

IntechOpen

Advanced Strategies for Robot Manipulators

Edited by S. Ehsan Shafiei



Advanced Strategies for Robot Manipulators

edited by

Seyed Ehsan Shafiei

Advanced Strategies for Robot Manipulators

<http://dx.doi.org/10.5772/278>

Edited by S. Ehsan Shafiei

© The Editor(s) and the Author(s) 2010

The moral rights of the and the author(s) have been asserted.

All rights to the book as a whole are reserved by INTECH. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECH's written permission.

Enquiries concerning the use of the book should be directed to INTECH rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 3.0 Unported License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are those 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 chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in Croatia, 2010 by INTECH d.o.o.

eBook (PDF) Published by IN TECH d.o.o.

Place and year of publication of eBook (PDF): Rijeka, 2019.

IntechOpen is the global imprint of IN TECH d.o.o.

Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Advanced Strategies for Robot Manipulators

Edited by S. Ehsan Shafiei

p. cm.

ISBN 978-953-307-099-5

eBook (PDF) ISBN 978-953-51-5954-4

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,400+

Open access books available

118,000+

International authors and editors

130M+

Downloads

151

Countries delivered to

Our authors are among the
Top 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Contents

Preface IX

- Chapter 1 **The Comparative Assessment of Modelling and Control of Mechanical Manipulator** 1
M. H. Korayem, H. N. Rahimi and A. Nikoobin
- Chapter 2 **Hyper Redundant Manipulators** 27
Ivanescu Mircea and Cojocaru Dorian
- Chapter 3 **Micro-Manipulator for Neurosurgical Application** 61
M. R. Arshad and Ya'akob Yusof
- Chapter 4 **DeLiA: A New Family of Redundant Robot Manipulators** 73
Jaime Gallardo-Alvarado
- Chapter 5 **Dynamic Modelling, Tracking Control and Simulation Results of a Novel Underactuated Wheeled Manipulator (WAcrobot)** 91
Mohsen Moradi Dalvand and Bijan Shirinzadeh
- Chapter 6 **Kinematics Synthesis of a New Generation of Rapid Linear Actuators for High Velocity Robotics with Improved Performance Based on Parallel Architecture** 107
Luc Rolland
- Chapter 7 **Sliding Mode Control of Robot Manipulators via Intelligent Approaches** 135
Seyed Ehsan Shafiei
- Chapter 8 **Supervision and Control Strategies of a 6 DOF Parallel Manipulator Using a Mechatronic Approach** 173
FJoão Mauricio Rosário, Didier Dumur, Mariana Moretti, Fabian Lara and Alvaro Uribe
- Chapter 9 **Collision Detection and Control of Parallel-Structured Flexible Manipulators Based on Unscented Kalman Filter** 197
Yuichi Sawada, Yusuke Watanabe and Junki Kondo
- Chapter 10 **On Saturated PID Controllers for Industrial Robots: The PA10 Robot Arm as Case of Study** 217
Jorge Orrante-Sakanassi, Víctor Santibáñez and Ricardo Campa

-
- Chapter 11 **Real-Time-Position Prediction Algorithm for Under-actuated Robot Manipulator Using of Artificial Neural Network** 249
Ahmad Azlan Mat Isa, Hayder M.A.A. Al-Assadi and Ali T. Hasan
- Chapter 12 **On Nonlinear Control Perspectives of a Challenging Benchmark** 261
Guangyu Liu and Yanxin Zhang
- Chapter 13 **A Unified Approach to Robust Control of Flexible Mechanical Systems Using \mathcal{H}_∞ Control Powered by PD Control** 273
Masayoshi Toda
- Chapter 14 **An Improved Adaptive Kinematics Jacobian Trajectory Tracking of a Serial Robot Passing Through Singular Configurations** 287
Ali T. Hasan, Hayder M.A.A. Al-Assadi and Ahmad Azlan Mat Isa
- Chapter 15 **Development of Fuzzy-logic-based Self Tuning PI Controller for Servomotor** 311
Oyas Wahyunggoro and Nordin Saad
- Chapter 16 **Distributed Particle Filtering over Sensor Networks for Autonomous Navigation of UAVs** 329
Gerasimos G. Rigatos
- Chapter 17 **Design and Control of a Compact Laparoscope Manipulator: A Biologically Inspired Approach** 365
Atsushi Nishikawa, Kazuhiro Taniguchi, Mitsugu Sekimoto, Yasuo Yamada, Norikatsu Miyoshi, Shuji Takiguchi, Yuichiro Doki, Masaki Mori and Fumio Miyazaki
- Chapter 18 **Open Software Architecture for Advanced Control of Robotic Manipulators** 381
J. Gomez Ortega, J. Gamez García, L. M. Nieto Nieto and A. Sánchez García
- Chapter 19 **Structure and Property of the Singularity Loci of Gough-Stewart Manipulator** 397
Y. Cao, Y. W. Li and Z. Huang

Preface

Robotic technology has grown beyond the boundaries of imagination during recent decades. Nowadays, it's not very surprising to see that a robot can hear, see and even talk and a servant robot is not a dream anymore. But now we confront newer challenges such as nano-robots, surgical manipulators and even robots who can make decisions which are employed for underwater or space missions.

Amongst the robotic systems, robot manipulators have proven themselves to be of increasing importance and are widely adopted to substitute humans in repetitive and/or hazardous tasks. Modern manipulators have a complicated design and need to do more precise, crucial and critical tasks. So, the simple traditional control methods cannot be efficient, and advanced control strategies with considering special constraints need to be established. In spite of the fact that groundbreaking researches have been carried out in this realm until now, there are still many novel aspects which have to be explored.

This book consists of a set of materials that introduces various strategies related to robot manipulators. Although the topics provided here are not in a rational order, they can be divided into three major subjects such as design and *modelling*, *control strategies* and *applications of robot manipulators*. These subjects cover different approaches like dynamic modelling, redundant manipulators, micro-manipulator, parallel manipulator, nonlinear control, intelligent control and many other valuable matters that are addressed here by different authors through 19 chapters.

I gratefully acknowledge the contributions made by each of my coauthors. They showed enthusiasm to contribute their knowledge that lead to creation of this book. However, this is not a text book for academic education, the book is addressed to graduate students as well as researchers in the field and I am sure they can benefit from its multidisciplinary chapters.

Editor

Seyed Ehsan Shafiei
Shahrood University of Technology
Iran

The Comparative Assessment of Modelling and Control of Mechanical Manipulator

M. H. Korayem, H. N. Rahimi and A. Nikoobin
*Robotic Research Lab, College of Mechanical Engineering,
Iran University of Science and Technology
Iran*

1. Introduction

1.1 Overview

In this book chapter a comparative assessment of modelling and control of mechanical manipulator is considered. First, kinematic and dynamic modelling of wide range of mechanical manipulators comprising flexible link, flexible joint and mobile manipulators are considered. Then, open-loop optimal control problem is formulated to control of the obtained system. Finally, some applications of method including motion planning and maximum payload determination are illustrated through the computer simulations.

1.2 Problem statement

Mechanical flexibilities can be classified into two categories: Link flexibility and joint flexibility. Link flexibility is a result of applying lightweight structure in manipulator arms designed to increase the productivity by fast motion and to complete a motion with small energy requirement. Joint flexibility arises from elastic behavior of the drive transmission systems such as transmission belts, gears and shafts. Mobile manipulators are combined systems consists of a robotic manipulator mounted on a mobile platform. Such systems are able to accomplish complicated tasks in large workspaces. In particular the greatest disadvantage of mobile robotic manipulators is that most of these systems are powered on board with limited capacity. Hence, incorporating light links can minimize the inertia and gravity effects on links and actuators and it results to decrease the energy consumption in the same motion. Hence, lightweight systems have primary importance in design and manufacturing stages of mobile manipulators.

1.3 Motivation

Unfortunately, reviewing of the recent literature on modelling and optimization of flexible and mobile manipulators shows that a very scant attention has been paid to study of model that describes both link and joint flexibility, particularly for mobile manipulators. The main motivation for this study is to present a comprehensive modelling and optimal control of flexible link-joint mechanical mobile manipulators. It can provide an inclusive reference for other researchers with comparative assessment view in the future studies.

1.4 Prior work

Analyzing of nonlinear dynamic motion of elastic manipulators is a very complex task that plays a crucial role in design and application of such robots in task space. This complexity arises from very lengthy, fluctuating and highly nonlinear and coupled set of dynamic equations due to the flexible nature of both links and joints. The original dynamics of robotic manipulators with elastic arms, being described by nonlinear coupled partial differential equations. They are continuous nonlinear dynamical systems distinguished by an infinite number of degrees of freedom. The exact solution of such systems does not exist. However, most commonly the dynamic equations are truncated to some finite dimensional models with either the assumed modes method (AMM) or the finite element method (FEM).

The assumed mode expansion method was used to derive the dynamic equation of the flexible manipulator (Sasiadek & Green, 2004). Dynamic modelling technique for a manipulator with multiple flexible links and flexible joints was presented based on a combined Euler-Lagrange formulation and assumed modes method (Subudhi & Morris, 2002). Then, control of such system was carried out by formulating a singularly perturbed model and using it to design a reduced-order controller. Combined Euler-Lagrange formulation and assumed modes method was used for driving the equation of motions of flexible mobile manipulators with considering the simply support mode shape and one mode per link (Korayem & Rahimi Nohooji, 2008). Then, open-loop optimal control method was proposed to trajectory optimization of flexible link mobile manipulator for a given two-end-point task in point-to-point motion.

In finite element method, the elastic deformations are analyzed by assuming a known rigid body motion and later superposing the elastic deformation with the rigid body motion (Usoro et al. 1986). One of the main advantages of FEM over the most of other approximate solution methods to modelling the flexible links is the fact that in FEM the connection are supposed to be clamp-free with minimum two mode shapes per each link (Korayem et al. 2009(a)). This ensures to achieve the results that display the nonlinearity of the system properly.

The Timoshenko beam theory and the finite element method was employed to drive the dynamic equation of flexible link planar cooperative manipulators in absolute coordinates (Zhang & Yu, 2004). Dynamic model of a single-link flexible manipulator was derived using FEM and then studied the feed-forward control strategies for controlling the vibration (Mohamed & Tokhi, 2004). Finite element method was used for describing the dynamics of the system and computed the maximum payload of kinematically redundant flexible manipulators (Yue et al., 2001). Then, the problem was formulated for finding the optimal trajectory and maximum dynamic payload for a given point-to-point task. Finally, numerically simulation was carried out for a planar flexible robot manipulator to validate the research work.

The review of the recent literature shows that extensive research has been addressed the elastic joints robotic arms (Korayem et al. 2009(b)). However, there is only limited research works have been reported on a comprehensive model that describes both link and joint elasticity (Rahimi et al. 2009). Moreover, in almost all cases, linearized models of the link flexibility are considered which reduced the complexity of the model based controller (Chen, 2001).

Mobile manipulators have recently received considerable attention with wide range of applications mainly due to their extended workspace and their ability to reach targets that are initially outside of the manipulator reach. A comprehensive literature survey on mobile manipulator systems can be found (Bloch, 2003). A host of issues related to mobile manipulators have been studied in the past two decade. These include for example: dynamic and static stability (Papadopoulos & Rey, 1996), force development and application (Papadopoulos & Gonthier, 1999), maximum payload determination (Korayem & Ghariblu, 2004). However, a vast number of research publications that deal with the mobile manipulators focus on techniques for trajectory planning of such robots (Korayem & Rahimi Nohooji, 2008).

Motion planning for mobile manipulators is concerned with obtaining open-loop or close-loop controls. It steers a platform and its accompanying manipulator from an initial state to a final one, without violating the nonholonomic constraints (Sheng & Qun, 2006). In most studies of trajectory planning for mobile manipulators the end effector trajectory is specified and the optimal motion planning of the base is considered (Mohri et al., 2001), or integrated motion planning of the base and the end effector is carried out (Papadopoulos, et al., 2002). However, because of designing limitation or environmental obstacle in majority of practical application of mobile manipulators especially in repetitive applications, the platform must follow a specified pose trajectory. In this case, designer must control the joint motions to achieve the best dynamic coordination that optimize the defined cost function such as energy consumption, actuating torques, traveling time or bounding the velocity magnitudes. Applications for such systems abound in mining, construction or in industrial factories.

Optimal control problems can be solved with direct and indirect techniques. In the direct method at first the control and state variables are discretized and the optimal control problem is transcribed into a large, constrained and often sparse nonlinear programming problem, then, the resulting nonlinear programming problem is treated by standard algorithm like interior point methods (Wachter & Biegler, 2006). Famous realizations of direct methods are direct shooting methods (Bock & Plitt, 1984) or direct collocation methods (Hargraves & Paris, 1987). However, direct methods are not yield to exact results. They are exhaustively time consuming and quite inefficient due to the large number of parameters involved. Consequently, when the solution of highly complex problems such as the structural analysis of optimal control problems in robotics is required, the indirect method is a more suitable candidate. This method is widely used as an accurate and powerful tool in analyzing of the nonlinear systems. The indirect method is characterized by a "first optimize, then discretize" strategy. Hence, the problem of optimal control is first transformed into a piecewise defined multipoint boundary value problem, which contains the full mathematical information about the respective optimal control problem. In the following step, this boundary value problem is discretized to achieve the numerical solution (Sentinella & Casalino, 2006). It is well known that this technique is conceptually fertile, and has given rise to far-reaching mathematical developments in the wide ranges of optimal dynamic motion planning problems. For example, it is employed in the path planning of flexible manipulators (Rahimi et.al, 2009), for the actuated kinematic chains (Bessonnet & Chessé, 2005) and for a large multibody system (Bertolazzi et al., 2005). A survey on this method is found in (Callies & Rentrop, 2008).

1.5 Layout

The balance of the remaining of the chapter is organized as follows. Section 2 provides background information about kinematic and dynamic analysis of the flexible mobile robotic manipulators. Hence, assumed mode and finite element methods are introduced and formulated to dynamic modelling of flexible link manipulators. Then, the flexible model is completed by adding the joint flexibility. After that, formulation is extended to comprise the mobile manipulators. Section 3 consists of a brief review of converting the problem from optimal control to optimization procedure with implementing of Pontryagin's minimum principle. some application examples with the two links flexible mobile manipulator is detailed in this section. Finally, the concluding remarks with a brief summary of the chapter is presented in the last section.

2. System modelling

2.1 Kinematic analysis

A mobile manipulator consisting of differentially driven vehicle with n flexible links and n flexible revolute joints is expressed in this section (Fig. 1). The links are cascaded in a serial fashion and are actuated by rotors and hubs with individual motors. The flexible joints are dynamically simplified as a linear torsional springs that works as a connector between the rotors and the links. A concentrated payload of mass m_p is connected to the distal link.

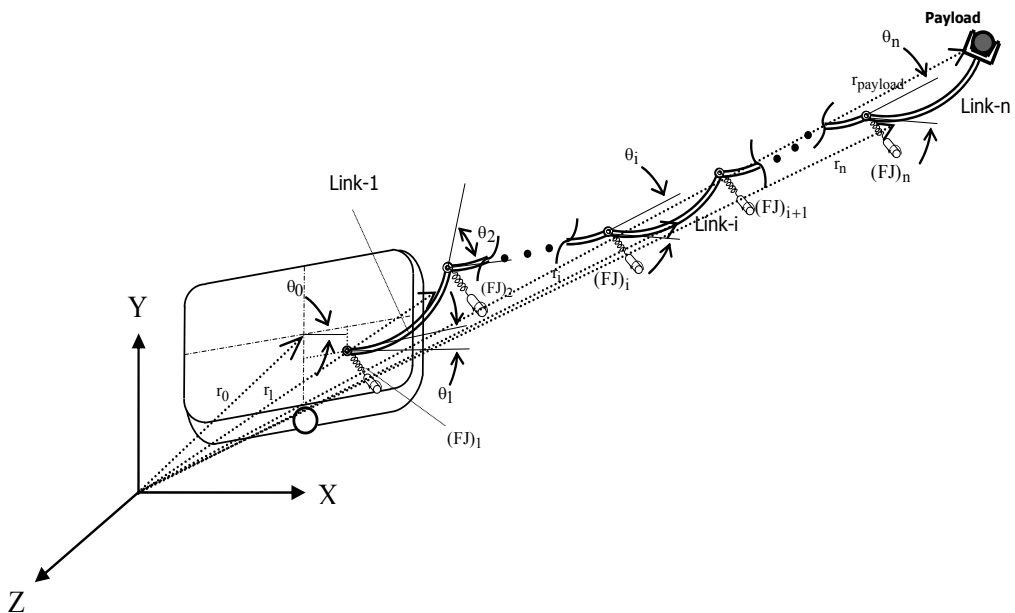


Fig. 1. A schematic view of a multiple flexible links - joints mobile manipulator

The following assumptions are made for the development of a dynamic model of the system.

- Each link is assumed to be long and slender.
- The motion of each link and its deformation is supposed to be in the horizontal plane.
- Links are considered to have constant cross-sectional area and uniform material properties.

- The inertia of payload is neglected.
- The backlash in the reduction gear and coulomb friction effects are neglected.
- It is assumed that the mobile base does not slide.

The generalized coordinates of the flexible links/joints mobile manipulator consist of four parts, the generalized coordinates defining the mobile base motion $\bar{q}_b = (q_{b1}, q_{b2}, \dots, q_{b_{m_b}})^T$, the generalized coordinates of the rigid body motion of links $\bar{q}_r = (q_1, q_2, \dots, q_n)^T$ and the generalized coordinates that related to the flexibility of the links $\bar{q}_f = (q_{11}, q_{12}, \dots, q_{1n_f}, q_{21}, \dots, q_{2n_f}, \dots, q_{n1}, \dots, q_{n_{m_f}})^T$, and the generalized coordinate corresponding to the flexibility of joints $\bar{q}_j = (q_{1+n}, q_{2+n}, \dots, q_{n+n})^T$. Where n , n_b and n_f are number of links, base degrees of freedom and manipulator mode shapes, respectively.

The notion of redundancy expresses that the number of generalized coordinates (v) is strictly greater than the (global) degree of freedom (d). Thus, the mechanical system is redundant if $d < v$; and the order of redundancy is $v-d$. Hence, it is comprehensible that in most mobile manipulator systems $v = n + n_b$ is greater than the end effector degree of freedom in the work space (d). Accordingly, these systems usually are subjected to some non-integrable kinematic constraints known as non-holonomic constraints. There are different techniques, which can be applied to a robotic system to solve the redundancy resolution. Some of these techniques are based on an optimization criterion such as overall torque minimization, minimum joint motion and so on. Hence, Seraji has used r additional user-defined kinematic constraint equations as a function of the motion variables (Seraji, 1998). This method results in a simple and online coordination of the control of a mobile manipulator during motion. The presenting study follows this method. Hence, some additional suitable kinematic constraint equations to the system dynamics are applied. Results are in simple and on-line coordination of the mobile manipulator during the motion. These constraints undertake the robot movement only in the direction normal to the axis of the driving wheels along with previously specification of the base trajectory during the motion.

2.2 Dynamic modelling

2.2.1 Dynamic modelling of flexible link manipulator

The original dynamics of robotic manipulators with elastic arms, being described by nonlinear coupled partial differential equations. They are continuous nonlinear dynamical systems distinguished by an infinite number of degrees of freedom. The exact solution of such systems does not exist. However, most commonly the dynamic equations are truncated to some finite dimensional models with either the assumed modes method (AMM) or the finite element method (FEM).

2.2.1.1 Assumed mode method

A large number of researchers use assumed modes of vibration to model robot dynamic in order to capture the interaction between flexural vibrations and nonlinear dynamics. In the assumed modes method, the dynamic model of the robot manipulator is described by a set of vibration modes other than its natural modes. Using assumed modes to model flexibility requires Euler-Bernoulli beam theory boundary conditions and accommodates changes in configuration during operation, whereas natural modes must be continually recomputed.

According to this method an approximate deflection of any continuous elastic beam subjected to transverse vibrations, can be expressed through truncated modal expansion, under the planar small deflection assumption of the link as

$$v_i(x_i, t) = \sum_{j=1}^{n_i} \varphi_{ij}(x_i) e_{ij}(t) \quad i = 1, \dots, n \quad (1)$$

where $v_i(x_i, t)$ is the bending deflection of the i^{th} link at a spatial point $x_i (0 \leq x_i \leq L_i)$ and L_i is the length of the i^{th} link. n_i is the number of modes used to describe the deflection of link i ; $\varphi_{ij}(x_i)$ and $e_{ij}(t)$ are the j^{th} assumed mode shape function and j^{th} modal displacement for the i^{th} link, respectively. Position and velocity of each point on link i can be obtained with respect to inertial coordinate frame using the transformation matrices between the rigid and flexible coordinate systems.

In the AMM there are numerous ways to choose the boundary conditions. The presenting study addresses four well-known conditions and chooses them with one mode shape per each link in the numerical simulations.

Ideally, the optimum set of assumed modes is that closest to natural modes of the system. Hence, there is no stipulation as to which set of assumed modes should be used. Natural modes depend on several factors such as size of hub inertia and size of payload mass. Choosing appropriate conditions is very important and it may cause better consequences in the results. Hence, the ultimate choice requires an assessment based on the actual robot structure and for example, anticipated range of payloads together with its natural modes. First four normal modes for some familiar mode conditions are described as following: Clamped-free mode shapes are given by

$$\begin{aligned} \varphi_i(x_i) &= \sin(B_i \cdot x_i) - \sinh(B_i \cdot x_i) + A(\cos(B_i \cdot x_i) - \cosh(B_i \cdot x_i)) \\ \text{where} \\ A_i &= \frac{\cos(B_i \cdot L_i) + \cosh(B_i \cdot L_i)}{\sin(B_i \cdot L_i) - \sinh(B_i \cdot L_i)} \\ B_i \cdot L_i &: 1.87 \quad 4.69 \quad 7.85 \quad 10.99 \dots \end{aligned} \quad (2)$$

Also, clamped – clamped mode shapes are determined as

$$\begin{aligned} \varphi_i(x_i) &= \sin(B_i \cdot x_i) - \sinh(B_i \cdot x_i) + A(\cos(B_i \cdot x_i) - \cosh(B_i \cdot x_i)) \\ \text{where} \\ A_i &= \frac{\cos(B_i \cdot L_i) - \cosh(B_i \cdot L_i)}{\sin(B_i \cdot L_i) + \sinh(B_i \cdot L_i)} \\ B_i \cdot L_i &: 4.73 \quad 7.85 \quad 10.99 \quad 14.13. \end{aligned} \quad (3)$$

In addition, mode shape functions with clamped-pinned boundary conditions are given by

$$\begin{aligned} \varphi_i(x_i) &= \sin(B_i \cdot x_i) - \sinh(B_i \cdot x_i) + A(\cos(B_i \cdot x_i) - \cosh(B_i \cdot x_i)) \\ \text{where} \\ A_i &= -\frac{\sin(B_i \cdot L_i) + \sinh(B_i \cdot L_i)}{\cos(B_i \cdot L_i) + \cosh(B_i \cdot L_i)} \\ B_i \cdot L_i &: 3.92 \quad 7.06 \quad 10.21 \quad 13.35. \end{aligned} \quad (4)$$

Similarly, this theory determines pinned–pinned mode shapes as:

$$\begin{aligned}\varphi_i(x_i) &= A_i \sin(B_i \cdot x_i) \\ A_i &= \frac{\cosh(B_i \cdot L_i)}{\cos(B_i \cdot L_i)} \\ B_i \cdot L_i &: 3.14 \quad 6.28 \quad 9.42 \quad 12.56.\end{aligned}\tag{5}$$

Choosing the appropriate set of assumed modes as a boundary condition may be quite valuable for robot to fit in a suitable application. Ideally, the optimum set of assumed modes is that closest to natural modes of the system. Natural modes depend on several factors within the robotic system ensemble including size of hub inertia and size of payload mass. For large joint gearing inertia and relatively small payload mass, the link may be considered clamped at the joint. Conversely, for smaller joint gearing inertia and larger payload mass both ends of the link may be considered pinned. The ultimate choice requires an assessment based on the actual robot structure and anticipated range of payloads together with its natural modes.

Although assume mode method has been widely used, there are several ways to choose link boundary conditions and mode eigen-functions. This drawback may increase drastically when finding modes for links with non-regular cross sections and multi-link manipulators is objected. In addition, using the AMM to derive the equations of motion of the flexible manipulators, only the first several modes are usually retained by truncation and the higher modes are neglected.

2.2.1.2 Finite element method

The finite element method is broadly used to derive dynamic equations of elastic robotic arms. Researcher usually used the Euler-Bernoulli beam element with multiple nodes and Lagrange shape function to achieve the reasonable finite element model. The node number can be selected according to requirement on precision. But, increasing the node number may enlarge the stiffness matrix and it cause to long and complex equations. Hence, choosing the proper node number is very important in the finite element analyzing.

The overall finite element approach involves treating each link of the manipulator as an assemblage of n elements of length L_i . For each of these elements the kinetic energy T_{ij} and potential energy V_{ij} , are computed in terms of a selected system of generalized coordinate q and their rate of change \dot{q} . Note that subscript ij refer to the j^{th} element of link i .

In summary the kinetic energy T_{ij} and potential energy V_{ij} are computed by the following equation:

$$T_{ij} = \frac{1}{2} \int_0^{L_i} m_i \left[\frac{\partial r_i^T}{\partial t} \cdot \frac{\partial r_i}{\partial t} \right] dx_{ij}\tag{6}$$

And

$$\begin{aligned}V_{ij} &= V_{gij} + V_{eij} \\ &= \int_0^{L_i} m_i g [0 \quad 1] T_0^1 \begin{bmatrix} (j-1)L_i + x_{ij} \\ y_{ij} \end{bmatrix} dx_{ij} + \frac{1}{2} \int_0^{L_i} EI_i \left[\frac{\partial^2 y_{ij}}{\partial x_{ij}^2} \right]^2 dx_{ij}\end{aligned}\tag{7}$$

In above equation, the potential energy is consisted of two parts. One part is due to gravity (V_{gij}) and another is related to elasticity of links (V_{eij}). r_i , m_i , L_i and EI_i are the position, mass, length and the flexural rigidity of i^{th} element respectively. x_{ij} and y_{ij} are specified the distances along body- fixed system $O_{ij}X_{ij}Y_{ij}$ from common junction between elements ' $i(j-1)$ '

and 'ij' of link i. $T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix}$ is transformation matrix from body-fixed system

attached to link 1 to inertial system of coordinates and θ_1 is it's correlated joint angle. These energies of elements are then combined to obtain the total kinetic energy T , and potential energy V , for the each link. Knowledge of the kinetic and potential energies is tantamount to specify the Lagrangian \mathcal{L} of the system, given by $\mathcal{L}=T-V$. Using of finite element method in modelling of the robotics system are details in (Usoro, 1986).

As it can be seen, modelling of flexural vibrations of robotic elements using finite element is a well-established technique. So, researchers can handle nonlinear conditions with this method. However, in order to solve a large set of differential equations derived by the finite element method, a lot of boundary conditions have to be considered, which are, in most situations, uncertain for flexible manipulators. Also, although significant advantages of FEM over analytical solution techniques such as easy to handle with which nonlinear conditions, this approach seems more complex over AMM. The main reason is that use of the finite element model to approximate flexibility usually gives rise to an overestimated stiffness matrix. Moreover, because of the large number of equations, the numerical simulation time may be exhausting for the finite element models.

2.2.1.3 Numerical simulations

The dynamic equations of the flexible robotic arms are verified in this section by undertaking a computer simulation. Hence, the case of harmonic motion of a nonlinear model of flexible robotic arms is selected to simulation. In this simulation, the robot is hanged freely and it influenced only under gravity effect. The physical parameters of the system used in this simulation study were $L_1 = L_2 = 1 \text{ m}$, $I_1 = I_2 = 5 \times 10^{-9} \text{ m}^4$, $m_1 = m_2 = 5 \text{ kg}$ and $E_1 = E_2 = 2 \times 10^{11} \text{ N / m}^2$. Simulating both FEM and AMM (pinned-pinned and clamped-pinned) models and comparing them with the rigid links in this simulation shows the oscillatory behavior of nonlinear robotic system advisably.

Now, considering the equations describe in the last section for FEM and AMM, also, using Lagrangian formulation, the set of equation of motion for each method is derived in compact form as

$$M(q)\ddot{q} + H(q, \dot{q}) = U \quad (8)$$

where M is the inertia matrix, H is the vector of Coriolis and centrifugal forces in addition to the gravity effects vector and U is the generalized force vector inserted into the actuator.

Open loop system response of changing the initial condition from normal equilibrium position to the relative angle between the first and second link of this system (θ_2) to the deviation of 5 degree is studied in this simulation (Fig. 2).

The responses of the system are presented in Figs. 3-5. Figures show the difference between rigid and flexible robotic arms also between the FEM and AMM with both pinned- pinned and clamped- pinned boundary conditions.

Figs. 3 and 4 show the angular positions and angular velocities of joints. It is obvious from figures that the link elasticity appears in velocity graph more and more than the position graph. Also, these figures restate the issue that the FEM model displays the nonlinearity of the system properly.

The corresponding amplitudes of vibration modes in the AMM are shown in Fig. 5. It is clear that link flexibility significantly affects the link vibrations. In addition, pictures shows that these effects are appeared more when clamped - pinned boundary condition is

considered. Figures are plotted in this section clearly show a good agreement between the obtained results in this study and those presented in (Usoro, 1986).

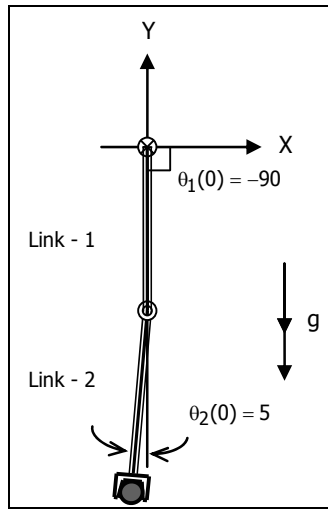


Fig. 2. Initial robot configuration

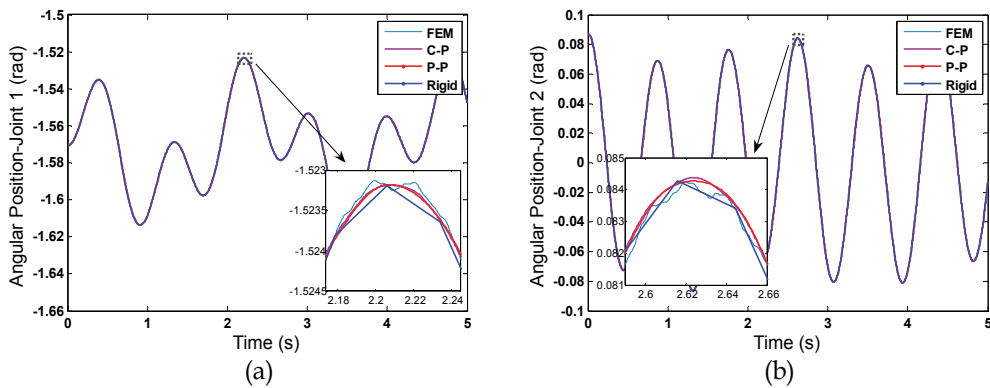


Fig. 3. Angular position of joints: (a) joint 1; (b) joint 2.

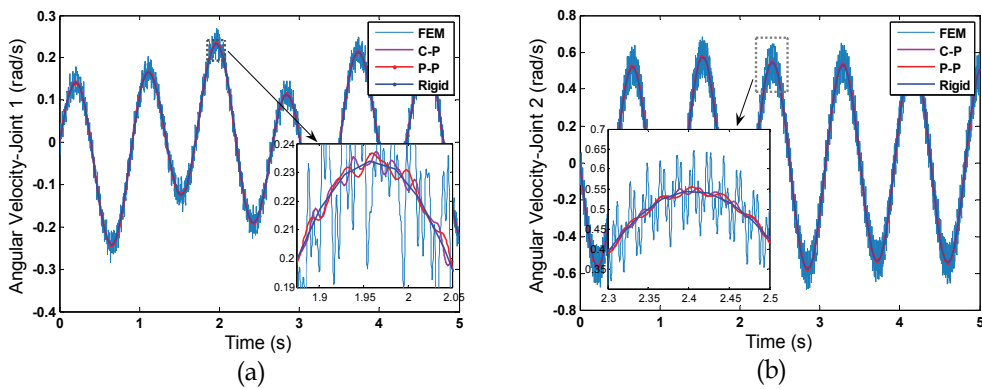


Fig. 4. Angular velocity of joints: (a) joint 1; (b) joint 2.

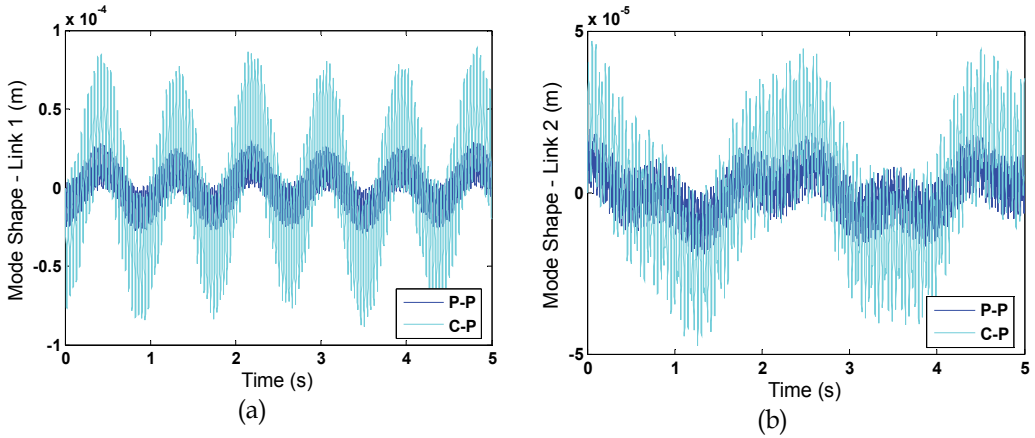


Fig. 5. Amplitudes of vibration's modes: (a) Link 1; (b) Link 2

2.2.2 Dynamic modelling of flexible joint manipulator

To model a flexible joint manipulator (FJM) the link positions are let to be in the state vector as is the case with rigid manipulators. Actuator positions must be also considered because in contradiction to rigid robots these are related to the link position through the dynamics of the flexible element. By defining the link number of a flexible joint manipulator is m , position of the i^{th} link is shown with $\theta_{2i-1} : i = 1, 2, \dots, m$ and the position of the i^{th} actuator with $\theta_{2i} : i = 1, 2, \dots, m$, it is usual in the FJM literature to arrange these angles in a vector as follows:

$$Q = [\theta_1, \theta_3, \dots, \theta_{2m-1} | \theta_2, \theta_4, \dots, \theta_{2m}]^T = [q_1^T, q_2^T]^T \quad (9)$$

So by adding the joint flexibility with considering the elastic mechanical coupling between the i^{th} joint and link is modeled as a linear torsional spring with constant stiffness coefficient k_i , the set of equation of motion comprising mobile base with both link and joint flexibility can be rearranged into the following form:

$$\begin{aligned} M(q_1)\ddot{q}_1 + H(q_1, \dot{q}_1) + G(q_1) + K(q_1 - q_2) &= 0 \\ J\ddot{q}_2 + K(q_2 - q_1) &= U \end{aligned} \quad (10)$$

where $K = \text{diag}[k_1, k_2, \dots, k_m]$ is a diagonal stiffness matrix which models the joint elasticity, $J = \text{diag}[J_1, J_2, \dots, J_m]$ is the diagonal matrix representing motor inertia.

A simulation is performed to investigate the effect of joint flexibility on the response of model by adding the elasticity at each joint as a linear spring. The case study with clamped-pinned boundary condition is modeled for that issue. Simulation is done at the overall time 5 seconds. Parameter values of joints are $k_1 = k_2 = 1500 \text{ N.m}$ and $J_1 = J_2 = 2 \text{ kg.m}^2$.

As shown in Fig. 6 the joint flexibility has considerable consequences on the robot behavior and link parameters have significant deviations from rotor's one. Hence, it can be conclude that the joint flexibility, considerably influences the performance of robotic arms and it can

be as a significant source of nonlinearity and system’s oscillatory behavior. Therefore, it is recommended that to improve the performance of the robotic systems, joint flexibility taken into account in modelling and control of such systems.

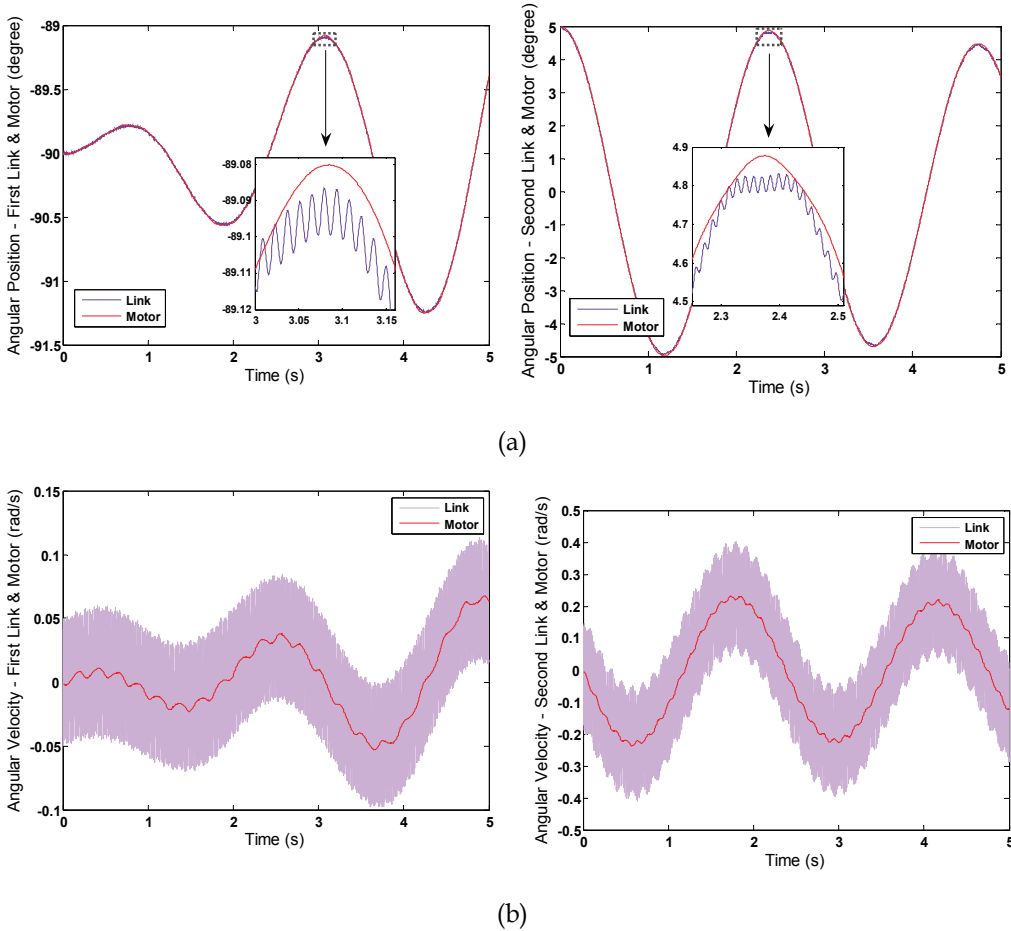


Fig. 6. Effect of joint flexibility in (a) Position and (b) Velocity of joints

2.2.4 Dynamic modelling of mobile manipulator

Consider an n DOFs rigid mobile manipulator with generalized coordinates $q = [q_i]$, $i = 1, 2, \dots, n$ and a task described by m task coordinates r_j , $j = 1, 2, \dots, m$ with $m < n$. By applying h holonomic constraints and c non-holonomic constraints to the system, $r = h + c$ redundant DOFs of the system can be directly determined. Therefore m DOFs of the system is remained to accomplish the desired task. As a result, we can decomposed the generalized coordinate vector as $q = [q_r \ q_{nr}]^T$, where q_r is the redundant generalized coordinate vector determined by applying constraints and q_{nr} is the non-redundant generalized coordinate vector. By considering the flexible link manipulators instead of the rigid ones, their related generalized coordinates, q_f , are added to the system; therefore, the overall decomposed generalized coordinate vector of system obtain as $q = [q_r \ q_{nrf}]^T$, where q_{nrf} is the combination vector of q_{nr} and q_f .

The system dynamics can also be decomposed into two parts: one is corresponding to redundant set of variables, q_r and the remained set of them, q_{nrf} . That is,

$$\begin{bmatrix} M_{r,r} & M_{r,nrf} \\ M_{r,nrf} & M_{nrf,nrf} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_{nrf} \end{bmatrix} + \begin{bmatrix} C_r + G_r \\ C_{nrf} + G_{nrf} \end{bmatrix} = \begin{bmatrix} U_r \\ U_{nrf} \end{bmatrix} \quad (11)$$

where by considering the second row in order to path optimization procedure leads to

$$U_{nrf} = A\ddot{q}_{nrf} + B. \quad (12)$$

Using redundancy resolution q_r will be obtained as a known vector in terms of the time (t). Therefore A is obtained as a function of time and q_{nrf} and B as a function of time, q_r and \dot{q}_{nrf} .

By defining the state vector as

$$X = [X_1 \quad X_2]^T = [q_{nrf} \quad \dot{q}_{nrf}]^T, \quad (13)$$

Eq. (5) can be rewritten in state space form as

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} X_2 \\ N(X) + D(X)U \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \quad (14)$$

where $D = M^{-1}$ and $N = -M^{-1}(C(X_1, X_2) + G(X_1))$. Then, optimal control problem is determined the position and velocity variable $X_1(t)$ and $X_2(t)$, and the joint torque U (t) which optimize a well-defined performance measure when the model is given in Eq. (14).

3. Optimal control

3.1 Defining the optimal control problem

Pontryagin's minimum principle provides an excellent tool to calculate optimal trajectories by deriving a two-point boundary value problem. Let the trajectory generation problem be defined here as determining a feasible specification of motion, which will cause the robot to move from a given initial state to a given final state. The method presented in this article adapts in a straightforward manner to the generation of such dynamic profiles.

There are known that nonlinear system dynamics stated as Eq. (14) be expressed in the term of states (X), controls (U) and time (t) as

$$\dot{X} = f(X, U, t) \quad (15)$$

Generating optimal movements can be achieved by minimizing a variety of quantities involving directly or not some dynamic capacities of the mechanical system. A functional is considered as the integral

$$J(u) = \int_{t_0}^{t_f} L(X, U, t) dt \quad (16)$$

where the function L may be specified in quite varied manners. There are initial and terminal constraints on the states:

$$X(t_0) = X_0 \quad X(t_f) = X_f \quad (17)$$

There may also be certain pragmatic constraints (reflecting such concerns as limited actuator power) on the inputs. For example:

$$|U(t)| \leq U_{\max}(t) \quad (18)$$

According to the minimum principle of Pontryagin (Kirk, 1970), minimization of performance criterion at Eq. (16), is achieved by minimizing the Hamiltonian (H) which is defined as follow:

$$H(X, U, \Psi, m_p, t) = L(X, U, m_p, t) + \Psi^T f(X, U, t) \quad (19)$$

where $\Psi(t) = [\psi_1(t)^T \quad \psi_2(t)^T]^T$ is the nonzero costate time vector-function.

Finally, according to the aforementioned principle, stating the costate vector-equation

$$\dot{\Psi}^T = -\partial H / \partial X \quad (20)$$

in addition to the minimality condition for the Hamiltonian as

$$\partial H / \partial U = 0 \quad (21)$$

$$\dot{X} = \partial H / \partial \Psi, \quad (22)$$

leads to transform the problem of optimal control into a non-linear multi-point boundary value problem.

Consequently, for a specified payload value, substituting obtained computed control equations from Eqs. (21) and Eq. (18) into Eqs. (20) and (22), sixteen nonlinear ordinary differential equations are obtained which with sixteen boundary conditions given in Eq. (17), constructs a Two Point Boundary Value Problem (TPBVP). Such a problem is solvable with available commands in different software such as MATLAB and MATEMATICA.

3.2 Application

3.2.1 Developing for two-link flexible mobile manipulator

3.2.1.1 Equations of motion

In this section, a mobile manipulator consists of a mobile platform with two flexible links / joints manipulator as depicted in Fig. 7 is considered to analysis. For study on the complete model, first, a mobile manipulator with two flexible links is considered to derive the dynamic equations, then, with applying the joint flexibility by modelling the elasticity at each joint as a linear torsional spring the model is developed for integrated link and joint flexible mobile manipulator.

To model the equations of motion of the system, assumed mode method is used. For this purpose, the total energy associated with the system must be computed to determine the Lagrangian function.

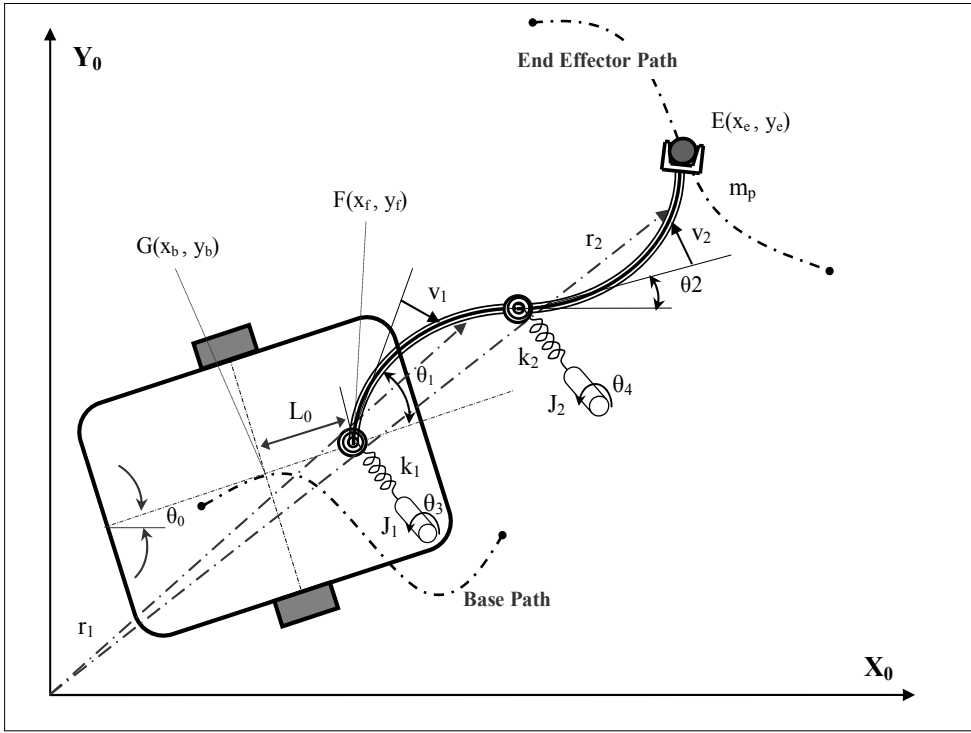


Fig. 7. Two links mobile manipulator with flexible links and joints

The total kinetic energy of the system (T) is given by

$$T = T_L + T_B + T_M, \quad (23)$$

The kinetic energy of flexible links can be found as

$$T_L = \sum_{i=1}^2 \frac{1}{2} \rho_i \int_0^{L_i} \dot{r}_i^T(x_i) \dot{r}_i(x_i) dx_i, \quad (24)$$

where r_i is the position vector that describes an arbitrary point along the i^{th} deflected link with respect to the global co-ordinate frame $(X_0 Y_0)$ and ρ_i is the linear mass density for the i^{th} link.

By defining r_b and r_m as position vectors of the base and the payload respectively, the associated kinetic energies are obtained as:

$$\begin{aligned} T_M &= \frac{1}{2} m_p \dot{r}_m^2 \\ T_B &= \frac{1}{2} m_b \dot{r}_b^2 + \frac{1}{2} I_b \dot{\omega}_b^2 \end{aligned} \quad (25)$$

where I_b and $\dot{\omega}_b$ are the moment of inertia and the angular velocity of base, respectively. Note that the moment of inertia of the end effector has been neglected.

Next, the potential energy associated with the flexibility of the links due to the link deformation is obtained as:

$$U_L = \sum_{i=1}^2 \frac{1}{2} \int_0^{L_i} (EI)_i \left(\frac{d^2 v_i}{dx_i^2} \right) dx_i, \quad (26)$$

where $(EI)_i$ is the flexural rigidity of the i^{th} link and $v_i(x_i, t)$ is the bending deflection of the i^{th} link at a point x_i , ($0 \leq x_i \leq L_i$). Now, by determining the gravity energy as:

$$U_g = \sum_{i=1}^2 \int_0^{L_i} \rho_i g x_i dx_i, \quad (27)$$

and adding this energy to those obtained in Eq. (26) the total potential energy of the system is obtained as $U = U_L + U_g$. Finally, by constructing the Lagrangian as $L = T - U$ and using the Lagrangian equation, the equations of motion for two-link flexible mobile manipulator can be obtained as Eq. (8). Hence, the overall generalized co-ordinate vector of the system can be written as: $q = [q_b \ q_r \ q_f] = [x_f \ y_f \ \theta_0 \ \theta_1 \ \theta_2 \ e_1 \ e_2]$, where $q_b = [x_f \ y_f \ \theta_0]$ is the base generalized coordinates vector, $q_r = [\theta_1 \ \theta_2]$ is the link angles vector and $q_f = [e_1 \ e_2]$ is the vector of link modal displacements.

There is one nonholonomic constraint for the mobile base that undertakes the robot movement only in the direction normal to the axis of the driving wheels:

$$\dot{x}_f \sin(\theta_0) - \dot{y}_f \cos(\theta_0) + L_0 \dot{\theta}_0 = 0. \quad (28)$$

Now, by predefining the base trajectory, the system dynamics can be decomposed into two parts: one is corresponding to redundant set of variables, q_r and the remained set of them, q_{nrf} . That is

$$\begin{bmatrix} M_{r,r} & M_{r,nrf} \\ M_{r,nrf} & M_{nrf,nrf} \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \ddot{q}_{nrf} \end{bmatrix} + \begin{bmatrix} H_r \\ H_{nrf} \end{bmatrix} = \begin{bmatrix} U_r \\ U_{nrf} \end{bmatrix}, \quad (29)$$

Now, by remaining the second row of above equation, the non-redundant part of system equations is considered to path optimization procedure.

