular, a particular attention is drawn on the integrated piezoactive elements taken from the active library as actuator. The design method consists in searching for an optimal distribution of allowed building blocks, as well as for the optimal set of structural parameters and materials. The location of fixed nodes and that of the piezoactive blocks can also be considered as optimisation parameters. The topology optimization method uses a genetic algorithm approach, which allows true multicriteria optimization and the use of these discrete variables (Fig. 4). The algorithm is structured as follows: discrete variable parameterization of compliant mechanisms considering conception requirements (mesh size, topology, material and thickness, boundary conditions), evaluation of individuals (design criteria calculation), and stochastic operators for the optimization (modification of compliant mechanisms description).

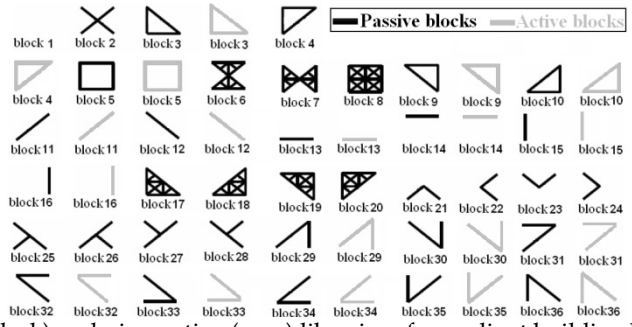


Fig. 3. Passive (black) and piezoactive (grey) libraries of compliant building blocks, for planar compliant mechanisms synthesis.

Many fitness functions are available in our method, thus allowing the optimal design of devices within a wide schedule of conditions: static mechanical fitness (free displacement and blocking force at the output port, geometric advantage, mechanical advantage, etc.), various dynamic control-oriented metrics have been newly implemented to meet specific control requirements for microrobotics devices. Obviously, the design strategy depends on the metrics chosen, which must be based on the real needs for the device use.

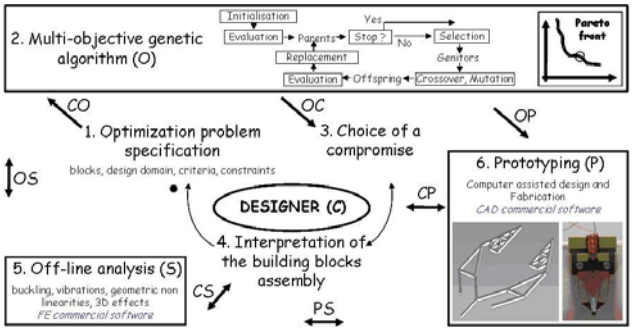


Fig. 4. Flowchart of the optimal design method of compliant structures (multicriteria optimization).

3.3 Electromechanical FE model of the piezoelectric structure

In our method, it is assumed that the compliant mechanisms are undergoing structural deformations, mainly due to the bending of the beams constituting the blocks. Thus, the models of the blocks are obtained considering Navier-Bernoulli beam type finite elements. Structural parameters of each rectangular block are height, width and thickness. Material characteristics of each block are parameterized by Young's modulus, Poisson's ratio, yield strength, density, and piezoelectric coefficients for the piezoactive blocks.

The piezoceramic beams constituting the active blocks are perfectly bonded to electrodes at their lower and upper faces (Fig. 5). Exploiting the transverse effect of piezoelectricity, longitudinal deformation S_{11} along L dimension is generated under the transverse electric field E_3. Considering the one-dimensional form of piezoelectricity equation along the length direction of the beam, the piezoelectric coupling matrix **d** and the stress-free electric

permittivity matrix **ε** are each represented by a single coefficient, d_{31} and ε_{33} respectively, and the electric-free compliance matrix **s** is represented by s_{11}. Hence, within the piezoelectric beam, the constitutive relations for the strain S_{11} and electric displacement D_3, as functions stress T_{11} and electric field E_3, take the form:

$$\begin{pmatrix} S_{11} \\ D_3 \end{pmatrix} = \begin{bmatrix} s_{11} & d_{31} \\ d_{31} & \varepsilon_{33} \end{bmatrix} \begin{pmatrix} T_{11} \\ E_3 \end{pmatrix} \tag{1}$$

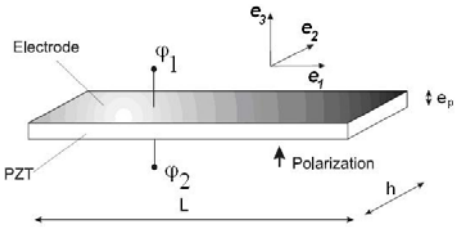


Fig. 5. Thickness-polarized piezoelectric beam transducer with electroded surfaces, and orientation in the material reference frame (e_1, e_2, e_3). φ_1 and φ_2 denotes the electric potential of the electrodes.