For developing the system to encounter the flexible joints manipulator, adding the actuator positions and their dynamic equations is required. Hence, the set of system dynamic equation is rearranged as explain in Eq. (10). This overall system is clearly established the equations that involve the flexible nature of both links and joints.

$$\left\{ \begin{array}{l} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ & m_{22} & m_{23} & m_{24} \\ & & m_{33} & m_{34} \\ Sym & & & m_{44} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 - \theta_3 \\ \theta_2 - \theta_4 \\ 0 \\ 0 \end{bmatrix} = 0 \\ \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_3 - \theta_1 \\ \theta_4 - \theta_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \end{array} \right. \quad (30)$$

These enhanced dynamic equations that involve dynamic of the two-link flexible mobile manipulator are considered in trajectory planning problem in the presenting study.

3.2.1.2 Stating an optimal control solution

Optimal control approach provides an excellent tool to calculate optimal trajectory with high accuracy for robots that include, in this case, two link flexible mobile manipulators.

Let the trajectory generation problem be defined here as determining a feasible specification of motion which will cause the robot to move from a given initial posture (state) to a given final posture (state) while minimize a performance criterion such as integral quadratic norm of actuating torques or velocities, which leads to minimize energy consumption or bounding the velocity magnitude.

For this reason, as it can be seen in Fig. 7 the state vectors can be defined as:

$$\begin{aligned} X_1 &= \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} & X_2 &= \begin{bmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_4(t) \end{bmatrix} \\ X_3 &= \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} x_5(t) \\ x_7(t) \end{bmatrix} & X_4 &= \begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} x_6(t) \\ x_8(t) \end{bmatrix} \\ X_5 &= \begin{bmatrix} \theta_3(t) \\ \theta_4(t) \end{bmatrix} = \begin{bmatrix} x_9(t) \\ x_{11}(t) \end{bmatrix} & X_6 &= \begin{bmatrix} \dot{\theta}_3(t) \\ \dot{\theta}_4(t) \end{bmatrix} = \begin{bmatrix} x_{10}(t) \\ x_{12}(t) \end{bmatrix}. \end{aligned} \quad (31)$$

where θ_1 and θ_2 are angular positions of links, e_1 and e_2 are links modal displacements, and θ_3 and θ_4 are angular positions of motors. The boundary condition can be expressed as:

$$\begin{aligned} x_1(0) = x_9(0) = x_{10}, \quad x_3(0) = x_{11}(0) = X_{30}; \\ x_1(f) = x_9(f) = X_{1f}, \quad x_3(f) = x_{11}(f) = X_{3f}; \end{aligned} \quad (32)$$

Other boundary conditions are assumed to be zero.

Now, with defining $Z_{4 \times 4} = M_{4 \times 4}^{-1}$ and $I_{2 \times 2} = J_{2 \times 2}^{-1}$ Eq. (30) can be rewritten in the compact form as:

$$\begin{aligned} \ddot{q}_1 &= Z(K(q_{21} - q_{11}) - H) = F_1 \\ \ddot{q}_2 &= I(U - K(q_{22} - q_{12})) = F_2, \end{aligned} \quad (33)$$

where $q_{21} = (x_1 \ x_3 \ 0 \ 0)$, $q_{11} = (x_9 \ x_{11} \ 0 \ 0)$, $q_{22} = (x_1 \ x_3)$, $q_{12} = (x_9 \ x_{11})$, and $U = (u_1 \ u_2)$. Remember that in this simulation the gravity effect is assumed to be zero.

Hence, by defining the vector F as: $F = [F_1 \ F_2] = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]$ the set of state space equations of system can be written as:

$$\begin{aligned} \dot{x}_{2i-1} &= x_{2i}; \\ \dot{x}_{2i} &= f_i; \quad i = 1 \dots 6. \end{aligned} \quad (34)$$

In order to derive the equations associated with optimality conditions, penalty matrices can be selected as follows:

$$\begin{aligned} W &= \text{diag}(w_1, w_2, w_3, w_4, w_5, w_6) \\ R &= \text{diag}(r_1, r_2). \end{aligned} \quad (35)$$

An important remark must be done here is that the study is planned a trajectory in the joint space rather than in the operating space. It means the control system acts on the manipulator joints rather than on the end effector. Trajectory planning in the joint space would allow avoiding the problems arising with kinematic singularities and manipulator redundancy. Moreover, it would be easier to adjust the trajectory according to the design requirements if working in the joint space. By controlling manipulator joints can achieve the best dynamic coordination of joint motions, while minimizing the actuating inputs together with bounding the velocity magnitudes. It causes to ensure soft and efficient functioning while improving the manipulator working performances. For that reason, the objective function is formed as:

$$L = \frac{1}{2} \left(r_1 u_1^2 + r_2 u_2^2 + \sum_{i=1}^6 w_i x_{2i}^2 \right). \quad (36)$$

Subsequently, with defining the auxiliary costate vector as: $\Psi = [\psi_1 \ \psi_2 \ \dots \ \psi_{12}] = [x_{13} \ x_{14} \ \dots \ x_{24}]$ results to the Hamiltonian function as:

$$H = \frac{1}{2} \left(r_1 u_1^2 + r_2 u_2^2 + \sum_{i=1}^6 w_i x_{2i}^2 \right) + \sum_{i=1}^{12} x_{12+i} \dot{x}_i, \quad (37)$$

Consequently, with differentiating the Hamiltonian function with respect to states, the costate equations are obtained as follow

$$\dot{\psi}_i = -\frac{\partial H}{\partial x_i}, \quad i = 1, \dots, 12. \quad (38)$$

Also, differentiating the Hamiltonian with respect to control and setting the derivative equal to zero, yields the following control equations:

$$\frac{\partial H}{\partial u_1} = r_1 u_1 + x_{23}/J_1 = 0 \quad ; \quad \frac{\partial H}{\partial u_2} = r_2 u_2 + x_{24}/J_2 = 0 \quad (39)$$

where by solving them, the expression for control values in the admissible interval, $u_i^- < u_i < u_i^+$; $i = 1, 2$ can be obtained as follow:

$$u_1 = -x_{23}/(r_1 J_1) \quad ; \quad u_2 = -x_{24}/(r_2 J_2). \quad (40)$$

Then, by considering the constraint on control input, the optimal control can be expressed as follows:

$$u_1 = \begin{cases} u_1^+ & -x_{23}/(r_1 J_1) \geq u_1^+ \\ -x_{23}/(r_1 J_1) & \text{otherwise} \\ u_1^- & -x_{23}/(r_1 J_1) \leq u_1^- \end{cases} \quad (41)$$

$$u_2 = \begin{cases} u_2^+ & -x_{24}/(r_2 J_2) \geq u_2^+ \\ -x_{24}/(r_2 J_2) & \text{otherwise} \\ u_2^- & -x_{24}/(r_2 J_2) \leq u_2^- \end{cases}$$

where the final bound of control for each motor is obtained as:

$$\begin{aligned} u_1^+ &= \tau_1 - S_1 x_{11} & ; & \quad u_1^- = -\tau_1 - S_1 x_{11} \\ u_2^+ &= \tau_2 - S_2 x_{12} & ; & \quad u_2^- = -\tau_2 - S_2 x_{12} \end{aligned} \quad (42)$$

where $S_i = (\tau_i / \omega_{mi})$, τ_i and ω_{mi} are the stall torque and maximum no-load speed of i^{th} motor respectively.

Finally, 24 nonlinear ordinary differential equations are obtained by substituting Eq.(33) into Eqs. (38) and (34), which with 24 boundary conditions given in Eq.(32) construct a two point boundary value problem (TPBVP).

There are numerous influential and efficient commands for solving such problems that are available in different software such as MATLAB, MATEMATICA or FORTRAN. These commands by employing capable methods such as finite difference, collocation and shooting method solve the problem. In this study, BVP4C command in MATLAB® which is based on the collocation method is used to solve the aforesaid problem. This numerical technique have been detailed by (Shampine et al.).

3.2.1.3 Required parameters

In all simulations the mobile base is initially at point ($x_{fi} = 0.5\text{m}$, $y_{fi} = 0.5\text{m}$, $\theta_{fi} = 0$) and moves along a straight-line path to final position ($x_{ff} = 1.5\text{m}$, $y_{ff} = 1\text{m}$). The necessary parameters used in the simulations are summarized in the Table 1.

Properties	Symbol	Value (Unit)
Length of Links	l	1(m)
Mass Density	ρ	6 (kg. m ⁻¹)
Flexural Rigidity	EI	100 (N.m ²)
Max. no Load Speed of Actuators	ω_s	6 (rad / s)
Actuator Stall Torque	τ_s	25 (N. m)
Moment of Inertia (Motor)	J	2 (kg. m ²)
Spring Constant	k	1000 Nm

Table 1. System parameters

Velocity at start and stop is considered to be zero. Other boundary conditions are assumed to be:

$$\begin{aligned} x_1(0) = x_9(0) = 120^\circ, \quad x_3(0) = x_{11}(0) = 90^\circ; \\ x_1(f) = x_9(f) = 30^\circ, \quad x_3(f) = x_{11}(f) = 30^\circ; \end{aligned} \quad (43)$$

Also, in all simulations, the penalty matrix of control efforts R assumes to be $R = \text{diag}[0.01]$. Note that in all simulations, the payload is calculated with the accuracy of 0.1 Kg.

3.2.2 Motion planning

3.2.2.1 Motion planning for different penalty matrixes

In the first case, effects of changing in performance index in the path planning problem are investigated. Hence, simulation is done for the different values of W and optimal paths for a given payload are obtained.

By considering penalty matrices as $W = (w, w, 0, 0, w, w)$ by zero the first path is determined. Other paths are drowning with scaling up the value of W as: 1, 100, and 1000. Note that in these simulations the penalty matrices refer to velocities of mode shapes are fixed in zero and the payload is assumed to be 1 Kg.

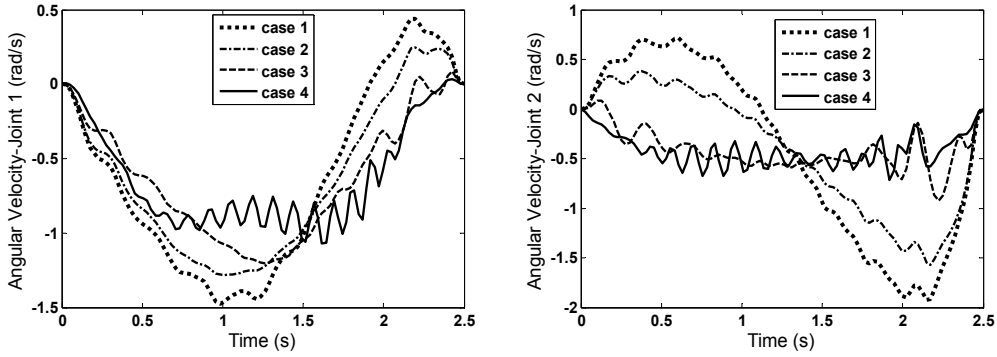


Fig. 8. Angular velocities of joints

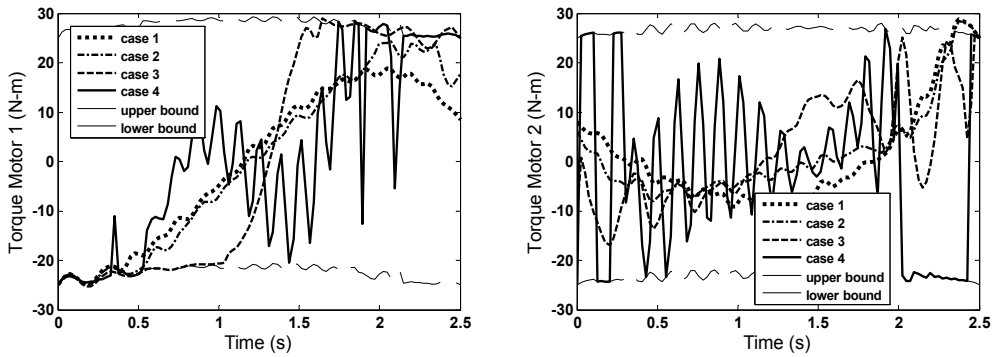


Fig. 9. Torques of motors

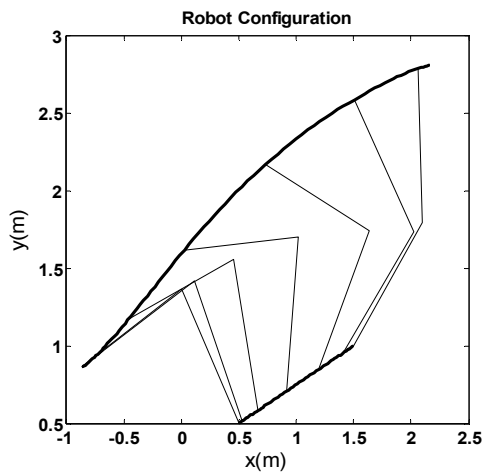


Fig. 10. Robot Configuration

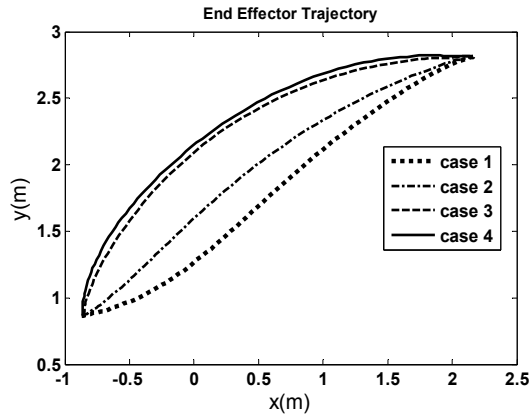


Fig. 11. End effector trajectory in XY plane

Fig. 8 shows the angular velocities of joints. The computed torques are plotted in Fig. 9. As shown in figures increasing W causes reducing the maximum velocity magnitude while the torques are growing. This issue is predictable, since, in the cost functional defined in the Eq. (16) increasing W causes to rise the role of velocity in path planning and it can decrease the proportion of R in such process. Furthermore, it can be found from figures, in order to attain a smoother path with smaller amount of velocity, more effort must be applied. Also, it is obvious that all the obtained graphs are satisfied the system cost function in Eq. (16). hence, they specify optimal trajectories of the system motion. Therefore, in the proposed method designer is able to choose most appropriate path among various optimal paths according to designing requirements. Robot configuration and end effector trajectory are depicted in Figs. 10 and 11 respectively.

3.2.2.2 Motion planning for different payloads

In this case W is assumed to be constant at $W=1$. Then, the robot path planning problem will be investigated by increasing the payload mass until maximum allowable load will be determined. This maximum payload is obtained as 8.4 kg (case 4). The obtained angular positions, angular velocities and torque curves graphs for a range of m_p given in Table 2 are shown in Figs. 12 - 14. It can be found that, increasing the m_p results to enlarge the velocity values as a consequence various optimal paths have been attained. As shown in figures, increasing the payload increases the required torque until the maximum payload. So that for the last case the torque curves lay on their limits. Hence, it is the most possible values of the torques and increasing the payload can lead to violate the boundary conditions. Finally, end effector trajectories in the Cartesian space are depicting in Fig. 15.

Case	1	2	3	4
m_p	1	3	7	8.4

Table 2. The values of m_p used in the simulation.

$m_{pmax} = 8.4$ kg is the maximum allowable payload for the selected penalty matrices while choosing the other penalty matrices, results in other optimal trajectories. To demonstrate that issue, simulations are carried out for different values of W given in Table 3.

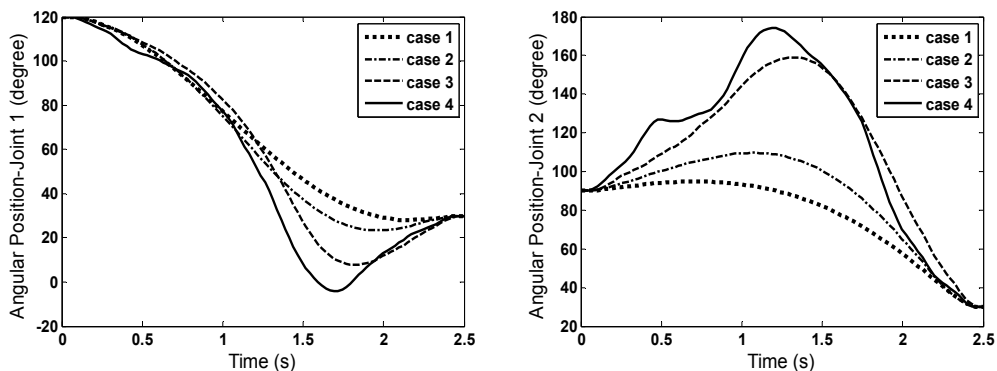


Fig. 12. Angular positions of joints

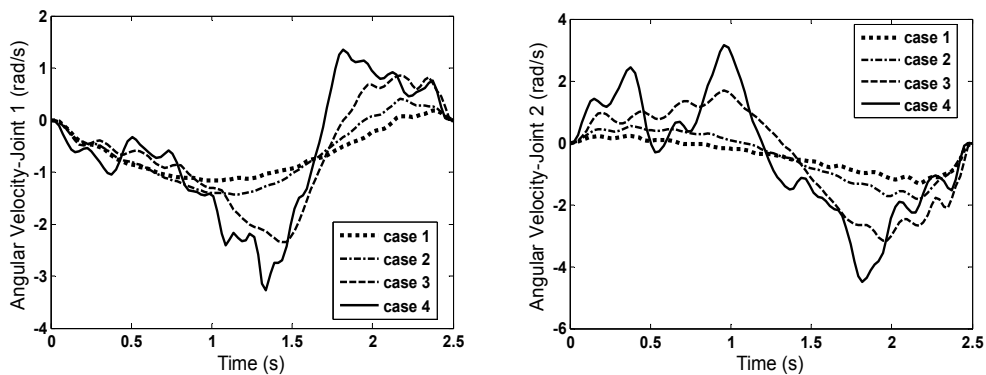


Fig. 13. Angular velocities of joints

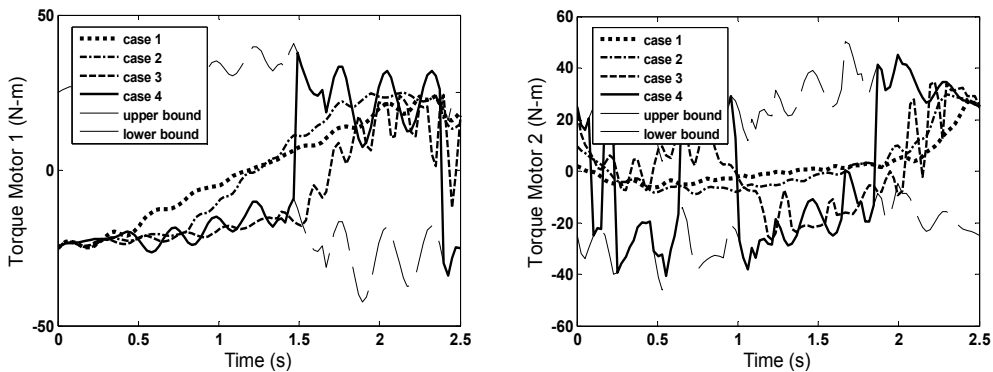


Fig. 14. Torques of motors

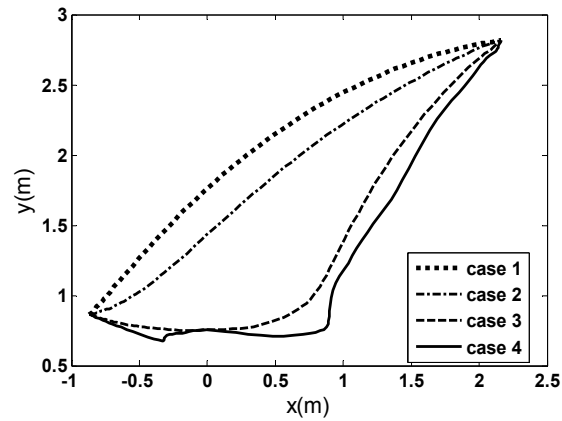


Fig. 15. End effector trajectory in XY plane

3.2.3 Maximum payload determination

In this case, the maximum payload of flexible mobile manipulator will be calculated and corresponding optimal trajectory at point-to-point motion will be illustrated for different values of W . Payload paths for these cases are shown in Fig. 16. Fig. 17 shows the robot configuration for the first and last cases. Also, the computed torques for these cases are plotted in Fig. 18. As it can be seen, increasing W causes to increase oscillatory behavior of the systems that results to reduce the maximum dynamic payload as shown in Table 3.

Case	1	2	3	4
W	1	400	600	800
m_{pmax}	8.4	7.9	7.5	6.3

Table 3. The values of W and corresponding calculated maximum payloads

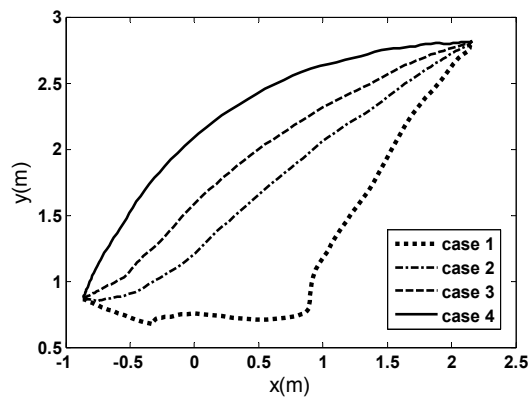


Fig. 16. End effector trajectory in XY plane

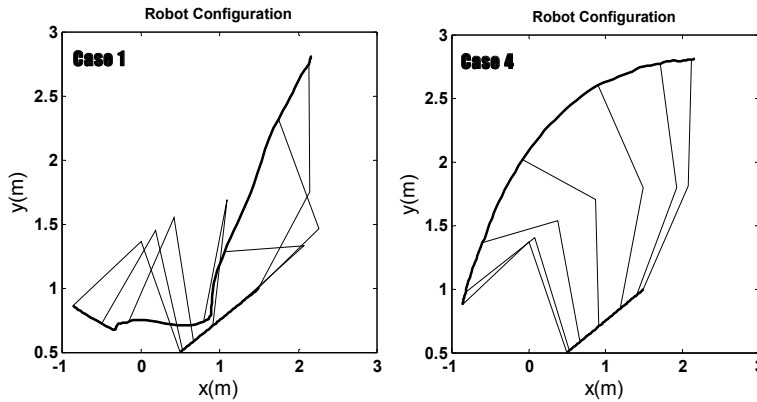


Fig. 17. Robot Configuration

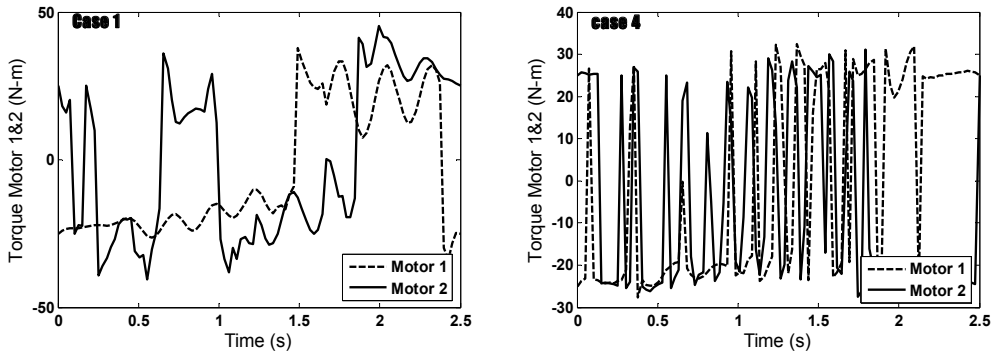


Fig. 18. Torques of motors

4. Conclusion

In this chapter, modelling and control of mechanical manipulator had been studied. First, kinematic and dynamic modelling of flexible link, flexible joint and mobile manipulators have been considered. Then, optimal control of a flexible mobile manipulator in point-to-point motion had been formulated based on the open-loop optimal control approach. The first objective of the chapter is to state the dynamic optimization problem under a quite generalized form in order to be applied to a variety of situations with any guess objective functions for the optimality solution. The second objective is consisting in developing the method for optimizing the applicable case studies, which results.

Using assumed mode and finite element methods oscillatory behavior of the mobile robotic manipulators had been described. The model equations had been verified for a two-link manipulator, and the model responses had been discussed. Then, joint flexibility had been added to the system and obtained model had been simulated. After that, an efficient solution on the basis of TPBVP solution had been proposed to path optimization – maximum payload determination in order to achieve the predefined objective. The solving strategy makes it possible to get any guess objective functions for the optimality solution. Attaining the minimum effort trajectory along with bounding the obtained velocity magnitude had been chosen at the application example. The obtained results illustrate the

power and efficiency of the method to overcome the highly nonlinearity nature of the optimization problem which with other methods, it may be very difficult or impossible.

Highlighting the main contribution of the chapter can be presented as:

- The proposed approach can be adapted to any general serial manipulator including both non-redundant and redundant systems with link flexibility and base mobility.
- In this approach the nonholonomic constraints do not appear in TPBVP directly, unlike the method given in (Mohri et al. 2001; Furuno et al. 2003).
- This approach allows completely nonlinear states and control constraints treated without any simplifications.
- The obtained results illustrate the power and efficiency of the method to overcome the high nonlinearity nature of the optimization problem, which with other methods, it may be very difficult or impossible.
- In this method, boundary conditions are satisfied exactly, while the results obtained by methods such as Iterative Linear Programming (ILP) have a considerable error in final time (Ghariblu & Korayem, 2006).
- In this method, designer is able to choose the most appropriate path among various optimal paths by considering the proper penalty matrices.

The optimal trajectory and corresponding input control obtained using this method can be used as a reference signal and feed forward command in the closed-loop control of such manipulators.

5. References

- Bertolazzi E.; Biral F. & Da Lio M. (2005). Symbolic-numeric indirect method for solving optimal control problems for large multibody systems, *Multibody System Dynamics*, Vol. 13, No. 2, pp. 233-252
- Bessonnet G. & Chessé S. (2005). Optimal dynamics of actuated kinematic chains, Part 2: Problem statements and computational aspects, *European J. of Mechanics A/Solids*, Vol. 24, pp. 472-490
- Bloch A. M. (2003). Nonholonomic mechanics and control. *Springer*, New York
- Bock H. G. & Plitt K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems, *Proc. 9th IFAC World Congress*, pp. 242-247
- Callies R. & Rentrop P. (2008). Optimal control of rigid-link manipulators by indirect methods, *GAMM-Mitt.*, Vol 31, No. 1, pp. 27 - 58
- Chen W. (2001). Dynamic modelling of multi-link flexible robotic manipulators, *Computers and Structures*, Vol. 79, (2), pp. 183-195
- Furuno S.; Yamamoto M. & Mohri A. (2003). Trajectory planning of mobile manipulator with stability considerations, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3403-3408
- Gariblu H. & Korayem M. H. (2006). Trajectory Optimization of Flexible Mobile Manipulators, *Robotica*, Vol. 24, No. 3, pp. 333-335
- Green A. & Sasiadek J.Z. (2004). Dynamics and Trajectory Tracking Control of a Two-Link Robot Manipulator, *Journal of Vibration and Control*, Vol. 10, No. 10, pp. 1415-1440
- Hargraves C. R. & Paris S. W. (1987). Direct trajectory optimization using nonlinear programming and collocation, *AIAA J. Guidance*, Vol. 10, No. 4, pp. 338-342, 1987.

- Korayem M. H. & Ghariblu H. (2004). Analysis of wheeled mobile flexible manipulator dynamic motions with maximum load carrying capacities, *Robotics and Autonomous Systems*, Vol. 48, No. 2-3, pp. 63-76
- Korayem M.H. & Rahimi Nohooji H. (2008). Trajectory optimization of flexible mobile manipulators using open-loop optimal control method, *LNAI*, Springer-Verlag Berlin Heidelberg, Vol. 5314, Part 1, pp. 54-63.
- Korayem M. H.; Haghpanahi M. ; Rahimi H. N. & Nikoobin A. (2009). Finite element method and optimal control theory for path planning of elastic manipulators, *New Advanced in Intelligent Decision Technology*, SCI, Springer-Verlag Berlin Heidelberg, Vol. 199, pp. 107-116
- Korayem M. H.; Rahimi H. N. & Nikoobin A. (2009). Analysis of Four Wheeled Flexible Joint Robotic Arms with Application on Optimal Motion Design, *New Advanced in Intelligent Decision Technology*, SCI, Springer-Verlag Berlin Heidelberg, Vol. 199, pp. 117-126
- Mohamed Z. & Tokhi M.O. (2004). Command shaping techniques for vibration control of a flexible robot manipulator, *Mechatronics*, Vol. 14, pp. 69-90.
- Mohri A.; Furuno S. & Yamamoto M. (2001). Trajectory planning of mobile manipulator with end-effector's specified path, *Proc. IEEE Int. Conf. on Intelligent Robots and systems*, pp. 2264-2269
- Papadopoulos E. & Rey, D. (1996). A New measure of tip over stability margin for mobile manipulators, *Proc. IEEE Int. Conference on Robotics and Automation*, pp. 3111-3116
- Papadopoulos E. & Gonthier, Y. (1999) A framework for large-force task planning of mobile redundant manipulators, *J. of Robotic Systems*, Vol. 16, No. 3, pp. 151-162
- Papadopoulos E.; Poulakakis I. & Papadimitriou I. (2002). On path planning and obstacle avoidance for nonholonomic platforms with manipulators: *A polynomial approach*, *Int. J. of Robotics Research*, Vol. 21, No. 4, pp. 367-383
- Rahimi H. N.; Korayem M. H. & Nikoobin A. (2009). Optimal Motion Planning of Manipulators with Elastic Links and Joints in Generalized Point-to-Point Task, *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, Vol. 7, Part B, 33rd Mechanisms and Robotics Conference, pp 1167-1174, San Diego, CA, USA
- Sentinella M. R. & Casalino L. (2006). Genetic algorithm and indirect method coupling for low-thrust trajectory optimization, *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, California
- Seraji H., "A unified approach to motion control of mobile manipulators, *Int. J. of Robotic Research*, Vol. 17, No. 12, pp.107-118 (1998).
- Shampine L. F.; Reichelt M. W. & Kierzenka J. Solving boundary value problems for ordinary differential equations in MATLAB with `bvp4c`, available at <http://www.mathworks.com/bvp tutorial>
- Sheng Ge Xin & Qun Chen Li. (2006). Optimal motion planning for nonholonomic systems using genetic algorithm with wavelet approximation, *Applied Mathematics and Computation*, Vol. 180, pp. 76-85
- Subudhi B. & Morris A.S. (2002). Dynamic Modelling, Simulation and Control of a Manipulator with Flexible Links and Joints, *Robotics and Autonomous Systems*, Vol. 41, pp. 257-270

- Usono P.B.; Nadira R., Mahil S. S. (1986). A finite element/Lagrange approach to modelling lightweight flexible manipulators, *J. of Dynamics Systems, Measurement, and Control*, Vol. 108, pp.198–205
- Wachter A. & Biegler L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming*, Vol. 106, No. 1, pp. 25–57
- Yue S., Tso S. K. & Xu W. L. (2001). Maximum dynamic payload trajectory for flexible robot manipulators with kinematic redundancy, *Mechanism and Machine Theory* 36, 785–800
- Zhang C. X. & Yu Y. Q. (2004). Dynamic analysis of planar cooperative manipulators with link flexibility, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 126, pp. 442–448

Hyper Redundant Manipulators

Ivanescu Mircea and Cojocaru Dorian
University of Craiova
Romania

1. Introduction

Modern industrial robots are mostly (human) arm-inspired mechanisms with serially arranged discrete links. When it comes to industrial environment where the workspace is structured and predefined this kind of structure is fine. This type of robots are placed in carefully controlled environments and kept away from human and their world.

When it comes to robots that must interact with the natural world, it needs to be able to solve the same problems that animals do. The rigid structure of traditional robots limit their ability to maneuver and in small spaces and congested environments, and to adapt to variations in their environmental contact conditions. For improving the adaptability and versatility of robots, recently there has been interest and research in “soft” robots. In particular, several research groups are investigating robots based on continuous body “continuum” structure. If a robot’s body is soft and/or continuously bendable it might emulate a snake or an eel with an undulating locomotion (Walker & Carreras, 2006).

An ideal tentacle manipulator is a non-conventional robotic arm with an infinite mobility. It has the capability of taking sophisticated shapes and of achieving any position and orientation in a 3D space. Behavior similar to biological trunks, tentacles, or snakes may be exhibited by continuum or hyper-redundant robot manipulators (Walker et al., 2005). Hence these manipulators are extremely dexterous, compliant, and are capable of dynamic adaptive manipulation in unstructured environments, continuum robot manipulators do not have rigid joints unlike traditional rigid-link robot manipulators. The movement of the continuum robot mechanisms is generated by bending continuously along their length to produce a sequence of smooth curves. This contrasts with discrete robot devices, which generate movement at independent joints separated by supporting links.

The snake-arm robots and elephant’s trunk robots are also described as continuum robots, although these descriptions are restrictive in their definitions and cannot be applied to all snake-arm robots (Hirose, 1993). A continuum robot is a continuously curving manipulator, much like the arm of an octopus (Cowan & Walker, 2008). An elephant’s trunk robot is a good descriptor of a continuum robot (Hutchinson, S.; Hager et al., 1996). The elephant’s trunk robot has been generally associated with an arm manipulation – an entire arm used to grasp and manipulate objects, the same way that an elephant would pick up a ball. As the best term for this class of robots has not been agreed upon, this is still an emerging issue. Snake-arm robots are often used in association with another device meant to introduce the snake-arm into the confined space.

However, the development of high-performance control algorithms for these manipulators is quite a challenge, due to their unique design and the high degree of uncertainty in their

dynamic models. The great number of parameters, theoretically an infinite one, makes very difficult the use of classical control methods and the conventional transducers for position and orientation." must be moved after the paragraph "An ideal tentacle manipulator is a non-conventional robotic arm with an infinite mobility. It has the capability of taking sophisticated shapes and of achieving any position and orientation in a 3D space. These systems are also known as hyper redundant manipulators and, over the past several years, there has been a rapid expanding interest in their study and construction.

An ideal tentacle manipulator is a non-conventional robotic arm with an infinite mobility. It has the capability of taking sophisticated shapes and of achieving any position and orientation in a 3D space. These systems are also known as hyper redundant manipulators and, over the past several years, there has been a rapid expanding interest in their study and construction.

The control of these systems is very complicated and a great number of researchers tried to offer solutions for this difficult problem. In (Hemami, 1984); (Suzumori et al., 1991) it analyses the control by cables or tendons meant to transmit forces to the elements of the arm in order to closely approximate the arm as a truly continuous backbone. Also, Mochiyama has investigated the problem of controlling the shape of an HDOF rigid-link robot with two-degree-of-freedom joints using spatial curves (Mochiyama & Kobayashi, 1999). Important results were obtained by Chirikjian (Chirikjian, 1993) who laid the foundations for the kinematic theory of hyper redundant robots. His results are based on a "backbone curve" that captures the robot's macroscopic geometric features.

The inverse kinematic problem is reduced to determining the time varying backbone curve behaviour (Takegaki & Arimoto, 1981). New methods for determining "optimal" hyper-redundant manipulator configurations based on a continuous formulation of kinematics are developed. In (Gravagne & Walker, 2001), Gravagne analysed the kinematic model of "hyper-redundant" robots, known as "continuum" robots. Robinson and Davies (Robinson & Davies, 1999) present the "state of art" of continuum robots, outline their areas of application and introduce some control issues. The great number of parameters, theoretically an infinite one, makes very difficult the use of classical control methods and the conventional transducers for position and orientation.

The lack of no discrete joints is a serious and difficult issue in the determination of the robot's shape. A solution for this problem is the vision based control of the robot, kinematics and dynamics.

The research group from the Faculty of Automation, Computers and Electronics, University of Craiova, Romania, started working in research field of hyper redundant robots over 25 years ago. The experiments started on a family of TEROB robots which used cables and DC motors. The kinematics and dynamics models, as well as the different control methods developed by the research group were tested on these robots. Starting with 2008, the research group designed a new experimental platform for hyper redundant robots. This new robot is actuated by stepper motors. The rotation of these motors rotates the cables which by correlated screwing and unscrewing of their ends determine their shortening or prolonging, and by consequence, the tentacle curvature (Blessing & Walker, 2004). Segments were cylindrical in the initial prototype, and cone-shaped in actual prototype. The backbone of the tentacle is an elastic cable made out of steel, which sustains the entire structure and allows the bending. Depending on which cable shortens or prolongs, the tentacle bends in different planes, each one making different angles (rotations) respective to the initial coordinate frame attached to the manipulator segment - i.e. allowing the movement in 3D. Due to the mechanical design it can be assumed that the individual cable torsion,

respectively entire manipulator torsion can be neglected. Even if these phenomena would appear, the structure control is not based on the stepper motors angles, but on the information given by the robotic vision system which is able to offer the real spatial positions and orientations of the tentacle segments.



Fig. 1. A tentacle arm prototype

2. Kinematics

In order to control a hyper-redundant robot it has to develop a method to compute the positions for each one of his segments (Immega & Antonelli, 1995). By consequence, given a desired curvature $S^*(x, t_f)$ as sequence of semi circles, identify how to move the structure, to obtain $s(x, t)$ such that

$$\lim_{t \rightarrow t_f} s(x, t) = S^*(x, t_f) \quad (1)$$

where x is the column vector of the shape description and t_f is the final time (see Fig. 2).

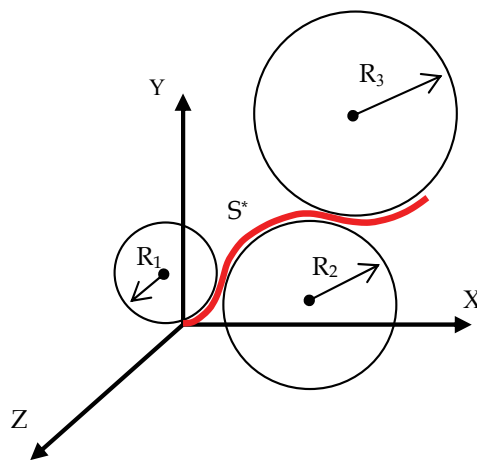


Fig. 2. The description of the desired shape

To describe the tentacle's shape we will consider two angles (α , θ) for each segment, where θ is the rotation angle around Z-axis and α is the rotation angle around the Y-axis (see Fig. 2). To describe the movement we can use the roto-translation matrix considering $\theta = 2\beta$ as shown in Fig. 3.

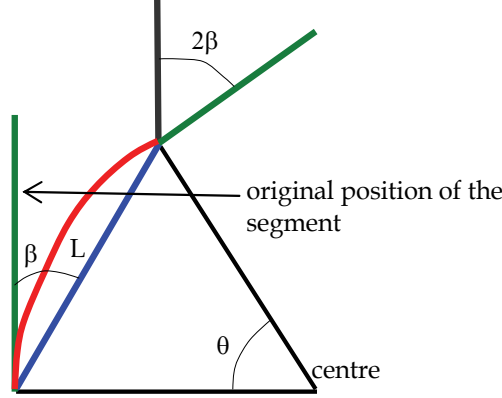


Fig. 3. Curvature and relation between θ and β

The generic matrix in 2D that expresses the coordinate of the next segment related to the previous reference system can be written as follow:

$$\begin{bmatrix} \cos(2 \cdot \beta) & \sin(2 \cdot \beta) & L \cdot \sin(\beta) \\ -\sin(2 \cdot \beta) & \cos(2 \cdot \beta) & L \cdot \cos(\beta) \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

In 3D space we cannot write immediately the dependence that exists between two segments. This relation can be obtained through the pre-multiplication of generic roto-translation matrix. One of the possible combinations to express the coordinate of the next segment related to the frame coordinate of the previous segment is the following:

$$R^i_{generic} := R_z^i(\theta^i) \cdot Tr_y(V^i) \cdot R_y^i(\alpha^i) \cdot R_z^i(\theta^i) \quad (3)$$

where $R_z^i(\theta^i)$ and $R_y^i(\alpha^i)$ are the fundamental roto-translation matrix having 4x4 elements in 3-D space, and $Tr_y(V^i)$ is a 4x4 elements matrix of pure translation in 3-D space and where V^i is the vector describing the translation between two segments expressed in coordinate of i-th reference system. The main problem remains to obtain an imposed shape for the tentacle arm. In order to control the robot, we need to obtain the relation between the position of the wires and the position of the segment.

Here, a decoupled approach is used for the robot control scheme. Thus the segments are controlled separately, without considering the interaction between them. Considering the segments of the tentacle separately, then $(\alpha, \theta)_i$ is the assigned coordinate of i-th segment. Having as purpose to command the robot to reach the position $(\alpha, \theta)_i$ the following relation is useful:

$$R = \frac{\bar{L}_{CB}}{\theta} \quad \forall \theta \neq 0 \quad (4)$$

where R represents the curvature's radius of the central bone and \bar{L}_{CB} is a constant, equal to the length of the central bone.

Once we have θ and α together as parameters of the desired shape, and after we obtained R , we can compute the corresponding lengths of the wires. Depending on the types of wires and on the structure of the tentacle, we must choose the way to compute the length of each wire.

For the hard wire, made from the same material as the central bone, and by consequence having the same elasticity, referring to Fig. 4, we can write:

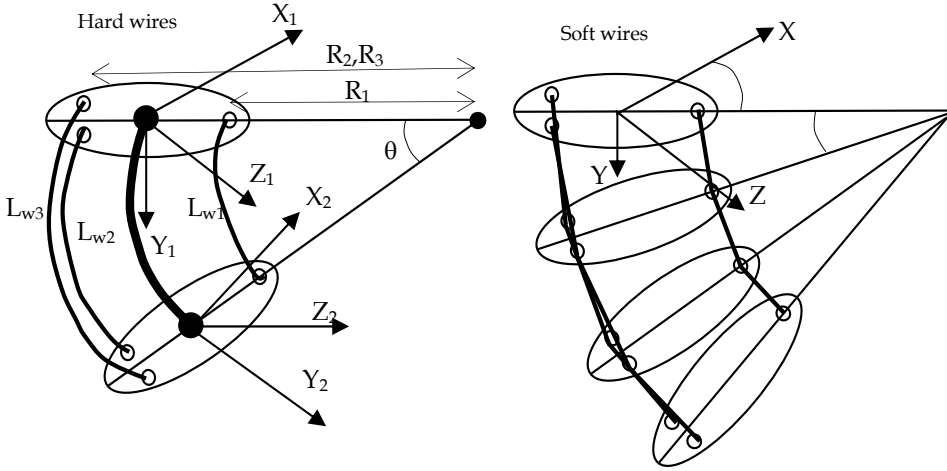


Fig. 4. Different types of wires.

$$\begin{cases} L_{w1} = R_1 \cdot \theta \\ L_{w2} = R_2 \cdot \theta \\ L_{w3} = R_3 \cdot \theta \end{cases} \quad (5)$$

For the soft wires, we can write:

$$\begin{cases} L_{w1} = [R_1 \cdot \theta] \cdot \frac{\sin(\theta/i)}{\theta/i} \\ L_{w2} = [R_2 \cdot \theta] \cdot \frac{\sin(\theta/i)}{\theta/i} \\ L_{w3} = [R_3 \cdot \theta] \cdot \frac{\sin(\theta/i)}{\theta/i} \end{cases} \quad (6)$$

where L_{wn} is the length of the n-th wire and R_i is the radius of the curvature of the real i-th wire.

Farther it can be written:

$$R_n = (R - \Delta R) \cdot \cos(\alpha_n) \quad (7)$$

where ΔR is constant equal to the distance between the center and the wires and α_n is:

$$\begin{cases} \alpha_1 = -\alpha \\ \alpha_2 = 120^\circ - \alpha \\ \alpha_3 = 240^\circ - \alpha \end{cases} \quad (8)$$

Obviously the equations (5) and (6), become the same for $i \rightarrow \infty$.

In order to reach the desired shape in a finite time t_f , we should choose the appropriate law for the time variation of the displacements and speed for the three wires, going from the home position to the final position. For each instant, the wires must be moved in order to avoid elongation or compression of it self.

The reference systems for each segment are oriented with the X -axes passing through the first wire. That means that the angles considered between the wires and the desired directions are as in the equation (8).

We can obtain the correlation between these angles and the bending direction of the segment. E.g. if the direction is $\alpha = 2/3\pi$, that means we intend to bend the tentacle in the direction of the second wire with the imposed value of θ degrees. In this case, if we will move the second wire of ΔL_{w2} , we should move the first and third wires with $\Delta L_{w2}/2$ and with the appropriate speed in order to maintain this relation during the movement.

Once we know the angle α , we can obtain the value $\Delta R_i = \Delta R \cdot \cos(\alpha_i)$, defining the displacements of the wires.

The algorithm that we are using, assigns the speed of the wires proportional to ΔR_i in order to go from the home position ($\theta=0, \alpha=0$) to the position $(\alpha, \theta)_i$ with a constant speed of the motors.

In fact, given the final time t_f and the starting time t_i , after we obtained the displacement of the wires we impose the speed in order to reach the desired position in $(t_f - t_i)$ seconds.

So the speed is:

$$\dot{L}_{wi} = \frac{L_{wi}(t_f) - \bar{L}_{CB}}{(t_f - t_i)} \quad (9)$$

Our structure does not have encoders. Counting the impulses given to the motors, we can evaluate the lengths $[L_{w1}, L_{w2}, L_{w3}]$. We use these values in order to obtain $(\alpha, \theta)_i$. The algorithm's steps are the following.

For the n -th rigid wire:

$$L_{wn} = \bar{L}_{CB} - \theta \cdot \Delta R \cdot \cos(\alpha_n) \quad (10)$$

Considering the equation (8) and (10), evaluating these for all the wires we can obtain:

$$\begin{cases} \sum_{i=1}^3 \cos(\alpha_i) = 0 \\ \frac{1}{3} \sum_{i=1}^3 R_i = R \\ \frac{1}{3} \sum_{i=1}^3 L_{wi} = L \end{cases} \quad (11)$$

Considering again the equation (10) for the first and second wires, we can write:

$$L_{w1} + \Delta R \cdot \theta \cdot \cos(\alpha_1) = L_{w2} + \Delta R \cdot \theta \cdot \cos(\alpha_2) \quad (12)$$

Replacing the (8) we obtain θ in function of α :

$$\theta = \frac{2}{\Delta R} \cdot \frac{L_{w1} - L_{w2}}{3 \cos(\alpha) - \sqrt{3} \sin(\alpha)} \quad (13)$$

And considering the eq. (10) for the third wire:

$$L_{w3} = L_{w1} + \frac{2 \cdot (L_{w1} - L_{w2}) \cdot (3 \cos(\alpha) - \sqrt{3} \sin(\alpha))}{3 \cos(\alpha) - \sqrt{3} \sin(\alpha)} \quad (14)$$

Finally the α angle can be obtained using the function atan2.

$$\alpha = \text{atan2}(\sqrt{3}(L_{w2} - L_{w3}), 2L_{w1} - L_{w2} - L_{w3}) \quad (15)$$

where atan2 is an extension of $\arctan(y/x)$ on more quadrant having the following form:

$$\left\{ \begin{array}{ll} \text{atan}(y/x) + \pi & \text{if } x < 0, y \geq 0 \\ \text{atan}(y/x) - \pi & \text{if } x < 0, y < 0 \\ \text{atan}(y/x) & \text{if } x > 0 \\ \frac{\pi}{2} & \text{if } x = 0, y > 0 \\ -\frac{\pi}{2} & \text{if } x = 0, y < 0 \end{array} \right. \quad (16)$$

The same methodology can be applied for a tronconical robot. The following paragraphs will show how the equations change. The geometry of one segment for the 2D case is described in Fig. 6. The curvature's angle θ of the segment is considered as the input parameter, while the lengths L_1 and L_2 of the control wires are the outputs.

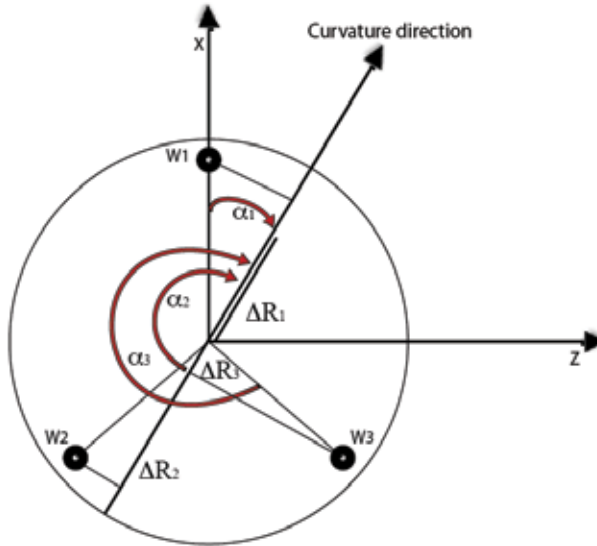


Fig. 5. Projection of the wire to get the α direction

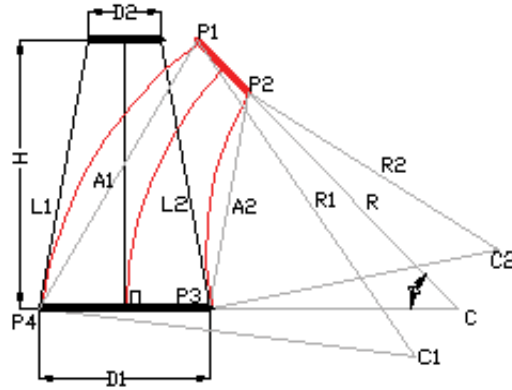


Fig. 6. The geometry of one segment.

The radius R of the segment curvature is obtained using equation (17):

$$R = \frac{H}{\theta} \quad (17)$$

where H is the height of the segment. The following lengths are obtained from Fig. 5, based on the segment curvature:

$$\begin{aligned} L_{11} &= \overline{CP_4} = R + D_1/2 & L_{12} &= \overline{CP_1} = R + D_2/2 \\ L_{21} &= \overline{CP_3} = R - D_1/2 & L_{22} &= \overline{CP_2} = R - D_2/2 \end{aligned} \quad (18)$$

where D_1 and D_2 are the diameters of the segment end discs. Based on the Carnot theorem, the lengths A_1 and A_2 are then obtained:

$$\begin{aligned} A_1 &= \sqrt{L_{11}^2 + L_{12}^2 - 2 \cdot L_{11} \cdot L_{12} \cdot \cos \theta} \\ A_2 &= \sqrt{L_{21}^2 + L_{22}^2 - 2 \cdot L_{21} \cdot L_{22} \cdot \cos \theta} \end{aligned} \quad (19)$$

The control wires curvature radius R_1 and R_2 are given by the relations (20):

$$R_1 = A_1/2 \cdot \sin \theta/2 \quad R_2 = A_2/2 \cdot \sin \theta/2 \quad (20)$$

Finally, the lengths of the control wires are obtained as in (21):

$$\begin{aligned} L_{w1} &= R_1 \cdot \theta = A_1 \cdot \theta / 2 \cdot \sin \frac{\theta}{2} \\ L_{w2} &= R_2 \cdot \theta = A_2 \cdot \theta / 2 \cdot \sin \frac{\theta}{2} \end{aligned} \quad (21)$$

For the 3D case, a virtual wire is considered, which gives the α direction of the curvature. Considering one virtual wire in the direction of the desired curvature having length calculated as follows. Firstly the following lengths are computed:

$$\begin{aligned} L_{11} &= R + D_1/2 \cdot \cos(\alpha_1) & L_{12} &= R + D_2/2 \cdot \cos(\alpha_1) \\ L_{21} &= R + D_1/2 \cdot \cos(\alpha_2) & L_{22} &= R + D_2/2 \cdot \cos(\alpha_2) \\ L_{31} &= R + D_1/2 \cdot \cos(\alpha_3) & L_{22} &= R + D_2/2 \cdot \cos(\alpha_3) \end{aligned} \quad (22)$$

where α_n is according to Fig. 5:

$$\begin{cases} \alpha_1 = -\alpha \\ \alpha_2 = 120^\circ - \alpha \\ \alpha_3 = 240^\circ - \alpha \end{cases} \quad (23)$$

Based on (19) and (20) the curvature radiuses R_1 , R_2 and R_3 of the three control wires are then obtained. Finally the lengths of the control wires are computed with (24):

$$\begin{aligned} L_{w1} &= R_1 \cdot \theta \\ L_{w2} &= R_2 \cdot \theta \\ L_{w3} &= R_3 \cdot \theta \end{aligned} \quad (24)$$

Apart from the system presented we can obtain two useful relations:

$$\begin{cases} \sum_{i=1}^3 \cos(\alpha_i) = 0 \\ \frac{1}{3} \sum_{i=1}^3 L_{wi} = L \end{cases} \quad (25)$$

The second equation of (25), can be utilized to estimate the virtual compression or the extension of the central bone. We call that virtual compression because before we compress the central bone, the robot will twist to find the shape to guaranty the wrong length of the wires.

3. Dynamics

3.1 Theoretical model

The essence of the tentacle model is a 3-dimensional backbone curve C that is parametrically described by a vector $r(s) \in \mathbf{R}^3$ and an associated frame $\varphi(s) \in \mathbf{R}^{3 \times 3}$ whose columns create the frame bases (Fig. 7a) (Ivănescu et al., 2006).

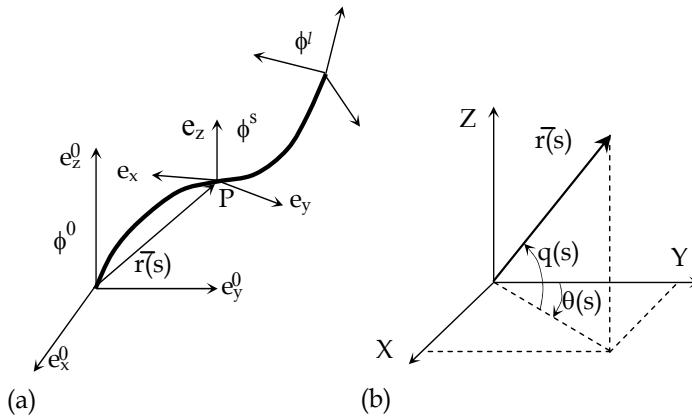


Fig. 7. Tentacle system parameters.

The independent parameter s is related to the arc-length from the origin of the curve C , a variable parameter, where

$$l = \sum_{i=1}^N (l_{0i} + \Delta l_i) \quad (26)$$

or

$$l = l_0 + u \quad (27)$$

where l_0 represents the length of the N elements of the arm in the initial position and

$$u = \sum_{i=1}^N \Delta l_i \quad (28)$$

determines the control variable of the arm length.

The position of a point s on curve C is defined by the position vector,

$$\bar{r} = r(s) \quad (29)$$

when $s \in [0, l]$. For a dynamic motion, the time variable will be introduced, $\bar{r} = \bar{r}(s, t)$.

We used a parameterization of the curve C based upon two "continuous angles" $\theta(s)$ and $q(s)$ and length variable u (Fig. 4).

At each point $\bar{r}(s, t)$, the robot's orientation is given by a right-handed orthonormal basis vector $\{\bar{e}_x, \bar{e}_y, \bar{e}_z\}$ and its origin coincides with point $\bar{r}(s, t)$, where the vector e_x is tangent and e_z is orthogonal to the curve C . The position vector on curve C is given by

$$\bar{r}(s, t) = [x(s, t) \ y(s, t) \ z(s, t)]^T \quad (30)$$

where

$$x(s, t) = \int_0^s \sin \theta(s', t) \cos q(s', t) ds' \quad (31)$$

$$y(s, t) = \int_0^s \cos \theta(s', t) \cos q(s', t) ds' \quad (32)$$

$$z(s, t) = \int_0^s \sin q(s', t) ds' \quad (33)$$

with $s' \in [0, s]$. We can adopt the following interpretation: at any point s the relations (31)-(33) determine the current position and $\theta(s)$ determines the robot's orientation, and the robot's shape is defined by the behaviour of functions $\theta(s)$ and $q(s)$. The robot "grows" from the origin by integrating to get $\bar{r}(s, t)$, $s \in [0, l_0 + u]$. The velocity components are

$$v_x = \int_0^s (-\dot{q}' \sin q' \sin \theta' + \dot{\theta}' \cos q' \cos \theta') ds' \quad (34)$$

$$v_y = \int_0^s (-\dot{q}' \sin q' \cos \theta' - \dot{\theta}' \cos q' \sin \theta') ds' \quad (35)$$

$$v_z = \int_0^s \dot{q}' \cos q' ds' \quad (36)$$

$$v_u = \dot{u} \quad (37)$$

For an element dm , kinetic and gravitational potential (Douskaia, 1998) energy will be

$$dT = \frac{1}{2} dm (v_x^2 + v_y^2 + v_z^2 + v_u^2) \quad (38)$$

$$dV = dm \cdot g \cdot z \quad (39)$$

Where

$$dm = \rho ds \quad (40)$$

From (13)-(15) we obtain

$$T = \frac{1}{2} \rho \int_0^l \left(\int_0^s (-\dot{q}' \sin q' \sin \theta' + \dot{\theta}' \cos q' \cos \theta') ds' \right)^2 + \quad (41)$$

$$+ \left(\int_0^s (-\dot{q}' \sin q' \cos \theta' - \dot{\theta}' \cos q' \sin \theta') ds' \right)^2 +$$

$$\left(\int_0^s \dot{q}' \cos q' ds' \right)^2 ds + \frac{1}{2} \rho \int_0^l \dot{u}^2 ds$$

$$V = \rho g \int_0^l \int_0^s \sin q' ds' ds \quad (42)$$

The elastic potential energy will be approximated by two components, one determined by the bending of the element

$$V_{eb} = k \frac{d^2}{4} \sum_{i=1}^N (q_i^2 + \theta_i^2) \quad (43)$$

and the other is given by the axial tension/compression energy component

$$V_{ea} = \frac{1}{2} k u^2 \quad (44)$$

where we assumed that each element has a constant curvature and a uniform equivalent elasticity coefficient k , assumed constant on all the length of the arm.

The total elastic potential energy will be

$$V_e = V_{eb} + V_{ea} \quad (45)$$

We will consider $F_\theta(s, t)$, $F_q(s, t)$ the distributed forces on the length that determine motion and orientation in the θ - plane, q - plane and $F_u(t)$, the force that determines axial motion, assumed constant along the length of the arm.

3.2 Dynamic model

In this paper, the manipulator model is considered a distributed parameter (Ivanescu, 2002). system defined on a variable spatial domain $\Omega = [0, l]$ and the spatial coordinate is denoted by s .

The dynamic model of this manipulator with hyper-redundant configurations can be obtained, in general form, from Hamilton partial differential equations of the distributed parameter model,

$$\frac{\partial \omega(t, s)}{\partial t} = \frac{\delta H}{\delta v(t, s)} \quad (46)$$

$$\frac{\partial v(t, s)}{\partial t} = -\frac{\delta H}{\delta \omega(t, s)} + F(t, s) \quad (47)$$

where ω and v are the generalized coordinates and momentum densities, respectively, and $\delta(\cdot) / \delta(\cdot)$ denotes a functional partial derivative.

The state of this system at any fixed time t is specified by the set $(\omega(t, s), v(t, s))$, where $\omega = [\theta \ q \ u]^T$. The set of all functions of $s \in \Omega$ that ω, v can take on at any time is state function space $\Gamma(\Omega)$. We will consider that $\Gamma(\Omega) \subset L_2(\Omega)$.

The control forces have the distributed components along the arm, $F_\theta(t, s)$, $F_q(t, s)$, $s \in [0, l]$ and a lumped component $F_u(t)$.

A practical form of dynamic model expressed only as a function of generalised coordinates is derived by using Lagrange equations developed for infinite dimensional systems,

$$\frac{\partial}{\partial t} \left(\frac{\delta T}{\delta \dot{\theta}(t, s)} \right) - \frac{\delta T}{\delta \theta(t, s)} + \frac{\delta V}{\delta \theta(t, s)} + \frac{\delta V_e}{\delta \theta(t, s)} = F_\theta \quad (48)$$

$$\frac{\partial}{\partial t} \left(\frac{\delta T}{\delta \dot{q}(t, s)} \right) - \frac{\delta T}{\delta q(t, s)} + \frac{\delta V}{\delta q(t, s)} + \frac{\delta V_e}{\delta q(t, s)} = F_q \quad (49)$$

$$\frac{\partial}{\partial t} \left(\frac{\delta T}{\delta \dot{u}} \right) - \frac{\delta T}{\delta u} + \frac{\delta V}{\delta u} + \frac{\delta V_e}{\delta u} = F_u \quad (50)$$

where $\partial / \partial(\cdot)$, $\delta / \delta(\cdot)$ denote classical and functional partial derivatives (in Gateaux sense), respectively.

In Appendix 1 the dynamic model of this ideal spatial tentacle manipulator will be developed and the difficulties to obtain a control law will be easily inferred.

The great number of parameters - theoretically an infinite number of parameters - the complexity of the dynamic model make the application of the classical algorithms meant to obtain the control law very difficult. In much of the literature concerned with the control of these systems, the complexity of the problem is emphasized and various methods that compensate all nonlinear terms in dynamics in real time are developed in order to reduce the complexity of control systems. Also, simplified procedures are introduced or the difficult components are neglected in order to generate a particular law for position or motion control. In all these cases, these methods require a large amount of complicated calculation so that it is difficult to implement these methods with usual level controllers. In addition, the reliability of these methods may be lost when a small error in computation or a small change in the system's parameters occurs.

3.3 Unconstrained control

The artificial potential is a potential function whose points of minimum are attractors for a dissipative controlled system. It was shown that the control of robot motion to a desired point is possible if the function has a minimum in the desired point. In this section we will extend this result for the infinite dimensional model of the tentacle manipulator with variable length.

We consider that the initial state of the system is given by

$$\omega_0 = \omega(0, s) = [\theta_0, q_0, l_0]^T \quad (51)$$

$$v_0 = v(0, s) = [0, 0, 0]^T \quad (52)$$

$$\theta_0 = \theta(0, s), \quad q_0 = q(0, s), \quad s \in [0, l_0] \quad (53)$$

$$l_0 = l(0) \quad (54)$$

corresponding to the initial position of the manipulator defined by the curve C_0

$$C_0 : (\theta_0(s), q_0(s), l_0), \quad s \in [0, l_0] \quad (55)$$

The desired point in $\Gamma(\Omega)$ is represented by a desired position of the arm, the curve C_d ,

$$\omega_d = [\theta_d, q_d, l_d]^T, \quad v_d = [0, 0, 0]^T \quad (56)$$

$$C_d : (\theta_d(s), q_d(s), l_d), \quad s \in [0, l_d]$$

The system motion (48)-(5) corresponding to a given initial state (ω_0, v_0) defines a trajectory in the state function space $\Gamma(\Omega)$. The control problem of the manipulator means the motion control by the forces F_θ, F_q, F_u from the initial position C_0 to the desired position C_d . From the viewpoint of mechanics, the desired position (ω_d, v_d) is asymptotically stable if the potential function of the system has a minimum at $(\omega, v)(s) = (\omega_d, v_d)(s), \quad s \in [0, l]$ and the

system is completely damped. As a control problem in this paper the results of will be extended for the infinite dynamic systems.

We will consider the control forces,

$$F_{\theta}(t,s) = \frac{\delta V}{\delta \theta(t,s)} + \frac{\delta V_e}{\delta \theta(t,s)} - F_{at} - \frac{\delta II}{\delta \theta(t,s)} \quad (57)$$

$$F_u(t) = \frac{\partial V}{\partial u(t)} + \frac{\partial V_e}{\partial u(t)} - F_{ud} - \frac{\partial II}{\partial u(t)} \quad (58)$$

The first two terms compensate the gravitational and elastic potential, the third components assure the damping control and the last terms define the new artificial potential introduced in order to assure the motion to the desired position. The minimum points of this potential must be identical with desired positions of the manipulator, as attractors of its motion. For example, the potential II can be selected as a functional of generalised coordinates,

$$II(\theta, q, u) = \int_0^l \left((\theta - \theta_d(s))^2 + (q - q_d(s))^2 \right) ds + (l_0 + u - l_d)^2 \quad (59)$$

The control law (57)-(59) modifies the system potential and the Lagrange equation (48)-(50) (Masoud & Masoud, 2000) become

$$\frac{\partial}{\partial t} \left(\frac{\delta T}{\delta \dot{\theta}(t,s)} \right) - \frac{\delta T}{\delta \theta(t,s)} + \frac{\delta II}{\delta \theta(t,s)} = F_{\theta_d} \quad (60)$$

$$\frac{\partial}{\partial t} \left(\frac{\delta T}{\delta \dot{q}(t,s)} \right) - \frac{\delta T}{\delta q(t,s)} + \frac{\delta II}{\delta q(t,s)} = F_{q_d} \quad (61)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial T}{\partial \dot{u}} \right) - \frac{\partial T}{\partial u} + \frac{\partial II}{\partial u} = F_{u_d} \quad (62)$$

The force components F_{θ_d} , F_{q_d} , F_{u_d} represent the damping components of the control and have the form

$$F_{\theta_d}(s,t) = - \int_0^l K_{\theta}(s,s') \dot{\theta}(s',t) ds' \quad (63)$$

$$F_{q_d}(s,t) = - \int_0^l K_q(s,s') \dot{q}(s',t) ds' \quad (64)$$

$$F_{u_d}(t) = -K_u \dot{u}(t) \quad (65)$$

where $K_{\theta}(s,s')$, $K_q(s,s')$ are positive definite specified spatial weighting functions on $(\Omega \times \Omega)$ and K_u is a positive constant. For practical reasons, the derivative components of the control have the form

$$K_\theta(s, s') = \delta(s - s') \cdot k_\theta(s) \quad (66)$$

$$K_q(s, s') = \delta(s - s') \cdot k_q(s) \quad (67)$$

3.4 Constrained control

Let B be the region of the state (Ceah & Wang, 2005) space where the mechanical system motion is not admissible, its complement \bar{B} is the region of admissible movements and ∂B is the boundary of B . The control problem is to determine the potential function $\Pi(\theta, q, u)$ which would determine the motion to the desired position $(\omega_d(s), v_d(s))$, $s \in [0, l]$ and it does not penetrate the constrained area B . In terms of the artificial potential, this means that this functional should have a single stationary point in \bar{B} and grows without limit when the system penetrates the boundary ∂B .

We will consider the following artificial potential,

$$\Pi(\theta, q, u) = \max\{\Pi_1(\theta, q, u), \Pi_2(\theta, q, u)\} \quad (68)$$

where $\Pi_1(\theta, q, u)$ is the artificial potential for unconstrained problem and $\Pi_2(\theta, q, u)$ is the potential for constrained control problem.

$\Pi_2(\theta, q, u)$ is a non-negative, continuous functional defined in \bar{B} and

$$\lim_{d \rightarrow 0} \Pi_2(\theta, q, u) = \infty \quad (69)$$

where d is the distance between the current state (θ, q, u) and the boundary ∂B .

3.5 Appendix 1

We will consider a spatial tentacle model, an ideal system, neglecting friction and structural damping. We assume a uniformly distributed mass with a linear density ρ [kg/m].

We will use the notations:

$$\begin{aligned} q &= q(s, t), \quad s \in [0, l], \quad t \in [0, t_f] & \theta &= \theta(s, t), \quad s \in [0, l], \quad t \in [0, t_f] \\ q' &= q(s', t), \quad s' \in [0, s], \quad t \in [0, t_f] & \dot{q} &= \frac{\partial q(s, t)}{\partial t}, \quad s \in [0, l], \quad t \in [0, t_f] \\ \dot{q}' &= \frac{\partial q(s', t)}{\partial t}, \quad s' \in [0, s], \quad t \in [0, t_f] & \ddot{q}' &= \frac{\partial^2 q(s', t)}{\partial t^2}, \quad s' \in [0, s], \quad t \in [0, t_f] \\ & & \ddot{q}'' &= \frac{\partial^2 q(s'', t)}{\partial t^2}, \quad s'' \in [0, s], \quad t \in [0, t_f] \end{aligned}$$

$$F_q = F_q(s, t), \quad s \in [0, l], \quad t \in [0, t_f] \quad F_u = F_u(t), \quad t \in [0, t_f]$$

From (60)-(62), it results,

$$\begin{aligned}
& \rho \int_0^s \int_0^s (\ddot{q}'(\sin q' \sin q'' \cos(q' - q'') + \cos q' \cos q'') - \ddot{\theta}' \cos q' \sin q'' \sin(\theta'' - \theta')) + \\
& + \dot{q}'^2 (\cos q' \sin q'' \cos(\theta'' - \theta') - \sin q' \cos q'') + \dot{\theta}'^2 \cos q' \sin q'' \cos(\theta'' - \theta') - \dot{q}' \dot{q}'' \sin(q'' - q') ds' ds'' + \\
& + \rho g \int_0^s \cos q' ds' + \frac{1}{2} k d^2 q = F_q \\
& \rho \int_0^s \int_0^s (\ddot{q}' \sin q' \cos q'' \sin(\theta'' - \theta') + \ddot{\theta}' \cos q' \cos q'' \cos(\theta'' - \theta') - \dot{q}'^2 \cos q' \cos q'' \sin(\theta'' - \theta') + \\
& + \dot{\theta}' \cos q' \cos q'' \sin(\theta'' - \theta') - \dot{\theta}' \dot{q}' \sin q' \cos q'' \cos(\theta'' - \theta')) ds' ds'' + \frac{1}{2} k d^2 \theta = F_\theta \\
& \rho \ddot{u} + \frac{1}{2} \rho \dot{u}^2 + k u = F_u
\end{aligned}$$

4. Visual servoing system

4.1 Camera system

In the Appendix 2 the dynamic model of the 3D spatial hyper redundant arm is determined. Two video cameras provide two images of the whole robot workspace. The two images planes are parallel with XOY and ZOY planes from robot coordinate frame, respectively (Fig. 8). The cameras provide the images of the scene stored in the frame grabber's video memory being displayed on the computer screens (Hannan & Walker, 2005); (Kelly, 1996). Related to the image planes, two dimensional coordinate frames, called screen coordinate frames or image coordinate systems are defined. Denote X_{S_1} , Y_{S_1} and Z_{S_2} , Y_{S_2} , respectively, the axes of the two screen coordinate frames provided by the two cameras. The spatial centers for each camera are located at the distances D_1 and D_2 , with respect to the XOY and ZOY planes, respectively. The orientation of the cameras around the optical axes with respect to the robot coordinate frame, are noted with ψ and ϕ , respectively. A point P in the coordinate frame is

$$P = [x, y, z]^T \quad (70)$$

The description of a point P in the two screen coordinate frames are denoted by

$$P_{S_2} = [x_{S_1}, y_{S_2}] \quad (71)$$

$$P_{S_2} = [z_{S_2}, y_{S_2}] \quad (72)$$

Geometric optics are used to model the mapping between the robot Cartesian space and the screen coordinate systems. We assume that the quantization and the lens distortion effects

are negligible. The description of the point $P=[x, y, z]^T$ in the robot coordinate frame is given in terms of screen coordinate frames as

$$\begin{bmatrix} x_{s_1} \\ y_{s_1} \end{bmatrix} = \alpha_1 \cdot \frac{\lambda_1}{\lambda_1 - (D_1 + x)} \cdot R(\phi) \cdot \left\{ \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} o_{11} \\ o_{12} \end{bmatrix} \right\} + \begin{bmatrix} c_{x_1} \\ c_{y_1} \end{bmatrix} \quad (73)$$

for the $Z_{s_1}O_{s_1}Y_{s_1}$ frame and

$$\begin{bmatrix} z_{s_2} \\ y_{s_2} \end{bmatrix} = \alpha_2 \cdot \frac{\lambda_2}{\lambda_2 - (D_2 + x)} \cdot R(\phi) \cdot \left\{ \begin{bmatrix} z \\ y \end{bmatrix} - \begin{bmatrix} o_{21} \\ o_{22} \end{bmatrix} \right\} + \begin{bmatrix} c_{z_2} \\ c_{y_2} \end{bmatrix} \quad (74)$$

for the $Z_{s_2}O_{s_2}Y_{s_2}$ frame, where $[c_{x_1}, c_{y_1}]^T$ and $[c_{z_2}, c_{y_2}]^T$ the image centers, α_1 and α_2 are the scale factors of the length units in the front image planes given in pixel/m, $R(\psi)$ and $R(\phi)$ are the rotation matrices generated by clockwise rotating the cameras about their optical axes by ψ and ϕ radians, respectively, and $[O_{11}, O_{12}]^T$ and $[O_{21}, O_{22}]^T$ represent the distances between the optical axes and the XOY and ZOY planes, respectively.

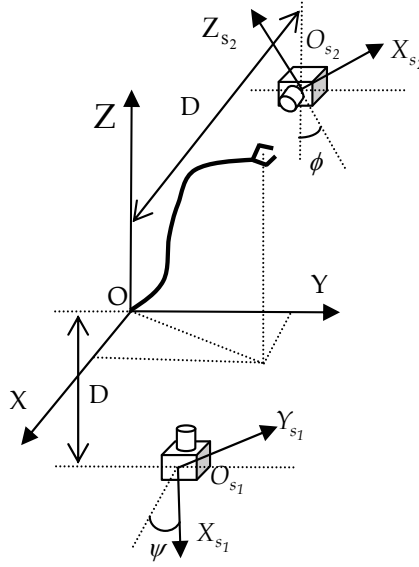


Fig. 8. Camera system

In Fig. 9 the frames corresponding to the screen images of the two cameras are presented. From the relations (73), (74), we obtain

$$\begin{bmatrix} \Delta x_{s_1} \\ \Delta y_{s_1} \end{bmatrix} = \alpha_1 \cdot \frac{\lambda_1}{\lambda_1 - (D_1 + x)} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (75)$$

$$\begin{bmatrix} \Delta z_{s_2} \\ \Delta y_{s_2} \end{bmatrix} = \alpha_2 \cdot \frac{\lambda_2}{\lambda_2 - (D_2 + x)} \cdot \begin{bmatrix} \Delta z \\ \Delta y \end{bmatrix} \quad (76)$$

and the orientation angles for each plane will be

$$\operatorname{tg} \theta_s = \frac{\Delta x_{s_1}}{\Delta y_{s_1}} = \frac{\Delta x}{\Delta y} = \operatorname{tg} \theta \quad (77)$$

hence

$$\theta_s(s') = \theta(s), \quad s \in [0, l], \quad s' \in [0, l'] \quad (78)$$

for the plane $Z_{s_1} O_{s_1} Y_{s_1}$ and $\operatorname{tg} q_s = \frac{\Delta z_{s_2}}{\Delta y_{s_2}} = \frac{\Delta z}{\Delta y}$

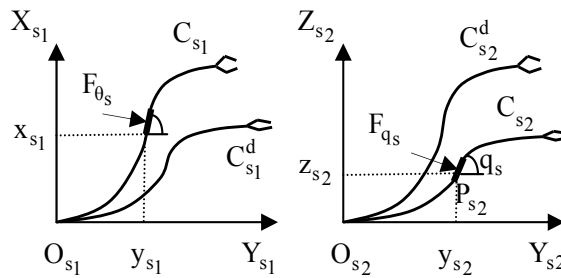


Fig. 9. Image frames

This relation allows the computation of the orientation angle q_s in the plane $Z_{s_2} O_{s_2} Y_{s_2}$

$$\operatorname{tg} q_s(s'') = \operatorname{tg} q(s) \cdot \frac{1}{\cos \theta(s)}, \quad s \in [0, l], \quad s'' \in [0, l''] \quad (79)$$

where, s', s'' and l', l'' represent the projections of the variable s and the length l in the two planes, respectively. The projection of the forces on the two planes can be easily inferred and the relations (77)-(79),

$$F_{\theta_s} = F_\theta \quad (80)$$

$$F_{q_s} = F_q \cdot \sqrt{\cos^2 q + \sin^2 q \cdot \cos^2 \theta} \quad (81)$$

4.2 Servoing system

The control system is an image - based visual servo control where the error control signal is defined directly in terms of image feature parameters. The desired position of the arm in the robot space is defined by the curve C_d ,

$$C : (\theta^d(s), q^d(s)), \quad s \in [0, l] \quad (82)$$

or, in the two image coordinate frames $Z_{s_1} O_{s_1} Y_{s_1}$ and $Z_{s_2} O_{s_2} Y_{s_2}$, by the projection of the curve C ,

$$C_{s_1}^d : (\theta_s^d(s')), s' \in [0, l'] \quad (83)$$

$$C_{s_2}^d : (q_s^d(s'')), s'' \in [0, l''] \quad (84)$$

Define the motion errors as

$$e_\theta(t, s) = \theta(t, s) - \theta_d(s), s \in [0, l] \quad (85)$$

$$e_q(t, s) = q(t, s) - q_d(s), s \in [0, l] \quad (86)$$

or, in the image coordinate frames, by $s' \in [0, l']$, $s'' \in [0, l'']$

$$e_{\theta_s}(t, s') = \theta_s(t, s') - \theta_s^d(s') \quad (87)$$

$$e_{q_s}(t, s'') = q_s(t, s'') - q_s^d(s'') \quad (88)$$

The global control system is presented in Fig. 10. The control problem of this system is a direct visual servocontrol but we do not use the classical concept of the position control where the error between the robot end-effector and target is minimized (Grosso et al., 1996).

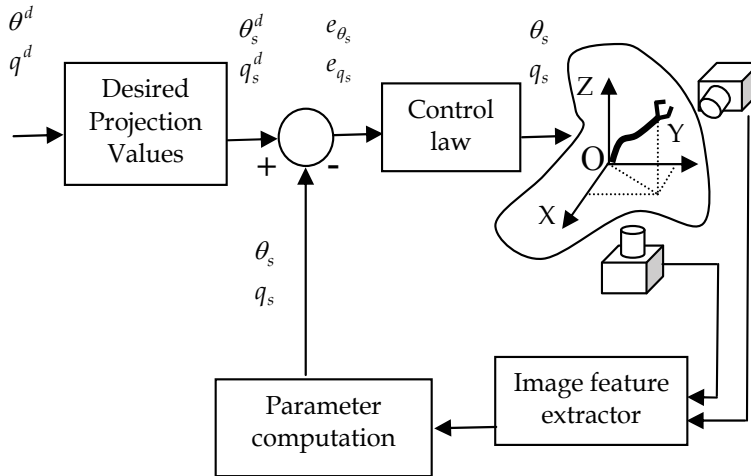


Fig. 10. The global control system

In this paper we will use the control of the curve's shape in each point of the mechanical structure. The method is based on the particular structure of the system defined as a "backbone with two continuous angles $\theta(s)$ and $q(s)$ ". The control of the system is based on the control of the two angles $\theta(s)$ and $q(s)$. These angles are measured directly or indirectly. The angle $\theta(s)$ is measured directly by the projection on the image plane $Z_{S_1}O_{S_1}Y_{S_1}$ (relation 78) and $q(s)$ is computed from the projection on the image plane $Z_{S_2}O_{S_2}Y_{S_2}$ (relation 79). The stability of the closed-loop system is proven by the Lyapunov's second method but, in order to avoid the complex problems derived from using the nonlinear derivation integral model, a method based on the energy-work relationship (Ge et al., 1996) was developed (see Appendix 2).

Proposition: The closed-loop hyper redundant arm system is stable if the control law is

$$F_\theta(s,t) = -k_\theta^1(s) \cdot e_{\theta_s}(s',t) - k_\theta^2(s) \cdot \dot{e}_{\theta_s}(s',t) \quad (89)$$

$$F_q(s,t) = -k_q^1(s) \cdot [tg^{-1}(\cos \theta_s(s',t)) \cdot tg q_s(s'',t) - q^d(s)] \quad (90)$$

where $s' \in [0, l]$, $s'' \in [0, l'']$ and $k_\theta^1(s), k_\theta^2(s), k_q^1(s)$ are positive coefficients of the control law for all $s \in [0, l]$. The parameter of the control law (88), (89), can be inferred from the image feature extraction of the two planes. The parameters e_{θ_s} can be directly calculated from equation (85-88) and \dot{e}_{θ_s} can be indirectly computed. Also θ_s , q_s and q_s^d are evaluated directly from the trajectory projections. We remark that the control law represents a robust control, independent of the camera parameters. No intrinsic camera parameters are assumed known.

4.3 Appendix 2

We will consider a spatial tentacle model, an ideal system, neglecting friction and structural damping. We assume a uniformly distributed mass with linear density $\rho [kg/m]$. We will consider a non-extensible arm with constant length.

We will use the notations:

$$q = q(s,t), \quad q' = q'(s',t), \quad \dot{q} = \frac{\partial q(s,t)}{\partial t},$$

$$\dot{q}' = \frac{\partial q'(s',t)}{\partial t}, \quad \ddot{q}' = \frac{\partial^2 q'(s',t)}{\partial t^2}, \quad \ddot{q}'' = \frac{\partial^2 q(s'',t)}{\partial t^2},$$

The position of a point P is given by (31-33) and the velocity components are given by (34-37). From an element dm , kinetic and potential energy are given by will be (38-40). Following (41-42) were computed.

The dynamic model is obtained by using Lagrange equation of motion

$$\frac{d}{dt} \left(\frac{\delta T}{\delta \dot{q}} \right) - \frac{\delta T}{\delta q} + \frac{\delta V}{\delta q} = F \quad (91)$$

where $\delta(\cdot)/\delta(\cdot)$ denotes a functional partial (variational) Gateaux derivate (Wang, 1965), as shown before, that is defined as the variation of the functional Ω with respect to the function θ at a point $s \in [0, l]$. From (41-42) it results,

$$\rho \cdot \int_0^l \int_0^s \left(\ddot{q}' \cdot (\sin q' \cdot \sin q'' \cdot \cos(q' - q'')) + \cos q' \cdot \cos q'' \right) - \ddot{\theta}' \cdot \cos q' \cdot \sin q'' \cdot \sin(\theta' - \theta'') +$$

$$+ \dot{q}'^2 \cdot (\cos q' \cdot \sin q'' \cdot \cos(\theta' - \theta'')) - \sin q' \cdot \cos q'' + \dot{\theta}'^2 \cdot \cos q' \cdot \sin q'' \cdot \cos(\theta' - \theta'') -$$

$$- \dot{q}' \cdot \dot{q}'' \cdot \sin(q'' - q') \Big) \cdot ds' \cdot ds'' + \rho \cdot g \cdot \int_0^s \cos q' \cdot ds' = F_q \quad (92)$$

$$\begin{aligned}
& \rho \cdot \int_0^l \int_0^s \left(\ddot{q}^1 \sin q^1 \cdot \cos^{l-1} \cdot \sin(q^1 - q^{l-1}) + \ddot{\theta}^1 \cdot \cos q^1 \cdot \cos^{l-1} \cdot \cos(\theta^1 - \theta^{l-1}) + \right. \\
& \left. + \dot{q}^{l-1} \cdot (\cos q^1 \cdot \cos q^{l-1} \cdot \sin(\theta^{l-1} - \theta^1)) + \dot{\theta}^1 \cdot \cos q^1 \cdot \cos q^{l-1} \cdot \sin(\theta^{l-1} - \theta^1) - \right. \\
& \left. - \dot{\theta}^1 \cdot \dot{q}^1 \cdot \cos q^1 \cdot \cos(\theta^{l-1} - \theta^1) \right) \cdot ds^1 \cdot ds^{l-1} = F_\theta
\end{aligned} \quad (93)$$

We consider the following Lyapunov function.

$$V^*(t) = T(t) + V(t) + \frac{1}{2} \cdot \int_0^l k_\theta^1(s) \cdot e_\theta^2(s, t) \cdot ds \quad (94)$$

where T , V represent the kinetic and potential energies of the system. $V^*(t)$ is positive defined because the terms that represent the energy T and V are always $T(t) \geq 0$, $V(t) \geq 0$. For the steady desired position, we have

$$\dot{V}^*(t) = \int_0^l \left(F_\theta(s, t) \cdot \dot{e}_\theta(s, t) + F_q(s, t) \cdot \dot{e}_q(s, t) + k_\theta^1(\theta) \cdot e_\theta(s, t) \right) \cdot ds \quad (95)$$

If we use the control law defined by the relations (89)-(90), where the parameters of motion are evaluated from (78)-(79), (85)-(88), we will have,

$$\dot{V}^*(t) = - \int_0^l k_\theta^1 \cdot \left(\dot{e}_\theta(s, t) \right)^2 \cdot ds \quad (96)$$

$$\dot{V}^*(t) \leq 0 \quad (97)$$

Q.E.D.

The derivative of the error in the control laws (89), (90) can be computed by an iteration procedure. The coordinate x_{s_1} on the projection C_{s_1} can be evaluated by the relation

$$x_{s_1}(i \cdot \Delta s^1) = \sum_{j=1}^i \sin \theta_{s_1}(j \cdot \Delta s^1) \cdot \Delta s^1 \quad (98)$$

$$\Delta x_{s_1}(i \cdot \Delta s^1) = \sum_{j=1}^i \cos \theta_{s_1}(j \cdot \Delta s^1) \cdot \Delta \theta(j \cdot \Delta s^1) \cdot \Delta s^1 \quad (99)$$

Assuming that $\theta_{s_i}(s_i) \neq \frac{\pi}{2}$, we obtain

$$\begin{aligned}
i=1 \quad \Delta \theta(\Delta s^1) &= \frac{1}{\Delta s^1} \left(\Delta x_{s_1}(\Delta s^1) / \cos \theta_{s_1}(\Delta s^1) \right) \\
i=2 \quad \Delta \theta(2 \cdot \Delta s^1) &= \left(\Delta x_{s_1}(2 \cdot \Delta s^1) / \Delta s^1 - \cos \theta_{s_1}(\Delta s^1) \cdot \Delta \theta(\Delta s^1) \right) / \cos \theta_{s_1}(2 \cdot \Delta s^1) \\
&\vdots \\
i=m \quad \Delta \theta(m \cdot \Delta s^1) &= \left(\Delta x_{s_1}(m \cdot \Delta s^1) / \Delta s^1 - \sum_{j=1}^{m-1} \cos \theta_{s_1}(j \cdot \Delta s^1) \cdot \Delta \theta(j \cdot \Delta s^1) \right) \cdot \frac{1}{\cos(m \cdot \Delta s^1)}
\end{aligned} \quad (100)$$

If $\theta_{s_i}(s_i) = \frac{\pi}{2}$, a similar procedure for $y_{s_i}(i \cdot \Delta s^i)$ can be used.

4.4 Camera calibration

The term “camera calibration” in the context of this paper refers to positioning and orienting the two cameras at imposed values (Fig. 11) (Tanasie et al., 2009). This calibration is performed only at the beginning, after that the cameras remain still. First, a zoom that maximizes the image resolution of the working space used by the manipulator is performed. Second, positioning of the two cameras brings the manipulator in the middle of the two images. Third, a pan / tilt orientation is performed (as described later in the paper). At this step the manipulator is moved in a test position that allows free of (or minimum) errors calibration. The test images are compared to the images generated by the graphic simulator (ideal images) which represent references for the calibration operation.

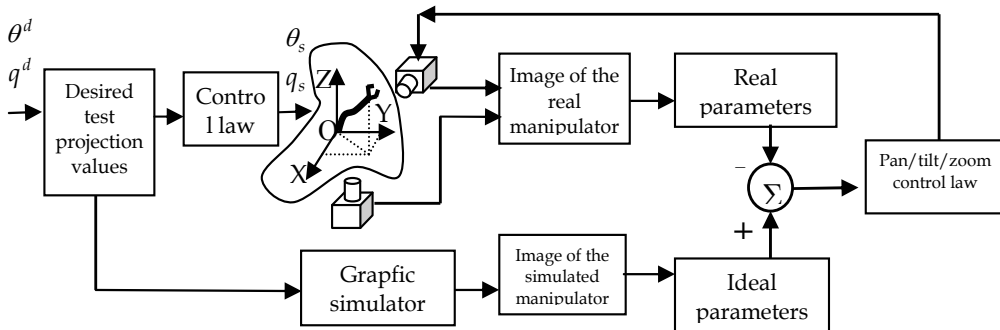


Fig. 11. Camera calibration system

In order, to ease the fulfil of the cameras calibration, a graphic simulator based on a 2D direct kinematics model was designed, implemented and used. By consequence, during the calibration procedure, the robot was commanded to bend in planes perpendicular to the cameras axes. Thus only the arching angle needs to be computed and a 2D model is sufficient to solve the problem. The next version of the software application introduces also the possibility to calibrate in 3D, the test positions corresponding to unrestricted planes orientation. A very important task in developing this application is to control the camera position and orientation. From this point of view, the calibration operation assures that the two cameras' axes are orthogonal. In the beginning, the tentacle manipulator receives the needed commands in order to stand in a test pose (imposed position and orientation). The same commands are sent to the graphic simulator. Two different sets of images are obtained: real images acquired by the real cameras and simulated images offered by the graphic simulator. From these two sets of images, two sets of parameters are computed: real parameters are computed from real images and, respectively, ideal parameters are computed from synthetic images. Comparing the two sets of parameters and knowing the image/parameters behavior for the camera orientation, the cameras are orientated (pan/tilt/zoom) in order to minimize the error.

A graphical simulator was designed and implemented in order to test the robot behavior under certain circumstances (Cojocaru et al., 2010). The simulator approximates the curved segments of the hyper redundant robot and considers constant the length of the median arc

of each segment. To ease the presentation, the term segment will be used in all that follows referring to the median segment (arched or un-arched). For the arched segment, its median arc remains constant. In this paper the term O-X angle will be used to denote the angle that the chord made by an arched element of the robot makes with the O-X axis of a selected reference system.

The inputs for the simulator are: robot configuration; robot initial position; control laws for each of the segments of the hyper redundant robot. The robot configuration consists of the number of segments the hyper redundant robot has, the length of each segment and the angles that the cords make with the O-X axis. The arching angles are computed from these angles. An arching angle is defined as the angle made by the cord (determined by the ends of the arched segment) and the original un-arched segment. For the direct kinematics problem, the control of the robot simulation is accomplished by giving the O-X angles for each of the segments in their final position and the output of the simulation is the hyper redundant robot's end-effector final position in the operation space. In order to compute the final position of the end-effector and the hyper redundant robot's behavior during its motion, a few elements must be computed: the relation between the arching angle and the angle at center determined by the arched segment (this angle determines the length of the arc); the cord length; the relation between an O-X angle and an arching angle; the final arching angles - recurrent set.

The computation of the relation between the arching angle and the angle at center determined by the arched segment is determined by the following axiom: For camera calibration a direct kinematics model was used, thus the rotation angles for each segment are given. For a robot that has only rotation joints, the O-X angle increases (or decreases, depending on the selected positive direction) for each segment with the sum of rotation angles of each of the previous segments (including the current segment). This is true because the orthogonal system attached to the i th segment is obtained from its initial position and applying all the anterior transformations. For a hyper redundant robot the problems are different. The arching angle is double the sum of each previous arching angle plus the current arching angle, because the un-arched segment is a prolongation of the previous segment.

In order to simulate the circular arched segments a series of intermediate points (that are connected by lines) between the segment origins must be determined. The Catmull-Rom interpolation algorithm was used for this simulator because it was need an interpolation algorithm that passes through the control points. Catmull-Rom splines are a family of cubic interpolating splines formulated such that the tangent at each point p_i is calculated using the previous and next point on the splines, $\tau(p_{i+1} - p_{i-1})$.

Camera calibration is the essential procedure for all such applications: positioning and orienting the cameras in order to support the accuracy of the image features extraction. Calibration for a pan/tilt/zoom camera shape is achieved by means of an engineered environment and a graphic simulation module.

Term "camera calibration" in the context of this paper refers to positioning and orienting the two cameras at imposed values. This calibration is performed only at the beginning, after that the cameras remain still. The general control method is an image based visual servoing one instead of position based. Camera calibration based on intrinsic parameters (classic

sense, not the one used in this paper) is not necessary. Calibration operation assures that the two cameras' axes are orthogonal.

Taking into account the presented structure of the tentacle - vision system, in order to apply the tested visual servoing algorithm, the two cameras must be positioned and oriented as: both focus on the robot, their axes are orthogonal, both have the same zoom factor.

Two different algorithms were implemented: one uses a cylindrical etalon, other uses the graphical simulator.

For the first algorithm, special starting conditions were imposed in order to support the image processing tasks: white background, dark grey cylinder, red vertical equidistant (90 degrees) axes, friendly initial camera's positions and orientations, zoom x1 (Fig. 12).

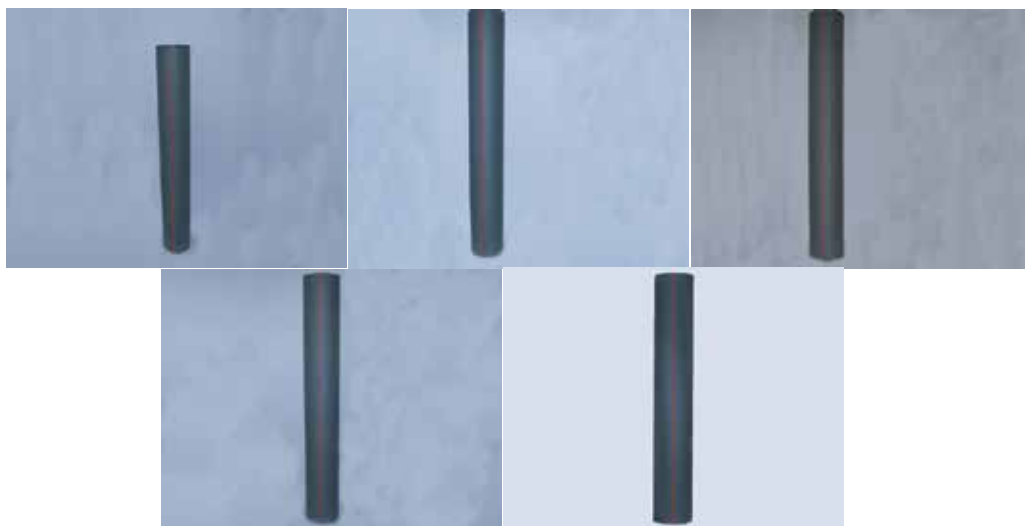


Fig. 12. The cylindrical etalon

Three successive and dependent calibrations are performed: Horizontal (pan): position and orientation are obtained in two successive, but dependent steps; Vertical (tilt): position and orientation are obtained in two successive, but dependent steps; Zoom: tuning the two cameras as both look to the cylinder from virtual equal distances.

Both offsets must be under the accepted thresholds. Else, the positioning destroyed the orientation and the procedure must be repeated. A similar algorithm is developed for the vertical orientation and positioning.

The second algorithm works together with the graphic simulator. It was proven that the two camera axes are orthogonal if, when both cameras are looking at the tentacle successively bended as circle's arcs in two orthogonal planes, are seeing also two circle's arcs (Fig. 13). The previous condition is fulfilled if each camera looks at the center of the circle containing the arc and the view line is orthogonal on the plane's circle.

Three calibration steps must be performed: Horizontal calibration - positioning and orienting the camera horizontally (pan); Vertical calibration - positioning and orienting the camera vertically (tilt); Zoom calibration - tuning the two cameras as both look at the robot from virtual equal distances.

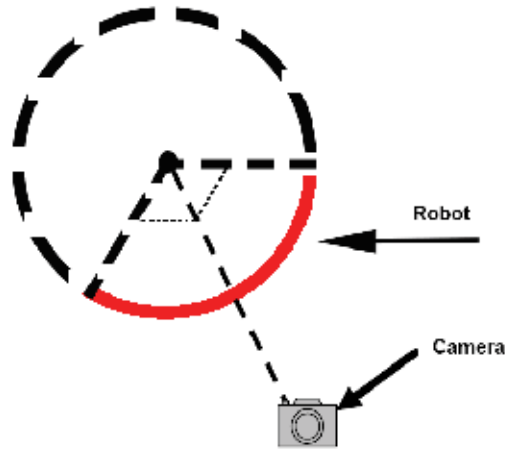


Fig. 13. Camera looks to the center of the circle

How to “move” the camera according to the steps of these algorithms? The image behavior in accordance with camera’s movements was studied. The effect of pan and tilt rotations on two points placed in a quadratic position on a circle was geometrically described. Coordinate transformation matrices corresponding to rotations with pan and tilt angles, respectively for perspective transformation were used. The variation of the distance between the two points, placed in a quadratic position on the circle, and the centre of the circle, depending of the tilt angle X , is plotted bellow in Fig. 14.

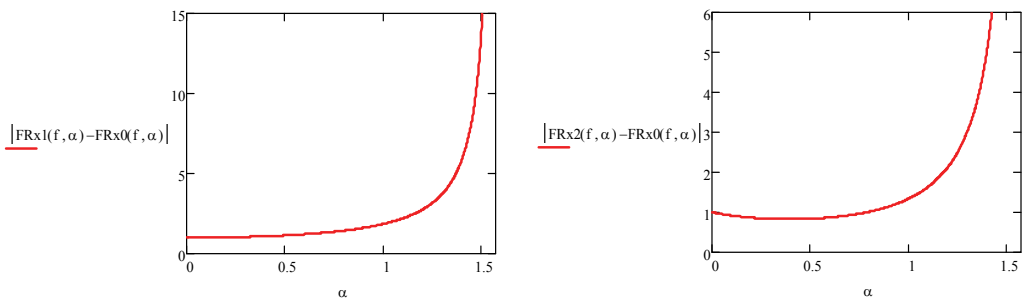


Fig. 14. Distance variation for quadratic positions

The variation of the ratio of the two distances is plotted bellow in Fig. 15a. The plot from Fig. 15b shows how is transformed a rectangle (inscribed in the circle and having the edges parallel with the axes OX and OY) when a tilt rotation is performed. Theoretically, by zooming, the distance between the two points varies in a linear way, as it is shown upper right.

The image’s segmentation is basically a threshold procedure applied to the image’s histogram. All the procedures included in the calibration algorithms were mathematically proven. If the calibration algorithm was successfully applied then the system is ready to perform the visual servoing tasks.

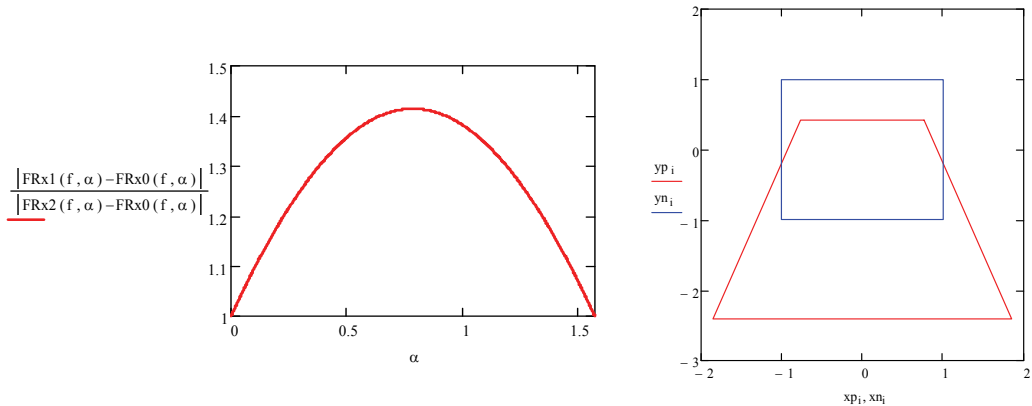


Fig. 15. a. Ratio distances variation b. Rectangle transformation and distance variation under zoom influence

5. A Compliance control of a hyper redundant robot

This section treats a class of hyper redundant arms can achieve any position and orientation in 3D space, and that can perform a coil function for the grasping. The arm is a high degree of freedom structure or a continuum structure, but in this chapter a different technological solution is assumed.

The general form of the arm is shown in Figure 16. It consists of a number (N) of elements, cylinders made of fibre-reinforced rubber.

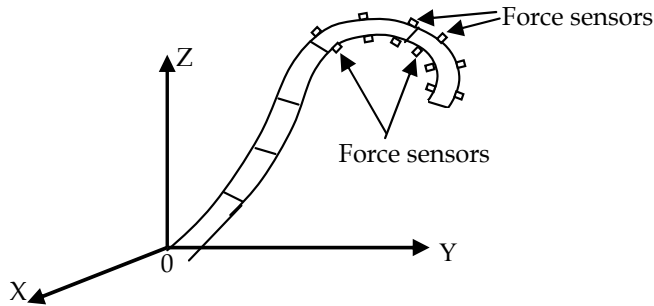


Fig. 16. The force sensors distribution

There are four internal chambers in the cylinder, each of them containing the ER fluid with an individual control circuit. The deformation in each cylinder is controlled by an independent electrohydraulic pressure control system combined with the distributed control of the ER fluid.

The last m elements ($m < N$) represent the grasping terminal. These elements contain a number of force sensors distributed on the surface of the cylinders. These sensors measure the contact with the load and ensure the distributed force control (Singh & Popa, 2005) during the grasping. The theoretical model is described as in Fig. 7 and equation (26)-(33).