From Hamilton's principle modified for general electromechanical system, the model of the active beam takes the following form:

$$M_b \ddot{\eta}_b + K_b \eta_b = G_b \Phi_b + Fr_b \tag{2}$$

where $\mathbf{M}_b$, $\mathbf{K}_b$ and $\mathbf{G}_b$ are respectively the mass, stiffness and electromechanical coupling beam matrices. $\mathbf{\Phi}_b = [\varphi_1 \ \varphi_2]^t$ is the vector representing the electric potentials on the upper and lower faces of the piezoelectric beam. Matrix $\mathbf{G}_b$ induces piezoelectric loads, which makes the actuator beam expand (or contract) proportionally to the external controlled potential difference (φ_1 - φ_2). The forces vector $\mathbf{Fr}_b$ is due to the variational mechanical work terms. Displacement field is related to the corresponding node values $\boldsymbol{\eta}_b$ by the mean of the shape functions, calculated under Euler-Bernoulli beam assumptions. Detailed derivations can be readily found in finite element textbooks, and corresponding matrices in (Grossard 2007).

The stiffness, damping, and mass matrices of each block are then calculated numerically, considering every combination of the discrete values allowed for the structural optimization variables. Then, the global dynamic behaviour of a structure results from the mass, damping, stiffness and electromechanical coupling matrices assembly of the constitutive blocks, and is done at each step for each individual during the optimization process.

The conservative dynamic behaviour of a structure is described through its mass $\mathbf{M}_g$, stiffness $\mathbf{K}_g$ and electromechanical coupling $\mathbf{G}_g$ matrices, obtained by the assembly in of the matrices of the blocks constituting the structure.

3.4 State-space model for flexible structure

Each flexible structure synthesized by our method is defined as a finite-dimension, controllable and observable linear system with small damping and complex conjugate poles (Lim 1993). Its undamped dynamic behavior is modeled by the following second-order differential matrix equations:

$$\begin{aligned} M_g \ddot{\eta}_g + K_g \eta_g &= E_g u \\ y &= F_g \eta_g \end{aligned} \tag{3}$$

η_g is the nodal displacement vector, u is the input vector which defines the controlled command of the actuator. For example, in case of a piezoelectric actuation scheme, u is defined by Φ_g. y is the output vector, defined from the output displacement matrix F_g. Each element of u (resp. y) denotes a physical actuator (resp. sensor) whose related degree of freedom is defined by the location of the nonzero entry in the corresponding column in E_g (resp. row in F_g).

By means of modal decomposition, a solution of the form

$$\eta_g(t) = \sum_i \psi_i q(t) = \psi q(t) \tag{4}$$

is considered, which consists of a linear combination of mode shapes ψ_i. The eigenvectors matrix ψ and corresponding eigenfrequencies ω_i are obtained as solutions of the vibration eigenproblem (Grossard 2007). Replacing η_g by ψq in (Eq. 3), multiplying on the left by ψ^t, the induced orthogonality relationships in modal form lead to:

$$\begin{aligned} \ddot{q} + \mathrm{diag}\left(2\xi_i\omega_i\right)\dot{q} + \mathrm{diag}\left(\omega_i^2\right)q &= \psi^t E_g u \\ y &= F_g \psi q \end{aligned} \tag{5}$$

In this equation, diagonal damping by using Basil's hypothesis has been introduced. Thus, ξ_i is here the i-th modal damping ratio. This hypothesis can be made because the system to control is slightly damped in the low-frequency band, where the modes are well separated. One interesting state vector x consists of modal velocities and frequency weighted modal displacements:

$$x = \begin{pmatrix} \dot{q}_1 & \omega_1 q_1 & \cdots & \dot{q}_n & \omega_n q_n \end{pmatrix}^t \tag{6}$$

with the advantage that the elements of state vector corresponding to each mode are about the same magnitude. This yields the matrices triplet (A, B, C) which denotes the modal state-space representation of a structure as stated below,

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX \end{aligned} \tag{7}$$

The matrices take the forms $A = \mathrm{diag}\left(A_1 \ \ldots \ A_n\right)$, $B = \left(B_1{}^t \ \ldots \ B_n{}^t\right)^t$ and $C = \left(C_1 \ \ldots \ C_n\right)$, with :

$$A_i = \begin{bmatrix} -2\xi_i\omega_i & -\omega_i \\ \omega_i & 0 \end{bmatrix},\ B_i = \begin{bmatrix} \psi_i{}^t E_g \\ 0 \end{bmatrix},\ C_i = \begin{bmatrix} 0 & \dfrac{1}{\omega_i} F_g \Psi_i \end{bmatrix} \tag{8}$$

Let us note that A matrix depends on the structure itself (eigenfrequencies and modal damping ratios), B matrix on the location and class of actuators, and C matrix on location and class of sensors. This modal state is considered to be a physical coordinate because of its direct physical link to structural mode shapes.