For an element dm , kinetic and gravitational potential energy will be:

$$dT = \frac{1}{2} dm (v_x^2 + v_y^2 + v_z^2), \quad dV = dm \cdot g \cdot z \quad (101)$$

where $dm = \rho \cdot ds$, and ρ is the mass density.

The elastic potential energy will be approximated by the bending of the element:

$$V_e = k \frac{d^2}{4} \sum_{i=1}^N (q_i^2 + \theta_i^2) \quad (102)$$

We will consider $F_\theta(s, t)$, $F_q(s, t)$ the distributed forces on the length of the arm that determine motion and orientation in the θ -plane, q -plane. The mechanical work is:

$$L = \int_0^l \int_0^t (F_\theta(s, \tau) \dot{\theta}(s, \tau) + F_q(s, \tau) \dot{q}(s, \tau)) d\tau ds \quad (103)$$

The energy-work relationship will be

$$L = \int_0^l \int_0^t (F_\theta(s, \tau) \dot{\theta}(s, \tau) + F_q(s, \tau) \dot{q}(s, \tau)) d\tau ds \quad (104)$$

where $T(t)$ and $T(0)$, $V^*(t)$ and $V^*(0)$ are the total kinetic energy and total potential energy of the system at time t and 0 , respectively.

In this chapter, the manipulator model is considered as a distributed parameter system defined on a variable spatial domain $\Omega = [0, L]$ and the spatial coordinate s .

From (101-103), the distributed parameter model becomes,

$$\begin{aligned} & \rho \int_0^s \int_0^s (\ddot{q}' (\sin q' \sin q'' \cos(q' - q'') + \cos q' \cos q'') - \ddot{\theta}' \cos q' \sin q'' \sin(\theta'' - \theta') + \\ & + (\dot{q}')^2 (\cos q' \sin q'' \cos(\theta' - \theta'') - \sin q' \cos q'') + (\dot{\theta}')^2 \cos q' \sin q'' \cos(\theta' - \theta'') - \\ & - \dot{q}' \dot{q}'' \sin(q'' - q')) ds' ds'' + \rho g \int_0^s \cos q' ds' + k^* q = F_q \end{aligned} \quad (105)$$

$$\begin{aligned} & \rho \int_0^s \int_0^s (\ddot{q}' \sin q' \cos q'' \sin(\theta'' - \theta') + \ddot{\theta}' \cos q' \cos q'' \cos(\theta'' - \theta') - (\dot{q}')^2 \cos q' \cos q'' \sin(\theta'' - \theta') + \\ & + (\dot{\theta}')^2 \cos q' \cos q'' \sin(\theta'' - \theta') - \dot{\theta}' \dot{q}' \sin q' \cos q'' \cos(\theta'' - \theta')) ds' ds'' + k^* \theta = F_\theta \end{aligned} \quad (106)$$

The control forces have the distributed components along the arm, $F_\theta(s, t)$, $F_q(s, t)$, $s \in [0, L]$ that are determined by the lumped torques,

$$\begin{cases} F_\theta(s, t) = \sum_{i=1}^N \delta(s - il) \tau_{\theta_i}(t) \\ F_q(s, t) = \sum_{i=1}^N \delta(s - il) \tau_{q_i}(t) \end{cases} \quad (107)$$

where δ is Kronecker delta, $l_1 = l_2 = \dots = l_N = l$, and

$$\tau_{\theta_i}(t) = (p_{\theta_i}^1 - p_{\theta_i}^2) S \cdot d / \delta \quad (108)$$

$$\tau_{q_i}(t) = (p_{q_i}^1 - p_{q_i}^2) S \cdot d / \delta, \quad i = 1, 2, \dots, N \quad (109)$$

In (107)-(108), $p_{\theta_i}^1$, $p_{\theta_i}^2$, $p_{q_i}^1$, $p_{q_i}^2$ represent the fluid pressure in the two chamber pairs, θ , q and S , d are section area and diameter of the cylinder, respectively (Fig. 17).

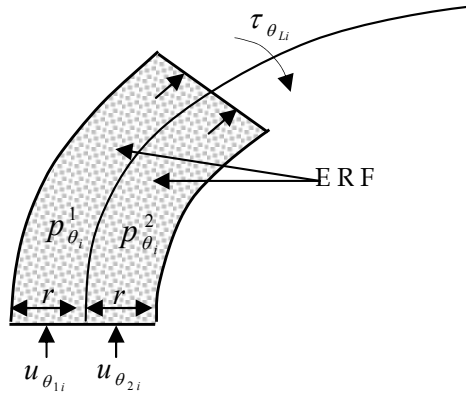


Fig. 17. The cylinder driving

The pressure control of the chambers is described by the equations:

$$a_{ki}(\theta) \frac{dp_{\theta_i}^k}{dt} = u_{\theta_{ki}} \quad (110)$$

$$b_{ki}(q) \frac{dp_{q_i}^k}{dt} = u_{q_{ki}}, \quad k = 1, 2, \quad i = 1, 2, \dots, N \quad (111)$$

where $a_{ki}(\theta)$, $b_{ki}(q)$ are determined by the fluid parameters and the geometry of the chambers and

$$a_{ki}(0) > 0, \quad b_{ki}(0) > 0 \quad (112)$$

The control problem of a grasping function by coiling is constituted from two subproblems: the position control of the arm around the object-load and the force control of grasping (Chiaverini et al., (1996). We consider that the initial state of the system is given by

$$\omega_0 = \omega(0, s) = [\theta_0, \quad q_0]^T \quad (113)$$

corresponding to the initial position of the arm defined by the curve C_0

$$C_0 : (\theta_0(s), \quad q_0(s)), \quad s \in [0, \quad L] \quad (114)$$

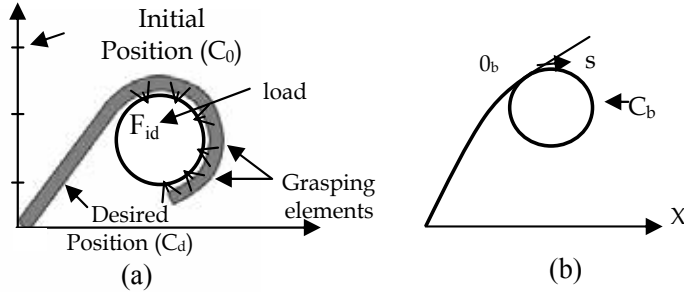


Fig. 18. (a) The grasping position; (b) The grasping parameters

The desired point is represented by a desired position, the curve C_d that coils the load,

$$\omega_d = [\theta_d, q_d]^T \quad (115)$$

$$C_d : (\theta_d(s), q_d(s)), s \in [0, L] \quad (116)$$

In a grasping function by coiling, only the last m elements ($m < N$) are used. Let l_g be the active grasping length, where $l_g = \sum_{i=m}^n l_i$. We define by $e_p(t)$ the position error

$$e_p(t) = \int_{L-l_g}^L ((\theta(s,t) - \theta_b(s)) + (q(s,t) - q_b(s))) ds \quad (117)$$

It is difficult to measure practically the angles θ, q for all $s \in [0, L]$. These angles can be evaluated at the terminal point of each element. In this case, the relation (117) becomes

$$e_p(t) = \sum_{i=m}^N ((\theta_i(t) - \theta_{bi}) + (q_i(t) - q_{bi})) \quad (118)$$

The error can also be expressed with respect to the global desired position C_d

$$e_p(t) = \sum_{i=1}^N ((\theta_i(t) - \theta_{di}) + (q_i(t) - q_{di})) \quad (119)$$

The position control of the arm means the motion control from the initial position C_0 to the desired position C_d in order to minimize the error. An area reaching control problem is discussed. The desired area is specified by the inequality function:

$$f(\delta r) \leq 0 \quad (120)$$

where f is a scalar function with continuous first partial derivatives, $\delta r = r_f - r_0$, $r_0 \in R^3$ is a reference point of the desired area and r_f is the position vector of the terminal point. The potential energy function for the area reaching control has the form:

$$\tau_{\theta_i}(t) = -k_{\theta_i} e_{\theta_i}(t) - k_{\dot{\theta}_i}^* \dot{e}_{\theta_i}^2(t) - \max \left(0, \frac{\partial V_P^T}{\partial r} \cdot k_{p_{\theta_i}} a^*(\theta_i, q_i) \right) \quad (121)$$

Theorem 1. The closed-loop control system for the desired reaching area problem is stable if the control forces are

$$\tau_{\theta_i}(t) = -k_{\theta_i} e_{\theta_i}(t) - k_{\theta_i}^* \dot{e}_{\theta_i}^2(t) - \max(0, \partial V_p^T / \partial r \cdot k_{p_{\theta_i}} a^*(\theta_i, q_i)) \quad (122)$$

$$\tau_{q_i}(t) = -k_{q_i} e_{q_i}(t) - k_{q_i}^* \dot{e}_{q_i}^2(t) - \max(0, \partial V_p^T / \partial r \cdot k_{p_{q_i}} a^*(\theta_i, q_i)) \quad (123)$$

Theorem 2. The closed-loop control system of the position (107)-(108), (110)-(111) is stable if the fluid pressures control law in the chambers of the elements given by:

$$u_{\theta_{ji}}(t) = -a_{ji}(\theta) (k_{\theta_i}^{j1} \dot{e}_{\theta_i}(t) + k_{\theta_i}^{j2} \ddot{e}_{\theta_i}(t)) \quad (124)$$

$$u_{q_{ji}}(t) = -b_{ji}(\theta) (k_{q_i}^{j1} \dot{e}_{q_i}(t) + k_{q_i}^{j2} \ddot{e}_{q_i}(t)) \quad (125)$$

where $j = 1, 2$; $i = 1, 2, \dots, N$, with initial conditions:

$$p_{\theta_i}^1(0) - p_{\theta_i}^2(0) = (k_{\theta_i}^{11} - k_{\theta_i}^{21}) e_{\theta_i}(0) \quad (126)$$

$$p_{q_i}^1(0) - p_{q_i}^2(0) = (k_{q_i}^{11} - k_{q_i}^{21}) e_{q_i}(0) \quad (127)$$

$$\dot{e}_{\theta_i}(0) = 0 \quad (128)$$

$$\dot{e}_{q_i}(0) = 0, \quad i = 1, 2, \dots, N \quad (129)$$

and the coefficients k_{θ_i} , k_{q_i} , $k_{\theta_i}^{mn}$, $k_{q_i}^{mn}$ are positive and verify the conditions

$$k_{\theta_i}^{11} > k_{\theta_i}^{21}; \quad k_{\theta_i}^{12} > k_{\theta_i}^{22} \quad (130)$$

$$k_{q_i}^{11} > k_{q_i}^{21}; \quad k_{q_i}^{12} > k_{q_i}^{22}, \quad i = 1, 2, \dots, N \quad (131)$$

The grasping by coiling of the continuum terminal elements offers a very good solution in the fore of uncertainty on the geometry of the contact surface. The contact between an element and the load is presented in Fig. 19. It is assumed that the grasping is determined by the chambers in θ -plane. The relation between the fluid pressure and the grasping forces can be inferred for a steady state from,

$$\int_0^l k \frac{\partial^2 \theta(s)}{\partial s^2} ds + \int_0^l f(s) \tilde{T} \tilde{\theta}(s) \int_0^s \tilde{T}^T \tilde{\theta}(s) ds = (p_1 - p_2) S \frac{d}{8} \quad (132)$$

where $f(s)$ is the orthogonal force on C_b , $f(s)$ is $F_\theta(s)$ in θ -plane and $F_q(s)$ in q -plane. For small variation $\Delta\theta_i$ around the desired position θ_{id} , in θ -plane, the dynamic model (118) can be approximated by the following discrete model,

$$m_i \Delta \ddot{\theta}_i + c_i \Delta \dot{\theta}_i + H_i(\theta_{id} + \Delta\theta_i, \theta_{id}, q_d) - H(\theta_{id}, q_d) = d_i (f_i - F_{ci}) \quad (133)$$

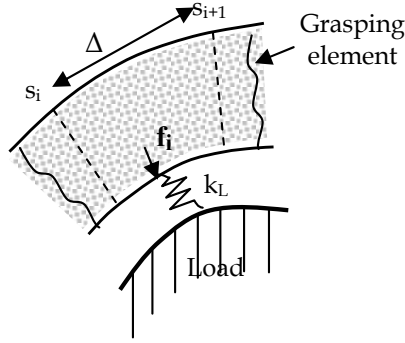


Fig. 19. The grasping force

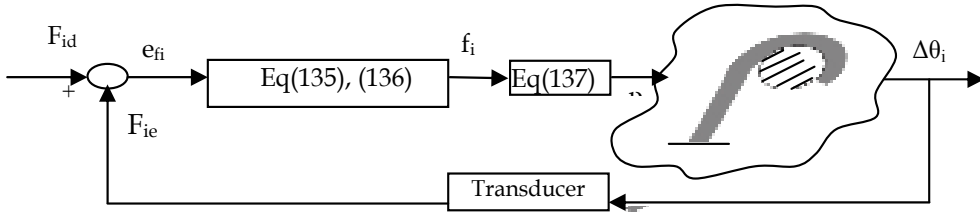


Fig. 20. The block scheme of the control system

where $m_i = \rho S \Delta$, $i = 1, 2, \dots, l_1$. $H(\theta_{id}, q_d)$ is a nonlinear function defined on the desired position (θ_{id}, q_d) , $c_i = c_i(v, \theta_i, q_d)$, $c_i > 0$, $\theta, q \in \Gamma(\Omega)$, where v is the viscosity of the fluid in the chambers. The equation (133) becomes:

$$m_i \Delta \ddot{\theta}_i + c_i(v, \theta_i, q_d) \Delta \dot{\theta}_i + h_i(\theta_{id}, q_d) \cdot \Delta \theta_i = d_i (f_i - F_{ei}) \quad (134)$$

The aim of explicit force control is to exert a desired force F_{id} . If the contact with load is modelled as a linear spring with constant stiffness k_L , the environment force can be modelled as $F_{ei} = k_L \Delta \theta_i$. The error of the force control may be introduced as

$$e_{fi} = F_{ie} - F_{id} \quad (135)$$

It may be easily shown that the equation (134) becomes

$$\frac{m_i}{k_L} \ddot{e}_{fi} + \frac{c_i}{k_L} \dot{e}_{fi} + \left(\frac{h_i}{k} + d_i \right) e_{fi} = d_i f_i - \left(\frac{h_i}{k} + d_i \right) F_{id} \quad (136)$$

Theorem 3. The closed force control system is asymptotic stable if the control law is

$$f_i = \frac{1}{k_L d_i} \left((h_i + k_L d_i + m_i \sigma^2) e_{fi} - (h_i - k_L d_i) F_{id} \right), \quad c_i > m_i \sigma \quad (137)$$

6. Conclusion

The research group from the Faculty of Automation, Computers and Electronics, University of Craiova, Romania, started working in research field of hyper redundant robots over 25 years ago. The experiments used cables and DC motors or stepper motors. The rotation of these motors rotates the cables which by correlated screwing and unscrewing of their ends determine their shortening or prolonging, and by consequence, the tentacle curvature.

The inverse kinematics problem is reduced to determining the time varying backbone curve behaviour. New methods for determining "optimal" hyper-redundant manipulator configurations based on a continuous formulation of kinematics are developed.

The difficulty of the dynamic control is determined by integral-partial-differential models with high nonlinearities that characterize the dynamic of these systems. First, the dynamic model of the system was inferred. The method of artificial potential was used for these infinite dimensional systems. In order to avoid the difficulties associated with the dynamic model, the control law was based only on the gravitational potential and a new artificial potential.

The control system is an image - based visual servo control. Servoing was based on binocular vision, a continuous measure of the arm parameters derived from the real-time computation of the binocular optical flow over the two images, and is compared with the desired position of the arm. The method is based on the particular structure of the system defined as a "backbone with two continuous angles". The control of the system is based on the control of the two angles. The error angle was used to calculate the spatial error and a control law was synthesized. The general control method is an image based visual servoing one instead of position based. By consequence, camera calibration based on intrinsic parameters is not necessary („calibration" in the classic sense of the term, not the one used in this paper). The term "camera calibration" in the context of this paper refers to positioning and orienting the two cameras at imposed values. This calibration is performed only at the beginning, after that the cameras remain still.

A new application investigates the control problem of a class of hyper-redundant arms with continuum elements that performs the grasping function by coiling. The control problem of a grasping function by coiling is constituted from two subproblems: the position control of the arm around the object-load and the force control of grasping.

7. Acknowledgement

The research presented in this paper was supported by the Romanian National University Research Council CNCSIS through the IDEI Research Grant ID93 and by FP6 MARTN through FREESUBNET Project no. 36186.

8. References

- Blessing, B.; & Walker, I.D. (2004). Novel Continuum Robots with Variable- Length Sections, *Proc. 3rd IFAC Symp. on Mechatronic Systems*, Sydney, Australia, pp. 55-60.
- Boccolato, G.; Dinulescu, I.; Predescu, A.; Manta, F.; Dumitru, S.; & Cojocaru, D.; (2010). 3D Control for a Triconic Tentacle, *12th International Conference on Computer Modelling and Simulation*, p380-386, ISBN 978-0-7695-4016-0,

Cambridge University, England.

- Ceah, C.C. & Wang, D.Q. (2005). Region Reaching Control of Robots: Theory and Experiments, *Proceedings of IEEE Intl Conf on Rob. and Aut.*, Barcelona, pp. 986-991.
- Chiaverini, C.; Siciliano, B. & Villani, L. (1996). Force and Position Tracking: Parallel Control with Stiffness Adaptation, *IEEE Control Systems*, Vol. 18, No 1, pp. 27-33.
- Chirikjian, G.S. (1993). A continuum approach to hyper-redundant manipulator dynamics, *Proc. 1993 Int. Conf. on Intelligent Robots and Syst.*, Yokohama, Japan, pp. 1059 - 1066.
- Cojocaru, D.; Ivanescu, M.; Tanasie, R.T.; Dumitru, S.; Manta, F. (2010), Vision Control for Hyperredundant Robots, *International Journal Automation Austria (IJAA)*, ISSN 1562-2703, IFAC-Beirat Österreich, Vol. 1, 18(2010), p52-66.
- Cowan, L. S. & Walker, I.D., 2008. "Soft" Continuum Robots: the Interaction of Continuous and Discrete Elements, *Artificial Life X*.
- Douskaia, N.V. (1998). Artificial potential method for control of constrained robot motion, *IEEE Trans. on Systems, Man and Cybernetics*, part B, vol. 28, pp. 447-453.
- Ge, S.S.; Lee, T.H. & Zhu, G. (1996). Energy-Based Robust Controller Design for Multi-Link Flexible Robots, *Mechatronics*, No 7, Vol. 6, pp. 779-798.
- Gravagne, I. D. & Walker, I.D. (2001). Manipulability, force, compliance analysis for planar continuum manip, *Proc. IEEE/RSI Intl. Conf. o Intell. Rob. and Syst.*, pp. 1846-1867.
- Grosso, E.; Metta, G. & a.o. (1996). Robust Visual Servoing in 3D Reaching Tasks, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 15, pp. 732-742.
- Hannan, M.W. & Walker, I.D. (2005). Real-time shape estimation for continuum robots using vision, *Robotica*, volume 23, pp. 645-651.
- Hemami, A. (1984). Design of light weight flexible robot arm, *Robots 8 Conference Proceedings*, Detroit, USA, pp. 1623-1640.
- Hirose, S. (1993). *Biologically Inspired Robots*, Oxford University Press.
- Hutchinson, S.; Hager, G. D. & Corke, P. F. (1996). A Tutorial on Visual Servo Control, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 15, pp. 651-670.
- Immega, G. & Antonelli, K. (1995). The KSI Tentacle Manipulator. *Proc. 1995 IEEE Conf. on Robotics and Automation*, pp. 3149-3154.
- Ivănescu, M.; Cojocaru, & a.o. (2006). Hyperredundant Robot Control by Visual Servoing, *Studies in Informatics and Control Journal*, Vol. 15, No. 1, ISSN 1220-1766, p93-102.
- Ivanescu, M. (2002). Position dynamic control for a tentacle manipulator, *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, A1-15, pp. 1531-1539.
- Kelly, R. (1996), Robust Asymptotically State Visual Servoing, *Proceedings IEEE International Conference on Robotics and Automation*, vol. 22, no. 15, pp. 759-765.
- Masoud, S. A. & Masoud, A.A. (2000). Constrained motion control using vector potential fields, *IEEE Trans. on Systems, Man and Cybernetics*, part A, vol. 30, pp. 251-272.
- Mochiyama, H. & Kobayashi, (1999). H. The shape Jacobian of a manip with hyper degrees of freedom, *Proc. 1999 IEEE Intl. Conf. on Rob. and Autom.*, Detroit, pp. 2837- 2842.
- Robinson, G. & Davies, J. B. C. (1999). Continuum robots—A state of the art. *In IEEE International Conference on Robotics and Automation*, pages 2849-2854. Detroit, MI.

- Singh, S.K. & Popa, D.O. (2005) An Analysis and Some Fundamental Problems in Adaptive Control of Force, *IEEE Trans. on Robotics and Automation*, Vol. 11 No 6, pp 912-922.
- Suzumori, K.; Iikura, S.; & Tanaka, H. (1991). Develop. of flexible microactuator and its appl. to robot mech, *IEEE Intl. Conf. on Rob. and Autom.*, Sacramento, pp. 1564 - 1573.
- Takegaki, T.; & Arimoto, S. (1981). A new feedback methods for dynamic control of manipulators, *Journal of Dynamic Systems, Measurement and Control*, pp. 119-125.
- Tanasie, R.T.; Ivănescu, M. & Cojocaru, D. (2009). Camera Positioning and Orienting for Hyperredundant Robots Visual Servoing Applications, *Journal of Control Engineering and Applied Informatics*, ISSN 1454-8658, Vol 11, No 1, p19-26.
- Walker, I.D., Dawson, D.M. & a.o. (2005), Continuum Robot Arms Inspired by Cephalopods, DARPA Contr. N66001-C-8043, <http://www.ces.clemson.edu>.
- Walker, I.D. & Carreras, C. (2006) Extension versus Bending for Continuum Robots, *Internl. Journal of Advanced Robotic Systems*, Vol. 3, No.2, ISSN 1729-8806, pp. 171-178.
- Wang, P.C. (1965). Control of distributed parameter systems, *Advance in Control Systems*, Academic Press.

Micro-Manipulator for Neurosurgical Application

M. R. Arshad and Ya'akob Yusof

*Underwater Robotics Research Group (URRG),
School of Electrical and Electronics Engineering, Universiti Sains Malaysia,
Malaysia*

1. Introduction

Neurosurgery is a part of the surgical field that focused in taking care of the diseases related to human's central peripheral nervous system and also their central spinal cord [20]. The term surgery refers to the operation of peripheral nervous system as well as the spinal cord, brain, blood vessel connected to it, spine, spinal cord, and also nerves that control our senses and body's movement [29]. There are lots of neuro diseases, which among them were brain tumors, head trauma, stroke, thalamic astrocytomas, and spinal cord trauma. These diseases, if not thrown away, will results the patient in body disorder, health problem, and of course, death. To put an end to these disorders, appropriate treatment is mandatory. Those diseases need to be cured and removed. Surgery, or specifically neurosurgery, is one of the effective methods to treat it.

Neurosurgery comes with risks. Any operation dealing with brain or the spinal cord can cause paralysis, brain damage, infection, psychosis, or even death if a mistake happens. These operations are also likely to cause mental impairment as of any surgical procedure dealing with the brain. Therefore, it is vital for neurosurgeon to make sure that this kind of surgery is performed in an almost perfect condition to minimize any risks or poor results as the consequences from it. Traditionally, starting from scalp removing, drilling and removing the skull, handling the lump, until sewing the skull and scalp back at its original location; surgeons put their efforts with their own hands and bare eyes. Tools and equipments did improved, for example with the usage of apparatus such as top-mount microscope and magnetic resonance imaging (MRI) machine. However, they still need to manipulate the surgical tools, the closest tools to the human brain, such as the knife and biopsy needles with naked hands. As a results, it will surely introduced limits to the tools manipulation. This is where robots can do a lot better. A very precise robotic device that can perform manipulation at much smaller or micro scale, plus the capability of the surgeon himself, will produce much superior results. These robotic devices are termed surgical micro-manipulator.

This chapter presents readers with information regarding the design of a micro-manipulator purposely for neurosurgical application. It also shares beneficial facts and particulars regarding current progress about micro-manipulator research around the globe. This chapter is organized as the following: Section 2 provides details justification of designing a robotic hand in an operating room based on the constraints for a neurosurgical procedures. Section 3 will discuss design considerations for a micro-manipulator for neurosurgery. This

includes the important hardware and software elements that contributed to the build-up of a micromanipulator. Section 4 briefly shares on the design and uniqueness of one of the recent and successful micro-manipulator for neurosurgical application. Section 5 will finally conclude this article.

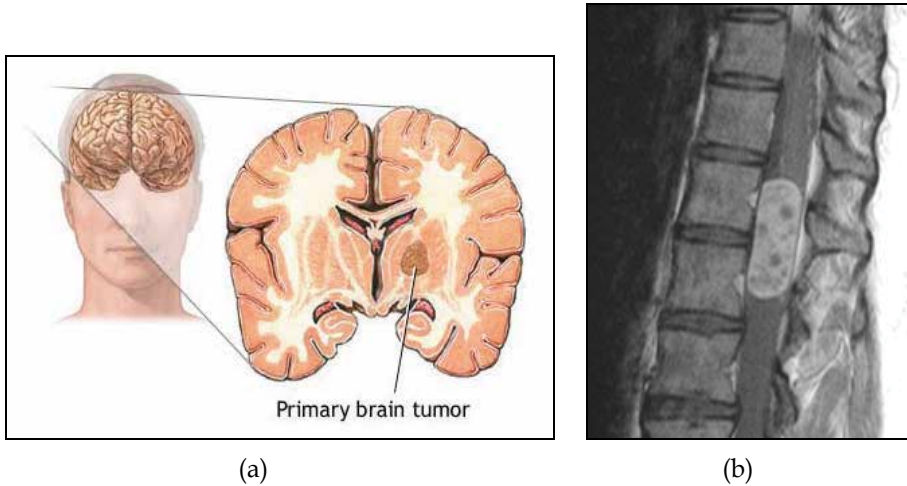


Fig. 1. (a) Graphical illustration of brain tumor. A primary brain tumor is a mass created by the growth or uncontrolled proliferation of cells in the brain. (b) Spinal tumor

2. Why robotics



Fig. 2. Da Vinci® tele-surgical system.

The idea of having robots inside the operation theatre is basically to assist the neurosurgeon perform the surgery. Fatigue experienced by surgeons, post-surgery trauma on the patient and human errors are among the challenges faced during neurosurgical operation [10]. According to [6], there are studies being done that shows during a long surgical operation, there will be substantial muscle fatigue. Neurosurgical or any surgical procedure usually takes a very long time, thus will decrease the effectiveness of a surgeon. In contrast, robots will never experience fatigue because their moves are controlled by devices. Moreover, they

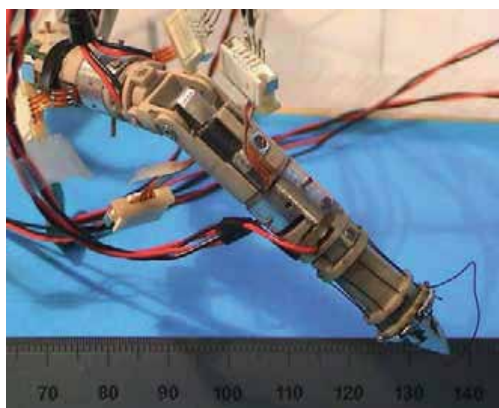
can be very precise and reliable because robot can filter the handshakes and keep the operation steady.

Another reason surgeons need to use such a system is that it can provide them with a minimally invasive surgery (MIS). This provides less trauma for the patient after the surgery and of course, a shorter recovery period. Moreover, human involvement is also a concern. In today's operating rooms, you'll find two or three surgeons, an anesthesiologist and several nurses, all needed for even the simplest of surgeries. Most surgeries require nearly a dozen people in the room. As with all automation, surgical robots will eventually eliminate the need for some personnel. Taking a glimpse into the future, surgery may require only one surgeon, an anesthesiologist and one or two nurses. In this nearly empty operating room, the doctor sits at a computer console, either in or outside the operating room, using the surgical robot to accomplish what it once took a crowd of people to perform.

The first use of robot in a neurosurgical procedure is in 1985, according to [30]. Researches from Department of Radiology, Memorial Medical Center employed a PUMA (Programmable Universal Machine for Assembly) robot in the operating room. Even though the task of the robot at that time is only to hold and manipulate biopsy cannulae, it marked the start of a robot's manipulator cooperation inside the operation room. Since then, various researches in various aspect of neurosurgery have been explored. Those included the micromanipulators design [22], vision and imaging scheme, sensors design [16], haptic technology [9], magnetic resonance imaging (MRI) compatibility equipments, telesurgery system [15], as well as controller technique and planning.

2.1 What is a micro-manipulator?

The term manipulator in robotics means a device or equipment that allows for movement of a part through multiple joints on the mechanical device. It is also better known as robotic arm [26]. Micro-manipulator also carries the same meaning, but the term 'micro' referred it to a more specific task, which is object handling in small (micro) scale. In dealing with neurosurgical procedure, precision and accuracy plays a very important role. This situation leads to the needs of micro-manipulation, using micro-manipulator. This further explains that the level of manipulation is very small and the accuracy in need is very high. It is not



(a)



(b)

Fig. 3. (a) A micro-manipulator for surgery, designed by Francesco Cepolina, [4] & [7]
(b) NeuroMaster, a stereotactic neurosurgery robot system by Beihang University, [13] & [18]

necessarily that the tool and manipulator must be small, but the whole system itself must be able to integrate and produce very precise micro-manipulation with a very minimal error in all the 3- axis's direction. This includes the sensors parts, vision and imaging system and also the controller technique.

2.2 Type of robotics involvement in operation theatre / operation room

According to [27], robotic involvement in surgical procedure can be divided into three categories. These categories were based on robot and surgeon interaction during the procedure.

The first category is *supervisory-controlled system*. This is where robots performed the surgery by implementing specific instructions and paths set earlier by the surgeon. Those paths were planned during planning and registration process before the operation, which integrates the images from the MRI scanning process. In this scheme, surgeon is still playing the important roles, which is planning and setting up the whole operation's path. Then, the robot will do everything that has been pre-planned, while the surgeon supervising the operation. Though, the surgeon did not directly use his hands during this procedure.

The second category, which is the *telesurgical system*, needs the surgeon to use his own hands during the process. Also known as remote surgery, this is different from the previous technique. However, it is the robotic manipulator who is doing the real operation on the patient. The surgeon, on the other hand, is manipulating the robot through some distance from a computer console. In this method, sensors including haptic feedback system are providing the surgeon with all the necessary data for the surgeon to react with. The computer console is the master, while the manipulator is called slave. The operation being done by the robotic manipulator is imitating the master controller's movement in real time. The most popular and widely used telesurgical system is the da Vinci® Surgical System manufactured by Intuitive Surgical, where more than 1000 units sold worldwide [5].

The last category is the *shared-control system*. This is where surgeon and robot works together at the same time, where a number of specifically designed tasks were done by the doctor and others by the robot. This system, compared to the previous two, has the most surgeon involvement in the operation theatre. Figure 3(b) is an example of this.

4. Discussion

The buildup of a neurosurgical type micro-manipulator usually consists of few important parts or elements, both hardwares and softwares. These elements were essential to ensure the specification and performance of the robot itself achieved the function of a micromanipulator. This description may act like a general guideline in developing the micromanipulator because some of the micro-manipulator might not have all the elements, because each of them has different specifications and design.

4.1 Modeling

Modeling is a process of using mathematical description to simulate real physical events [17]. It allows complex systems to be understood and thus their behavior can be predicted and simulated. In modeling, usually some details will be ignored or assumed, due to the shortcomings occurred in the process. Micro-manipulator is a very expensive system to build.

Thus, it needs to be modeled before it is fabricated. From the model, we can investigate much information, such as the kinematic and dynamic behavior of the model, the workspace of the model, materials to build the model, suitable actuator to achieve the design objectives and the controller technique that is most efficient to the system. There are lots of software that can be used to model a micro-manipulator system, including Robotics Toolbox® and SimMechanics from Matlab, AutoCAD, SolidWorks and Rhinoceros®.

One common technique to model a manipulator system is to use Denavit-Hartenberg (DH) method [12]. From the DH model, either the classical or the modified version of it, we can simulate and further investigate the behavior of the micro-manipulator, both the kinematic and dynamic. Kinematics relates the motion behavior of the robot without regards to the forces that causes it, whereas dynamic considers the effect of internal and external forces or torques applied to the micro-manipulator. With the information from both the kinematic and dynamic behavior, we can have a good knowledge on how the micro-manipulator moves, which path it follows, how many micro-Newton of forces applied at the patient's head, how precise the robot is, as well as the speed of the robot movements. Those are among the vital information needed by the surgeon during his usage of the micro-manipulator during a surgery. In addition, we can always estimate the workspace of the robot. Workspace is the region where the end effector of the micro-manipulator can possibly reach, which a surgeon needs to know prior to an operation to estimate the tools arrangements and movements.

Equation 1 below represents the transformation matrices associated with modified DH method. The parameters ' α ' and ' θ ' represents the angular behavior of the micromanipulator's links, while the ' a ' and ' d ' parameters represents the prismatic aspect.

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & \alpha_{i-1} \\ \sin \theta_i \times \cos \alpha_{i-1} & \cos \theta_i \times \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} \times d_i \\ \sin \theta_i \times \sin \alpha_{i-1} & \cos \theta_i \times \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} \times d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Equation 2 shows the equations of motion, in general, of a micro-manipulator.

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) \quad (2)$$

where;

τ = torques vector

M = inertia matrix

C = Coriolis and Centrifugal matrix (these are types of internal forces)

N = gravity terms and other forces act on the joints (all external forces defines here)

4.2 Trajectory planning

Trajectory refers to time domain of the position, velocity and acceleration of a system [23]. It described the motion's behavior of the micro-manipulator in all of the 3-dimensional (3-D) axis. The trajectories were generated through interpolation or approximation of the desired path by a polynomial or any other smooth function. The function is used to approximate and provide the mathematical description of the trajectory.

There are two categories of trajectory planning techniques. They are joint space technique and Cartesian space technique. Joint space technique is suitable for point-to-point motion, where the motion planning is done at the joint level. This technique describes the time function of all the joints' variables including the speed and acceleration. Equation 3 below shows an example of a five-degree or quintic polynomial equations with its joints' variables, C_i . The position of the end effector was computed by using forward kinematics. This is just one of many smooth functions that can be used to interpolate a trajectory.

Cartesian space technique is a method that most suited with continuous path type of motion, and therefore suit neurosurgical application better than the previous technique. While joint space method focus on the joint position, Cartesian space method controls the end effector itself, with respect to the base of the robot. By using inverse kinematics, the joints variables were computed.

In general, to set a trajectory, we must define the starting and end points, as well as the mathematical function that the joints and end effector will undertake during its movement. The time taken to complete the trajectory is also important as it will affect the speed and smoothness of the manipulation. This is important in designing a micro-manipulator since we need to specify each and every point on the route of the end-effector and the joints as well. Failing to do this will let us lose control of the surgical tools.

$$\theta(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \quad (3)$$

where;

$$c_0 = \theta_0$$

$$c_1 = \dot{\theta}_0$$

$$c_2 = \frac{\ddot{\theta}_0}{2}$$

$$c_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3}$$

$$c_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}$$

$$c_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}$$

4.3 Actuator

This is an equipment that allows a robot to move by conversion of different energy types such as electrical or mechanical processes [26]. This includes human muscles, propellers, and hydraulic cylinders. Actuators are very important because it's the main mechanism to make the robot being in specified motion. For micro-manipulator, type of actuator that is widely used is the electrically controlled motors. Electrical motors are favored because it is much more precise and accurate in terms of their generated motion, as compared to the hydraulic and pneumatic-actuated motor.

Accuracy to the highest level is very important for a micro-manipulator design. Thus, the selection of a suitable actuator is very important. In [1], Adha Cahyadi et al use piezo

electric actuator in their micro-manipulator design. With the special capability of piezo material, it can produce a very fine displacement, down till micrometers. This shows the importance of selecting the right actuator. It really depends on the micro-manipulator design. Moreover, suitable controller implementation technique can also contribute well to manage the output of an actuator.

4.4 Sensors

Sensors are very important devices that allow the analog world to communicate with the digital environment. By definition, it converts physical signals such as heat, light, sound, rotary motion, and force into electrical signal [24]. The resulting electrical signal will be send to the controller and the required calculation or assigned resulting action will be taken. A precise rotary encoder for example, can provide the system with the exact location of each joints and the end effector. Then by using various types of control method, their locations can be corrected if it does slightly differ from the desired one.

In micro-manipulator design, there are many types sensor that is very useful to be incorporated with. Among them is vision or imaging sensor and force or haptic devices. Imaging is a key element for a robotic neurosurgery. It can be used during the registration process before the surgery to organize the surgery through coordinate relationship between the robot frame and the patient head's frame [13]. In addition, MRI images can also help the surgeon. Using images from the MRI and special software, the patient head can be redraw in the computer, and allows the surgeon to use it for a rehearsal before the real surgery [8]. From the MRI images as well, the location of the tumor can be captured and locked for operation. This can act as a simulation tool for the surgeon, as well as it can help the surgeon during the registration process.

On the other hand, the term haptic is referring to the tactile or sense of touch information that is required during a surgery [11]. By using the tele-operated surgery system, this kind of information is not there with the surgeon. Thus, haptic or force feedback sensor must be put at the slave robot so that the surgeon who is manipulating the master robot can have better control and awareness of the operation undertaken. Excessive force applied on the head of a patient might well damage important tissues or nerves.

4.5 End-effector

It is a device or tool specifically designed and attached to the last link of the robot or to the robot wrist [25]. It enables the robot to perform its intended tasks, for example cutting the skull or taking samples inside a body. The end-effector is loosely comparable to a human's hand. Its size is depending on the tasks assigned and also the working area. For a MIS operation, the end effector must be small enough to get to the body through the specified path or hole created.

4.6 Controller design

Control technique or controller is a tools used to cause the micro-manipulator perform the desired motions and actions, for example to executes planned trajectories. There are two class of manipulator control, namely the linear control and the nonlinear control. If a system can be defined using linear differential equations, than a linear control method is used. Otherwise, nonlinear control will come into action. There are various types of control method or technique. Among them were closed loop and open loop, classical, adaptive and intelligent control technique.

4.7 Size and materials

The micro-manipulator's type of materials used, its shape and size depends on the purpose and intended usage. The size and shape can be as shown in Figure 3(a) or as big as the DaVinci® system. It goes back to the user or surgeon, and the use of the system. The most important aspect is that it can work as it was intended to be, in a neurosurgical procedure. Moreover, the material is also very important. Let say it is going to be used inside an MRI machine, thus it must be made from MRI compatible type of materials.

5. Case study: NeuroArm

NeuroArm is a research project that was organized by the University of Calgary and MacDonald Dettwiler Associates (MDA). It was aimed to develop a MR-compatible micro-manipulator system for neurosurgical operation. This surgical robot system provides an immersive robotic system with full complement of planning and assistive software. Besides being a MR-compatible, this system is also incorporated with powerful image-guided system and haptic devices in its architecture ([2], [6] & [8]).

The system consists of three main parts. They are the robot itself, the workstation, and its cabinet for control system. The robot has with two arms with a moveable platform. Each arm has 8 degree of freedom (dof). The size of the arm is small and can operate in a 68cm working area inside the MRI scanner. The end-effector of the robot can fit various surgical tools required by the surgical procedure, such as forceps, needle drivers, suction, microscissor, and dissector. It is using the master-slave concept, means it incorporated the telesurgical system. The type of actuator used on the surgical side is a piezoceramic motors. This is primarily used to ensure safe functioning of the robot at the operative site and to avoid image distortion.



(a)



(b)

Fig. 4. (a) NeuroArm workstation. (b) NeuroArm surgical robot.

The master and the surgeon workstation are placed in a room adjacent to the operation room, while the slave or the robot itself was put inside the operation room. The master and slave are bit different in design, where the master was only a 6-dof controller with 3-dof positional force feedback. The workstation also consists of high-resolution binoculars where the surgeon can have direct access to the surgical binoculars, haptic hand-controllers, microphone to communicate with the operation room's personnel, and four monitors that display the information needed to know by the surgeon of the surgery that is going on. The master manipulator, as shown in Figure 6 has high-fidelity haptic capability, so the surgeon that is using the master controller will know the force currently being exerted to the patient at real time. In addition, tremor filter was installed in the system, to improve accuracy and precision as well as the stamina of the surgeon. This also enhances the surgical ability of the robot. The force exerted during operation can be limited. And for security purpose, if the robot fails itself, intrinsic braking system will automatically freeze the robot. Besides, the robot's actuators are functioning at low torque and low force in order to reduce any risks of injury. It can also move on a slow pace as 1mm/s, and can go up to 200mm/s, depending on the needs.

TABLE 1. NeuroArm mechanical specifications	
Parameters	Specification
Degrees of freedom	8 (including tool actuation) for each arm; 16 total
Payload	0.5 kg
Force (static)	10 N
Tool tip speed	Surgery: 0.5–50 mm/s Tool change: 200 mm/s
Positional accuracy	
Payload < 100 g	± 1 mm absolute 100 μm resolution
Payload > 100 g	± 2 mm absolute 1 mm resolution
Optical force sensors	Sensitivity: 0.02–5 N Dynamic range: 450:1
Continuous operation time	>10 h

TABLE 2. NeuroArm workstation specifications ^a	
Parameters	Specification
Hand controller	6-DOF position sensing 3-DOF translational force feedback using direct current motors
Workspace	Workspace (tool tip) x × y × z (ellipsoid) 40 × 25 × 50 cm Pitch, ± 130 degrees; yaw, ± 150 degrees; roll, ± 168 degrees
Microscope	Counterbalanced microscope equipped with motorized tilt and high-quality optics Beam splitter with two high-resolution JVC cameras High-definition 1080i format
Visual display	Binoculars using miniature display technology XGA resolution
Voice communication	Simultaneous talk/listen voice communication Wireless digital headset

^aDOF, degrees of freedom; XGA, extended graphics array.

Fig. 5. Specifications and accuracy of NeuroArm

This system has a simulation software that allows the surgeon to do a simulation operation before actually going for the real surgery. During the simulation, he can calculate the safe region for the robot's arm to operate. The virtual boundaries were also being able to define before the surgery. This can prevent injuries to the neural area surrounding the operation area.

Among the uniqueness of this NeuroArm is that it can fit into an MRI magnet bore. This means that the material used in building this robot is not affected by the effect of the magnet bore. Its upper arm is made of titanium and its lower arm is made from polyetheretherketone (PEEK) materials. In addition it ensures that the image generated by

the MRI is not significantly affected by the surgical tools. Even though some of the procedure is still being done by the surgeon himself, like burr holes and cranial exposure, NeuroArm has started creating its milestones in neurosurgery. With tested accuracy of 30micron, it is so great that it will surely enhance surgical capability. In addition to that, NeuroArm has successfully performed a neurosurgical operation in May 2008. The operation is to remove a tumor from a 21-year old's women brain in United States of America.

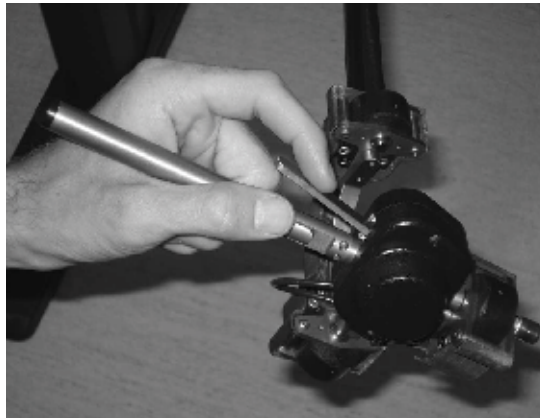


Fig. 6. Master manipulator of NeuroArm, that has the haptic interface

6. Conclusion

Researches on various aspects of neurosurgical operation were still going on. The purpose is to improve the existing system and also human lives. While surgical robots offer some advantages over the human hand, we are still a long way from the day when autonomous robots will operate on people without human interaction. With advances in our technology, it is not an impossible thing to be done. However, besides the excitement in this research field, the safety of the patient inside the operation theatre must be the highest priority.

7. References

- [1] Adha Cahyadi and Yoshio Yamamoto, Hysteretic Modelling of Piezoelectric Actuator Attached on Flexure Hinge Mechanism, IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 9-15 October 2006
- [2] Alexander D. Greer, Perry M. Newhook, and Garnette R. Sutherland, Human-Machine Interface for Robotic Surgery and Stereotaxy, IEEE/ASME Transactions on Mechatronics, Vol. 13, No. 3, JUNE 2008
- [3] Da Liu and Tianmiao Wang, A Workflow for Robot Assisted Neurosurgery, IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 9-16 October 2006
- [4] Damien Sallé, Francesco Cepolina and Philippe Bidaud, Surgery grippers for Minimally Invasive Heart Surgery, Proceedings of IEEE International Conference on Intelligent Manipulation and Grasping, Genova, Italy, 1-2 July 2004
- [5] da Vinci Surgical System, http://en.wikipedia.org/wiki/Da_Vinci_Surgical_System

- [6] Deon F. Louw, Tim Fielding, and Paul B. McBeth et al, Surgical Robotics: A Review and Neurosurgical Prototype Development, *Neurosurgery*, Vol. 54, No. 3, March 2004
- [7] F. Cepolina, Development of Micro Tools for Surgical Applications, Ph.D. Thesis, Universita' Degli Studi De Genova/ Uiversite Piere Et marie Currie: Paris 2005
- [8] Garnette R. Sutherland, Isabelle Latour, and Alexander D. Greer, Integrating an Image-Guided Robot with Intraoperative MRI, *IEEE Engineering in Medicine and Biology Megazine*, May/June 2008
- [9] Ho-seok Song, Ki-young Kim and Jung-ju Lee, Development of the Dexterous Manipulator and the Force Sensor for Minimally Invasive Surgery, *IEEE International Conference on Autonomous Robots and Agents*, Wellington, New Zealand, 10-12 February 2009
- [10] How Robotic Surgery Will Work,
<http://science.howstuffworks.com/roboticsurgery.htm/printable>
- [11] James C. Gwilliam, Mohsen Mahvash and Balazs Vagvolgyi et al, Effects of Haptic and Graphical Force Feedback on Teleoperated Palpation, *IEEE International Conference of Robotics and Automation*, Kobe, Japan, 12-17 May 2009
- [12] John J. Craig, *Introduction to Robotics Mechanics and Control*, 3rd edition, 2005, Pearson Prentice Hall
- [13] Junchuan Liu, Yuru Zhang, Zengmin Tian, Tianmiao Wang and Hongguang Xing, NeuroMaster: A Robot System for Neurosurgery, *IEEE International Conference on Robotics and Automation*, 2004
- [14] Junchuan Liu, Yuru Zhang and Zhen Li, The Application Accuracy of NeuroMaster:A Robot System for Stereotactic Neurosurgery, *Proceedings of the 2nd IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 2006
- [15] Koji Ikuta, Keiichi Yamamoto and Keiji Sasaki, Development of RemoteMicrosurgery Robot and New Surgical Procedure for Deep and Narrow Space, *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 14-19 September 2003
- [16] M. Tanimoto, F. Arail and T. Fukuda et al, Micro Force Sensor for Intravascular Neurosurgery and In Vivo Experiment, *The Eleventh Annual International Workshop on Micro Electro Mechanical Systems*, 1998
- [17] Mathematical Model, http://en.wikipedia.org/wiki/Mathematical_model
- [18] Meng Cai, Wang Tianmiao, Chou Wusheng and Zhang Yuru, A Neurosurgical Robotic System under Image-Guidance, *IEEE International Conference on Industrial Informatics*, 2006
- [19] NeuroArm, <http://www.neuroarm.org/project.php>
- [20] Neurosurgery, <http://en.wikipedia.org/wiki/Neurosurgery>
- [21] Paul B. McBeth, Deon F. Louw, Peter R. Rizun and Garnette R. Sutherland, *Robotics in Neurosurgery*, *The American Journal or Surgery* 188 (Suppl to October 2004)
- [22] Peter R. Rizun, Paul B. McBeth, Deon F. Louw and Garnette R. Sutherland, *Robot-Assisted Neurosurgery*, <http://www.sagepublications.com>
- [23] R. K. Mittal and I. J. Nagrath, *Robotics and Control*; 2005, Tata McGraw Hill
- [24] *Robot and Robotic Glossary*,
http://www.krobotics.com/robot_information/robot_glossary.php
- [25] Robot End Effector, http://en.wikipedia.org/wiki/End_effector

- [26] Robot Glossary – Industrial Terminology Defined,
<http://www.robots.com/glossary.php>
- [27] Robotic Surgery,
http://biomed.brown.edu/Courses/BI108/BI108_2005_Groups/04/index.html
- [28] Victor M. Becerra, Callum N. J. Cage, William S. Harwin and Paul M. Sharkey,
Hardware Retrofit and Computed Torque Control of a PUMA 560 Robot, IEEE
Control System Magazine: pp. 78-82, October 2004
- [29] What is Neurosurgery by Sanjay Mongia (Dr.),
<http://www.neurosurgeon.co.in/neurosurgery.html>
- [30] Yik San Kwok, Joahin Hou, Edmond A. Jonckheere and Samad Hayati, A Robot with
Improved Absolute Positioning Accuracy for CT Guided Stereotactic Brain Surgery,
IEEE Transactions on Biomedical Engineering, Vol. 35, No. 2, February 1988

DeLiA: A New Family of Redundant Robot Manipulators

Jaime Gallardo-Alvarado
*Instituto Tecnológico de Celaya,
México*

1. Introduction

A fully decoupled parallel manipulator is a mechanism in which one output kinematic joint, degree of freedom, is affected by only one active or input kinematic pair, the perfect mechanism from a kinematic point of view due to the possibility to generate linear input-output kinematic constraint equations. Parallel manipulators with fewer than six-degrees-of-freedom frequently referred as limited-dof or defective parallel manipulators were the first class of parallel manipulators to be considered in that trend. Kong & Gosselin (2002a) introduced a class of translational fully decoupled parallel manipulators called Tripteron family. Carricato & Parenti-Castelli (2004) invented a two-degrees-of-freedom parallel wrist in which two interconnected linkages independently actuate one of the two angles associated to the orientation of the moving platform. Recently, Briot & Bonev (2009) proposed a fully decoupled translational parallel manipulator, called Pantopteron, for simple pick-and-place operations. Certainly, there is a significative number of contributions dealing with the study of limited-dof fully decoupled parallel manipulators, see for instance Carricato & Parenti-Castelli (2002), Kong & Gosselin (2002b, 2002c), Gosselin et al., (2004), Gogu (2005), Li et al., (2005), Ruggiu (2009) and so on. On the other hand, a fully decoupled six-degrees-of-freedom parallel manipulator is maybe, still in our days, an unrealistic task. In fact, the dream that in a Gough-Stewart platform one degree of freedom shall be affected by only one active kinematic joint is a far away reality, if sensors are not considered. In order to diminish such drawback, the term fully can be removed from the original concept meaning that a decoupled parallel manipulator is a mechanism in which the position and orientation, pose, of the moving platform with respect to the fixed platform can be computed separately. The decoupled motion can be achieved by introducing geometric conditions, e.g. Wohlhart (1994) studied a Gough-Stewart platform in which three of the six limbs share a common spherical joint over the moving platform, other topologies with uncoupled rotations and translations were investigated by Innocenti & Parenti-Castelli (1991), Zabalza et al., (2002), Yang et al., (2004), Takeda (2005) and so on. Despite the indisputable recent valuable advances in this subject, the development of decoupled parallel manipulators with simplified architectures preserving the well-known benefits of parallel manipulators such as higher stiffness and payload/capacity is a rather complicated task. At this point, and mainly due to the lack of an efficient mathematical resource to approach the forward kinematics of a general Gough-Stewart platform capable to determine the actual configuration of the manipulator, without using sensors, one can take into account that if

there is not essential the fully decoupled motion, then the development of partially decoupled parallel manipulators is a viable option to apply the benefits of mechanisms with nearly parallel kinematic structures, see for instance Briot et al., (2009), Altuzarra et al., (2010). It is interesting to note that mechanisms with mixed motions can be included in the class known as partially decoupled parallel manipulators.