3.5 Useful measures for fitting the frequency response of flexible systems

From the computation of the linear state model of compliant systems, an optimal topology design strategy is derived taking into account control considerations. In the method, numerical criteria help reaching input-output open-loop system performances with specific operation requirements.

Since the dynamic model of a flexible structure is characterized by a large number of resonant modes, accurate identification of all the dominant system dynamics often leads to very high order model. Thus, a model reduction is required.

A first criterion has been drawn to optimize the reduced-model accuracy of the systems, while limiting spillover effects (Fig. 6). Given a set of structures to optimize, the optimal structures are chosen as the ones guaranteeing the highest joint controllability and observability for all the modes in the bandwidth of interest (i.e. resonance peaks amplitudes must be maximized in the frequencies bandwidth [0; ω_c] to increase authority control on these dominant modes), while providing the minimum joint controllability and observability of the neglected modes (i.e. the amplitudes of resonance peaks after cut-off frequency must be minimized to increase gain margin and to limit modes destabilization in this area). This criterion will enable the rise of structures with accurate reduced model, based on a few highly dominant modes, allowing the easy identification and computation of state model, well adapted to further design and implementation of the control system.

To improve simultaneously the control authority on the k first dominant modes and the accuracy of the reduced order model, the first new criteria implemented in the method is the following:

$$J_1^k = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=k+1}^{n} \sigma_i} \tag{9}$$

where σ_i are the Hankel Singular Values (HSV) defined in their modal form for flexible structures (Lim 1996):

$$\sigma_i = \frac{\sqrt{\left(\psi_i F_g\right)^t \left(\psi_i F_g\right)\left(\psi_i^t E_g\right)\left(\psi_i^t E_g\right)^t}}{4\xi_i \omega_i^2}. \tag{10}$$

The corresponding i-th HSV is proportionally linked to the maximum amplitude value of the frequency response at ω_i natural pulsation. The k first resonant modes (where k < n) will be optimized to guarantee high HSV compared to the ones out of the bandwidth. The modal states with small HSV are both weakly controllable and weakly observable, and will be removed from the reduced-system.

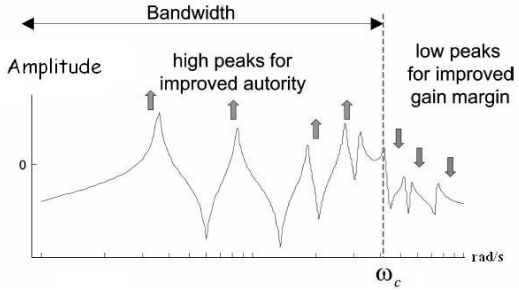


Fig. 6. Desired form of the open-loop frequency response function.

A second criteria relating to control of flexible is particularly interesting. One major characteristic of a collocated system is the interlacing of poles and zeros along the imaginary axis. For a lightly damped structure, poles and zeros are located in the left half-part in the pole-zero map. Such systems are minimum of phase, so that collocated systems are known to possess interesting properties. Vibration control of flexible structures involving collocated characteristics was discussed in (Martin 1978). Control was shown to have simple stability criteria due to the alternating poles and zeros pattern.

An evaluation function was implemented in our method to be used in the optimization process in order to obtain systems designs with collocated type open-loop transfer function, forcing the resonances (poles) and antiresonances (zeros) alternating in the reduced model. Inspired by (Martin 1976), it can be shown that the maximization of the following discrete criterion will imply the interlacing pole-zero pattern exhibited by a collocated transfer function:

$$J_2^k = \left| \sum_{i=1}^{k} \operatorname{sign}\left(\left(F_g \psi_i\right)\left(\psi_i^t E_g\right)\right) \right| \tag{11}$$

where sign(.) = {-1; 0;+1} according to the argument sign. The sum over i concerns all the modes contained in the frequency spectrum of the first k dominant modes, where the alternative is desired.

3.6 Example of an optimal synthesis of an integrated flexible piezoelectric actuator

The concepts presented previously have been applied to the design of a microgripper actuator, considering a multi-criteria optimization problem, with both static mechanical (free stroke and blocking force at the output node of the structure) and control-oriented J_1 and J_2 fitnesses.