In this chapter a new family of partially decoupled parallel manipulators endowed with an extra active kinematic joint is introduced. One member of this new family of robot manipulators is selected with the purpose to illustrate the methodology of kinematic analysis chosen to characterize the angular and linear kinematic properties, up to the acceleration analysis, of it. The forward position analysis of the robot, a challenging task for most parallel manipulators, is carried-out in a semi-closed form solution applying recursively the Sylvester dialytic elimination method that allows to determine all the feasible locations that the output platform can reach with respect to the fixed platform given a set of generalized coordinates. On the other hand, the velocity and acceleration analyses of the robot are approached by means of the theory of screws. With this mathematical tool, simple and compact expressions for computing the velocity and reduced acceleration states of the output platform are obtained taking advantage of the properties of reciprocal screws, via the Klein form of the Lie algebra $e(3)$. Finally, the robot is simulated as a virtual five-degrees-of-freedom parallel kinematic machine using special commercially available software like ADAMS©.

2. Description of the DeLiA robot family

Before the transcendental contributions of Gough (1957), Gough & Whitehall (1962) and Stewart (1965), it seems that a five-degrees-of-freedom spray painting machine was the first promissory industrial application of a parallel manipulator (Pollard, 1940; Bonev, 2003). Furthermore, many practical applications do not require the six degrees of freedom of a general Gough-Stewart platform, particularly five-degrees-of-freedom parallel manipulators had been proposed, among simple pointing devices, as multi-axis machine tools (Bohez, 2002; Zheng et al., 2005; Gao et al., 2005, 2006), bio-mechanical devices (Zhu et al., 2008; Gallardo-Alvarado, 2010) or new architectures for medical applications (Vlachos & Papadopoulos, 2005; Piccina, 2009).

With these considerations in mind and with the motivation that not always is essential a pure parallel kinematic topology, this work is intended to be a viable option to the development of a new class of five-degrees-of-freedom robots with a nearly parallel kinematic architecture, preserving the advantages of serial-parallel manipulators but with the possibility to mount all the active limbs on the fixed platform.

The proposed general topology is depicted in Fig. 1, it consists of a fixed platform, a coupler platform and an end-effector-platform also called output- platform. Please note that while in a general Gough-Stewart platform the generalized coordinates or active joints are necessarily coupled, in the proposed topology these motors can be decoupled into two different groups which allows to simplify the forward kinematics of the mechanisms at hand. Furthermore, the end-effector-platform is connected at the fixed platform by means of an active 6-dof three-legged parallel manipulator (XYS-type limb with $X=RR,U,C,S$; $Y=P,R,C$) whereas the coupler platform is connected at the fixed platform by means of an active 3-dof parallel manipulator (XYS-type limb with $X=R,P$; $Y=R,P$) and at the end-effector platform through a passive 3-dof parallel manipulator (XYS-type limb with $X=R,P$; $Y=R,P$). Interesting benefits can be observed in this topology:

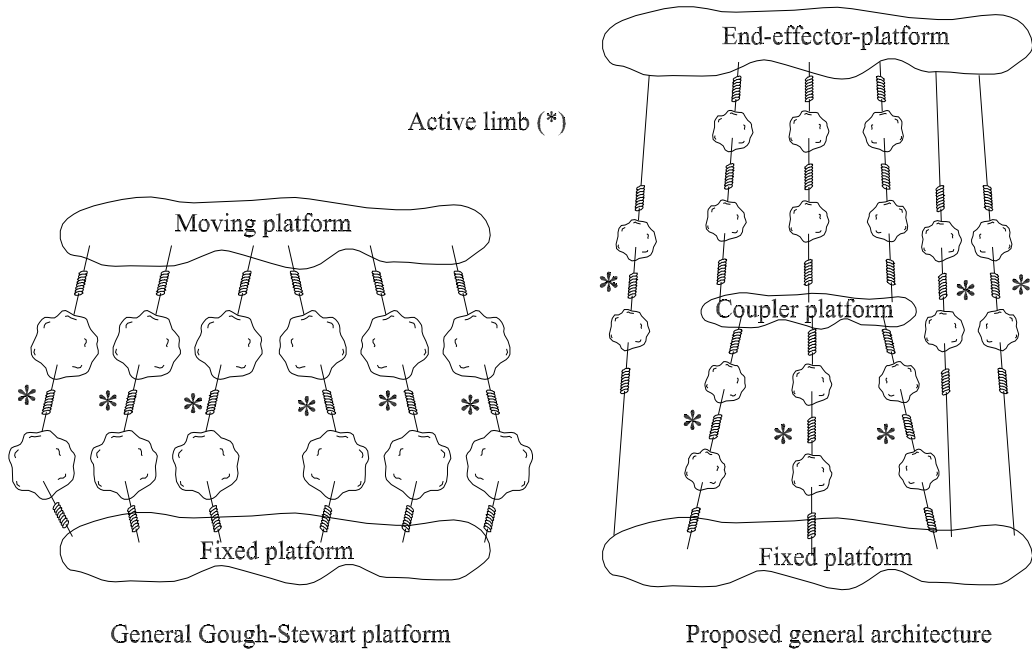


Fig. 1. General Gough–Stewart platform and the general proposed topology

- The motors can be mounted on the fixed platform
- The forward finite kinematics can be carried-out solving two decoupled systems of non-linear kinematic constraint equations
- The spherical joints attached at the end-effector platform allow to affirm that this topology is a non-overconstrained mechanism, e.g. does not require of additional tolerances of manufacture that ensure the intersection of screws

On the other hand, the 3-dof parallel manipulators chosen for this research, belong to the class known as zero-torsion parallel manipulators (Bonev, 2002).

Several combinations can be generated with the considerations above-mentioned and one of them, here after called D1 robot, is presented in Fig. 2.

D1 is a robot formed with an active 3-UPS parallel manipulator and two 3-RPS parallel manipulators, one active and the other passive. The nominal coordinates of the universal, prismatic, spherical and revolute joints of the chosen architecture are denoted respectively by U , P , S and R and are located by vectors \mathbf{U} , \mathbf{P} , \mathbf{S} and \mathbf{R} . In the rest of this work, the analysis is focused on the D1 robot.

3. Mobility analysis of the D1 robot

An exhaustive review of formulae addressing the mobility analysis of closed kinematic chains can be found in Gogu (2005) and the following is a variant of the well-known Kutzbach–Grübler formula for computing the degrees-of-freedom of spatial parallel manipulators

$$F = 6(n - j - 1) + \sum_{i=1}^j f_i \quad (1)$$

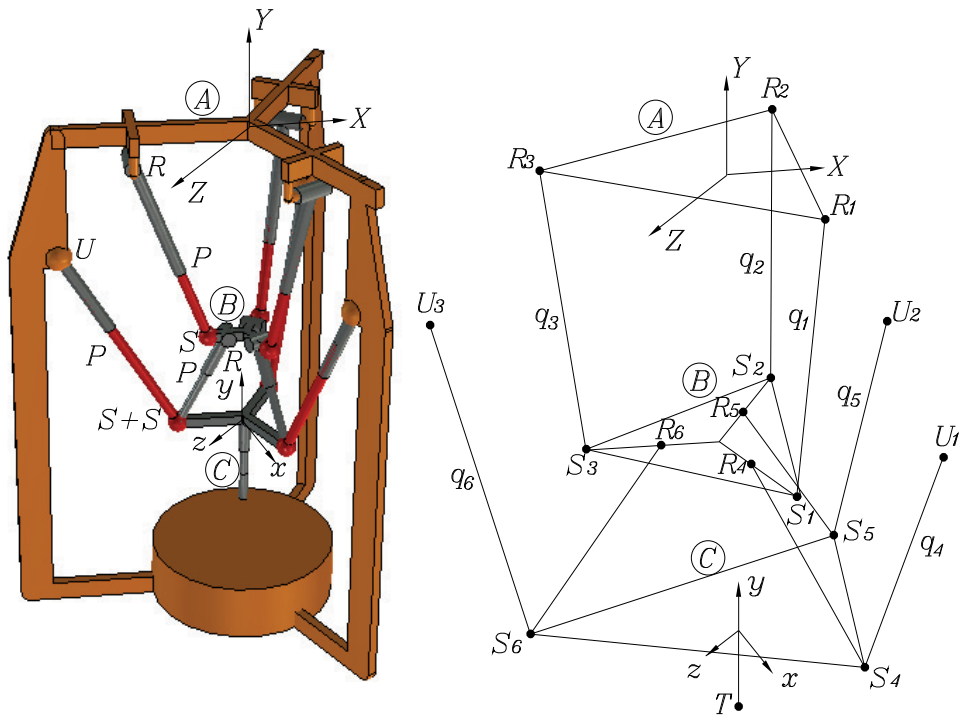


Fig. 2. D1, a member of the DeLiA robot family and its geometric scheme

where n is the number of links, j is the number of kinematic pairs, and f_i is the number of freedoms of the i -th kinematic pair. For the robot D1 $n = 21$, $j = 6R + 9P + 3U + 9S = 27$, $\sum_{i=1}^j f_i = 48$ and therefore $F = 6$, which is a wrong result. In fact, Dai et al. (2006) proved that in a 3-RPS parallel manipulator a basis representing the motions of the moving platform, with respect to the fixed platform, consists of three elements, two non-parallel coplanar rotations and one translation along an axis perpendicular to the plane formed by the spherical joints. According to this basis, the coupler platform of the robot D1 cannot rotate with respect to the fixed platform along an axis perpendicular to the plane formed by the spherical joints attached at the coupler platform. It is straightforward to demonstrate that such argument is valid too for the end-effector-platform and the coupler platform, in other words, the end-effector-platform has a rotation restricted with respect to the coupler platform due to the passive 3-RPS parallel manipulator connecting both platforms and the coupler platform has a rotation restricted with respect to the fixed platform due to the 3-RPS active parallel manipulator. With these considerations in mind, although the computed degrees of freedom of the robot at hand is six, the end-effector-platform does not accept arbitrary orientations with respect to the fixed platform, and therefore D1 is in reality a five-degrees-of-freedom redundant robot.

4. Finite kinematics

In this section the position analysis of the proposed robot is presented.

4.1 Forward position analysis

The forward position analysis (FPA) consists of finding the pose of the end-effector-platform with respect to the fixed platform given a set of six generalized coordinates $q_i (i = 1, 2, \dots, 6)$. Due to the decoupled architecture, the pose of the coupler platform, body B, with respect to the fixed platform, body A, is controlled by means of the internal generalized coordinates $q_i (i = 1, 2, 3)$. Furthermore, the pose of the coupler platform is easily determined through the computation of the coordinates of the centers of the spherical joints attached at the coupler platform, points $S_i (i = 1, 2, 3)$.

Let XYZ and xyz be two reference frames attached, respectively, at the fixed platform and at the end-effector-platform. With reference to Fig. 2, since the revolute joints attached at the fixed platform have a tangential arrangement, then it is possible to write three kinematic constraint equations as

$$\left. \begin{aligned} (S_1 - R_1) \bullet (R_2 - R_3) &= 0 \\ (S_2 - R_2) \bullet (R_3 - R_1) &= 0 \\ (S_3 - R_3) \bullet (R_2 - R_1) &= 0 \end{aligned} \right\} \quad (2)$$

where the dot (\bullet) denotes the usual inner product of the three-dimensional vectorial algebra. Furthermore, three closure equations can be written as

$$(S_i - R_i) \bullet (S_i - R_i) = q_i^2 \quad i = 1, 2, 3 \quad (3)$$

Finally, according to the triangle $\Delta S_1 S_2 S_3$ three compatibility equations are given by

$$(S_i - S_j) \bullet (S_i - S_j) = \overline{S_i S_j}^2 \quad i, j = 1, 2, 3 \quad \text{mod}(3) \quad (4)$$

Expressions (2), (3) and (4) are solved by applying the Sylvester dialytic elimination method (Tsai, 1999; Gallardo et al., 2007). Once the coordinates of the points $S_i (i = 1, 2, 3)$ are calculated, the center of the triangle $\Delta S_1 S_2 S_3$, vector ${}^A \rho^B$, results in

$${}^A \rho^B = (S_1 + S_2 + S_3) / 3 \quad (5)$$

Finally, the pose of the coupler platform with respect to the fixed platform is summarized in the 4×4 homogeneous transformation matrix ${}^A \mathbf{T}^B$:

$${}^A \mathbf{T}^B = \begin{bmatrix} {}^A \mathbf{R}^B & {}^A \rho^B \\ \mathbf{O}_{1 \times 3} & 1 \end{bmatrix} \quad (6)$$

where ${}^A \mathbf{R}^B$ is the rotation matrix which is computed by means of the coordinates of the points $S_i (i = 1, 2, 3)$, for details see Gallardo-Alvarado et al. (2008). Following a similar procedure, the homogeneous transformation matrix of the end-effector-platform with respect to the fixed platform, ${}^A \mathbf{T}^C$, is computed. To this end, consider the following closure equations

$$\left. \begin{aligned} (S_4 - R_4) \bullet (R_5 - R_6) &= 0 \\ (S_5 - R_5) \bullet (R_6 - R_4) &= 0 \\ (S_6 - R_6) \bullet (R_5 - R_4) &= 0 \\ (S_{i+3} - U_i) \bullet (S_{i+3} - U_i) &= q_{i+3}^2 \quad i = 1, 2, 3 \\ (S_i - S_j) \bullet (S_i - S_j) &= \overline{S_i S_j}^2 \quad i, j = 4, 5, 6 \quad \text{mod}(3) \end{aligned} \right\} \quad (7)$$

where the vectors $R_i (i = 4, 5, 6)$ are computed by using the matrix ${}^A T^B$. Furthermore, the homogeneous transformation matrix between the end-effector- platform and the fixed platforms, ${}^B T^C$, can be calculated from

$${}^A T^C = {}^A T^B {}^B T^C \quad (8)$$

4.2 Inverse position analysis

The inverse position analysis consists of finding the active limb lengths $q^i (i = 1, 2, \dots, 6)$ of the robot given the pose of the end-effector- platform with respect to the fixed platform, matrix ${}^A T^C$. To this end, it is necessary to compute, as an intermediate step, the vectors $S^i (i = 1, 2, \dots, 6)$.

Immediately emerges that the coordinates of the points $S_i (i = 4, 5, 6)$, expressed in the fixed reference frame XYZ , attached at the end-effector- platform can be obtained from

$$\begin{bmatrix} S_i \\ 1 \end{bmatrix} = {}^A T^C \begin{bmatrix} S_i \\ 1 \end{bmatrix} \quad (9)$$

where $s_i (i = 4, 5, 6)$ is the i -th point but expressed according to the moving reference frame xyz . Furthermore, the unknown vectors $R_i (i = 4, 5, 6)$ and $S_i (i = 1, 2, 3)$ can be computed according to the following closure equations

$$\left. \begin{aligned} (S_6 - R_6) \bullet (R_5 - R_4) &= 0 \\ (S_5 - R_5) \bullet (R_6 - R_4) &= 0 \\ (S_4 - R_4) \bullet (R_6 - R_5) &= 0 \\ (S_1 - R_1) \bullet (R_2 - R_3) &= 0 \\ (S_2 - R_2) \bullet (R_3 - R_1) &= 0 \\ (S_3 - R_3) \bullet (R_2 - R_1) &= 0 \\ S_1 + S_2 + S_3 &= R_4 + R_5 + R_6 \\ (S_i - S_j) \bullet (S_i - S_j) &= \overline{S_i S_j}^2 \quad i, j = 1, 2, 3 \quad \text{mod}(3) \\ (R_i - R_j) \bullet (R_i - R_j) &= \overline{R_i R_j}^2 \quad i, j = 4, 5, 6 \quad \text{mod}(3) \\ (S_i - R_{i+3}) \bullet (S_i - R_{i+3}) &= \overline{S_i R_{i+3}}^2 \quad i = 1, 2, 3 \end{aligned} \right\} \quad (10)$$

Finally, the limb lengths $q_i (i = 1, 2, \dots, 6)$ result in

$$\left. \begin{aligned} q_i^2 &= (S_i - R_i) \bullet (S_i - R_i) \\ q_{i+3}^2 &= (S_{i+3} - U_i) \bullet (S_{i+3} - U_i) \quad i = 1, 2, 3 \end{aligned} \right\} \quad (11)$$

5. Infinitesimal kinematics

In this section the velocity and acceleration analyses of the proposed robot are approached by means of the theory of screws. For detailed information of the kinematic analysis of closed chains and parallel manipulators, using such mathematical resource, the reader is referred to (Rico & Duffy, 2000; Gallardo et al., 2003). In particular screw theory is an

efficient mathematical resource to analyze five-degrees-of-freedom parallel manipulators (Li & Huang, 2002; Zhu et al., 2008; Gallardo-Alvarado et al., 2009). Furthermore, as a consideration for readers unfamiliar with the theory of screws an explanation of basic concepts dealing with it is also included in this section.

5.1 Preliminary concepts. Basic concepts of the screw theory

A screw $\$ = (\hat{s}, s_O)$ is a six-dimensional vector composed of a vector \hat{s} , namely the primal part, denoting the direction of the screw axis and a vector s_O , namely the dual part, which is the moment produced by \hat{s} about a point O fixed to the reference frame. The moment s_O is calculated as follows

$$s_O = \hat{s}h + \hat{s} \times r_{O/P} \quad (12)$$

where h is the pitch of the screw and $r_{O/P}$ is a vector directed from a point P , fixed to the screw axis, to point O . Note that if the pitch of the screw goes to infinity, then the screw represents a prismatic joint and it is represented by $\$ = (0, \hat{s})$. Any lower kinematic pair can be represented either by a screw or a group of screws. A cylindrical joint can be simulated by the combination of one revolute joint and one prismatic joint, whereas a spherical joint results of the action of three revolute joints whose axes, usually mutually orthogonal, intersect a common point.

Screw theory, which is isomorphic to the Lie algebra $e(3)$ also referred as motor algebra, is the set of elements of the form $\$ = (\hat{s}, s_O)$ with the following operations.

Let $\$1 = (\hat{s}_1, s_{O1})$, $\$2 = (\hat{s}_2, s_{O2})$, and $\$3 = (\hat{s}_3, s_{O3})$ be elements of the Lie algebra $e(3)$ with $\lambda_1, \lambda_2, \lambda_3 \in \mathfrak{R}$. Then

1. Addition, $\$1 + \$2 = (\hat{s}_1 + \hat{s}_2, s_{O1} + s_{O2})$
2. Multiplication by a scalar, $\lambda \$1 = (\lambda \hat{s}_1, \lambda s_{O1})$
3. Lie product or dual motor product, $[\$1 \$2] = (\hat{s}_1 \times \hat{s}_2, \hat{s}_1 \times s_{O2} - \hat{s}_2 \times s_{O1})$.

The Lie product exhibits interesting properties like

- a. Nilpotent, $[\$1 \$1] = (0, 0)$
- b. Non-commutative, $[\$1 \$2] = -[\$2 \$1]$
- c. Distributive

$$[\$1 \lambda_2 \$2 + \lambda_3 \$3] = \lambda_2 [\$1 \$2] + \lambda_3 [\$1 \$3]$$

$$[\lambda \$1 + \lambda_2 \$2 \$3] = \lambda_1 [\$1 \$3] + \lambda_2 [\$2 \$3]$$

4. Jacobi identity, $[\$1 [\$2 \$3]] + [\$3 [\$1 \$2]] + [\$2 [\$3 \$1]] = (0, 0)$

Furthermore, the Lie algebra $e(3)$ is endowed with two symmetric bilinear forms

1. The Killing form, $\{\$1; \$2\} = \hat{s}_1 \cdot \hat{s}_2$
2. The Klein form, $\{\$1; \$2\} = \hat{s}_1 \cdot s_{O2} + \hat{s}_2 \cdot s_{O1}$

It is said that the screws $\$1$ and $\$2$ are reciprocal if $\{\$1; \$2\} = 0$, an interesting property that allows to simplify the forward infinitesimal kinematics of parallel manipulators.

Screw theory is a powerful mathematical tool modeling the kinematics of rigid bodies.

The velocity state ${}^nV_O^m$ of a rigid body m as it is observed from another body n or reference frame is a twist about screw (Ball, 1900), indeed ${}^nV_O^m = \omega^n \m , given by

$${}^n \mathbf{V}_O^m = \begin{bmatrix} {}^n \boldsymbol{\omega}^m \\ {}^n \mathbf{v}_O^m \end{bmatrix} \quad (13)$$

where ${}^n \boldsymbol{\omega}^m$ and ${}^n \mathbf{V}_O^m$ are, respectively, the angular and linear velocities of the body under study and O is a point of the body m that is instantaneously coincident with the origin of the reference frame n , point O is also known as the reference pole. Furthermore, in an open kinematic chain, e.g. a serial manipulator, the velocity state of the end-effector, labeled body m , with respect to the base link, labeled body n , can be written as a linear combination of the involved infinitesimal screws associated to the kinematic pairs as follows

$${}^n \omega_{n+1} {}^n \$^{n+1} + {}_{n+1} \omega_{n+2} {}^{n+1} \$^{n+2} + \dots + {}_{m-1} \omega_m {}^{m-1} \$^m = {}^n \mathbf{V}_O^m \quad (14)$$

On the other hand, the reduced acceleration state ${}^n \mathbf{A}_O^m$ of body m with respect to body n , also known as accelerator, is a six-dimensional vector given by

$${}^n \mathbf{A}_O^m = {}_n \dot{\omega}_m {}^n \$^m = \begin{bmatrix} {}^n \dot{\boldsymbol{\omega}}^m \\ {}^n \mathbf{a}_O^m - {}^n \boldsymbol{\omega}^m \times {}^n \mathbf{v}_O^m \end{bmatrix} \quad (15)$$

where $\dot{\omega}$ and ${}^n \mathbf{a}_O^m$ are the angular and linear accelerations of body m with respect to body n taking O as the reference pole. Furthermore, in a serial manipulator the reduced acceleration state of the end-effector with respect to the base link is given by

$${}^n \dot{\omega}_{n+1} {}^n \$^{n+1} + {}_{n+1} \dot{\omega}_{n+2} {}^{n+1} \$^{n+2} + \dots + {}_{m-1} \dot{\omega}_m {}^{m-1} \$^m + {}^n \mathcal{L}^m = {}^n \mathbf{A}_O^m \quad (16)$$

where

$$\begin{aligned} {}^n \mathcal{L}^m = & [{}^n \omega_{n+1} {}^n \$^{n+1} \quad {}_{n+1} \omega_{n+2} {}^{n+1} \$^{n+2} + \dots + {}_{m-1} \omega_m {}^{m-1} \$^m] \\ & + [{}_{n+1} \omega_{n+2} {}^{n+1} \$^{n+2} + {}_{n+2} \omega_{n+3} {}^{n+2} \$^{n+3} + \dots + {}_{m-1} \omega_m {}^{m-1} \$^m] \\ & + \dots + [{}_{m-2} \omega_{m-1} {}^{m-2} \$^{m-1} \quad {}_{m-1} \omega_m {}^{m-1} \$^m] \end{aligned} \quad (17)$$

is the Lie screw or complementary six-dimensional vector of the reduced acceleration state.

It is worth mentioning that eventhough its compactness, Eq. (16) contains all the terms involved in the acceleration analysis of a rigid body. In fact, e.g. Eq. (16) contains the terms of the acceleration of Coriolis and one not need to make a distinction of it. Furthermore, Eq. (16) can be easily translated into computer codes approaching the kinematic analysis of robot manipulators. This expression was introduced by the first time by Rico-Martínez & Duffy (1996) and its correctness was validated by the author of this work with the publication of several papers in well known journals. Before the pioneering contribution of Rico-Martínez & Duffy (1996) the screw theory was confined to the so-called first order analysis (velocity analysis). Its introduction almost fifteen years ago open the possibility to extend the screw theory to the so-called higher order kinematic analyses.

5.2 Velocity analysis

The modeling of the screws of three representative limbs of the robot is depicted in Fig. 3. It must be noted that due to existence of compound joints in the output platform, e.g.

Spherical + Spherical, these spherical joints require each one more of the usual three infinitesimal screws indicating concurrent revolute joints. Furthermore, before do any further, in order to solve the inverse velocity and acceleration analyses, it is necessary the introduction of auxiliary screws with the purpose to satisfy an algebraic requirement, with this consideration in mind the revolute joints are modeled as cylindrical joints, in which the corresponding translational velocities are equal to zero. In other words, ${}^i_0\omega_1 = {}^i_6\omega_7 = 0 (i = 1, 2, 3)$.

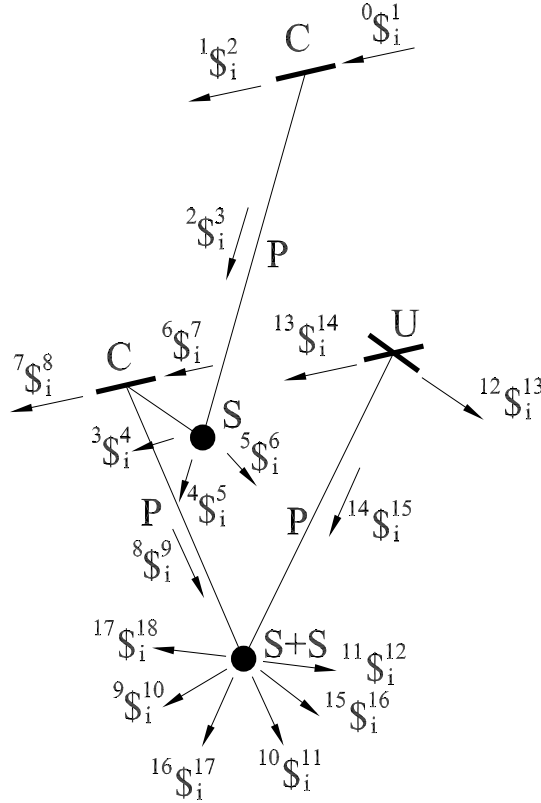


Fig. 3. Infinitesimal screws of three representative limbs of the robot D1

Let ${}^A\omega^C$ and ${}^A v_O^C$ be, respectively, the angular and linear velocities of a point O attached at the end-effector-platform. The velocity state, or twist about a screw, of the end-effector-platform with respect to the fixed platform, six-dimensional vector ${}^A V_O^C = [{}^A\omega^C, {}^A v_O^C]^T$, can be obtained through the coupler and fixed platforms as

$${}^A V_O^C = {}^A V_O^B + {}^B V_O^C \quad (18)$$

where ${}^A V_O^B$ is the velocity state of the coupler platform with respect to the fixed platform, and ${}^B V_O^C$ is the velocity state of the end-effector-platform with respect to the coupler platform. Furthermore, these kinematic states can be written in screw form as

$$V = J_i \Omega_i, V \in \{{}^A V_O^B, {}^B V_O^C, {}^A V_O^C\} \quad i = 1, 2, 3 \quad (19)$$

where the Jacobians $\mathbf{J}_i \in \{ {}^A \mathbf{J}_i^B, {}^B \mathbf{J}_i^C, {}^A \mathbf{J}_i^C \}$ are given by

$$\left. \begin{aligned} {}^A \mathbf{J}_i^C &= [{}^{12} \underline{\$}_i^{13}, {}^{13} \underline{\$}_i^{14}, {}^{14} \underline{\$}_i^{15}, {}^{15} \underline{\$}_i^{16}, {}^{16} \underline{\$}_i^{17}, {}^{17} \underline{\$}_i^{18}] \\ {}^B \mathbf{J}_i^C &= [{}^6 \underline{\$}_i^7, {}^7 \underline{\$}_i^8, {}^8 \underline{\$}_i^9, {}^9 \underline{\$}_i^{10}, {}^{10} \underline{\$}_i^{11}, {}^{11} \underline{\$}_i^{12}] \\ {}^A \mathbf{J}_i^B &= [{}^0 \underline{\$}_i^1, {}^1 \underline{\$}_i^2, {}^2 \underline{\$}_i^3, {}^3 \underline{\$}_i^4, {}^4 \underline{\$}_i^5, {}^5 \underline{\$}_i^6] \end{aligned} \right\} \quad (20)$$

whereas $\mathbf{\Omega}_i \in \{ {}^A \mathbf{\Omega}_i^B, {}^B \mathbf{\Omega}_i^C, {}^A \mathbf{\Omega}_i^C \}$ are matrices containing the joint rate velocities. In fact:

$$\left. \begin{aligned} {}^A \mathbf{\Omega}_i^C &= [{}_{12} \overline{\omega}_i^{13}, {}_{13} \overline{\omega}_i^{14}, {}_{14} \overline{\omega}_i^{15}, {}_{15} \overline{\omega}_i^{16}, {}_{16} \overline{\omega}_i^{17}, {}_{17} \overline{\omega}_i^{18}]^T \\ {}^B \mathbf{\Omega}_i^C &= [{}_6 \overline{\omega}_i^7, {}_7 \overline{\omega}_i^8, {}_8 \overline{\omega}_i^9, {}_9 \overline{\omega}_i^{10}, {}_{10} \overline{\omega}_i^{11}, {}_{11} \overline{\omega}_i^{12}]^T \\ {}^A \mathbf{\Omega}_i^B &= [{}_0 \overline{\omega}_i^1, {}_1 \overline{\omega}_i^2, {}_2 \overline{\omega}_i^3, {}_3 \overline{\omega}_i^4, {}_4 \overline{\omega}_i^5, {}_5 \overline{\omega}_i^6]^T \end{aligned} \right\} \quad (21)$$

The inverse velocity analysis consists of finding the joint rate velocities of the robot given a prescribed velocity state of the end-effector-platform with respect to the fixed platform, ${}^A \mathbf{V}_O^C$. This analysis is solved directly by means of expressions (18) and (19), however the loss freedom of the end-effector-platform must be taken into proper account in order to obtain the desired velocity state. Furthermore, it is important to emphasize that the Jacobians ${}^A \mathbf{J}_i^B$, ${}^B \mathbf{J}_i^C$ and ${}^A \mathbf{J}_i^C$ must be invertible, otherwise the robot is at singular configuration.

On the other hand, the forward velocity analysis consists of finding the velocity state ${}^A \mathbf{V}_O^C$, given the active joint rate velocities of the robot. It is interesting to note that due to the decoupled architecture, the velocity state ${}^A \mathbf{V}_O^B$ depends only of the three active joints ${}^2 \underline{\omega}_3^i (i=1,2,3)$. Furthermore, since ${}^3 \underline{\$}_i^4$ and ${}^4 \underline{\$}_i^5$ are reciprocal to the remaining screws representing the revolte joints in the same limbs, the application of the Klein form, $\{*; *\}$, between the velocity state ${}^A \mathbf{V}_O^B$ and these reciprocal screws, the reduction of terms leads to

$$\underline{\mathbf{J}}_1^T \Delta {}^A \mathbf{V}_O^B = [0 \quad {}^2 \underline{\omega}_3^1 \quad 0 \quad {}^2 \underline{\omega}_3^2 \quad 0 \quad {}^2 \underline{\omega}_3^3]^T \quad (22)$$

where

$$\underline{\mathbf{J}}_1 = [{}^3 \underline{\$}_1^4, {}^4 \underline{\$}_1^5, {}^3 \underline{\$}_2^4, {}^4 \underline{\$}_2^5, {}^3 \underline{\$}_3^4, {}^4 \underline{\$}_3^5] \quad (23)$$

is the active Jacobian matrix between the middle and fixed platforms and

$$\Delta = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix} \quad (24)$$

is an operator of polarity defined by the 3×3 identity matrix \mathbf{I} and the 3×3 zero matrix \mathbf{O} . Hence, the velocity state ${}^A \mathbf{V}_O^B$ is obtained directly from the input/output velocity equation (22). In order to compute the velocity state ${}^A \mathbf{V}_O^C$ please note that the screws ${}^{16} \underline{\$}_i^{17} (i=1,2,3)$ are reciprocal to the remaining screws, in the same limb, representing the revolte joints of the UPS-type limbs. Thus, after applying the Klein form between these screws and the velocity state ${}^A \mathbf{V}_O^C$, the reduction of terms yields

$$\{^{16}\$^i; {}^A V_O^C\} = {}_{14}\underline{\omega}_{15}^i \quad i = 1, 2, 3 \quad (25)$$

Similarly, the application of the Klein form of the screws ${}^9\i ($i = 1, 2, 3$) to both sides of Eq. (18) allows to write

$$\{^9\$^i; {}^A V_O^C\} = \{^9\$^i; {}^A V_O^B\} \quad i = 1, 2, 3 \quad (26)$$

Finally, casting in a matrix–vector form Eqs. (25) and (26) one obtains

$$\underline{\mathbf{J}}_2^T \Delta {}^A V_O^C = \begin{bmatrix} {}_{14}\underline{\omega}_{15}^1 \\ {}_{14}\underline{\omega}_{15}^2 \\ {}_{14}\underline{\omega}_{15}^3 \\ \{^9\$^1; {}^A V_O^B\} \\ \{^9\$^2; {}^A V_O^B\} \\ \{^9\$^3; {}^A V_O^B\} \end{bmatrix} \quad (27)$$

where

$$\underline{\mathbf{J}}_2 = [{}^{16}\$^1, {}^{16}\$^2, {}^{16}\$^3, {}^9\$^1, {}^9\$^2, {}^9\$^3] \quad (28)$$

is the active Jacobian matrix between the output and fixed platforms.

Therefore the velocity state ${}^A V_O^C$ can be computed directly from the input/ output velocity equation (27). Please note that the forward velocity analysis requires that the active Jacobian matrices $\underline{\mathbf{J}}_1$ and $\underline{\mathbf{J}}_2$ must be invertible, otherwise the manipulator is at a singular configuration.

5.3 Redundancy analysis of the robot D1

In what follows the redundancy of the robot under study is briefly explained. Firstly, consider that according to section 3 ${}^A \omega^B \cdot {}^A \tau^B = {}^B \omega^C \cdot {}^B \tau^C = 0$ where ${}^A \tau^B$ and ${}^B \tau^C$ are, respectively, normal vectors to the planes $S_1S_2S_3$ and $S_4S_5S_6$. Furthermore, taking into account that ${}^A \omega^C = {}^A \omega^B + {}^B \omega^C$, then the loss rotation of the robot leads to

$${}^A \omega^C \cdot ({}^A \tau^B + {}^B \tau^C) - {}^A \omega^B \cdot {}^A \tau^B - {}^B \omega^C \cdot {}^B \tau^C = 0 \quad (29)$$

Equation (29) is called a *zero-torsion condition* and indicates that one element of the angular velocity ${}^A \omega^C$ can be written as a linear combination of its remaining components. With this consideration in mind the velocity state ${}^A V_O^C$ can be considered, by using a proper reference frame, as a five-dimensional vector which implies that it is possible to write, according to Eqs. (22) and (27), ${}^A V_O^C$ in terms of first order coefficients (Gallardo–Alvarado & Rico–Martínez, 2001) as

$${}^A V_O^C = \mathbf{G}\mathbf{Q} + \mathbf{Q}_* \quad (30)$$

where \mathbf{Q} is a 5×1 matrix containing five of the six generalized or active joint rate velocities which is affected by the 5×5 matrix \mathbf{G} whose elements are the corresponding first order coefficients of the chosen active joints while \mathbf{Q}_* is a 5×1 matrix formed with the remaining active joint multiplied by its corresponding first order coefficients. Given a prescribed

velocity state ${}^A V_O^C$, expression (30) indicates that the user can select five of the six active joints and the remaining one can be used in order to avoid/escape from possible singularities, if any. Furthermore, the extra active joint can be used with the purpose to optimize trajectories. This feature is one of the main benefits of the robot D1.

5.4 Acceleration analysis

Let ${}^A \dot{\omega}^C$ and ${}^A a_O^C$ be, respectively, the angular and linear accelerations of a point O of the end-effector-platform. The reduced acceleration state, or accelerator, of the end-effector-platform with respect to the fixed platform, ${}^A A_O^C = [{}^A \dot{\omega}^C, {}^A a_O^C - {}^A \omega^C \times {}^A v_O^C]^T$, can be obtained through the coupler and fixed platforms as

$${}^A A_O^C = {}^A A_O^B + {}^B A_O^C + [{}^A V_O^B \quad {}^B V_O^C] \quad (31)$$

where ${}^B A_O^C$ is the accelerator of the coupler platform with respect to the fixed platform, ${}^B A_O^C$ is the accelerator of the end-effector-platform with respect to the coupler platform. Furthermore, these accelerators can be written in screw form as follows

$$A = J_i \dot{\Omega}_i + \mathcal{L}_i \quad A \in \{{}^A A_O^B, {}^B A_O^C, {}^A A_O^C\} \quad i = 1, 2, 3 \quad (32)$$

where $\dot{\Omega}_i \in \{{}^A \dot{\Omega}_i^B, {}^B \dot{\Omega}_i^C, {}^A \dot{\Omega}_i^C\}$ are matrices containing the joint rate accelerations of the corresponding limbs, whereas $\mathcal{L}_i \in \{{}^A \mathcal{L}_i^B, {}^B \mathcal{L}_i^C, {}^A \mathcal{L}_i^C\}$ are composed Lie products given by

$$\left. \begin{aligned} {}^A \mathcal{L}_i^C &= \sum_{j=12}^{16} \left[{}^j \omega_{j+1} i^j \mathcal{S}_i^{j+1} \sum_{k=j+1}^{17} {}^k \omega_{k+1} i^k \mathcal{S}_i^{k+1} \right] \\ {}^B \mathcal{L}_i^C &= \sum_{j=6}^{10} \left[{}^j \omega_{j+1} i^j \mathcal{S}_i^{j+1} \sum_{k=j+1}^{11} {}^k \omega_{k+1} i^k \mathcal{S}_i^{k+1} \right] \\ {}^A \mathcal{L}_i^B &= \sum_{j=0}^4 \left[{}^j \omega_{j+1} i^j \mathcal{S}_i^{j+1} \sum_{k=j+1}^5 {}^k \omega_{k+1} i^k \mathcal{S}_i^{k+1} \right] \end{aligned} \right\} \quad (33)$$

The inverse acceleration analysis consists of finding the joint rate accelerations of the robot given a prescribed accelerator ${}^A A_O^C$. This analysis is carried-out by means of expressions (31) and (32).

On the other hand the forward acceleration analysis consists of finding the accelerator of the end-effector-platform with respect to the fixed platform, ${}^A A_O^C$, given the active joint rate accelerations of the robot. This analysis is very close to the presented to solve the forward velocity analysis, therefore only the obtained expressions are included here.

The accelerator ${}^A A_O^B$ can be computed upon the input/output acceleration expression

$$J_1^T \Delta {}^A A_O^B = \begin{bmatrix} \{ {}^3 \Phi_1^4, {}^A \mathcal{L}_1^B \} \\ {}_2 \dot{\omega}_3^1 + \{ {}^4 \Phi_1^5, {}^A \mathcal{L}_1^B \} \\ \{ {}^3 \Phi_2^4, {}^A \mathcal{L}_2^B \} \\ {}_2 \dot{\omega}_3^2 + \{ {}^4 \Phi_1^5, {}^A \mathcal{L}_2^B \} \\ \{ {}^3 \Phi_3^4, {}^A \mathcal{L}_3^B \} \\ {}_2 \dot{\omega}_3^3 + \{ {}^4 \Phi_3^5, {}^A \mathcal{L}_3^B \} \end{bmatrix} \quad (34)$$

whereas the reduced acceleration state ${}^A A_O^C$ can be obtained from the input/ output acceleration relationship

$$\mathbf{J}_2^T \Delta {}^A A_O^C = \begin{bmatrix} {}_{14}\dot{\omega}_{15}^1 + \{ {}^{16}\$1^A, {}^A \mathcal{L}_1^C \} \\ {}_{14}\dot{\omega}_{15}^2 + \{ {}^{16}\$1^A, {}^A \mathcal{L}_2^C \} \\ {}_{14}\dot{\omega}_{15}^3 + \{ {}^{16}\$1^A, {}^A \mathcal{L}_3^C \} \\ \{ {}^9\$1^A, {}^A A_O^B + [{}^A V_O^B, {}^B V_O^C] + {}^B \mathcal{L}_1^C \} \\ \{ {}^9\$2^A, {}^A A_O^B + [{}^A V_O^B, {}^B V_O^C] + {}^B \mathcal{L}_2^C \} \\ \{ {}^9\$3^A, {}^A A_O^B + [{}^A V_O^B, {}^B V_O^C] + {}^B \mathcal{L}_3^C \} \end{bmatrix} \quad (35)$$

Finally, please note that the computation of the accelerators ${}^A A_O^B$ and ${}^A A_O^C$, by means respectively of expression (34) and (35), does not require the values of the passive joint rate accelerations of the robot. Furthermore, once these reduced acceleration states are calculated, the accelerator ${}^B A_O^C$ is obtained using expression (31).

6. Computer aided kinematic simulations

With the purpose to exemplify the performance of the D1 robot, in this section the kinematic behavior of a virtual prototype, left image provided in Fig. 2, is simulated by means of the commercially available software ADAMS[®].

The parameters of the robot, using hereafter SI units, are given by

$$\left. \begin{array}{l} R_1 = (0.3, 0, 0) \\ R_2 = (-0.125, 0, -0.216) \\ R_3 = (-0.125, 0, 0.216) \\ U_1 = (0.5, -0.25, 0) \\ U_2 = (-0.215, -0.25, -0.337) \\ U_3 = (-0.215, -0.25, 0.337) \\ \overline{S_1 S_2} = \overline{S_1 S_3} = \overline{S_3 S_2} = 0.173 \\ \overline{S_4 S_5} = \overline{S_4 S_6} = \overline{S_5 S_6} = 0.311 \\ \overline{S_1 R_4} = \overline{S_2 R_5} = \overline{S_3 R_6} = 0.125 \end{array} \right\}$$

whereas in the home position the coordinates of the spherical joints $S_i (i = 1, 2, \dots, 6)$ and of the tool tip T are given by

$$\left. \begin{array}{l} S_1 = (0.1, -0.5, 0) \\ S_2 = (-0.05, -0.05, 0.086) \\ S_3 = (-0.05, -0.05, -0.086) \\ S_4 = (0.18, -0.75, 0) \\ S_5 = (-0.09, -0.75, -0.155) \\ S_6 = (-0.09, -0.75, 0.155) \\ T = (0, -1.0, 0) \end{array} \right\}$$

The inverse kinematics of the robot is proved simulating the D1 robot as a parallel kinematic machine tool. To this end, two tasks are assigned to the tool tip:

- The robot will drill three holes
- The robot will mill a hexagon

In order to achieve these tasks, only five of the six available generalized coordinates are required, therefore one of them, e.g. q_3 , should be locked. After, the required instantaneous variations of the generalized coordinates satisfying such operations are provided in Fig. 4.

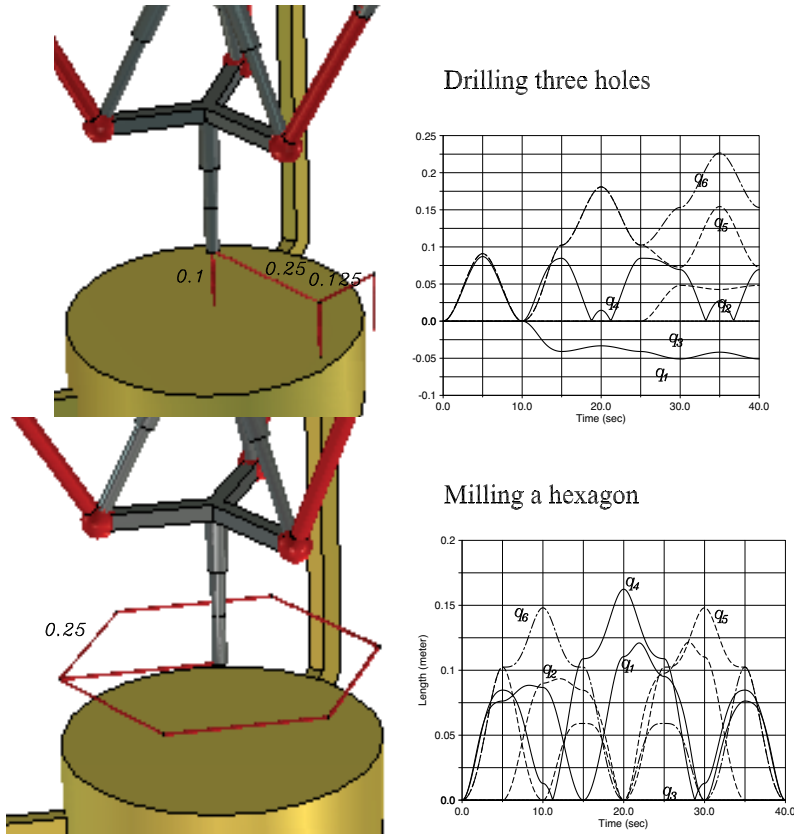


Fig. 4. The D1 robot working as a non-redundant five-degrees-of-freedom parallel kinematic machine

On the other hand, the forward kinematics of the robot is simulated by using the six generalized coordinates. In other words, the robot D1 is used as a redundant manipulator with six active joints to realize five degrees of freedom in the output platform. To this end, upon the home position of the robot D1, the active limbs are conditioned to the following periodical variations

$$\left. \begin{aligned} \Delta q_1 &= -0.01 \sin(t), \Delta q_2 = 0.0125 \sin(t) \\ \Delta q_3 &= 0.015 \sin(t) \cos(t), \Delta q_4 = 0.1 \sin(t) \cos(t) \\ \Delta q_5 &= 0.075 \sin(t), \Delta q_6 = 0.125 \sin^2(t) \\ 0.0 &\leq t \leq 2\pi \end{aligned} \right\}$$

With these data the most representative results of the simulation are provided in Fig. 5.

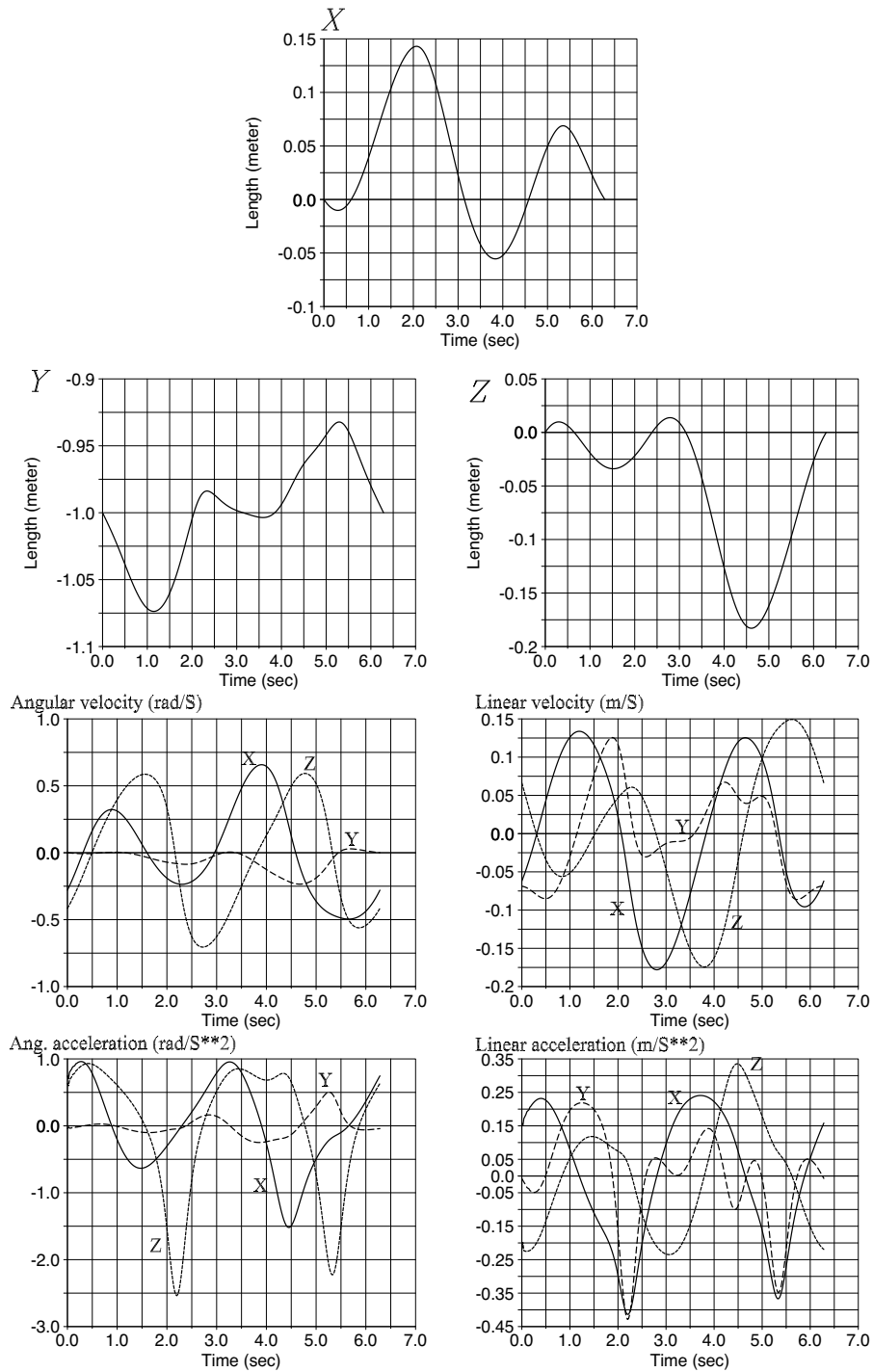


Fig. 5. Forward kinematics, six active joints to realize five degrees of freedom

Finally, of course a virtual prototype is not the final word about the correctness performance of a proposed manipulator, but it is an advisable option before the construction of a real prototype.

7. Conclusions

In this work a new class of redundant robot manipulators called DeLiA is introduced. The main features of the proposed robots are:

- Symmetry
- Decoupled architecture. Only three of the six active limbs connect the end-effector-platform to the fixed platform
- The forward position analysis, a challenging task of most parallel manipulators, is carried-out by solving two sets of non-linear equations
- The proposed robot does not require of special conditions of mechanical assembly like intersection of screws or similar
- The six active limbs are mounted on the fixed platform, simplifying the kinematics and control of the robot
- Redundancy, the robot is endowed with an extra degree of freedom which can be used with the purpose to avoid/escape from singular configurations, as well to optimize trajectories. Any of the six active joints can play this role

Finally, D1, a member of the DeLiA robot family, is simulated as a five-degrees- of-freedom parallel kinematic machine tool with the aid of commercially available software like ADAMS©.

8. Acknowledgement

This work was supported by National Council of Science and Technology of México, Conacyt.

9. References

- Altuzarra, O.; Loizaga, M.; Pinto, C. & Petuya, V. (2010). Synthesis of partially decoupled multi-level manipulators with lower mobility, *Mechanism and Machine Theory*, Vol. 45, No. 1, 106–118.
- Ball, R.S. (1999, Reprinted 1998). *A Treatise on the Theory of Screws*, Cambridge University Press, Cambridge.
- Bohez, E.L.J. (2002). Five-axis milling machine tool kinematic chain design and analysis, *International Journal of Machine Tools and Manufacture*, Vol. 42, No. 4, 505–520.
- Bonev, I.A. (2002). *Geometric Analysis of Parallel Mechanisms*, thèse de doctorat, Université Laval, Canada.
- Bonev, I. (2003). The true origins of parallel robots. Available at Parallel Mic Center, ParalleMIC online review, available at <http://www.parallemic.org/Reviews/Review007.html>
- Briot, S.; Arakelian, V. & Guégan, S. (2009). PAMINSA: a new family of partially decoupled parallel manipulators, *Mechanism and Machine Theory*, Vol. 44, No. 2, 425–444.
- Briot, S. & Bonev, C.M. (2009). Pantopteron: a new fully decoupled 3DOF translational parallel robot for pick-and-place applications, *ASME Journal of Mechanisms and Robotics*, Vol. 1, May, paper 021001, 1–9.

- Carricato, M. & Parenti-Castelli, V. (2002). Singularity-free fully-isotropic translational parallel mechanisms, *The International Journal of Robotics Research*, Vol. 21, No. 2, 161-174.
- Carricato, M. & Parenti-Castelli, V. (2004). A novel fully decoupled two- degrees-of-freedom parallel wrist, *The International Journal of Robotics Research*, Vol. 23, No. 6, 661-667.
- Dai, J.S.; Huang, Z. & Lipkin, H. (2006). Mobility of overconstrained parallel mechanisms, *ASME Journal of Mechanical Design*, Vol. 128, No. 1, 220-229.
- Gallardo, J.; Orozco, H.; Rodríguez, R. & Rico, J.M. (2007). Kinematics of a class of parallel manipulators which generates structures with three limbs, *Multibody System Dynamics*, Vol. 17, No. 1, 27-46.
- Gallardo, J.; Rico, J.M.; Frisoli, A.; Checcacci, D. & Bergamasco, M. (2003). Dynamics of parallel manipulators by means of screw theory, *Mechanism and Machine Theory*, Vol. 38, No. 11, 1113-1131.
- Gallardo-Alvarado, J., & Rico-Martínez, J.M. (2001) Jerk influence coefficients, via screw theory, of closed chains, *Meccanica*, Vol. 36, No. 2, 213-228.
- Gallardo-Alvarado, J.; Aguilar-Nájera, C.R.; Casique-Rosas, L.; Pérez-González, L. & Rico-Martínez, J.M. (2008). Solving the kinematics and dynamics of a modular spatial hyper-redundant manipulator by means of screw theory, *Multibody System Dynamics*, Vol. 20, No. 4, 307-325.
- Gallardo-Alvarado, J.; Rojas-Garduño, H. & Arroyo-Ramírez, B. (2009). Kinematics of a five-degrees-of-freedom parallel manipulator using screw theory, *The International Journal of Advance Manufacturing Technology*, Vol. 45, No. (7-8), 830-840.
- Gallardo-Alvarado, J. & Lesso-Arroyo, R. (2009). Jerk analysis of a module of an artificial spine by means of screw theory, *Journal of Advanced Research and Technology*, Vol. 7, No. 3, 249-258.
- Gao, F.; Peng, B.; Li, W. & Zhao, H. (2005). Design of a novel 5-DOF fully parallel kinematic machine tool based on workspace, *Robotica*, Vol. 23, No. 1, 35-43.
- Gao, F.; Peng, B.; Zhao, H. & Li, W. (2006). A novel 5-DOF fully parallel kinematic machine tool, *The International Journal of Advanced Manufacturing Technology*, Vol. 31, No. (1-2), 201-207.
- Gogu, G. (2005). Singularity-free fully-isotropic parallel manipulators with Schönflies motions, *Proceedings of the ICAR International Conference on Ad-vanced Robotics*, pp. 194-201, Seattle, WA.
- Gogu, G. (2005). Mobility of mechanisms: a critical review, *Mechanism and Machine Theory*, Vol. 40, No. 9, 1068-1097.
- Gosselin, C.M.; Kong, X.; Foucault, S. & Bonev, I.A. (2004). A fully decoupled 3-DOF translational parallel mechanism, *Proceedings PKM International Conference*, pp. 595610, Chemnitz, Germany.
- Gough, V.E. (1957). Contribution to discussion to papers on research in automobile stability and control and in type performance, *Proceedings Automation Division Institution of Mechanical Engineers*, 392-395.
- Gough, V.E. & Whitehall, S.G. (1962). Universal tyre test machine, *Proceedings of the FISITA Ninth International Technical Congress*, pp. 117-137.
- Innocenti, C. & Parenti-Castelli, V. (1991). Direct kinematics of the 6-4 fully parallel manipulator with position and orientation uncoupled, *Proceedings European Robotics and Intelligent Systems Conference*, pp. 23-28, Corfou.
- Kong, X. & Gosselin, C.M. (2002a). A class of 3-DOF translational parallel manipulators with linear input-output equations, *Proceedings of the Workshop on Fundamental*

- Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, pp. 2532, Quebec City, Quebec, Canada.
- Kong, X. & Gosselin, C.M. (2002b). Type synthesis of linear translational parallel manipulators, In: *Advances in Robot Kinematics Theory and Applications*, Lenarčič, J. & Thomas, F. (Eds.), Kluwer Academic Publishers, 411420.
- Kong, X. & Gosselin, C.M. (2002c). Kinematics and singularity analysis of 3- CRR 3-DOF translational parallel manipulators, *The International Journal of Robotics Research*, Vol. 21, No. 9, 791798.
- Li, Q. & Huang, Z. (2003). Type synthesis of 5-DOF parallel manipulators, *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, pp. 1203-1208, Taipei, Taiwan.
- Li, W.M.; Gao, F. & Zhang, J.J. (2005). R-CUBE, a decoupled parallel manipulator only with revolute joints, *Mechanism and Machine Theory*, Vol. 40, No. 4, 467-473.
- Piccina, O.; Bayle, B.; Maurin B. & de Mathelin, M. (2009). Kinematic modeling of a 5-DOF parallel mechanism for semi-spherical workspace, *Mechanism and Machine Theory*, Vol. 44, No. 8, 1485-1496.
- Pollard, W.L.G. (1940). *Spray painting machine*, US Patent No. 2,213,108, August 26.
- Rico-Martínez, J.M. & Duffy, J. (1996). An application of screw algebra to the acceleration analysis of serial chains, *Mechanism and Machine Theory*, Vol. 31, No. 4, 445-457.
- Rico, J.M. & Duffy, J. (2000). Forward and inverse acceleration analyses of in-parallel manipulators, *ASME Journal of Mechanical Design*, Vol. 122, No. 3, 299-303.
- Ruggiu, M. (2009). Kinematic analysis of a fully decoupled translational parallel manipulator, *Robotica*, Vol. 27, 961-969.
- Stewart, D. (1965). A platform with six degrees of freedom, *Proceedings Institution of Mechanical Engineers Part I*, Vol. 180, No. 15, 371-386.
- Takeda, Y.; Kamiyama, K.; Maki, Y.; Higuchi, M. & Sugimoto, K. (2005). Development of position-orientation decoupled spatial in-parallel actuated mechanisms with six degrees of freedom, *Journal of Robotics and Mechatronics*, Vol. 17, No. 1, 59-68.
- Tsai, L.W. (1999). *Robot Analysis*, John Wiley & Sons, New York.
- Vlachos, K. & Papadopoulos, E. (2005). Endpoint-Side optimization of a five degrees-of-freedom haptic mechanism, *Proceedings of the 13th Mediterranean Conference on Control and Automation Limassol*, pp. 27-29 Cyprus.
- Wohlhart, K. (1994). Displacement analysis of the general spherical Stewart platform, *Mechanism and Machine Theory*, Vol. 29, No. 4, 581-589.
- Yang, G.; Chen, I-M.; Chen, W. & Lin, W. (2004). Kinematic design of a six-dof parallel-kinematics machine with decoupled-motion architecture, *IEEE Transactions on Robotics*, Vol. 20, No. 5, 876-884.
- Zabalza, I.; Ros, J.; Gil, J.J.; Pintor, J.M. & Jiménez, J.M. (2002). Tri-Scott. A new kinematic structure for a 6-dof decoupled parallel manipulator, *Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, pp. 12-15, Québec, Canada.
- Zheng, K.-J.; Gao, J.-S. & Zhao, Y.S. (2005). Path control algorithms of a novel 5-DOF parallel machine tool, *Mechatronics and Automation, 2005 IEEE International Conference*, pp. 1381-1385, Niagara Falls, Ont., Canada.
- Zhu, S.J.; Huang, Z. & Zhao, M.Y. (2008). Feasible human-spine motion simulators based on parallel manipulators, In: *Parallel Manipulators, Towards New Applications*, Wu, H., ed., I-Tech Education and Publishing, Vienna, Austria.

Dynamic Modelling, Tracking Control and Simulation Results of a Novel Underactuated Wheeled Manipulator (WAcrobot)

Mohsen Moradi Dalvand and Bijan Shirinzadeh

*Robotics and Mechatronics Research Laboratory (RMRL), Department of Mechanical and Aerospace Engineering, Monash University, Clayton, Victoria
Australia*

1. Introduction

Being an inherently open loop unstable mechanical system with highly nonlinear dynamics and with the number of actuators less than the number of degrees of freedom, the inverted pendulum system is a perfect benchmark for the design of a wide range of classical and contemporary control techniques. There are a number of different versions of the inverted pendulum systems offering a variety of control challenges. The most common types are the single inverted pendulum on a cart (Ohsumi & Izumikawa, 1995; Åström & Furuta, 2000; Yoshida, 1999), the double inverted pendulum on a cart (Zhong & Rock, 2001), the double inverted pendulum with an actuator at the first joint only (Pendubot) (Spong, 1996; Graichen & Zeitz, 2005; Fantoni et al., 2000), the double inverted pendulum with an actuator at the second joint only (Acrobot) (Spong, 1994; 1995; Hauser & Murray, 1990), the rotational single-arm pendulum (Furuta et al., 1991; 1992) and the rotational two-arm pendulum (Yamakita & Furuta, 1999). Beyond non-mobile inverted pendulum robots, wheeled inverted pendulum robots or commonly known as balancing robots (e.g., Segway (Browning et al., 2005), Quasimoro (Salerno & Angeles, 2003), and Joe (Grasser et al., 2002)) have induced much interests by researchers.

The control techniques involved in various types of inverted pendulum systems are also numerous, ranging from simple conventional controllers to advanced control techniques based on modern nonlinear control theory. A vast range of contributions exists for the stabilization of different types of inverted pendulums (Mori et al., 1976; Chaturvedi et al., 2008; Angeli, 2001). Besides the stabilization aspect, the swing-up of various types of single and double inverted pendulum(s) is also addressed in the literature. Examples include classic single pendulum on a cart (Åström et al., 2008; Åström & Furuta, 2000), Acrobot and Pendubot (Fantoni et al., 2000; Spong, 1994; 1995; Graichen et al., 2007; Brown & Passino, 1997) and the rotary double inverted pendulum (Yamakita et al., 1993; 1995). In addition to the stabilization and swing-up of different kinds of inverted pendulum robots, trajectory tracking of these underactuated systems has gained attention by researches (Cho & Jung, 2003; Chanchareon et al., 2006; Hung et al., 1997; Magana & Holzapfel, 1998). There are two major approaches to construct the trajectory tracking controller for such nonlinear systems. The first one is based on system inversion (Devasia et al., 1996; Wang & Chen, 2006) and the

second approach is based on output regulation theory (Isidori & Byrnes, 1990; Qian & Lin, 2002; Hirschorn & Aranda-Bricaire, 1998). Extensive controller developments have also been achieved by researchers for mobile inverted pendulum robots over the last decade (Salerno & Angeles, 2007; Pathak et al., 2005; Tsuchiya et al., 1999).

This chapter studies a novel underactuated wheeled manipulator (WAcrobot) comprising an underactuated 2-DOF planar manipulator or an unstable double inverted pendulum (Acrobot) combined with a balancing robot. The WAcrobot has two independent driving wheels in same axis, and two gyro type sensors to determine the inclination angular velocity of two arms and rotary encoders to know wheels and arms rotation individually. Due to its configuration with two coaxial wheels, each wheel is coupled to a geared dc motor. The manipulator is able to do stationary U-turns while keeping balance and manipulating. Such manipulator is of interest because it has a small foot-print and can turn on dime. The design, dynamic modeling and tracking control of this novel mobile manipulator is discussed in this chapter for the first time. This chapter aims at achieving three different types of trajectory tracking control tasks for *a)* wheels, *b)* first or second arm and *c)* wheels and one of the arms simultaneously, while the WAcrobot stabilization is guaranteed by the system internal equilibria calculation. The tracking controller is designed using the Gain Scheduling method that is based on the idea of the linearisation of the system equations around certain operating points and design of a linear controller for each region of operation (Lawrence & Rugh, 1993; Shamma & Athans, 1990a)]. For the design of the linear controller, we consider the Linear Quadratic Regulator (LQR) model to stabilize the WAcrobot around any point over the equilibrium manifold. We verified the effectiveness of the designed control system via numerical simulation visualized by graphical simulation to illustrate the physical response of the WAcrobot.

In the following sections of this chapter the dynamic model of the wheeled manipulator (WAcrobot) is firstly presented. Then the equilibrium manifold of the WAcrobot is investigated. After that the stabilization controller based on LQR technique is proposed. Then by employing Gain Scheduling method, for any given trajectory of wheels and/or arm(s), the trajectories of the rest of DOF of the WAcrobot is determined such that during the trajectory tracking the WAcrobot system is stabilized. Numerical and graphical simulations for three types of tracking control tasks are given to show the effectiveness of the proposed scheme.

2. Dynamics of WAcrobot

The mechanism of the WAcrobot is shown in Figure 1 schematically. The WAcrobot (Wheeled Acrobot) is an underactuated mechanical system consisting of an underactuated planar manipulator (Acrobot), a double inverted pendulum robot with an actuator at the second joint only (Figure 1-a), which is combined with a balancing robot (Figure 1-b) or equipped with two actuated wheels and has the capability to be as an underactuated wheeled manipulator. The mathematical model of the WAcrobot can be derived using the Euler-Lagrange equation. The form of the Euler-Lagrangian equation used here is:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = \tau \quad (1)$$

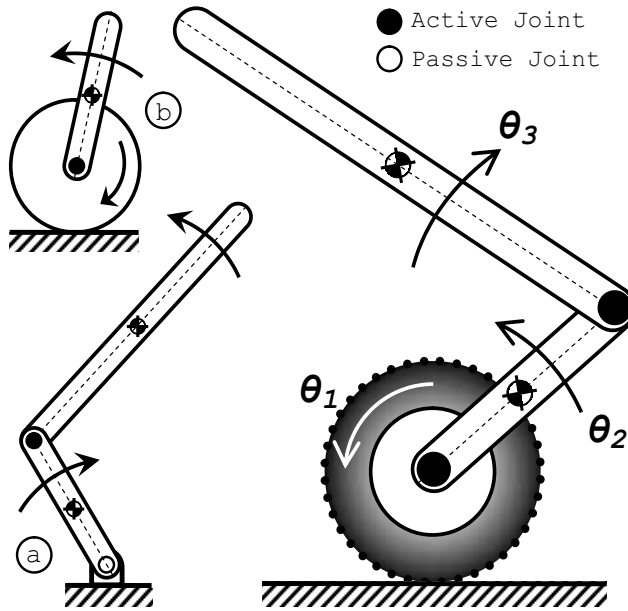


Fig. 1. WAcrobot, Acrobot (a) and Wheeled Inverted Pendulum (b)

where $L = T - V$ is a Lagrangian, T is kinetic energy, V is potential energy, $\tau = [\tau_1 \ 0 \ \tau_2]^T$ is the input generalized force vector produced by two actuators at wheels and second arm, $q = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^3$ is generalized coordinate vector which is selected as $q = [\theta_1 \ \theta_2 \ \theta_3]^T$ where θ_1 , θ_2 and θ_3 are angular positions of wheels, first arm, and second arm of the WAcrobot, respectively. The kinetic and potential energies of the WAcrobot's components in terms of generalized coordinates can be determined as:

$$\begin{aligned} & \frac{1}{2}(I_1 + (m_1 + m_2 + m_3)l_1^2)\dot{\theta}_1^2 + \frac{1}{2}(m_3l_{c3}^2 + I_3)\dot{\theta}_3^2 + m_3l_{c3}l_2\dot{\theta}_2\dot{\theta}_3\cos(\theta_3) = T \quad (2) \\ & + \frac{1}{2}(2m_3\cos(\theta_3)l_2l_{c3} + m_2l_{c2}^2 + m_3l_2^2 + m_3l_{c3}^2 + I_2 + I_3)\dot{\theta}_2^2 + m_3l_{c3}l_{c3}\dot{\theta}_2\dot{\theta}_3 \\ & + l_1((m_2l_{c2} + m_3l_2)\cos(\theta_2) + m_3l_{c3}\cos(\theta_2 + \theta_3))\dot{\theta}_1\dot{\theta}_2 + m_3l_1l_{c3}\cos(\theta_2 + \theta_3)\dot{\theta}_1\dot{\theta}_3 \\ & (m_1 + m_2 + m_3)l_1g + m_3l_{c3}\cos(\theta_2 + \theta_3)g + (m_2l_{c2} + m_3l_2)\cos(\theta_2)g = V \quad (3) \end{aligned}$$

Differentiating the Lagrangian ($L = T - V$) by generalized coordinate vector θ and $\dot{\theta}$ yields Euler-Lagrange Equation (1) as:

$$\begin{aligned} & ((m_1 + m_2 + m_3)l_1^2 + I_1)\ddot{\theta}_1 + l_1(l_{c3}m_3\cos(\theta_2 + \theta_3) + \cos(\theta_2)(m_2l_{c2} + m_3l_2))\ddot{\theta}_2 = \tau_1 \quad (4) \\ & + m_3l_1l_{c3}\cos(\theta_2 + \theta_3)\ddot{\theta}_3 - l_1(l_{c3}m_3\sin(\theta_2 + \theta_3) + (l_2m_3 + l_{c2}m_2)\sin(\theta_2))\dot{\theta}_2^2 \\ & - l_1l_{c3}m_3\sin(\theta_2 + \theta_3)\dot{\theta}_3^2 - 2l_1l_{c3}m_3\sin(\theta_2 + \theta_3)\dot{\theta}_2\dot{\theta}_3 \end{aligned}$$

$$l_1(l_{c3}m_3\cos(\theta_2 + \theta_3) + (m_2l_{c2} + m_3l_2)\cos(\theta_2))\ddot{\theta}_1 + m_3l_{c3}(l_{c3} + l_2\cos(\theta_3))\ddot{\theta}_3 = 0 \quad (5)$$

$$\begin{aligned} & + (m_2l_{c2}^2 + m_3(l_{c3}^2 + l_2^2) + I_2 + I_3 + 2m_3l_2l_{c3}\cos(\theta_3))\ddot{\theta}_2 - m_3l_2l_{c3}\sin(\theta_3)\dot{\theta}_3^2 \\ & - (m_3l_{c3}\sin(\theta_2 + \theta_3) + (m_3l_2 + m_2l_{c2})\sin(\theta_2))g - 2m_3l_2l_{c3}\sin(\theta_3)\dot{\theta}_2\dot{\theta}_3 \\ & l_1l_{c3}m_3\cos(\theta_2 + \theta_3)\ddot{\theta}_1 + m_3l_{c3}(l_{c3} + l_2\cos(\theta_3))\ddot{\theta}_2 + (m_3l_{c3}^2 + I_3)\ddot{\theta}_3 = \tau_2 \quad (6) \\ & + l_2l_{c3}m_3\sin(\theta_3)\dot{\theta}_2^2 - l_{c3}m_3\sin(\theta_2 + \theta_3)g \end{aligned}$$

Equations (4), (5) and (6) can be put into the frequently used compact form (Spong & Block, 1995):

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (7)$$

where $\theta = [\theta_1 \ \theta_2 \ \theta_3]^T \in \mathbb{R}^3$ is the generalized coordinate vector, $M(\theta) \in \mathbb{R}^{3 \times 3}$ is the symmetric positive definite inertia matrix, $C(\theta, \dot{\theta}) \dot{\theta} \in \mathbb{R}^3$ contains Coriolis and centrifugal terms, $G(\theta) \in \mathbb{R}^3$ contains gravitational terms and $\tau = [\tau_1 \ 0 \ \tau_2]^T$ is the input generalized force vector. Furthermore,

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}, \quad (8)$$

where

$$\begin{aligned} M_{11} &= (m_1 + m_2 + m_3)l_1^2 + I_1 \\ M_{12} = M_{21} &= l_1(l_{c3}m_3\cos(\theta_2 + \theta_3) + \cos(\theta_2)(m_2l_{c2} + m_3l_2)) \\ M_{13} = M_{31} &= m_3l_1l_{c3}\cos(\theta_2 + \theta_3) \\ M_{22} &= m_2l_{c2}^2 + m_3(l_{c3}^2 + l_2^2) + I_2 + I_3 + 2m_3l_2l_{c3}\cos(\theta_3) \\ M_{23} = M_{32} &= m_3l_{c3}(l_{c3} + l_2\cos(\theta_3)) \\ M_{33} &= m_3l_{c3}^2 + I_3 \end{aligned}$$

and

$$C(\theta, \dot{\theta})\dot{\theta} = [H_1 \ H_2 \ H_3]^T \quad (9)$$

where

$$\begin{aligned} H_1 &= -l_1(l_{c3}m_3\sin(\theta_2 + \theta_3) + (l_2m_3 + l_{c2}m_2)\sin(\theta_2))\dot{\theta}_2^2 \\ & - l_1l_{c3}m_3\sin(\theta_2 + \theta_3)\dot{\theta}_3^2 - 2l_1l_{c3}m_3\sin(\theta_2 + \theta_3)\dot{\theta}_2\dot{\theta}_3 \\ H_2 &= -2m_3l_2l_{c3}\sin(\theta_3)\dot{\theta}_2\dot{\theta}_3 - m_3l_2l_{c3}\sin(\theta_3)\dot{\theta}_3^2 \\ H_3 &= +l_2l_{c3}m_3\sin(\theta_3)\dot{\theta}_2^2 \end{aligned}$$

and

$$G(\theta) = [G_1 G_2 G_3]^T \quad (10)$$

where

$$G_1 = 0$$

$$G_2 = -(m_3 l_{c3} \sin(\theta_2 + \theta_3) + (m_3 l_2 + m_2 l_{c2}) \sin(\theta_2))g$$

$$G_3 = -l_{c3} m_3 \sin(\theta_2 + \theta_3)g$$

and g is the gravitational acceleration. The parameters of the WAcrobot are defined in Table 1. Equation (7) represents the underactuated and nonlinear system of the WAcrobot including two input torques applied to wheels and second arm (τ_1 and τ_2), two active DOFs (θ_1 and θ_3) and one passive DOF (θ_2).

$\theta_i (i = 1, 2, 3)$	Angular rotation of the wheels and arms
$m_i (i = 1, 2, 3)$	Mass of wheels and arms
$l_{ci} (i = 2, 3)$	Length from the joint to the center of the gravity of the arms
$l_i (i = 1, 2, 3)$	Radius of the wheels and length of arms
$I_i (i = 1, 2, 3)$	Inertia moment around the center of gravity

Table 1. Definition of Parameters

3. Tracking control

The tracking controller of the WAcrobot is designed using the Gain Scheduling method based on the linearisation of the system equations around certain equilibrium points in a first stage followed by the design of a linear controller for each region of tracking operation in a second stage. For the design of the linear controller, we consider the Linear Quadratic Regulator (LQR) model to stabilize the WAcrobot around any operating point over the equilibrium manifold.

3.1 Equilibrium manifold

Underactuated mechanical systems generally have equilibria which depend on both their kinematic and dynamic parameters (Bortoff & Spong, 1992). In these systems, to track a trajectory while balancing is guaranteed, it is vital to consider the equilibrium manifold. Beyond the unforced equilibria of the WAcrobot, $(\theta, \dot{\theta}) = (\theta_1, \pi, 0, 0, 0, 0)$ (lower or pendent equilibrium) and $(\theta, \dot{\theta}) = (\theta_1, 0, 0, 0, 0, 0)$ (upper or inverted equilibrium), it has a manifold of forced equilibrium points. Generally the WAcrobot is at rest or particularly at equilibrium point whenever $\theta_{1_{eq}}, \dot{\theta}_{2_{eq}}$ and $\dot{\theta}_{3_{eq}}$ are zero and the joint torque $\tau_{eq} = [\tau_{1_{eq}} \ 0 \ \tau_{2_{eq}}]^T$ is such that to equalize $G(\theta)$ in Equation (7). So this set of equilibrium points consists of all states where

$$\ddot{\theta}_{eq} = \dot{\theta}_{eq} = 0 \quad (11)$$

$$G(\theta_{eq}) = \tau_{eq} \quad (12)$$

If the outputs that are required to track a trajectory include the first arm, it follows from Equations (7), (10), (11) and (12) that:

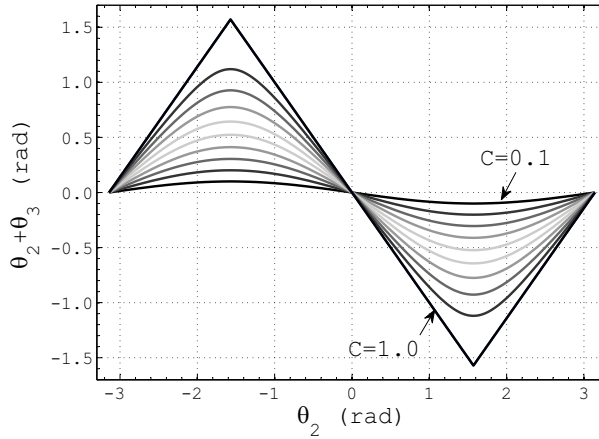


Fig. 2. Equilibrium set points according to the different values of parameter C

$$\tau_{1_{eq}} = 0$$

$$\tau_{2_{eq}} = (m_3 l_2 + m_2 l_{c2}) g \sin(\theta_{2_{eq}}) \quad (13)$$

$$\theta_{3_{eq}} + \theta_{2_{eq}} = \arcsin\left[-\left(\frac{m_3 l_2 + m_2 l_{c2}}{m_3 l_{c3}}\right) \sin(\theta_{2_{eq}})\right] \quad (14)$$

Since the value of the absolute angular position of the second arm with respect to the vertical direction $\theta_{3_{eq}}^a = \theta_{3_{eq}} + \theta_{2_{eq}}$ cannot be imaginary, the condition for the existence of the equilibrium from Equation (14) may be written:

$$C = \frac{m_3 l_2 + m_2 l_{c2}}{m_3 l_{c3}} \leq 1 \quad (15)$$

Figure 2 illustrates equilibrium set points derived from Equation (14) according to different values of parameter C from Equation (15). It demonstrates how value of parameter C affects $\theta_{3_{eq}}^a$ corresponding to any given $\theta_{2_{eq}}$ in which the WAcrobot is stabilized. By decreasing parameter C, the required $\theta_{3_{eq}}^a$, corresponding to the desired $\theta_{2_{eq}}$ to stabilize the robot, decreases and to decrease parameter C, the second arm should be long and heavy which is not suitable. On the other hand, for any given angular position of the first arm, if the second arm is long and heavy, it needs smaller angular changes to stabilize the WAcrobot and vice versa. Therefore there needs to be a trade-off between the ranges of the rotational motions of arms and the volume and weight of the WAcrobot. In particular, for any given m_3 , if the specification of the first arm (m_2 , l_2 and l_{c2}) are given, Equation (15) is only true if $l_{c3} \geq (m_3 l_2 + m_2 l_{c2}) / m_3$ and for any given l_{c3} , it is only true if $m_3 \geq m_2 l_{c2} / (l_{c3} - l_2)$. Considering $l_2 = 2l_{c2}$ and $l_3 = 2l_{c3}$, we can simplify Equation (15) as:

$$\frac{l_3}{l_2} \geq \left(2 + \frac{m_2}{m_3}\right) \quad (16)$$

From the other point of view, if the trajectory tracking of the second arm is desired, it follows from Equations (7), (10), (11) and (12) that:

$$\begin{aligned}\tau_{1_{eq}} &= 0 \\ \tau_{2_{eq}} &= -m_3 l_{c3} g \sin(\theta_{3_{eq}}^a)\end{aligned}\quad (17)$$

$$\theta_{2_{eq}} = \arcsin\left[-\left(\frac{m_3 l_{c3}}{m_3 l_2 + m_2 l_{c2}}\right) \sin(\theta_{3_{eq}}^a)\right] \quad (18)$$

Since the value of the angular position of the first arm ($\theta_{2_{eq}}$) cannot be imaginary, the condition for the existence of the equilibrium from Equation (18) is:

$$C^{-1} = \frac{m_3 l_{c3}}{m_3 l_2 + m_2 l_{c2}} \leq 1 \quad (19)$$

Considering $l_2 = 2l_{c2}$ and $l_3 = 2l_{c3}$, we can simplify Equation (19) as:

$$\frac{l_3}{l_2} \leq \left(2 + \frac{m_2}{m_3}\right) \quad (20)$$

3.2 Stabilization

The balancing controller is designed using the well known Linear Quadratic Regulator (LQR) method based on the linearised plant model around any equilibrium point. The LQR is a controller for state variable feedback in such a way that $u = -Kx$ is the input so that the value of K is obtained from minimization of the cost function $J = \int_0^{\infty} (x'Qx + u'Ru)dt$ where matrix Q and R are positive semidefinite matrix and symmetric positive definite matrix that penalize the state error and the control effort, respectively.

3.3 Gain scheduling

Jacobian linearisation or linearisation about an equilibrium point is the technique for transforming original system models into equivalent models with simpler form. Since the linearization is about a single point, trajectory tracking can only be guaranteed in a sufficiently small region of states about that point. There are several methods for circumventing this problem; one of the most common is Gain Scheduling (Shamma & Athans, 1990b). Control of nonlinear systems by Gain Scheduling is based on the idea of the linearising the system equations around certain operating points, and the design of a linear controller for each region of operation over the entire motion envelope (Cloutier et al., 1996; Dorato et al., 1994; Langson, 1997). The controller coefficients are varied continuously according to the value of the scheduling variable. In fact, this can be performed in a more or less continuous fashion using a technique called extended linearisation (Baumann & Rugh, 1986).

In broad terms, according to (WJ & Shamma, 2000), the design of a gain scheduled controller for nonlinear plant of the WAcrobot can be described with a six-step procedure, though

various technical methods are available in each step. The first step involves finding $\theta_{3_{eq}}^a$ or $\theta_{2_{eq}}$ in all operating points, for each desired $\theta_{2_{eq}}$ or $\theta_{3_{eq}}^a$, using Equations 14 or 18 respectively. The second step is the calculation of the joint torque required to keep the WAcrobot at desired $\theta_{2_{eq}}$ (or $\theta_{3_{eq}}^a$) and calculated $\theta_{3_{eq}}^a$ (or $\theta_{2_{eq}}$). The third step is the computation of a linear parameter varying model for the plant. The most common approach is to linearise the nonlinear plant around a selection of equilibrium points. This results in a family of operating points. The fourth step is to design a family of controllers for the linearised models in each operating point. Because of the linearised model, linear controller design methods such as LQR can be used to stabilize the system around the operating point. The fifth step is the actual Gain Scheduling. Gain Scheduling involves the implementation of the family of linear controllers such that the controller coefficients are scheduled according to the current value of the scheduling variables which are $\theta_{2_{eq}}$ or $\theta_{3_{eq}}^a$. The last step is the performance assessment that can be performed analytically or by using extensive computational analysis and simulation.

3.4 Computational analysis and simulation

In order to verify the validity of the Gain Scheduling method for trajectory tracking of different types of reference trajectories in the WAcrobot, we carried out computational analyses and visual simulations using *MATLAB/Simulink*[®] package integrated with *ADAMS*[®] simulation software. Three types of tracking control tasks for wheels and/or arm(s) have been evaluated which are presented in this section. The simulations are performed with the following parameters given in Table 2.

Wheels/Arms	Wheels	First arm	Second arm
m_i [kg]	1.22	0.28	0.72
l_i [m]	0.05	0.15	0.45
l_{ci} [kg]	—	0.075	0.225
I_i [kg.m ²]	1.53E-003	5.98E-004	1.3138E-002

Table 2. Parameters of the WAcrobot

In table 2, parameter l , for wheels, means radius while for arms means length. From Equation (15) and Table 2, we obtain $C=0.763$. It is supposed that the WAcrobot starts the trajectory tracking from its unforced inverted equilibrium position. Therefore the initial conditions are as follow:

$$\theta_1 = 0 \quad \theta_2 = 0 \quad \theta_3 = 0 \quad \dot{\theta}_1 = 0 \quad \dot{\theta}_2 = 0 \quad \dot{\theta}_3 = 0 \quad \tau_1 = 0 \quad \tau_2 = 0$$

Q and R in the optimal regulators for simulations are designed as:

$$Q = \text{diag}([10, 100, 100, 0, 0, 0])$$

$$R = \text{diag}([0.1, 0.1])$$

It must be noted that in order to have better sense of motion, the angular position and velocity of the second arm are considered as absolute states and are plotted with respect to the vertical direction not relative to the first arm.

3.4.1 Wheels tracking

The control objective is that the wheels to follow a trajectory with linear segments and parabolic blends while both arms are balancing inverted close to their initial positions. Practically this task is that the WAcrobot smoothly starts moving at $x = 0$ (m) and gently stops at $x = 1.5$ (m) while both arms are stabilized during the movement. Figure 3 shows the computational analysis results. In this figure the responses of angular position and velocity of the wheels and arms as well as applied torques to actuated DOFs are shown respectively. Tracking errors are calculated for the linear position and velocity of the WAcrobot, as shown in Figure 4. It should be noted that this control task can be defined as tracking problem for wheels while the arms, instead of being at inverted position, are at any point over the equilibrium manifold. Assume that the WAcrobot is balanced while the first arm is at 1 (rad). In this case the absolute angular position of the second arm and the input torque for the second joint required to keep the arms balanced at the specified angular positions, are calculated using Equations (13) and (14). Therefore the initial conditions for this simulation are as follow:

$$\theta_1 = 0 \quad \theta_2 = 1 \quad \theta_3 = -0.7343 \quad \dot{\theta}_1 = 0 \quad \dot{\theta}_2 = 0 \quad \dot{\theta}_3 = 0 \quad \tau_1 = 0 \quad \tau_2 = -1.065$$

Figure 5 demonstrates a superimposed snapshot of the graphical simulation for two tracking problems of the WAcrobot's wheels while the arms are at the inverted position and are at another point over the equilibrium manifold. It is clear from both numerical and graphical simulations that the WAcrobot's wheels track a specified trajectory while both arms are close to the inverted position or a defined position over the equilibrium manifold at all times during the movement.

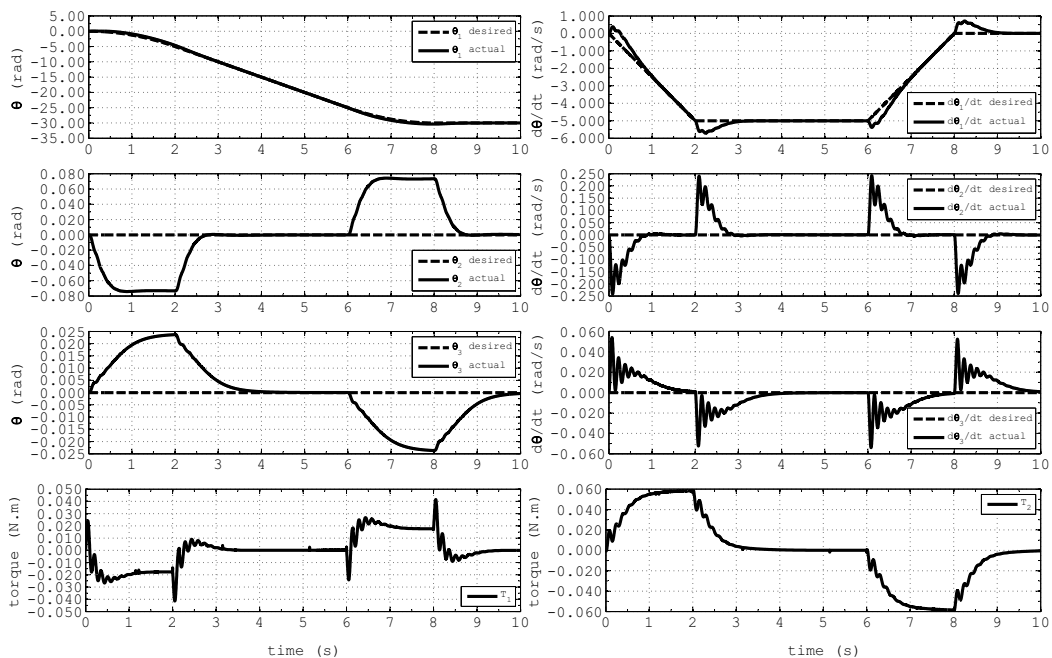


Fig. 3. The simulation responses of positions, velocities and torques for the wheels tracking

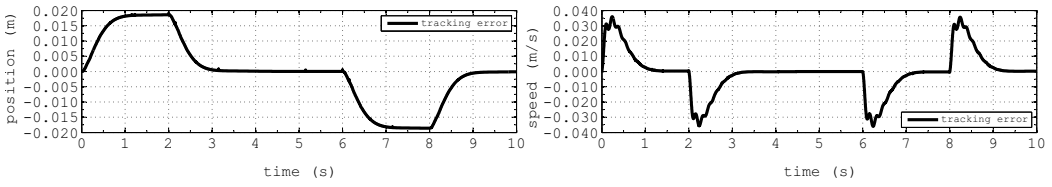


Fig. 4. Tracking errors for the linear position and velocity of the wheels for wheels tracking

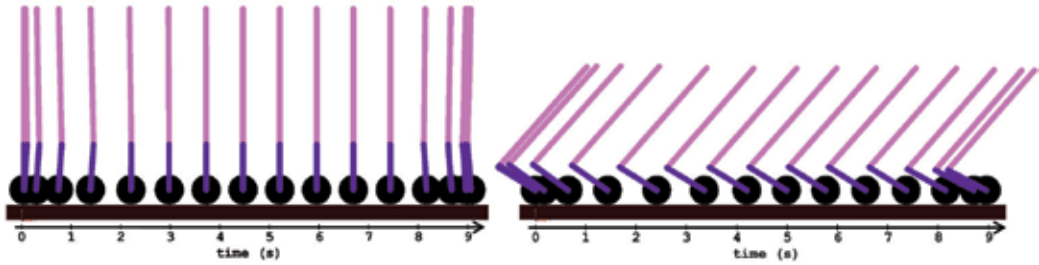


Fig. 5. Superimposed snapshot of the visualized simulation of the wheels tracking task while the arms are in inverted position (left) and are not in inverted position (right)

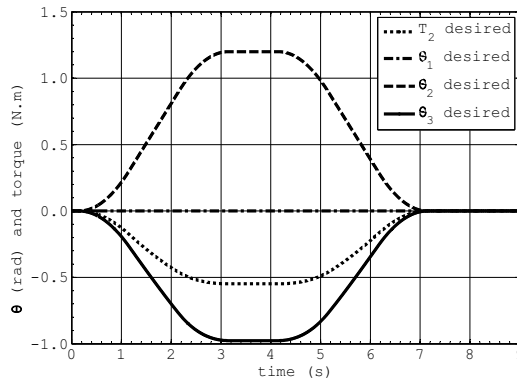


Fig. 6. The desired and calculated trajectories for the wheels, arms and time-varying torque

3.4.2 Arms tracking

Another tracking control objective defined for the WAcrobot is that the first arm to follow a trajectory with linear segments and parabolic blends while the WAcrobot's wheels are fixed in the initial position. Practically this task is that the first arm to smoothly start rotating from the inverted position and stop at a special angular position while wheels have no rotation during the tracking. To make the first arm to track the desired trajectory, the second arm should also track a calculated trajectory to make the WAcrobot stabilized during the tracking motion. Therefore the tracking problem for the first arm is also a tracking problem for the second arm. Figure 6 shows the desired and calculated trajectories for the first and second arm as well as the calculated time-varying torque required to be applied at the second joint to keep the WAcrobot balanced. Tracking errors of the linear and angular positions and velocities of the wheels, first arm and second arm are displayed in Figure 7 from top to bottom, respectively.

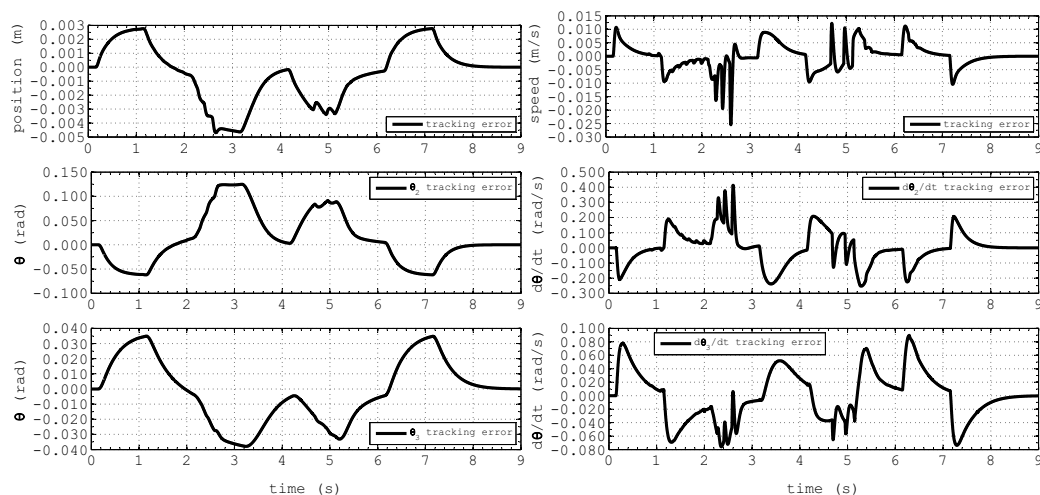


Fig. 7. Tracking errors of the positions and velocities of the wheels and arms for arms tracking

Simulation results are shown in Figure 8. In this figure the simulation responses of angular positions and velocities of wheels and arms as well as applied torques to actuated degrees of freedom are demonstrated, respectively. To show the correlation between the computational analysis results and the WAcrobot physical response, the graphical simulation is prepared and is shown in Figure 9.

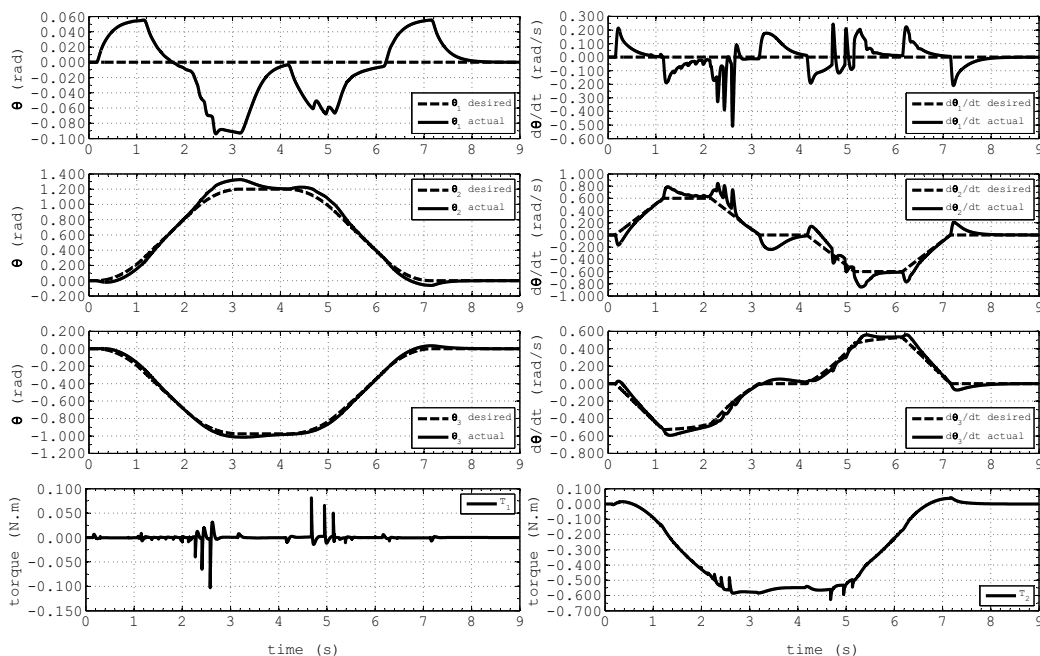


Fig. 8. The simulation responses of positions, velocities and torques for the first arm tracking

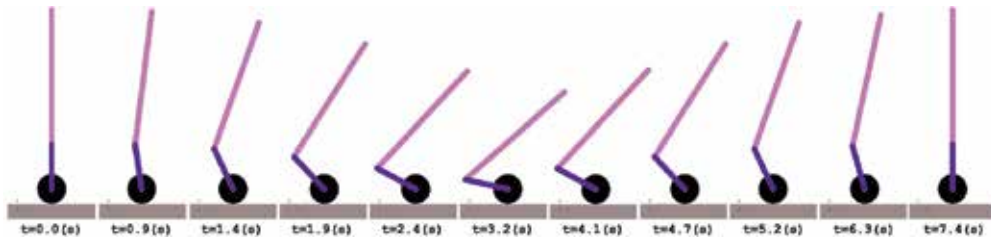


Fig. 9. Snapshots of the visualized simulation for the arms tracking task

3.4.3 Wheels and arm tracking

The most complicated task is that the first arm to follow a trajectory while the wheels are tracking another specified trajectory. In other words, this task is that the first arm to track the trajectory while the WAcrobot starts moving from the initial position ($x = 0 (m)$) and stop at $x = 1.6 (m)$ smoothly. Both specified trajectories are linear segments with parabolic blends and are shown in Figure 10. Also the calculated trajectories for the angular position of the second arm as well as the input torque required at the second joint are also plotted in Figure 10.

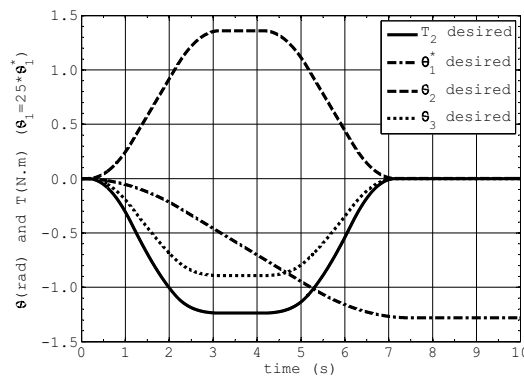


Fig. 10. The desired and calculated trajectories for the wheels, arms and time-varying torque

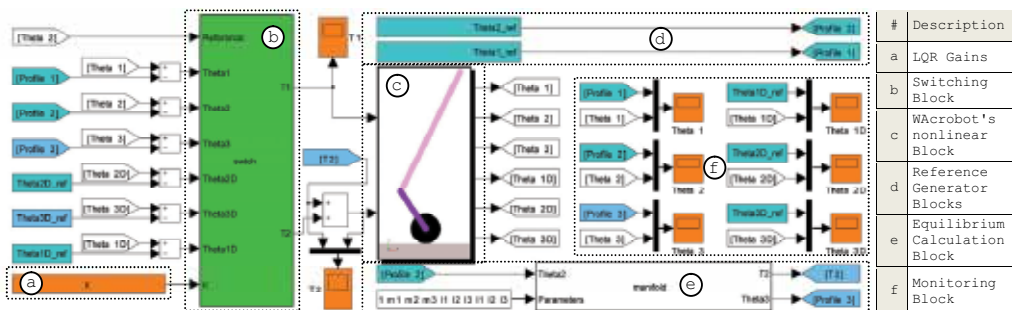


Fig. 11. The Simulink[®] block diagram of the WAcrobot system with tracking controller

The WAcrobot system with gain scheduling tracking controller simulated in Simulink[®] is illustrated in Figure 11. Figure 12 shows the simulation results of the wheels and arm tracking task. In this figure the simulation responses of the angular positions and velocities

of wheels and arms as well as applied torques to actuated DOFs are shown. Figure 13 respectively displays the tracking errors of the linear and angular positions and velocities of the wheels and arms from top to bottom. Figure 14 shows a superimposed snapshot of the visualized simulation of the WAcrobot while wheels are tracking the specified trajectory and the first arm is tracking another specified trajectory simultaneously. The simulation results illustrate the effectiveness of the proposed control methodology and the developed theory.