The synthesis of a symmetric monolithic microactuation mechanism, made of a single piezoelectric material (PIC151 from PI Piezo Ceramic Technology) has been made using our method. From the set of structures results, one pseudo-optimal solution, whose topology is presented on Fig. 7, is chosen to illustrate performances.

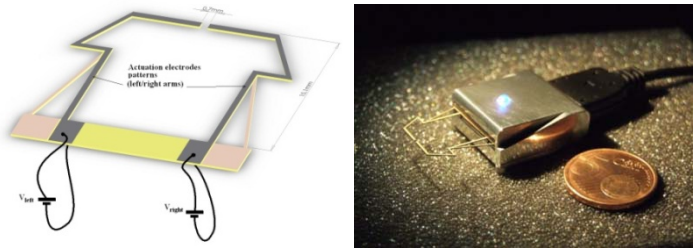


Fig. 7. On the left, model of the piezoeletric device with top face electrode patterns - V_{left} (resp. V_{right}) is the controlled input for actuating the left (resp. right) arm. On the right, photo of the prototyped piezoelectric device, obtained by laser cutting.

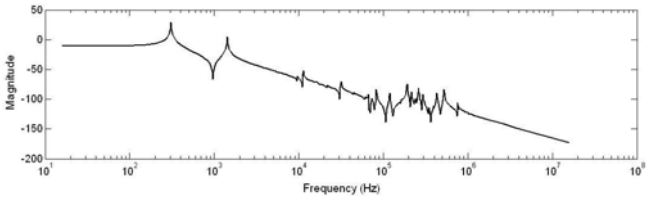


Fig. 8. Bode amplitude diagram of the chosen solution between input (voltage u, in V) and arm output (deflection in µm) simulated by our method.

Each arm of such a microgripper is able to produce ±10.69µm movement range when ±100V is applied on the actuation electrodes. A blocking force of about 840mN is also produced. Moreover, this solution is an example of structure with interesting control-oriented criteria (Fig. 8): the authority control on the two first resonant modes is well optimized, resulting in an important roll-off after the second resonance. As expected, this structure exhibits an alternating pole/zero pattern in the spectrum of interest.

4. Conclusion

A brief overview of design specificities and strategies including mechanical and control considerations for micromechatronic structures has been presented. Designing, modelling and controlling flexible microscale structures actuated by active materials are a quite complex task, partly because the designer has to deal with several problems. Amongst them,

specific mechanical performances, spillover treatment, model reduction techniques and robust control have been highlighted in this chapter.
To help the design of such systems, an example of a systematic optimal design method for smart compliant mechanisms has been particularly presented here. This method can consider a smart compliant mechanism as an assembly of passive and active compliant building blocks made of PZT, so that actuators are really integrated in the structure. Complex multi-objective design problems can be solved, taking advantage of versatile criteria to synthesize high performance microrobotic flexible mechanisms designs. In addition to classical mechanical criteria, currently encountered in topology optimization (i.e. force and displacement maximization), our method considers now simultaneously efficient control-based criteria.
Open-loop transfer considerations lead to two new efficient numerical criteria. A first criterion can modulate resonances amplitudes of its frequency response function in a spectrum of interest. A second criterion can force minimum-phase system property. These two criteria, coupled with mechanical ones, help designing non-intuitive compliant mechanisms. This optimization strategy was tested for the optimal design of a microgripper actuator. The results obtained have proved that the method can furnish innovative and efficient solutions.

5. References

Abdalla M., and al. (2005), Design of a piezoelectric actuator and compliant mechanism combination for maximum energy efficiency, *Smart Material and Structures*, vol. 14, pp. 1421-1430, 2005

Abreu G. L. C., Ribeiro M., Steffen J. F. (2003), Experiments on optimal vibration control of a flexible beam containing piezoelectric sensors and actuators, *Journal of Shock and Vibration*, vol. 10, pp. 283-300, 2003

Agnus J., Nectoux P. , Chaillet N. (2005), Overview of microgrippers and micromanipulation station based on a MMOC microgripper, *Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 117-123, 2005

Aphale S. S., Fleming A.J., Moheimani S. O. R. (2007), Integral resonant control of collocated smart structures, Smart Materials and Structures, vol.16, pp. 439-446, 2007

Barboni R., and al. (2000), Optimal placement of PZT actuators for the control of beam dynamics, *Smart Material and Structures*, pp. 110-120, 2000

Bernardoni P., and al. (2004a), A new compliant mechanism design methodology based on flexible building blocks, *Smart Material and Structures*, vol. 5383, pp. 244-254, USA, 2004