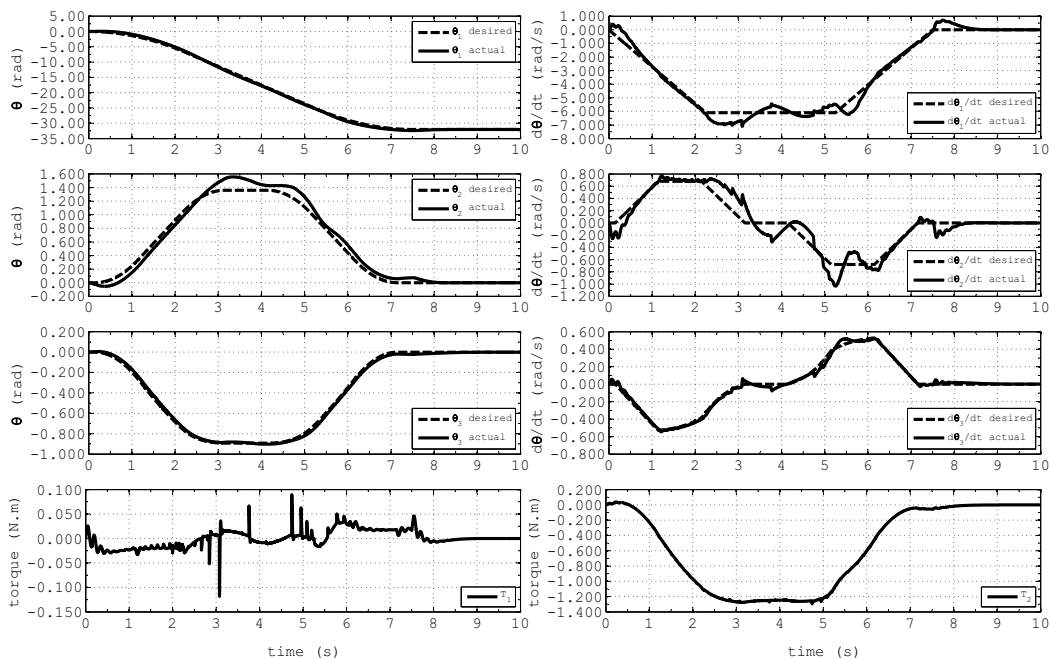


Fig. 12. The simulation responses of positions, velocities and torques for the manipulator

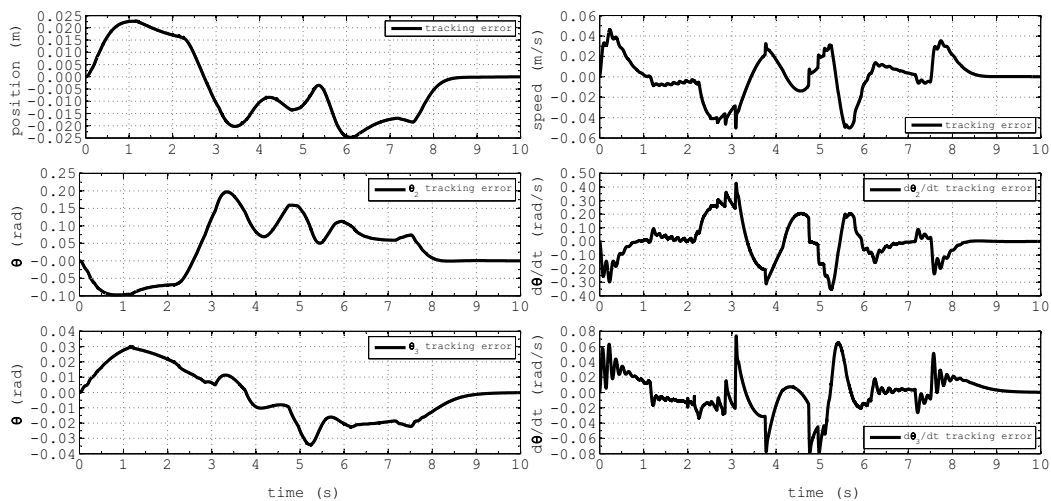


Fig. 13. Tracking errors of the positions and velocities of the wheels and arms

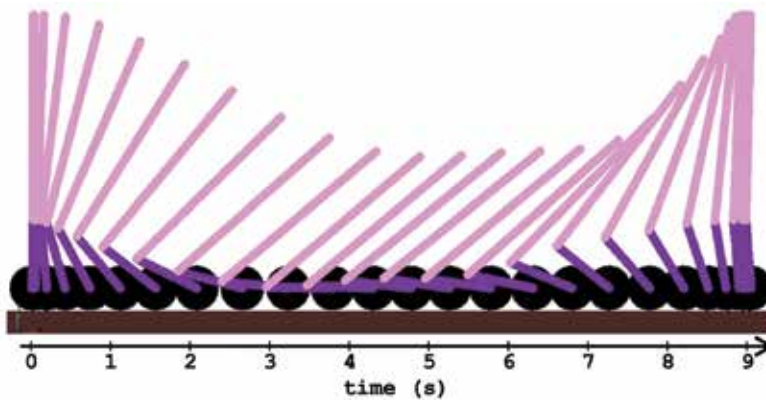


Fig. 14. Superimposed snapshot of the graphical simulation of the wheels and arms tracking

4. Conclusion

In this chapter the WAcrobot, a novel underactuated manipulator which is the combination of a well-known double inverted pendulum (Acrobot) and a wheeled inverted pendulum, was proposed and the tracking control algorithm of this mobile manipulator was investigated. The balancing controller is designed using the well known Linear Quadratic Regulator (LQR) method and the tracking controller was designed on the basis of the Gain Scheduling control strategy. Three different types of trajectory tracking tasks were investigated including tracking of *a*) wheels, *b*) first or second arm and *c*) wheels and first or second arm simultaneously.

This chapter also provided numerical and graphical simulation results to validate the obtained theoretical results and to demonstrate the correlation between the numerical results and the WAcrobot physical response. Simulation results illustrated good performance results for different tracking controls designed based on the Gain Scheduling method.

Research into the control of this novel robotic system is just in the beginning and there are a number of research problems that remain to be addressed. It would be desirable to develop the theory of robust and adaptive controller for swing-up control problem of the WAcrobot.

5. References

- Angeli, D. (2001). Almost global stabilization of the inverted pendulum via continuous state feedback, *Automatica* 37(7): 1103–1108.
- Åström, K., Aracil, J. & Gordillo, F. (2008). A family of smooth controllers for swinging up a pendulum, *Automatica* 44(7): 1841–1848.
- Åström, K. & Furuta, K. (2000). Swinging up a pendulum by energy control, *Automatica* 36: 287–295.
- Baumann, W. & Rugh, W. (1986). Feedback control of nonlinear systems by extended linearization, *IEEE Transactions on Automatic Control* 31(1): 40–46.
- Bortoff, S. & Spong, M. (1992). Pseudolinearization of the acrobot using spline functions, *Proceedings of the 31st IEEE Conference on Decision and Control*, pp. 593–598.
- Brown, S. & Passino, K. (1997). Intelligent control for an acrobot, *Journal of Intelligent and Robotic Systems* 18(3): 209–248.

- Browning, B., Searock, J., Rybski, P. & Veloso, M. (2005). Turning segways into soccer robots, *Industrial Robot: An International Journal* 32(2): 149–156.
- Chanchareon, R., Sangveraphunsiri, V. & Chantranuwathana, S. (2006). Tracking Control of an Inverted Pendulum Using Computed Feedback Linearization Technique, *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1–6.
- Chaturvedi, N., McClamroch, N. & Bernstein, D. (2008). Stabilization of a 3D axially symmetric pendulum, *Automatica* 44(9): 2258–2265.
- Cho, H. & Jung, S. (2003). Balancing and position tracking control of an inverted pendulum on a xy plane using decentralized neural networks, *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, Vol. 1.
- Cloutier, J., D'Souza, C. & Mracek, C. (1996). Nonlinear regulation and nonlinear H-infinity control via the state-dependent Riccati equation technique. I- Theory, *Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace*, pp. 117–130.
- Devasia, S., Chen, D. & Paden, B. (1996). Nonlinear inversion-based output tracking, *IEEE Transactions on Automatic Control* 41(7): 930–942.
- Dorato, P., Cerone, V. & Abdallah, C. (1994). *Linear-Quadratic Control: An Introduction*, Simon & Schuster.
- Fantoni, I., Lozano, R. & Spong, M. (2000). Energy based control of the pendubot, *IEEE Transactions on Automatic Control* 45(4): 725–729.
- Furuta, K., Yamakita, M. & Kobayashi, S. (1991). Swing up control of inverted pendulum, *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation (IECON'91)*, pp. 2193–2198.
- Furuta, K., Yamakita, M. & Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback, *Proceedings of the Institution of Mechanical Engineers* 206: 263–9.
- Graichen, K., Treuer, M. & Zeitz, M. (2007). Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation, *Automatica* 43(1): 63–71.
- Graichen, K. & Zeitz, M. (2005). Nonlinear feedforward and feedback tracking control with input constraints solving the pendubot swing-up problem, *Preprints of 16th IFAC world congress, Prague, CZ*.
- Grasser, F., D' Arrigo, A., Colombi, S. & Rufer, A. (2002). JOE: a mobile, inverted pendulum, *IEEE Transactions on industrial electronics* 49(1): 107–114.
- Hauser, J. & Murray, R. (1990). Nonlinear controllers for non-integrable systems: The acrobat example, *American Control Conference*, pp. 669–671.
- Hirschorn, R. & Aranda-Bricaire, E. (1998). Global approximate output tracking for nonlinear systems, *IEEE Transactions on Automatic Control* 43(10): 1389–1398.
- Hung, T., Yeh, M. & Lu, H. (1997). A PI-like fuzzy controller implementation for the inverted pendulum system, *Proceedings of the IEEE International Conference on Intelligent Processing Systems (ICIPS'97)*, Vol. 1.
- Isidori, A. & Byrnes, C. (1990). Output regulation of nonlinear systems, *IEEE Transactions on Automatic Control* 35(2): 131–140.
- Langson, W. (1997). *Optimal and suboptimal control of a class of nonlinear systems*, PhD thesis, University of Illinois.
- Lawrence, D. & Rugh, W. (1993). Gain scheduling dynamic linear controllers for a nonlinear plant, *Proceedings of the 32nd IEEE Conference on Decision and Control*, pp. 1024–1029.

- Magana, M. & Holzapfel, F. (1998). Fuzzy-logic control of an inverted pendulum with vision feedback, *IEEE Transactions on Education* 41(2): 165–170.
- Mori, S., Nishihara, H. & Furuta, K. (1976). Control of unstable mechanical system Control of pendulum, *International Journal of Control* 23(5): 673–692.
- Ohsumi, A. & Izumikawa, T. (1995). Nonlinear control of swing-up and stabilization of an inverted pendulum, *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 4.
- Pathak, K., Franch, J. & Agrawal, S. (2005). Velocity and position control of a wheeled inverted pendulum by partial feedback linearization, *IEEE Transactions on Robotics* 21(3): 505–513.
- Qian, C. & Lin, W. (2002). Practical output tracking of nonlinear systems with uncontrollable-unstable linearization, *IEEE Transactions on Automatic Control* 47(1): 21–36.
- Salerno, A. & Angeles, J. (2003). On the nonlinear controllability of a quasiholonomic mobile robot, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03)*, Vol. 3.
- Salerno, A. & Angeles, J. (2007). A new family of two-wheeled mobile robots: Modeling and controllability, *IEEE Transactions on Robotics* 23(1): 169–173.
- Shamma, J. & Athans, M. (1990a). Analysis of gain scheduled control for nonlinear plants, *IEEE Transactions on Automatic Control* 35(8): 898–907.
- Shamma, J. & Athans, M. (1990b). Analysis of nonlinear gain-scheduled control systems, *IEEE Transactions on Automatic Control* 35(8): 898–907.
- Spong, M. (1994). Swing up control of the acrobot, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2356–2361.
- Spong, M. (1995). The swing up control problem for the acrobot, *IEEE Control Systems Magazine* 15(1): 49–55.
- Spong, M. (1996). The control of underactuated mechanical systems, *Proceedings of the First international conference on mechatronics*, pp. 26–29.
- Spong, M. & Block, D. (1995). The Pendubot: a mechatronic system for control research and education, *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 1.
- Tsuchiya, K., Urakubo, T. & Tsujita, K. (1999). A motion control of a two-wheeled mobile robot, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, Vol. 5.
- Wang, X. & Chen, D. (2006). Output tracking control of a one-link flexible manipulator via causal inversion, *IEEE Transactions on Control Systems Technology* 14(1): 141–148.
- WJ, R. & Shamma, J. (2000). Research on gain scheduling, *Automatica* 36(10): 1401–1425.
- Yamakita, M. & Furuta, K. (1999). Toward robust state transfer control of titech double pendulum, *Proceedings of the Åström Symposium on Control*, pp. 73–269.
- Yamakita, M., Iwashiro, M., Sugahara, Y. & Furuta, K. (1995). Robust swing up control of double pendulum, *Proceedings of the American Control Conference*, Vol. 1.
- Yamakita, M., Nonaka, K. & Furuta, K. (1993). Swing up control of double pendulum, *Proceedings of the American Control Conference*, Vol. 3, pp. 2229–2229.
- Yoshida, K. (1999). Swing-up control of an inverted pendulum by energy-based methods, *Proceedings of the American control conference*, Vol. 6, pp. 4045–4047.
- Zhong, W. & Rock, H. (2001). Energy and passivity based control of the double inverted pendulum on a cart, *Proceedings of the 2001 IEEE international conference on control applications*, pp. 896–900.

Kinematics Synthesis of a New Generation of Rapid Linear Actuators for High Velocity Robotics with Improved Performance Based on Parallel Architecture

Luc Rolland

*Ecole Normale Supérieure des Arts et Métiers, Metz
France*

1. Introduction

This article studies several classes of linear actuators based on parallel topology featuring lower mobility.

Translation actuator design represents a very important issue in manipulator design in areas like machine tools for example and more recently hexapods. Actual designs are usually limited to low accelerations actually limited to 2 g. Moreover, alignment problems are difficult to circumvent and usually lead to non-uniform friction in the translation motion referred as hard spots. Despite important breakthroughs, linear motors are still limited to accelerations of 5g and they are plagued by problems such as surrounding magnetisation and limited torque. As for any parallel mechanisms, the proposed architectures do provide for a more rigid linkage. Their rigidity advantage leads to larger actuator bandwidth, thereby allowing for increased accelerations which result in larger forces being applicable to the extremity while keeping overall mass very low. The main disadvantage will be their transverse encumbrance which will be minimized through mechanism networking.

Two diamond and one rhombus configurations have been designed, analyzed, constructed and compared verifying their ability for very fast accelerations. Their kinematics are investigated allowing to write the forward and inverse problems for position, velocity and accelerations where closed-form solutions are determined. Motion limitations and singularity analysis are also provided from which configuration recommendations can be derived. These actuators will then be easily controllable despite their non-linear nature.

In parallel manipulators, the prismatic pairs are usually encountered as the linear actuators for several architectures such as the planar 3RPR, the general Gough platform, (Gough & Whitehall, 1962; Fichter, Kerr and Rees-Jones) and the Kanuk (Rolland, 1999) for examples.

These prismatic actuators, may they be guided or not, do play a very important role in robotics design and their performance has been an issue. According to the author's observation on several high speed milling projects, these actuators have been hampering the advent of high speed milling by being unable to provide for adequate accelerations in low inertia and high rigidity packages.

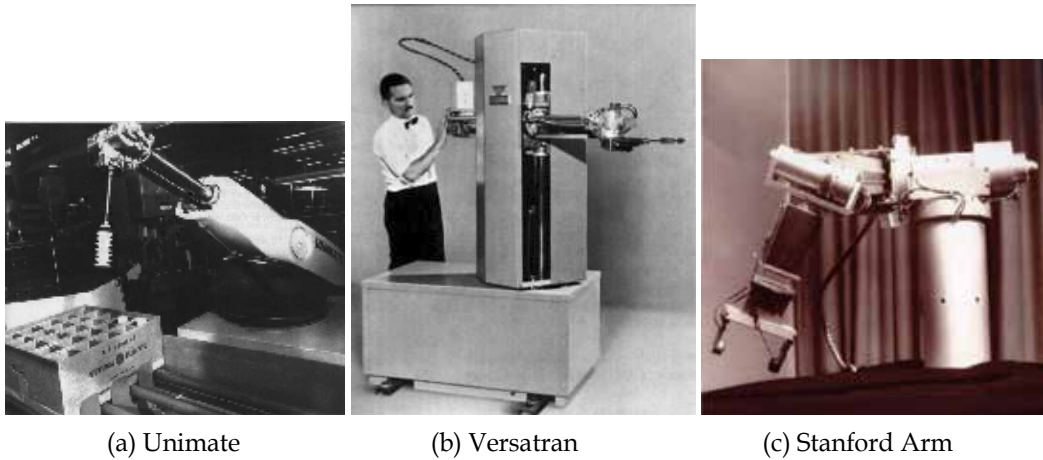


Fig. 1. Cylindrical robots

Typically, contemporary linear actuators have generally evolved in devices which can be classified under the nine following categories:

- Piston in a Cylinder or diaphragm being driven by a fluid
- Linear motors
- machine screw and nut
- A worm gear and screw
- Rack and pinion
- Belts and pulleys
- Cam and plunger
- Crank-slider
- Linkages

Lets consider Euclid's definition of a straight line: "A straight line is a line which lies evenly with the points on itself", (Euclid, 2002).

A straight line mechanism is defined as a mechanism that generates a straight-line output motion from an input actuator which rotates, oscillates, or moves in a straight line.

Inventing a straight line mechanism, referred as SLM hereafter, has been the concern of many researchers and engineers long before the industrial revolution. The use of linkages as SLM can be traced as far back as in the XIII century when sawmill drawings showed mechanisms for changing circular motion to straight-line motion. Even Da Vinci himself has drawn one mechanism to convert rotation to translation having slides acting as guides (DaVinci, 1493). Door locking mechanisms are other old examples where the rotation of the key was converted into translation motion of the lock element.

In 1603, Christopher Scheiner invented the pantograph, (Scheiner, 1631). It may be regarded as the first example of the four-bar linkage. The pantograph is a device for copying and enlarging drawings. Knowing that the actuator is located to one end, this device can be made to move on a straight-line providing that the input follows a straight-line, therefore becoming a pure amplification linkage.

In the late seventeenth century, it was extremely difficult to machine straight line and flat surfaces. Knowing that prismatic pair construction without backlash had become an important and difficult challenge, much effort was then diverted towards the coupler curve of a linkage comprising only revolute joints which were much easier to produce.



Fig. 2. Typical linear actuator

Later, James Watt proposed a four-bar mechanism generating motion approximating roughly a straight-line. We would have to wait until 1864 when Peaucellier introduced the first planar linkage capable of transforming rotary motion into exact straight-line motion. Until this invention, no planar mechanism existed for producing straight-line motion without reference guideways which could not be made very straight themselves.

It was soon followed by the grasshopper linkage which also provided for an exact straight-line. These mechanisms were essential in the development of steam engines and machine tools. Then, Hart's Linkage and A-frame both reduce the link number to only five.

The Kmoddl library from Cornell University presents 39 linkages imagined to produce linear motion which come from Franz Reuleaux Collection of Kinematic Mechanisms (Reuleaux, 1876; Moon, 2007). Most of them feature relatively complex architectures where linkages cannot easily be practically applied in systems such as robots or milling machines.

Several proposals were patented trying to simplify the linkage producing straight-line motion. In the class of nearly straight line linkages, one can identify several linkages by inventors such as Hoekens, Chebyshev, Evans, Roberts and Burmester. With appropriate linkage dimensions, part of the motion can be a straight line. Hoekens linkage can be considered a Cognate linkage of the Chebyshev linkage since it produces a similar motion pattern. These simpler designs always applied the properties of special points on one of the links of a four-bar linkage. They could often produce straight-lines over some limited range of their motion. The commonality of all these ingenious mechanisms is in the fact that they feature linkages based on closed loops or so-called parallel topology.

Very early on, the designers were faced with the fact that a prismatic pair or joint is much more difficult to build than a revolute joint. This is even more the case when trying to have a linear actuator, (Soylemez, 1999). This observation holds on even today.

Practically, parallel mechanism architectures have been able to provide solutions to industrial problems and needs with improved performance manipulators.

Theoretically, they may even improve accuracy but this is still an open problem at the moment, especially when control is concerned.

When the proposed configurations allow to bring the actuators fixed on or jointed to the base, the inertia of mobile elements can be significantly reduced so the extremity or end-effector can move at higher accelerations resulting in the deliverance of larger forces.

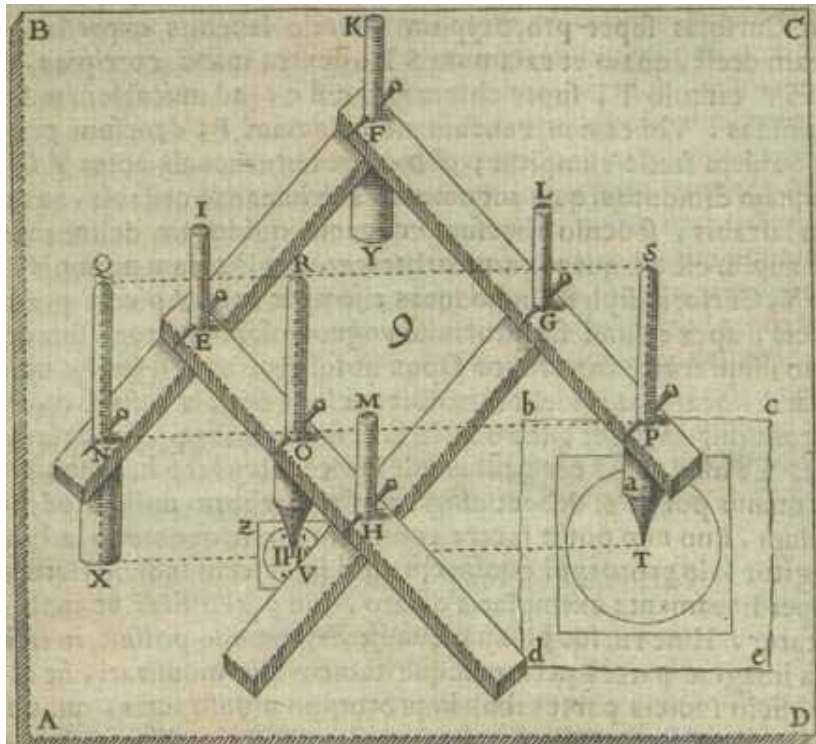


Fig. 3. Scheiner pantograph

The principal drawback which shall be studied is the construction of kinematics models characterized by non-linear equations where an implicit relationship is produced between the manipulator configuration parameters, actuator joint positions and the end-effector position and orientation.

The simplest forms of parallel manipulators are the ones producing one degree-of-freedom. Performance evaluation for these single DOF mechanisms includes the four following criterias:

- workspace
- singularity avoidance
- linkage ecumbrance
- linearity in motion transmission

In a one DOF problem, the workspace criteria is then reduced to a simple range with two extremum values to be determined: the minimal and maximal positions.

Moreover, the design of machine-tools based on parallel robots have been concerned by problems related to inherent difficulties with prismatic actuator designs which have hampered their successful implementation.

This paper original contribution is in the design of a new generation of linear actuators for increased performance where planar parallel linkages are applied. For example, if implemented to replace classical linear actuators on Gough platform or even planar 3RPR manipulators, they allow to bring the motors in positions directly jointed to the base.

The introduction is followed by chapter on kinematic topology synthesis consisting of a review of various kinds of mechanisms to provide straight-line motion where mobility is

analyzed. Chapter 3 is dedicated to the kinematics analysis of several promising alternatives based on the four-bar mechanism. Then, chapter four investigates the selected performance criterias. This paper closes on a design chapter where prototypes are shown with motion analysis in terms of position, velocity and acceleration.

2. Kinematics topology synthesis

Firstly, in this section, we shall make a review of some interesting planar mechanisms which can perform the specified set of functional requirements. In this case the tasks shall be to achieve straight-line motion.

2.1 Background study

We need two definitions related to degree-of-freedom.

The DOF of the space is defined as the number of independent parameters to define the position of a rigid body in that space, identified as λ .

The DOF of a kinematic pair is defined as the number of independent parameters that is required to determine the relative position of one rigid body with respect to the other connected rigid body through the kinematic pair.

The term mechanism is defined as a group of rigid bodies or links connected together to transmit force and motion.

Mobility and kinematics analyses are possible under some assumptions:

- Ideal mechanisms with rigid bodies reducing the mechanism motion to the geometric domain.
- Elastic deformations are neglected
- Joint clearance and backlash are insignificant

2.2 Functional requirements

Historically, the need for straight-line motion has resulted on linkages based on closed loops or so-called parallel topology. The idea is to convert rotation motion into translations or straight-line motions. It is usually considered that prismatic pairs are much harder to build than revolute joints, (Soylemez, 1999).

Prismatic actuators as well as slides have the following problems:

- the side reactions of prismatic pairs produce friction leading to wear
- these wears are uneven, non-uniform and unpredictable along the path of the slide since the flat surfaces in contact are not well defined due to construction imperfections.

Some mechanisms are designed to generate a straight-line output motion from an input element which rotates, oscillates or moves also in a straight line.

The kinematic pair DOF is defined as the number of independent parameters necessary to determine the relative position of one rigid body with respect to the other connected to the pair, (Soylemez, 1999).

The linkages are designed to generate motion in the plane and are then limited to three DOFs, therefore the only available joints are either with one or 2 DOFs only.

The actual problem is addressed from a robotics or even machine-tool point of view. It can be summarized by this question: how can you draw a straight line without a reference edge? Most robotics manipulators or machine tools are applying referenced linear motions with guiding rails and even now linear motors. In design of parallel manipulators such as 3RPR

or Gough platforms, the actuators have especially to generate straight lines without any guiding rails.

This question is not new and it actually comes from the title taken from the book written by Kempe, where he describes plane linkages which were designed to constrain mechanical linkages to move in a straight line (Kempe, 1877).

2.3 Mobility analysis of linkages

Here is the mobility formula that is applied for topology investigation, (Rolland, 1998):

$$m = \sum j_i - \lambda n \tag{1}$$

where $\sum j_i$ is the sum of all degree-of-freedom introduced by joints and $\lambda = 3$ is the available DOF of the planar space in which the actuator is evolving.

Finally, the number of closed loops in the system is n . This number can be multiplied and shall be a natural number $n \in \{1,2,3, \dots\}$

2.4 Four-bar mechanisms

If $n = 1$ and only revolute joints are selected, then the mechanisms can be selected in the large variety of four-bar mechanisms. These linkages feature one closed-loop or one mechanical circuit. According to Grashof’s law, the sum of the shortest and longest link cannot exceed the sum of the remaining two links if there is to be continuous relative motion between the links. Hence, they can be classified as four types as shown in figure 4.

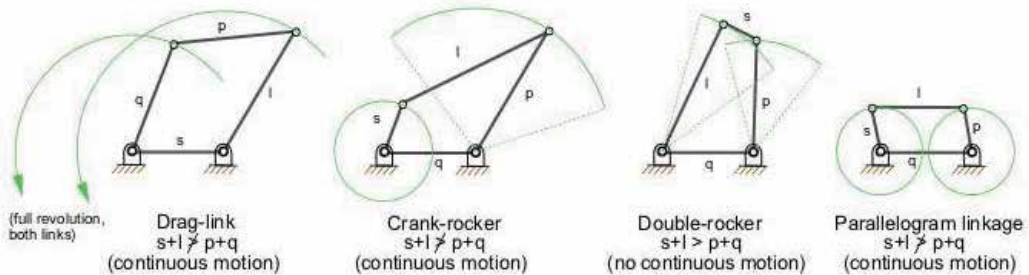


Fig. 4. Four-bar mechanism classification (from Wikipedia)

Three four-bar mechanisms can produce partial straight-line motion. They are characterized by two joints connected to the fixed base.

The Chebyshev linkage is the epitome of the four-bar mechanical linkage that converts rotational motion to approximate straight-line motion. It was invented by the 19th century mathematician Pafnuty Chebyshev. It is a four-bar linkage therefore it includes 4 revolute joints such that $\sum j_i = 4 * 1$ where $n = 1$ since there is only one closed loop. The resulting mobility: $m = 4-3 * 1=1$. Hoekens linkage happens to be a Cognate linkage of the Chebyshev linkage. It produces a similar motion pattern. With appropriate linkage dimensions, part of the motion can be an exact straight line.

Robert’s linkage can have the extremity P set at any distance providing it is layed out on that line perpendicular to the coupler, i-e link between A and B. This means that P can be positioned on top of the coupler curve instead of below.

This mobility calculation holds for any four-bar mechanism including the free ones, i-e not being attached to the base.

If properly designed and dimensionned, four-bar linkages can become straight-line motion generators as will be seen in the next section on kinematics. This is one of the contribution of this work.

2.5 True straight-line mechanisms

If $n > 1$ and only revolute joints are selected, then the mechanisms become more complex and will integrate two closed loops or two mechanical circuits.

Three mechanisms can produce exact straight-line motion: the Peaucelier linkage, the Grasshoper mechanism and a third one which has no name.

This linkage contains nine revolute joints such that $\Sigma j_i = 9 * 1 = 9$. Please note that where three links meet at one point, two revolute joints are effectively existing. Three closed loops can be counted for $n = 3$. The resulting mobility: $m = 9 - 3 * 3 = 0$. The linkage designed by Peaucelier is one of those mechanisms which cannot meet the mobility criterion but do provide the required mobility. Very recently, Gogu has reviewed the limitations of mobility analysis, (Gogu, 2004).

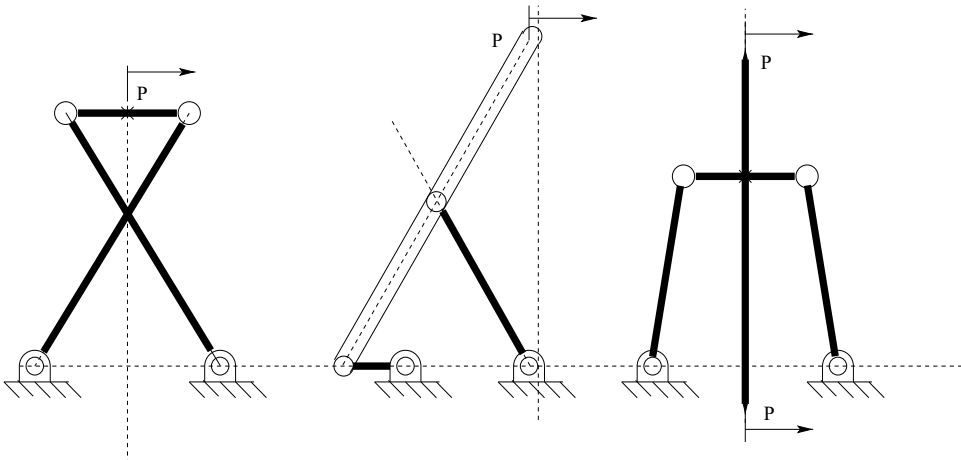


Fig. 5. Four-bar mechanisms: Chebyshev, Hoekens and Robert linkages

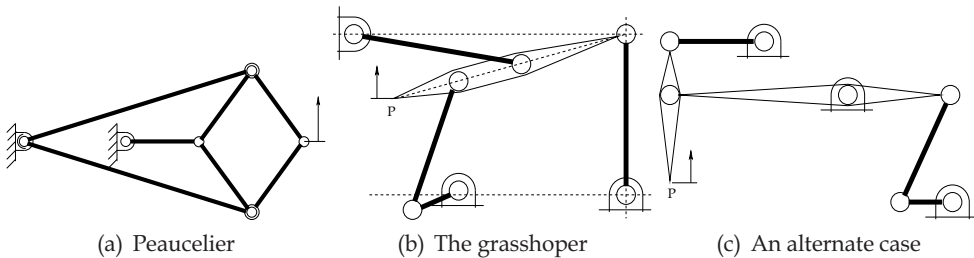


Fig. 6. Exact straight-line mechanisms

The two other linkages do provide for seven revolute joints for $\Sigma j_i = 7 * 1 = 7$ and two closed loops for $n = 2$. The resulting mobility: $m = 7 - 3 * 2 = 1$ which is verified by experiments.

These three mechanisms do provide for straight-line motion at the cost of complex linkages which do occupy very valuable space. This makes them less likely to be applied on robots.

3. Kinematics analysis

A mechanism is defined as a group of rigid bodies connected to each other by rigid kinematics pairs to transmit force and motion. (Soylemez, 1999).

Kinematics synthesis is defined as the design of a mechanism to yield a predetermined set of motion with specific characteristics.

We shall favor dimensional synthesis of function generation implementing an analytical method. The function is simply a linear function describing a straight-line positioned parallel to one reference frame axis.

The method will implement a loop-closure equation particularly expressed for the general four bar linkage at first. The first step consists in establishing the fixed base coordinate system.

3.1 Four-bar mechanism

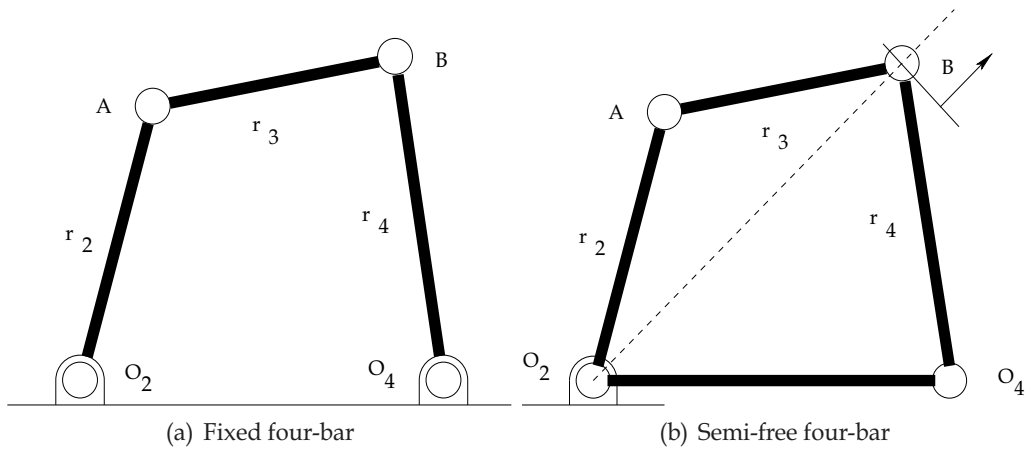


Fig. 7. General four-bar linkages

Lets define the position vectors and write the vector equation. Taking O_2 and O_4 as the link connecting points to the fixed base located at the revolute joint center, taking A and B as the remainder mobile revolute joint centers, the general vectorial formulation is the following, (Uicker, Pennock and Shigley, 2003):

$$(r_1 + r_2 + r_3 + r_4 = 0) \quad (2)$$

This last equation is rewritten using the complex algebra formulation which is available in the textbooks, (Uicker, Pennock and Shigley):

$$r_1 e^{j\theta_1} + r_2 e^{j\theta_2} + r_3 e^{j\theta_3} - r_4 e^{j\theta_4} = 0 \quad (3)$$

where θ_1 , θ_2 , θ_3 and θ_4 are respectively the fixed base, crank, coupler and follower angles respective to the horizontal X axis.

If we set the x axis to be colinear with O_2O_4 , if we wish to isolate point B under study, then the equation system becomes:

$$r_3 e^{j\theta_3} = r_4 e^{j\theta_4} - r_1 - r_2 e^{j\theta_2} \quad (4)$$

Complex algebra contains two parts directly related to 2D geometry. We project to the x and y coordinate axes, in order to obtain the two algebraic equations. The real part corresponds to the X coordinates and the imaginary part to the Y coordinates. Thus, the equation system can be converted into two distinct equations in trigonometric format.

For the real or horizontal part:

$$r_3 \cos(\theta_3) = r_4 \cos(\theta_4) - r_1 - r_2 \cos(\theta_2) \quad (5)$$

For the imaginary or vertical part:

$$r_3 \sin(\theta_3) = r_4 \sin(\theta_4) - r_2 \sin(\theta_2) \quad (6)$$

When O_2O_4 is made colinear with the X axis, as far as r_1 is concerned, there remains only one real part leading to some useful simplification.

The general four bar linkage can be configured in floating format where the O_4 joint is detached from the fixed base, leaving one joint attached through a pivot connected to the base. Then, a relative moving reference frame can be attached on O_2 and pointing towards O_4 . This change results in the same kinematic equations.

Since, the same equation holds and we can solve the system:

$$(\theta_4) = 2 \arctan \left(\frac{B + \sqrt{A^2 + B^2 - C^2}}{A + C} \right) \quad (7)$$

where the A,B,C parameters are:

$$\begin{aligned} A &= \frac{r_1}{r_2} - \cos(\theta_2) \\ B &= -\sin(\theta_2) \\ C &= \frac{r_1 \cos(\theta_2)}{r_4} - \frac{1}{2} \frac{r_1^2 + r_2^2 - r_3^2 + r_4^2}{r_2 r_4} \end{aligned} \quad (8)$$

To determine the position of joint center B in terms of the relative reference frame O:

$$O_2B = [r_1 + r_4 \cos(\theta_4), r_4 \sin(\theta_4)]^t \quad (9)$$

Then, the norm of the vector OB gives the distance between O and B:

$$x = |O_2B| = \sqrt{(r_1 + r_4 \cos(\theta_4))^2 + r_4^2 (\sin(\theta_4))^2} \quad (10)$$

This explicit equation gives the solution to the forward kinematics problem. An expression spanning several lines if expanded and which cannot be shown here when the expression of θ_4 , equation 7, is substituted in it. This last equation gives the distance between O and B, the output of the system in relation to the angle θ_2 , the input of the system as produced by the rotary motor. The problem can be defined as: **Given the angle θ_2 , calculate the distance x between O and B.**

The four-bar can be referred as one of the simplest parallel manipulator forms, featuring one DOF in the planar space ($\lambda = 3$). One family of the lowest mobility parallel mechanisms. The important issue is the one of the path obtained by point B which is described by a coupler curve not being a straight line in the four-bar general case. However, in the floating case, if applied as an actuator, the general four-bar can be made to react like a linear actuator. The drawbacks are in its complex algebraic formulation and non-regular shape making it prone for collisions.

3.2 Specific four bar linkages

We have two questions if we want to apply them as linear actuators:

- Can we have the four-bar linkage to be made to move in a straight-line between point O_2 , the input, where the motor is located and B, the output, where the extremity or end-effector is positioned?
- Can simplification of resulting equations lead to their inversions?

As we have seen earlier, specific four bar linkages can be made to produce straight-line paths if they use appropriate dimensions and their coupler curves are considered on link extensions. In this case, we still wish to study the motion of B with the link lengths made equal in specific formats to produce specific shapes with interesting properties. Three solutions can be derived:

- the parallelogram configuration,
- the rhombus configuration,
- the kite or diamond shape configuration, (Kempe, 1877).

3.2.1 The parallelogram configuration

Parallelograms are characterized by their opposite sides of equal lengths and they can have any angle. They even include the rectangle when angles are set to 90 degrees. They have been applied for motion transmission in the CaPaMan robot, (Ceccarelli, 1997).

The parallelogram four-bars are characterized by one long and one short link length. They can be configured into two different formats as shown in figure 8.

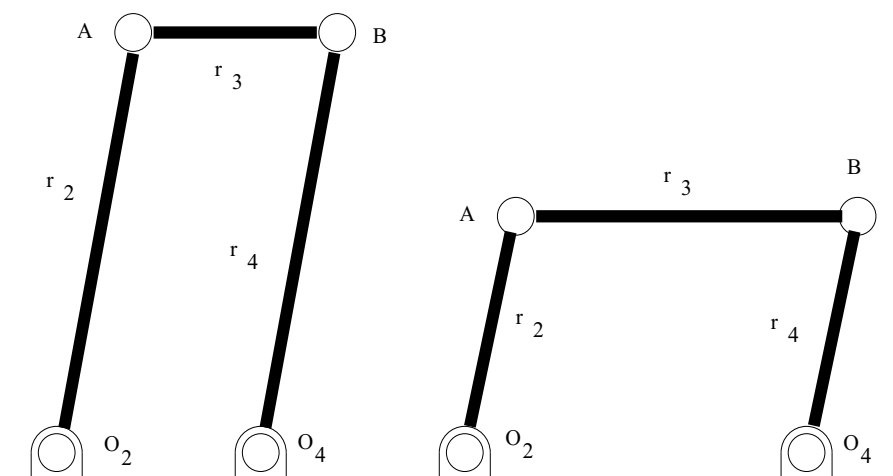


Fig. 8. The two parallelogram four-bar cases

The follower follows exactly the crank. This results in the equivalence of the input and following angles: $\theta_4 = \theta_2$.

If we set R and r as the link lengths respectively, then to determine the position of joint center B in terms of the relative reference frame O_2 ; an simple expression is derived from the general four-bar one:

$$O_2B = [R + r \cos(\theta), r \sin(\theta)] \quad (11)$$

Then, the norm of the vector O_2B gives the distance between O_2 and B :

$$x = |O_2B| = \sqrt{R^2 + r^2 + 2Rr \cos(\theta)} \quad (12)$$

This last equation is the result of the forward kinematics problem.

Isolation of the θ variable will lead to the inverse kinematics problem formulation:

$$\theta = \arccos\left(\frac{x^2 - R^2 - r^2}{2Rr}\right) \quad (13)$$

Detaching joint O_4 from the fixed base, the parallelogram becomes a semi-free linkage which can be considered as one prismatic actuator.

3.2.2 The rhombus configuration

The rhombus configuration can be considered a special case of the parallelogram one. All sides of a Rhombus are congruent and they can have any angle. Therefore, $r_1 = r_2 = r_3 = r_4$ or even one can write $r = R$ as for the parallelogram parameters. The mechanism configuration even includes the square when angles are set to 90 degrees.

The forward kinematics problem becomes:

$$x = 2r \cos\left(\frac{\theta}{2}\right) \quad (14)$$

The Inverse kinematics problem is expressed as:

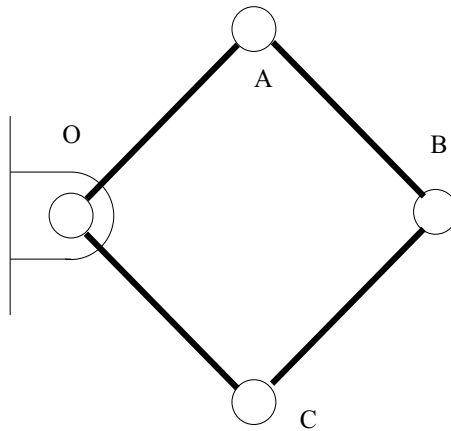


Fig. 9. The Rhombus detailed configuration

$$\theta = 2\arccos\left(\frac{x}{2r}\right) \quad (15)$$

Simple derivation will lead to differential kinematics.

The forward differential kinematics is expressed by the following equation:

$$v = -r\omega\sin\left(\frac{\theta}{2}\right) \quad (16)$$

where $\omega = \frac{d\theta}{dt}$

We take the following geometric property:

$$\cos\left(\frac{\theta}{2}\right) = \frac{x}{2} \quad (17)$$

We apply Pythagore's theorem:

$$\sin\left(\frac{\theta}{2}\right) = \frac{1}{2}\sqrt{4 - \frac{x^2}{r^2}} \quad (18)$$

Then, the FDP can be rewritten in terms of the length x:

$$v = -\omega\frac{r}{2}\sqrt{4 - \frac{x^2}{r^2}} \quad (19)$$

Inversion of equation 19 lead to the inverse differential kinematics problem being expressed as:

$$\omega = -\frac{v}{r\sin\left(\frac{\theta}{2}\right)} \quad (20)$$

Substituting equation 17 and equation 18 into the former, we obtain:

$$\omega = -\frac{v}{r}\sqrt{1 - \frac{x}{2r}} \quad (21)$$

Further derivation will give the extremity acceleration where the FDDP can be expressed as:

$$a = -r\alpha\sin\left(\frac{\theta}{2}\right) - \frac{1}{2}r\omega^2\cos\left(\frac{\theta}{2}\right) \quad (22)$$

Substituting equation 17 into the former lead to the following expression of the FDDP:

$$a = -\frac{1}{2}r\alpha\sqrt{4 - \frac{x^2}{r^2}} - \frac{1}{4}\omega^2x \quad (23)$$

The IDDP:

$$\alpha = \frac{-a - \frac{1}{2}r\omega^2 \cos\left(\frac{\theta}{2}\right)}{r \sin\left(\frac{\theta}{2}\right)} \quad (24)$$

Substituting equation 21, equation 17 and equation 18 into the former, we obtain:

$$\alpha = -2ar^{-1} \frac{1}{\sqrt{4 - \frac{x^2}{r^2}}} - \frac{1}{2}vx\sqrt{2r^{-4}}\left(\frac{x}{r}\right)^{-3/2} \quad (25)$$

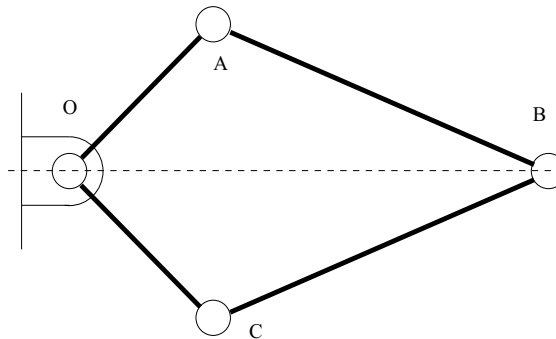


Fig. 10. The diamond shape four-bar

3.2.3 The kite or diamond shape configuration

The kite configuration is characterized by two pairs of adjacent sides of equal lengths, namely R and r.

Then, two configurations into space depending on which joint the motor is attached. The motor is also located on the joint attached on the fixed base.

To obtain the first configuration, the first pair is located at O_2 , the crank joint center where the motor is located, as its articulation center and the second pair at B, the extremity joint, as its center.

The second configuration integrates the actuator on O_4 . However, the actuator x output is defined as the linear distance between O_2 and B making this actuator moving sideways. The problem will be that the change of four-bar width is going to introduce parasitic transverse motion which will in turn prevent real linear motion due to the pivot effect caused by the motor joint. This approach is thus rejected.

To obtain the second disposition, one can mount the driven joint between two unequal links and have the output on the opposite joint also mounted between two unequal links. This results in sideways motion. However, this would also result in parasitic transverse motion which would mean that the final motion would not be linear being their combination. Therefore, this last configuration will not be retained further.

Lets R be the longest link length, the links next to B, and r be the smallest link one, the links next to O_2 .

Since this configuration is symmetric around the axis going through O_2 and B, it is thus possible to solve the problem geometrically by cutting the quadrilateral shape into two mirror triangles where the Pythagorean theorem will be applied to determine the distance between O_2 and B giving:

$$x = r\sqrt{1 - \left(\sin\left(\frac{\theta}{2}\right)\right)^2} + R\sqrt{1 - \frac{r\left(\sin\left(\frac{\theta}{2}\right)\right)^2}{R}} \quad (26)$$

This equation expresses then the forward kinematics problem.

Using the law of cosines on the general triangle where the longest side is that line between O_2 and B , it is possible to write a more compact version for the FKP:

$$x = \sqrt{R^2 - r^2 - 2r\cos\left(\frac{1}{2}\theta\right)} \quad (27)$$

The inverse kinematics problem requires the distance or position x as input which completes the two triangle lengths into the diamond shape. Hence, the cosine laws on general triangles can be applied to solve the IKP:

$$\theta = 2\arccos\left(\frac{1}{2} \frac{R^2 - r^2 - x^2}{r}\right) \quad (28)$$

To obtain the differential kinematics models, the kinematics models are differentiated. FDP:

$$v = \frac{1}{2} \frac{r^2 \sin\left(\frac{1}{2}\theta\right)\omega}{\sqrt{R^2 - r^2 - 2r\cos\left(\frac{1}{2}\theta\right)}} \quad (29)$$

Differentiation of the IKP leads to the following IDP expression:

$$\omega = v^* \frac{4\sqrt{R^2 - r^2 - 2r\cos\left(\frac{1}{2}\theta\right)}}{r\sqrt{4 - 4\cos\left(\frac{1}{2}\theta\right)^2}} \quad (30)$$

After testing several approach for obtaining the differential model leading to accelerations, it was observed that starting with the inverse problem leads to more compact expressions: The IDDP is obtained by differentiating the IDP:

$$\alpha = a(A1 + A2) \text{ where } A1 = \frac{4}{r\sqrt{4 - \frac{(R^2 - r^2 - x^2)^2}{r^2}}}, \quad A2 = -\frac{8x^2(R^2 - r^2 - x^2)}{r^3\left(4 - \frac{(R^2 - r^2 - x^2)^2}{r^2}\right)^{3/2}} \quad (31)$$

Inverting the IDDP produces the FDDP but it cannot be shown in the most compact form.

The Kite configuration models are definitely more elaborate and complex than for the rhombus configuration without necessarily leading to any kinematics advantages.

3.2.4 The rhombus configuration repetition or networking

The rhombus four-bar linkage can be multiplied as it can be seen in platform lifting devices. The repetition of these identical linkages helps reduce the encumbrance and this will be studied in this section in the context of linear actuator design.

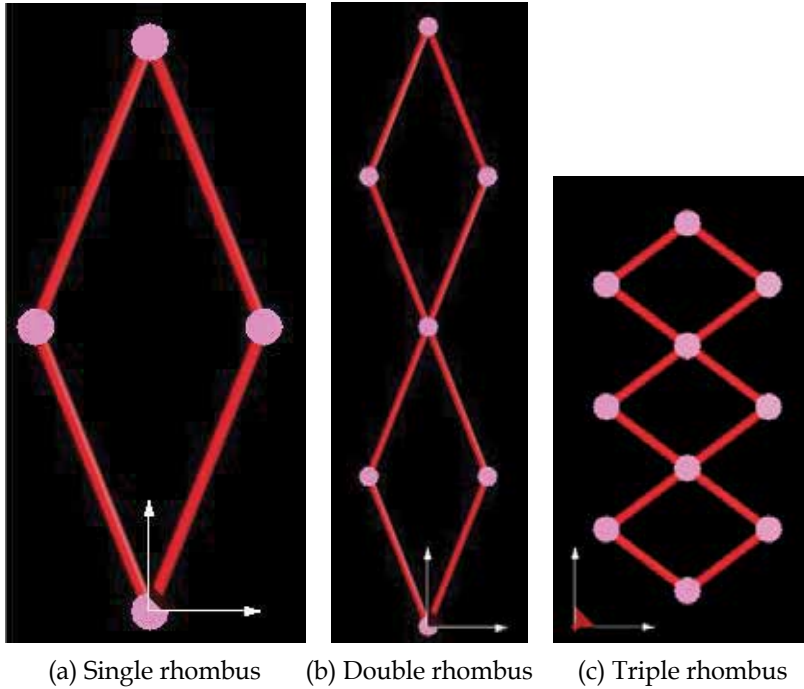


Fig. 11. Rhombus networking

The distance traveled by the first moving central joint (FKP) is:

$$x_1 = 2r \cos\left(\frac{\theta}{2}\right) \quad (32)$$

This problem can be solved just like solving the original single rhombus FKP.