Bernardoni P. (2004b), Outils et méthodes de conception de structures mécaniques à déformations réparties et actionnement discret – applications en microrobotique, *PhD Thesis realized at the CEA*, University Paris 6, France, 2004

Breguet J.M. (1997), and al., Monolithic piezoceramic flexible structure for micromanipulation, *9th International Precision Engineering Seminar and 4th International Conference on Ultraprecision in Manufacturing Engineering*, pp. 397-400, Braunschweig Germany, 1997

Chang H.C., Tsai J.M.L., Tsai H.C., Fang W. (2006), Design, fabrication, and testing of a 3-DOF HARM micromanipulator on (111) silicon substrate, *Sensors and Actuators*, vol. 125, pp. 438-445, 2006

Frecker M., Canfield S. (2000), Optimal design and experimental validation of compliant mechanical amplifiers for piezoceramic stack actuators, *Journal of Intelligent Material Systems and Structures*, vol. 11, pp. 360-369, 2000

Frecker M., Haluck R. (2005), Design of a multifunctional compliant instrument for minimally invasive surgery, *Journal of Biomedical Engineering*, vol. 127, pp. 990-993, November 2005

Grossard M., Rotinat-Libersa C., Chaillet N. (2007a), Redesign of the MMOC microgripper piezoactuator using a new topological method, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Zürich, Switzerland, 2007

Grossard M., Rotinat-Libersa C., Chaillet N., Perrot Y. (2007b), Flexible building blocks method for the optimal design of compliant mechanisms using piezoelectric material, *12th IFToMMWorld Congress*, Besançon, France, 2007

Halim D., Moheimani S. O. R. (2002a), Experimental implementation of spatial H1 control on a piezoelectric laminate beam, *IEEE/ASME Transactions on Mechatronics*, vol. 4, pp. 346-356, 2002

Halim D., Moheimani S. O. R. (2002b), Spatial H2 control of a piezoelectric laminate beam: experimental implementation, *IEEE Transactions on Control System Technology*, vol. 10, pp. 533-546, 2002

Houston K., Sieber A., Eder C., Tonet O., Menciassi A., Dario P. (2007), Novel Haptic Tool and Input Device for Real Time Bilateral Biomanipulation addressing Endoscopic Surgery , *Proc. of the 29th Annual International Conference of the IEEE EMBS*, Lyon, France, August 23-26, pp. 198-201, 2007

Hurlebaus S. (2005). Smart Structures – Fundamentals and Applications, Lecture Notes, Texas A&M University, Zachry Department of Civil Engineering

Janocha, H. (2007). *Adaptronics and smart structures – Basics, Materials, design, and Applications,* Springer Editor, ISBN 978-3-540-71965-6, Berlin Heildeberg New-York

Kota S., Ananthasuresh G.K., Crary S.B., and Wise K. D. (1994), Design and fabrication of micro-electromechanical systems, *ASME Journal of Mechanical Design*, vol. 116, pp. 1081-1088, 1994

Kota S. (1999), "Tailoring unconventional actuators using compliant transmissions: design methods and applications", *IEEE/ASME Transactions on Mechatronics*, vol. 4, pp. 396-408, December 1999

Lau G. K., and al. (2000), Systematic design of displacement – amplifying mechanisms for piezoelectric stacked actuators using topology optimization, *Journal of Intelligent Material Systems and Structures*, vol. 3985, pp. 583-591, 2000

Lee W. H., Kang B. H., Oh Y. S., Stephanou H., Sanderson A.C., Skidmore G., Ellis M. (2003), Micropeg manipulation with a compliant microgripper, *Proceedings of IEEE Int. Conf. on Robotics and Automation,* pp. 3213-3218, Taipei, Taiwan, September 2003

Lim K. B., Gawronski W. (1993), Actuators and sensor placement for control of exible structures, *Control and Dynamics Systems: Advances in Theory and Applications*, ed. London, Academic Press, 1993

Lim K. B., Gawronski W. (1996), Balanced control of Flexible structures, *ed. London, Springer*, 1996.

Maddisetty H., Frecker M. (2002), Dynamic topology optimization of compliant mechanisms and piezoceramic actuators, *ASME Journal of Mechanical Design*, vol. 126, pp. 975-983, 2002

Martin G. D. (1978), On the control of flexible mechanical systems, *PhD Dissertation, Stanford University*, USA, 1978

Moore B.C. (1981), Principal component analysis in linear systems: controllability, observability, and model reduction, *IEEE Transactions on Automatic Control*, vol. 26, 1981

Nelli Silva E.C., Kikuchi N. (1999), Design of piezoelectric transducers using topology optimization, *Smart Material and Structures*, vol. 8, pp. 350 -365, USA, 1999