The distance traveled by the second moving central joint (FKP) is:

$$x_2 = 2x_1 = 4r \cos\left(\frac{\theta}{2}\right) \quad (33)$$

The impact of adding the second rhombus is doubling the distance or position reach.

The distance traveled by the third moving central joint or the solution of the FKP of a triple rhombus is:

$$x_3 = 3x_1 = 6r \cos\left(\frac{\theta}{2}\right) \quad (34)$$

This trend can be generalized to a repetition of n identical rhombuses.

$$x_n = nx_1 = n2r \cos\left(\frac{\theta}{2}\right) \quad (35)$$

The result of the four-bar rhombus repetition is the linear motion amplification by that repetition number.

To obtain the inverse kinematics problem, one can proceed with inversion of the FKP.

The double rhombus angular position of the actuator can then be deduced:

$$\theta = 2\arccos\left(\frac{1}{4} \frac{x}{r}\right) \quad (36)$$

This equation can then also be extrapolated to a repetition of n identical rhombuses.

$$\theta = 2\arccos\left(\frac{1}{2n} \frac{x}{r}\right) \quad (37)$$

The forward differential model is obtained by derivation of the forward kinematics model. For a double rhombus configuration, the relative speed of the second central joint is equal to the absolute speed of the first central joint:

$$v_{2/r} = v_1 \quad (38)$$

$$v_2 = v_1 + v_{2/r} \quad (39)$$

$$v_2 = 2v_1 \quad (40)$$

where

$$v_1 = -r\omega\sin\left(\frac{\theta}{2}\right) \quad (41)$$

Hence, the actual speed of the second extremity or the final end-effector becomes:

$$v_2 = -2r\omega\sin\left(\frac{\theta}{2}\right) \quad (42)$$

The impact of adding the second rhombus is doubling the end-effector velocity.

The same result would be obtained by derivation of the equation for x_2 .

We now calculate the velocity of the third moving central joint which corresponds to the solution of the FDP of a triple rhombus.

$$v_3 = 3v_1 = -3r\omega\sin\left(\frac{\theta}{2}\right) \quad (43)$$

This trend can be generalized to a repetition of n identical rhombuses:

$$v_n = nv_1 = -nr\omega\sin\left(\frac{\theta}{2}\right) \quad (44)$$

The inverse differential model can be obtained in two ways, either by derivation of the inverse kinematics model or inversion of the forward differential model.

By inversion of the FDP, the double rhombus angular position of the actuator can then be deduced:

$$\omega = vr^{-1} \frac{1}{\sqrt{4 - \frac{x^2}{r^2}}} \quad (45)$$

For the triple rhombus, we extrapolate:

$$\omega = v \frac{2}{3r} \frac{1}{\sqrt{4 - \frac{x^2}{r^2}}} \quad (46)$$

For a linkage with the repetition of n rhombuses, we obtain the following equation:

$$\omega = v \frac{2}{nr} \frac{1}{\sqrt{4 - \frac{x^2}{r^2}}} \quad (47)$$

To determine the accelerations, we will again differentiate the former differential models. We calculate derivation of the equation for v_2 for the second rhombus; it results in doubling the end-effector acceleration.

The FDDP for the case where we are doubling the rhombus leads to:

$$a_2 = 2a_1 = -2r\alpha \sin\left(\frac{\theta}{2}\right) - \frac{1}{2}r\omega^2 \cos\left(\frac{\theta}{2}\right) \quad (48)$$

For the triple rhombus, we can determine that:

$$a_3 = 3a_1 = -3r\alpha \sin\left(\frac{\theta}{2}\right) - \frac{1}{2}r\omega^2 \cos\left(\frac{\theta}{2}\right) \quad (49)$$

For n rhombuses, it is possible to extrapolate:

$$a_n = na_1 = -nr\alpha \sin\left(\frac{\theta}{2}\right) - \frac{1}{2}r\omega^2 \cos\left(\frac{\theta}{2}\right) \quad (50)$$

Multiplying n times the rhombus linkage results in multiplying the acceleration likewise.

The IDDP, inverse model for a double rhombus, through derivation of the IDP or inversion of the FDDP, the calculation returns this equation:

$$\alpha = 2ar^{-1} \frac{1}{\sqrt{16 - \frac{x^2}{r^2}}} - \frac{1}{8}v^2xr^{-4} \left(1 - \frac{1}{16} \frac{x^2}{r^2}\right)^{-\frac{3}{2}} \quad (51)$$

For three rhombuses, the angular acceleration can then be determined:

$$\alpha = 2ar^{-1} \frac{1}{\sqrt{36 - \frac{x^2}{r^2}}} - \frac{1}{2}7v^2xr^{-4} \left(1 - \frac{1}{36} \frac{x^2}{r^2}\right)^{-\frac{3}{2}} \quad (52)$$

We have then extrapolated for a linear actuator constructed with n rhombuses:

$$\alpha = 2 \frac{a}{nr} \frac{1}{\sqrt{4 - \frac{x^2}{n^2r^2}}} - v^2xn^{-3}r^{-4} \left(1 - \frac{1}{4} \frac{x^2}{n^2r^2}\right)^{-\frac{3}{2}} \quad (53)$$

3.2.5 The kite configuration repetition or networking

There seems to be no advantage to gain from networking the kite configuration. This will even add complexity to the kinematics models. Therefore, this prospect has not been investigated further.

4. Kinematics performance

4.1 Singularity analysis

4.1.1 General four bar linkage

For the general four bar linkage, singularities can be found when $A + C = 0$ using the values of equation 8. The solution to this equation results in:

$$\theta_2 = \arccos\left(\frac{1 - 2r_1 r_4 + r_1^2 + r_2^2 - r_3^2 + r_4^2}{2 r_2 (-r_4 + r_1)}\right) \quad (54)$$

4.1.2 The parallelogram configuration

Singularities could be found only when $Rr = 0$ which is impossible since all links obviously have lengths larger than zero.

From the kinematics point of view, no limitations apply on the application of parallelograms since the rocker can follow the crank in any position allowing full rotation capability, therefore having no kinematics singularity whatsoever.

This mechanism could be considered somewhat similar or equivalent to the belt and pulley one where the two pulleys are of equal lengths if the belt is considered without elasticity.

4.1.3 The rhombus configuration

For the IDP, singularities exist and they can be determined by cancelling the denominator in the equations 20 and 21 leading to the two following equations.

The first one calculates the singularity in terms of the input angle θ :

$$\sin\left(\frac{\theta}{2}\right) = 0 \quad (55)$$

Hence, we find a singularity at $\theta = 0$ and its counterpart $\theta = 360$ degrees.

For the second one determines the singularity in terms of the extremity position x :

$$\sqrt{4 - \frac{x^2}{r^2}} = 0 \quad (56)$$

Hence, the singular position $x = 2r$ corresponds to the same posture as $\theta = 0$.

From a geometric point of view, links have no material existence (no mass) and they can occupy the same position in space. In reality, the masses do not allow such cases and therefore the singularity will be alleviated by bar width as will be explained later in the design section. The IDDP models bring singularities. Observation of the denominator allows us to determine that the singular configurations are just the same as the one studied for the IDP since the equations feature the same denominators under the power.

4.1.4 The kite or diamond shape configuration

If $R > r$, then this results then into an amplified motion without any singularity with full 360 degrees rotation of the input crank. This configuration has an advantage over the other

types of four-bars. This would surely represent one reason to apply this mechanism as a linear actuator.

If $R < r$, then the mechanism cannot reach an input angle of 180 degrees since this would mean $2R > 2r$ in contradiction with stated configurations. Hence, the system will block into position $\theta_{max} < 180^\circ$ unable to go further. The angular range will be limited to $[0, \theta_{max}]$ where:

$$\theta_{max} = 2\arcsin\left(\frac{R}{r}\right) \quad (57)$$

This posture also yield a singularity which can also enforce mechanism blockage. Hence, this type will not be retained.

4.1.5 The rhombus configuration repetition or networking

In terms of singularities, finding the roots of the FDP and IDP will lead to the same singularities as for a single rhombus as it would seem logical. In terms of singularities, finding the roots of the FDDP and IDDP is equivalent to finding the same singularities solving the roots of only the IDDP as for a single rhombus.

Therefore, networking rhombuses will not introduce any singularity.

4.2 Workspace

The second important performance criterion for robotic design is usually the workspace. In the case of single DOF device, this narrows down to a simple range which we wish to maximize.

4.2.1 The general four-bar linkage

The mechanism can reach the following maximum length where two links are aligned, either r_1 and r_4 or r_2 and r_3 . Then, the mechanism reach will be x_{max} and is calculated by the length of the extension of the two shortest links going from O_2 and leading to the extremity B:

$$x_{max} = \min(r_1 + r_4, r_2 + r_3) \quad (58)$$

The mechanism can also reach a minimum length which is a far more difficult problem to determine depending upon the configuration and relative link lengths. This is where Graschoff's formulas could help solve this problem. Despite the fact that link lengths value could be found leading to a coupler curve being a straight line, this constitutes another reason to avoid the general four-bar mechanisms.

4.2.2 The parallelogram configuration

The maximum and minimum actuator values of x can be determined by looking for the roots of the $x(\theta)$ function derivative or by geometric reasoning. Hence, using the simplest, i-e the second approach, we can determine that the extremas are found at $\theta = k\pi$ where $k \in \{0, 1, 2, 3, \dots\}$. With $n = 0$, the maximum value is found $x_{max} = R + r$ and with $n = 1$, the minimum value is $x_{min} = |R - r|$. We do not need to go further because of the repetitive nature of the trigonometric signal. These correspond to the posture where the four-bar is folded on itself: one fold to the left and one to the right.

4.2.3 The rhombus configuration

To determine the maximum and minimum values, several methods lead to the same results.

Taking the FKP, equation 14, the maximum value is obtained when $\cos(\frac{\theta}{2}) = 1$ and the minimum value will be when $\cos(\frac{\theta}{2}) = -1$. Hence, $x_{max} = 2r$ and the related angle is then $\theta = 0$. Moreover, $x_{min} = -2r$ and the related angle is then $\theta = -2\pi$. These values imply that the pure geometric nature of the kinematics analysis allows the mechanism to reverse by going unto itself. Hence, the minimum can be seen on the left or negative side of the reference frame and the maximum is located on the right or positive side.

With considerations of the linkage dimensions, the geometric analysis can be augmented by taking into account the linkage width.

Firstly, two linkages cannot occupy the same space, therefore, the rhombus linkage configuration will have a pair of opposite linkages below and one pair above. This lead to physical constraints equations. This property can also be translated into geometric information.

These opposite links are parallel pairs which will eventually touch each other alongside at two mechanism rotations. These postures could be considered as folded ones. The first corresponds to the minimum rotation and the second to the maximum rotation.

Let the rhombus linkage be constructed by four bars of identical width w .

The minimal rotation angle increases to:

$$\theta_{min} > 0 \tag{59}$$

$$\theta_{min} = 2\arcsin\left(\frac{w}{r}\right) \tag{60}$$

Taking into account that the kinematics chain cannot reverse by going unto itself, the maximal rotation angle reduces to:

$$\theta_{max} < \pi \tag{61}$$

$$\theta_{max} = 2\arccos\left(\frac{w}{r}\right) \tag{62}$$

The final range of the linear actuator is then the interval determined by: $[2 \arcsin(\frac{w}{r}), 2 \arccos(\frac{w}{r})]$.

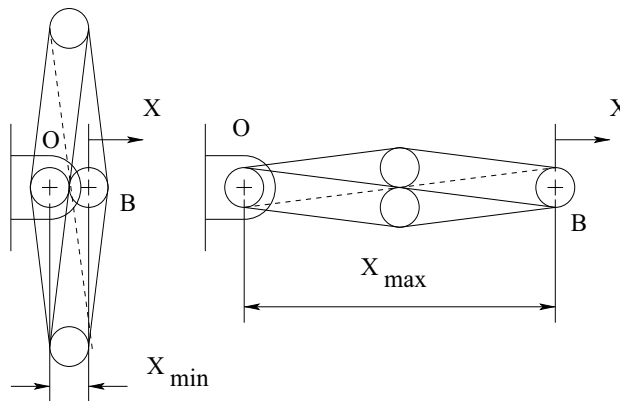


Fig. 12. The rhombus extreme positions

The extreme positions can be determined from these values. Minimal length is going to occur at maximal angular displacement.

$$x_{min} = 2r \cos\left(\frac{\theta}{2_{max}}\right) \quad (63)$$

Substituting θ_{max} into the equation, we get:

$$x_{min} = 2r \cos\left(2\arccos\left(\frac{w}{r}\right)\right) \quad (64)$$

Similarly, maximal length is going to occur at minimal angular displacement.

$$x_{max} = 2r \cos\left(\frac{\theta}{2_{min}}\right) \quad (65)$$

Substituting θ_{min} into the equation, we get:

$$x_{max} = 2r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (66)$$

4.2.4 The kite or diamond configuration

Let the kite linkage be constructed by four bars of identical width w . The minimal rotation angle is exactly the same as the rhombus:

$$\theta_{min} > 0 \quad (67)$$

$$\theta_{min} = 2\arcsin\left(\frac{w}{r}\right) \quad (68)$$

Taking into account that the kinematics chain cannot reverse by going unto itself, the maximal rotation angle reduces to the case where the long bars touch each other in the negative sense of the reference frame. We have to take then the angle outside the triangle formed by these long bars:

$$\theta_{max} < 2\pi \quad (69)$$

$$\theta_{max} = 2\pi - 2\arcsin\left(\frac{w}{R}\right) \quad (70)$$

The extreme positions can be determined from these values. In the case of maximal position, the two geometric triangles formed by the long and short links add up: Minimal length is going to occur at maximal angular displacement.

$$x_{min} = \sqrt{R^2 - \left(\frac{w}{2}\right)^2} - r \cos\left(\frac{\theta}{2_{max}}\right) \quad (71)$$

Maximal length is going to occur at minimal angular displacement.

$$x_{max} = \sqrt{R^2 - \left(\frac{w}{2}\right)^2} + r \cos\left(\frac{\theta}{2_{max}}\right) \quad (72)$$

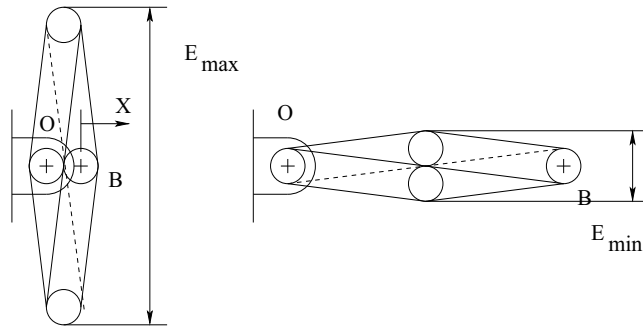


Fig. 13. The rhombus encumbrance

4.3 Encumbrance

4.3.1 The rhombus configuration

The proposed linear actuators are based on four-bar linkage where encumbrance becomes an issue considering that the mechanism spread sideways making them subject to collisions if other actuators would be located in the vicinity such as it is often the case with parallel manipulators.

Encumbrance is defined as the distance from one side of the mechanism to the other side taking into account the linkage width.

A Rhombus would have minimum encumbrance of $E_{min} = 2w$ when the angle is at θ_{min} . This characteristic is relatively unimportant compared to the maximum encumbrance occurring at the maximum input angle posture θ_{max} :

$$E_{max} = 2\sqrt{r^2 - w^2} + w \quad (73)$$

4.3.2 The kite configuration

A folded kite would have minimum encumbrance of $E_{min} = 2w$ when the angle is at θ_{min} just like the rhombus. The maximum encumbrance is occurring at the maximum input angle posture $\theta_{max} = \frac{\pi}{2}$ when the smaller links are aligned:

$$E_{max} = 2r + w \quad (74)$$

As can be observed, the kite encumbrance only really depends on the dimension of the shortest links.

4.4 The repeated rhombus configuration

The problem of encumbrance justifies the design of a mechanism based on the repetition of identical rhombuses as it is done for lifting platforms.

The repetition of the four-bar rhombuses is not affecting the rotation input and the θ_{min} and θ_{max} values are only related to the first rhombus, therefore these extrema are unchanged.

4.4.1 The double rhombus

For the double rhombus, the minimum position is determined by:

$$x_{min} = 4r \cos\left(2\arccos\left(\frac{w}{r}\right)\right) \quad (75)$$

And the maximum position is calculated using:

$$x_{max} = 4r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (76)$$

4.4.2 The triple rhombus

For the triple rhombus, the minimum position is determined by:

$$x_{max} = 6r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (77)$$

And the maximum position is calculated using:

$$x_{max} = 6r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (78)$$

4.4.3 The multiple rhombus

For the generalized case with n rhombuses, the minimum position is determined by:

$$x_{max} = n2r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (79)$$

And the maximum position is calculated using:

$$x_{max} = n2r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) \quad (80)$$

4.4.4 Encumbrance of the multiple rhombus

The networking of rhombuses is not affecting encumbrance in the sense that the values are exactly the same. However, the main advantage is that the reach which can be defined as the maximum position is increasing while the encumbrance remains unchanged. This could not happen with a simple rhombus where we would need to increase the link lengths in order to increase reach resulting in larger encumbrance.

Lets define the encumbrance ratio.

The encumbrance ratio is defined as the ratio of reach divided by the transverse encumbrance perpendicular to the axis of motion.

$$e = \frac{X_{max}}{E_{max}} \quad (81)$$

For the repeated rhombus, this occurs when $\theta = \theta_{min}$ and the encumbrance ratio becomes:

$$e = n2r \cos\left(2\arcsin\left(\frac{w}{r}\right)\right) 2\sqrt{r^2 - w^2} + w \quad (82)$$

Hence, the motion to encumbrance ratio is increasing proportionally with the rhombus repetition.

5. Design examples

5.1 Initial prototypes

A first group of prototypes were constructed and tested using a Meccano set while the author was working at the Ecole Nationale des Arts et Metiers in Metz. This resulted in the construction of a planar parallel manipulator as seen in figure 14. The DC motor was the typical Meccano 36 VDC.

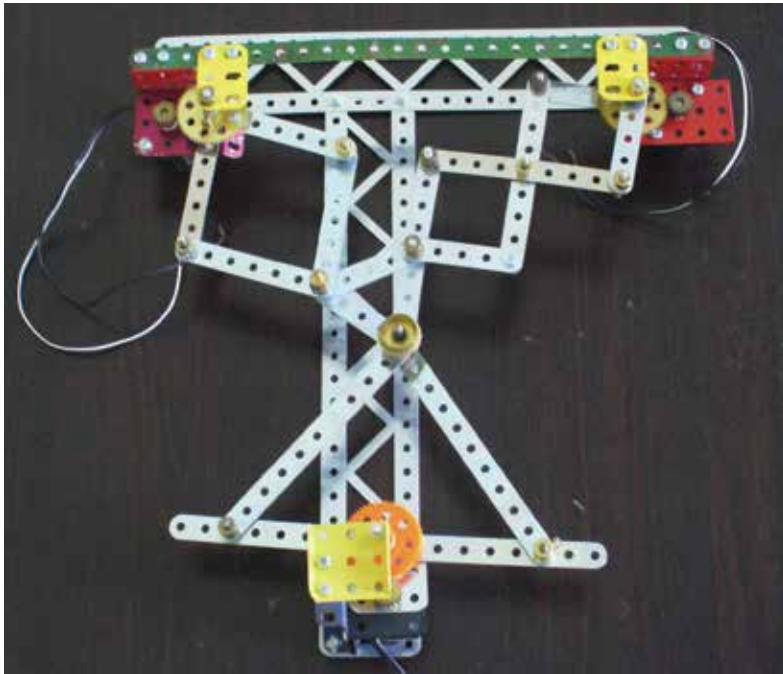


Fig. 14. Planar parallel manipulator with four-bar actuators

They were sufficient to prove and validate the concept. In effect, one rhombus four-bar with a Meccano motor could not be seen moving due to very high accelerations.

5.2 Actual prototypes

This work was then completed during the author stay at Middle East Technical University, Northern Cyprus Campus.

Three typical linear actuators were constructed as seen in figure 15. One comprising one rhombus, one with two rhombuses and one with a kite configuration using the same links and motors whenever possible.

5.2.1 Configuration

Here are the mechanism geometric parameters. They were constructed with two standardized bars. The short bars have length $r = 10$ cm and width $w = 3$ cm. The long bars have length $R = 20$ cm with same width.

The geared electrical motors were selected to provide maximum rotation speed of 120 RPM. Hence, $\omega_{max} = 4\pi$ radians/s.

5.2.2 Extreme positions

For the rhombus, from the proposed equations, the minimum input angle is then $\theta_{min} = 0,301$ radian. The maximum input angle is $\theta_{max} = 2,84$ radian. These values are confirmed by measurements on the prototypes.

From these angular positions, we calculate the maximum position or reach as $x_{max} = 19,77$ cm and the minimum value is $x_{min} = 3$ cm. Again these calculated values are confirmed by measurements.

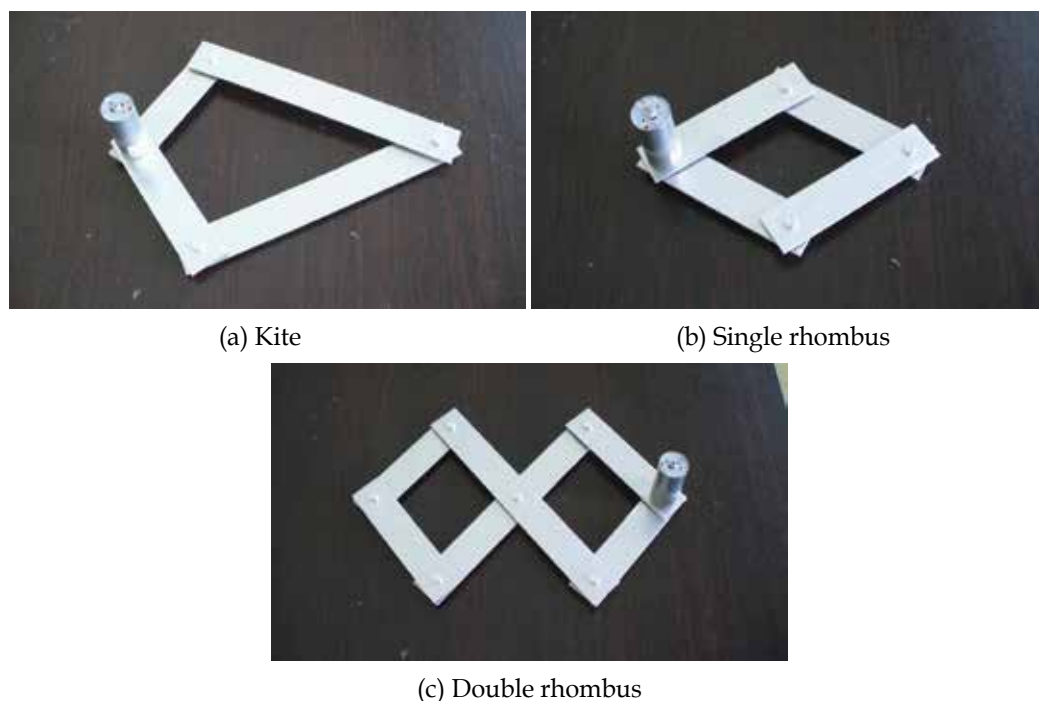


Fig. 15. The three four-bar actuator prototypes

The encumbrance can also be deduced. At θ_{max} the maximum encumbrance is calculated as $E_{max} = 22,77$ cm. This was confirmed by measurements.

5.2.3 Motion analysis

In the curves of figure 16, t is in fact θ_2 , the angular input position in radian, and ω is the angular velocity made to change from -4π to 4π radians/s.

The first question is about the actuator linearity and this issue can be answered by plotting the extremity position in relation to the input angle position. The motion becomes non-linear at the angular extremities and it becomes almost linear for a large number of angular positions from -1 to 1 radian corresponding to the position range of -18 to 18 cm.

The second question is to determine the extremity velocity profile according to input angle position at angular velocities going from the minimum until the maximum. This last value comes from the geared motor specifications. End-effector velocity changes almost linearly with the angular velocity but changes non-linearly with the angular position. It cancels out at angular extremities and it becomes maximal at $\theta = \pi$.

This velocity profile also corresponds to the accuracy profile. This means that for a predefined encoder accuracy located on the gearmotor shaft, the resulting extremity accuracy will be changing accordingly. One can foresee that the lowest accuracy is attained at $\theta = \pi$ and the best accuracies are achieved near the angular position extremities.

The third question involves the extremity acceleration in relation to input angle position at angular velocities going from the minimum until the maximum. We have to fix some angular acceleration and the value $1rad/s^2$ has been selected arbitrarily. Extremity

acceleration changes non-linearly with the angular velocity. It changes almost linearly with the angular position. It reaches very high values when the linkage reaches very close to the maximum position.

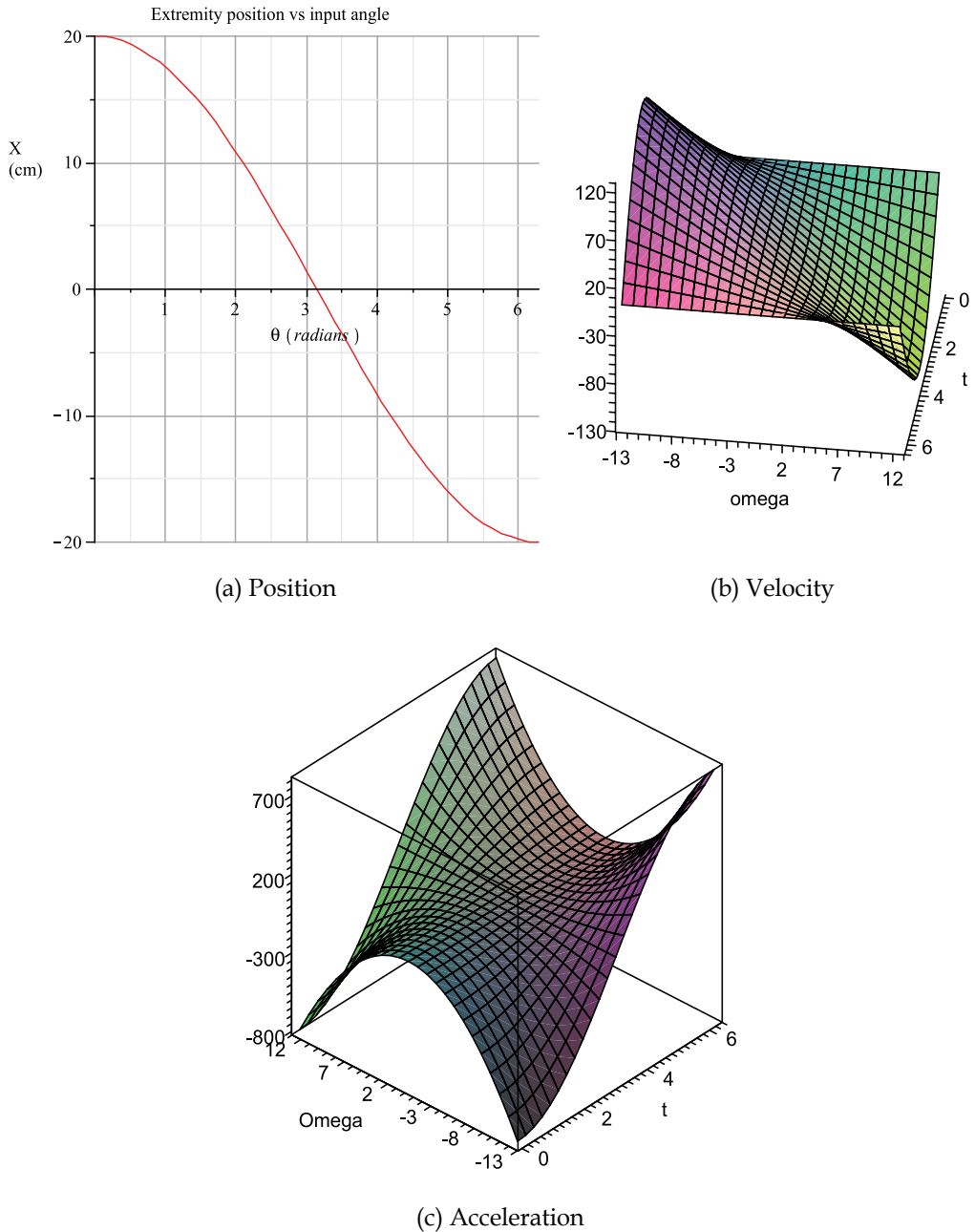


Fig. 16. The rhombus kinematics performance

6. Conclusion

As can be seen by the results obtained by calculations as well by experiments on the prototypes, four-bar mechanisms can be rearranged in rhombus and kite configurations which lead to very performant prismatic pairs allowing to design very fast linear actuators. Moreover, to improve performance and to reduce emcumbrance, networking the rhombus four-bars can lead to very good results.

In the author's knowledge, this is the first time that four-bars were envisaged to be applied as linear actuators.

The next step will be to analyze their dynamics design integrating force analysis.

Then, to design a large scale parallel robot prototype will help investigate their worthiness towards the design a very high speed milling machine.

Finally, several optimization problems may arise to determine proper linkage sizing.

7. References

- Ceccarelli M. (1997) A new 3 d.o.f. spatial parallel mechanism. *Mechanism and Machine Theory*, vol.32(no.8), pp 896-902.
- DaVinci, L. (1493) *Tratado de Estatica y Mechanica en Italiano*. CodexMadrid 1, National Library Madrid.
- Euclid (2002) *Euclid's Elements - All thirteen books in one volume*. Green Lion Press. Based on Heath's translation, Greek original from c. 300 BC
- Fichter, E.F., Kerr, D.R. and Rees-Jones, J. (2009) The Gough-Stewart platform parallel manipulator: a retrospective appreciation. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Volume 223, Number 1, pp. 243-281.
- Gogu, G. (2004) Chebychev-Grübler-Kutzbach's criterion for mobility calculation of multiloop mechanisms revisited via theory of linear transformations. *European Journal of Mechanics - A/Solids*. Volume 24, Issue 3, May-June 2005, Pages 427-441
- Gough, V. E. and Whitehall, S. G. (1962) Universal tyre testing machine. *Proceedings of the 9th International Automobile Technical Congress, discussion*, Federation Internationale des Societes d'Ingenieurs des Techniques de l'Automobile (FISITA) (Ed. G. Eley), (IMechE 1, London, UK), pp. 117137.
- Kempe A.B. (1877) *How to draw a straight line; a lecture on linkages*. Macmillan and Co, London
- Moon, F.C. (2007) *The Machines of Leonardo Da Vinci and Franz Reuleaux*. Volume on Kinematics of Machines from the Renaissance to the 20th Century, Series on History of Mechanism and Machine Science, Springer Netherlands, 417 pages
- Reuleaux, F. (1876) *Kinematics of Machinery: Outlines of a Theory of Machines*. Macmillan and Co., London.
- Rolland, L. (1998) Conception de mécanismes, élaboration des principes demobilit'e. Technical report number 98-02, ISR, EPFL, Lausanne.
- Rolland, L. (1999) The Manta and the Kanuk novel 4-dof parallel mechanisms for industrial handling. *Proceedings of the ASME International Mechanical Engineering Congress*, Nashville, 14-19 Novembre 1999.

- Scheiner, C. (1631) *Pantographice, seu ars delineandi res quaslibet per parallelogrammum lineare seu cavum, mechanicum, mobile*. Romae: Ex typographia Ludouici Grignani, sumptibus Hermanni Scheus, vol. 12, 108 pages.
- Soylemez E. (1999) *Mechanisms*. METU Press, Ankara, 350 pages.
- Uicker, J.J., Pennock, G.R. and Shigley, 2003, J.E. *Theory of Machines and Mechanisms, third edition*. Oxford University Press, New-York, 2003, 734 pages.

Sliding Mode Control of Robot Manipulators via Intelligent Approaches

Seyed Ehsan Shafiei
Shahrood University of Technology
Iran

1. Introduction

1.1 Robot manipulators

Robot manipulators are well-known as nonlinear systems including strong coupling between their dynamics (Craig, 1996). These characteristics, in company with: 1) *structured uncertainties* caused by model imprecision of link parameters, payload variation, etc., and 2) *unstructured uncertainties* produced by un-modeled dynamics –such as nonlinear friction and external disturbances– make the motion control of rigid-link manipulators a complicated problem (Spong & Vidyasagar, 1989). Practice trajectory control is required in many of the sophisticated applications of manipulators (e.g. machining, welding, complex assembly). On the other hand, robot manipulators have to face various uncertainties in their dynamics and they are required to handle various tools and, hence, the dynamic parameters of the robots vary during operation. Thus, it is difficult to initiate an appropriate mathematical model for employing model-based control strategies.

In general, the intelligent control approaches can attenuate the effects of structured parametric uncertainty and unstructured disturbance by using their powerful learning ability without a detailed knowledge of the controlled plant in the design processes. On the other hand, many intelligent control algorithms could have been found for the robot control system without including the actuator dynamics, while, actuator dynamics carry out a significant role in the complete robot dynamics and ignoring them may cause detrimental effects, especially in the case of high-velocity moment, highly varying loads, friction, and actuator saturations (Chang et al., 2008), (Chang & Yen, 2009). Since the electrical actuators are highly controllable in comparison with the other one, they are more convenient for driving manipulators. Also, in practical applications, the voltages or currents of the electrical actuators are accessible for applying control commands and consequently, torque-based control design confronts implementation problems when one intends to apply the torque control commands directly to actuators. Additionally, one constraint in the robot controller designs is saturation nonlinearity of actuators which is less considered in control design of robot manipulators.

1.2 Sliding mode control

Sliding mode control (SMC) is a variable-structure, robust control strategy which is capable in controlling different class of uncertain systems including nonlinear systems, MIMO systems, and even discrete time systems (Utkin, 1978), (Zhang et al., 2008). Such

uncertainties may be structured, unstructured, or may result from nondeterministic features of the plant. A sliding mode controller is essentially high gain switching controller. The idea is to keep the trajectory of the system on a particular surface in the phase space. In a two dimensional system this would reduce to following a line in the phase plane. The SMC law is formulated using a Lyapunov approach that guarantees robustness despite the presence of bounded modeling uncertainties (Slotin & Li, 1991).

However, sliding mode control has a good deal of advantages such as insensitivity to parameter variations, disturbance rejection and fast dynamic responses (Zhang et al., 2008). Despite these merits, SMC suffers from some disadvantages. Actually, the sliding mode control law consists of two main parts. The first part is the *equivalent control law* which involves inverse dynamics of model nonlinearities that demonstrates the dependency of SMC on the dynamical model of the plant. The second part is the *robustifying term* which has discontinuous nature and may employ unnecessary high control gain to overcome uncertainties and disturbances. However, this discontinuity may lead to chattering phenomenon that can excite un-modeled high-frequency plant dynamics and harm the overall control system. Also, using high control gain may cause saturating the actuators. Accordingly, several methods have been developed for improving the SMC performance which the most significant of them is intelligent control approach (Kaynak et al., 2001) mainly includes fuzzy logic control and neural network control.

1.3 Fuzzy control

Fuzzy control is based on fuzzy logic and is a nonlinear control strategy which uses heuristic information. In the fuzzy control design methodology, human thinking and expert knowledge are incorporated into a fuzzy system that emulates the decision-making process of the human. Basically, a fuzzy system in general or fuzzy control in especial comprises five main parts: 1) fuzzyfication of inputs, 2) fuzzy control rules, 3) fuzzy implication, 4) fuzzy reasoning and 5) defuzzification (Lee, 1990), (Wang, 1997).

Fuzzy control represents efficient performance in absence of uncertainties and disturbance and where the plant dynamics were well-described with mathematical equations. Moreover, stability of the fuzzy control systems is hard to analyze and needs strong mathematical procedures. Therefore, it seems reasonable to enhance fuzzy control efficiency by using of incorporating well-organized nonlinear control methods (e.g. sliding mode control).

1.4 Neural network control

Prominent features of neural networks (NN) have drawn much attention in control research areas especially in robot control systems (Lewis, 1998). Some of this features that are closely related to control design strategies are:

- *Universal approximation:* neural networks can approximate smooth nonlinear functions with any degree of accuracy. This feature may be utilized in nonlinear control systems.
- *Learning and adaptation:* neural networks can be trained off-line with adequate amount of data or they can be adapted on-line with appropriate adaptation laws. This property is applied to identification concerns.
- *MIMO characteristic:* neural networks can accept many inputs and can produce required number of outputs. So they are appropriate for MIMO control systems.

There are many other distinguished features as parallel processing, hardware implementation and data fusion etc. that we neglect them here. Also, fuzzy logic may be employed for constructing special networks like fuzzy-neural-networks. Alternatively, neural networks may be exerted to fuzzy control design like neuro-fuzzy control systems.

In the remainder of this chapter three methods are proposed for controller designs. In the first case, sliding mode control plays the main role and fuzzy logic is employed for tuning the controller gains. In the second case, fuzzy control and sliding mode control have the parallel mission in control strategy. Finally, the third case proposes the sliding mode control method by using adaptive neural network approach.

2. Sliding mode control using fuzzy approach

2.1 Sliding_mode_PID controller design by using of fuzzy tuning

This section addresses a chattering free sliding mode control (SMC) for a robot manipulator including PID part with a fuzzy tunable gain. The main idea is that the robustness property of SMC and good response characteristics of PID are combined with fuzzy tuning gain approach to achieve more acceptable performance. For this purpose, in the first stage, a PID sliding surface is considered such that the robot dynamical equations can be rewritten in terms of sliding surface and its derivative and the related control law of the SMC design will contain a PID part. The stability guarantee of this sliding mode PID-controller is proved by a lemma using Lyapunov direct method. Then, in the second stage, in order to decrease the reaching time to the sliding surface and deleting the oscillations of the response, a fuzzy tuning system is used for adjusting both controller gains including sliding controller gain parameter and PID coefficients (Ataei & Shafiei, 2008).

2.1.1 Mathematical model of the system

The dynamical equation of an n-link robot manipulator in the standard form is as follows (Spong & Vidyasagar, 1989):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_d = \tau \quad (1)$$

where $M(q) \in R^{n \times n}$ is the completed inertia matrix, the vectors $q, \dot{q}, \ddot{q} \in R^n$ are the position, velocity and angular acceleration of the robot joints, respectively. Moreover, the matrix $C(q, \dot{q}) \in R^{n \times n}$ is the matrix of Coriolis and centrifugal forces and $G(q) \in R^n$ is the gravity vector. Also, $\tau_d \in R^n$ denotes the vector of disturbance and un-modeled dynamics, and finally, τ is the torque vector. In the following, two conventional properties of the robot manipulators are considered.

Property 2.1. The inertia matrix $M(q)$ is symmetric and positive definite, $M^T = M$.

Property 2.2. The matrix of $(\dot{M} - 2C)$ is skew-symmetric, i.e. for any vector of X , we have $X^T(\dot{M} - 2C)X = 0$.

2.1.2 Sliding mode control with PID

The objective of the tracking control is to design such a control law, for obtaining the suitable input torque τ , that the position vector q could track the desired trajectory q_d . In this regard, the tracking error vector is defined as follows:

$$e = q_d - q \quad (2)$$

In order to apply the SMC, the sliding surface is considered as relation (3) which contains the integral part in addition to the derivative term:

$$s = \dot{e} + \lambda_1 e + \lambda_2 \int_0^t e dt \quad (3)$$

where λ_i is diagonal positive definite matrix. Therefore, $s = 0$ is a stable sliding surface and $e \rightarrow 0$ as $t \rightarrow \infty$. The robot dynamical equations can be rewritten based on the sliding surface (in term of filtered error) as:

$$M\dot{s} = -Cs + f + \tau_d - \tau \quad (4)$$

Where

$$f = M(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) + C(\dot{q}_d + \lambda_1 e + \lambda_2 \int_0^t e dt) + G \quad (5)$$

Now, the control input can be considered as:

$$\tau = \hat{f} + K_v s + K \text{sgn}(s) \quad (6)$$

where

$$\hat{f} = \hat{M}(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) + \hat{C}(\dot{q}_d + \lambda_1 e + \lambda_2 \int_0^t e dt) + \hat{G} \quad (7)$$

is an estimation of f and $K_v s = K_v \dot{e} + K_v \lambda e + K_v \lambda \int_0^t e dt$ is the outer PID tracking loop, and K_v, K are diagonal positive definite matrices and are defined such that the stability conditions are guaranteed. The $\text{sgn}(s)$ is also the sign function.

We have also:

$$|\tilde{f}| = \left| \tilde{M}(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) + \tilde{C}(\dot{q}_d + \lambda_1 e + \lambda_2 \int_0^t e dt) + \tilde{G} \right| \leq F \quad (8)$$

where $\tilde{f} = f - \hat{f}$, $\tilde{M} = M - \hat{M}$, $\tilde{C} = C - \hat{C}$, and $\tilde{G} = G - \hat{G}$. Vector F can also be selected as the following relation:

$$F = \left| \tilde{M}(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) \right| + \left| \tilde{C}(\dot{q}_d + \lambda_1 e + \lambda_2 \int_0^t e dt) \right| + \left| \tilde{G} \right| \quad (9)$$

In order to govern the system states (e, \dot{e}) to reach the sliding surface $s = 0$ in a limited time and to remain there, the control law should be designed such that the following sliding condition is satisfied (Slotin & Li, 1991):

$$\frac{1}{2} \frac{d}{dt} [s^T M s] < -\eta (s^T s)^{1/2}, \quad \eta > 0 \quad (10)$$

This aim is fulfilled in the following lemma.

Lemma 2.1. In the SMC design of a system with dynamical equation (1) and sliding surface (3), if the control input τ is selected as (6), by considering F as (9) and $K = \text{diag}(K_{11}, K_{22}, \dots, K_{mm})$ with the following components:

$$K_{ii} = [F + |K_v s| + T_D + \eta]_i, \quad i = 1, 2, \dots, n \quad (11)$$

Then, the sliding condition (10) is satisfied by equation (4).

Proof: Consider the following Lyapunov function candidate:

$$V = \frac{1}{2} s^T M s \quad (12)$$

Since M is positive definite, for $s \neq 0$ we have $V > 0$ and by taking derivative from relation (12) and regarding the symmetric property of M , it can be written:

$$\dot{V} = \frac{1}{2} s^T \dot{M} s + s^T M \dot{s} \quad (13)$$

By substituting (4) into (13) and considering that $s^T (\dot{M} - 2C)s = 0$, we have:

$$\dot{V} = \frac{1}{2} s^T \dot{M} s - s^T C s + s^T (f + \tau_d - \tau) = s^T (f + \tau_d - \tau) \quad (14)$$

By replacing the relation (6) into (14), \dot{V} can be rewritten as:

$$\dot{V} = s^T (f + \tau_d - \hat{f} - K_v s - K \operatorname{sgn}(s)) = s^T (\tilde{f} + \tau_d - K_v s) - \sum_{i=1}^n K_{ii} |s_i| \quad (15)$$

Since the following inequality (16) is valid and by regarding the relation (11), we have:

$$F + |K_v s| + T_D \geq |\tilde{f} + \tau_d - K_v s| \quad (16)$$

$$K_{ii} \geq |[\tilde{f} + \tau_d - K_v s]_i| + \eta_i \quad (17)$$

Finally, it can be concluded that:

$$\dot{V} \leq -\sum_{i=1}^n \eta_i |s_i| \quad (18)$$

This indicates that V is a Lyapunov function and the sliding condition (10) has been satisfied.

The use of sign function in the control law leads to high oscillations in control torque which is undesired phenomenon and is called chattering. To overcome this drawback, there are some solutions that one of them is using the following saturation function instead of sign function in the discontinuous part of the control law:

$$\operatorname{sat}\left(\frac{s}{\varphi}\right) = \begin{cases} 1 & s \geq \varphi \\ \frac{s}{\varphi} & -\kappa < s < \varphi \\ -1 & s \leq -\varphi \end{cases} \quad (19)$$

By this, there is a boundary layer φ around the sliding surface such that once the state trajectory reaches this layer, then it will be remaining there.

2.1.3 Fuzzy gain tuning

As mentioned before, by using a high gain in SMC, i.e. K , the sensitivity of the controller to the model uncertainties and external disturbances can be reduced. Moreover, a high gain in PID part of the control system (K_v) can reduce the reaching time to sliding surface and tracking error. However, increasing the gain causes the increment of the oscillations in the input torque around the sliding surface. Therefore, if this gain can be tuned based on the distance of the states to the sliding surface, a more acceptable performance can be achieved. In the other words, the value of gain should be selected high when the state trajectory is far from the sliding surface and when the distance is decreasing, its value should be decreased. This idea can be accomplished by using fuzzy logic in combination with SMC to tune the gain adaptively.

For this purpose, two-input one-output fuzzy system is designed whose inputs are s and \dot{s} which are the distance of state trajectories to the sliding surface and its derivative, respectively. The membership functions of these two inputs are shown in Fig. 1. The output of the fuzzy system is denoted by K_{fuzz} and has been shown in Fig. 2. For applying these gains to the control input, the normalization factors N and N_v are used as the following relations:

$$K = N \cdot K_{fuzz} \quad (20)$$

$$K_v = N_v \cdot K_{fuzz} \quad (21)$$

These factors can be selected by trial and error such that the stability condition (17) is satisfied.

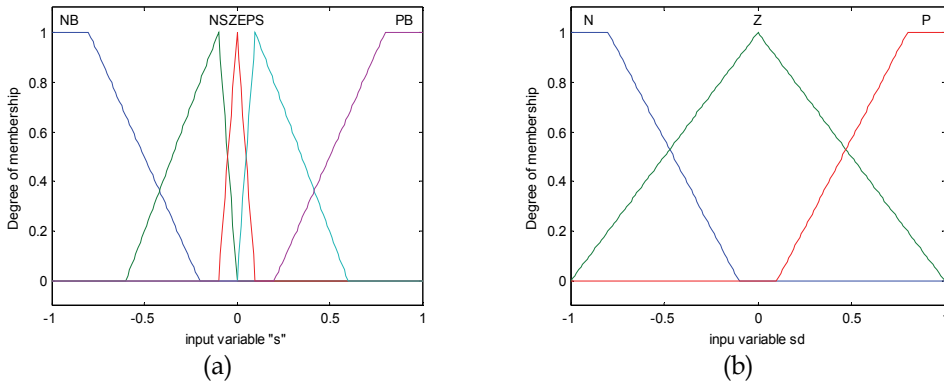


Fig. 1. The membership functions, (a) input s , (b) input \dot{s}

The maximum values of K and K_v are limited according to the system actuators power, and the minimum value of K should not be less than the provided amount in relation (17). The fuzzy rule base has been given in table 1 in which the following abbreviations have been used: NB: Negative Big; NS: Negative Small; Z: Zero; PS: Positive Small; PB: Positive Big; M: Medium. For example, when s is negative small (NS) and \dot{s} is positive (P), then K_{fuzz} is small (S).

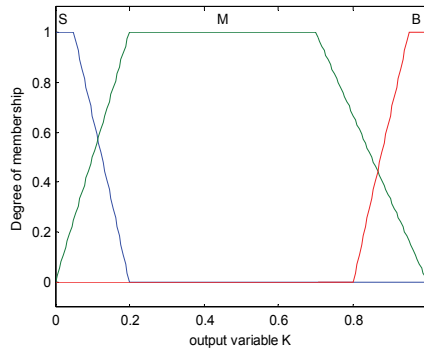


Fig. 2. The membership functions of the output K_{fuzz}

s s_i	NB	NS	Z	PS	PB
N	B	B	M	S	B
Z	B	M	S	M	B
P	B	S	M	B	B

Table 1. The fuzzy rule base for tuning K_{fuzz}

Simulation example 2.1. In order to show the effectiveness of the proposed control law, it is applied to a two-link robot with the following parameters:

$$M(q) = \begin{bmatrix} \alpha + \beta + 2\gamma \cos q_2 & \beta + \gamma \cos q_2 \\ \beta + \gamma \cos q_2 & \beta \end{bmatrix} \quad (22)$$

$$C(q, \dot{q}) = \begin{bmatrix} -\gamma \dot{q}_2 \sin q_2 & -\gamma (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ \gamma \dot{q}_1 \sin q_2 & 0 \end{bmatrix} \quad (23)$$

$$G(q) = \begin{bmatrix} \alpha \delta_1 \cos q_1 + \gamma \delta_1 \cos(q_1 + q_2) \\ \gamma \delta_1 \cos(q_1 + q_2) \end{bmatrix} \quad (24)$$

where $\alpha = (m_1 + m_2)a_1^2$, $\beta = m_2 a_2^2$, $\gamma = m_2 a_1 a_2$, $\delta = g/a_1$, and $m_1, m_2, a_1 = .7, a_2 = .5$ are the masses and lengths of the first and second links, respectively. The masses are assumed to be in the end of the arms and the gravity acceleration is considered as $g = 9.8$. Moreover, the masses are considered with 10% uncertainty as follow:

$$\begin{aligned} m_1 &= m_{1_0} + \Delta m_1, \quad |\Delta m_1| \leq .4 \\ m_2 &= m_{2_0} + \Delta m_2, \quad |\Delta m_2| \leq .2 \end{aligned} \quad (25)$$

where $m_{1_0} = 4$ and $m_{2_0} = 2$, and \hat{M} , \hat{C} , and \hat{G} are estimated. The desired state trajectory is:

$$q_d = \begin{bmatrix} 1 - \cos \pi t \\ 2 \cos \pi t \end{bmatrix} \quad (26)$$

and the disturbance torque is considered as:

$$\tau_d = \begin{bmatrix} 0.5 \sin 2\pi t \\ 0.5 \sin 2\pi t \end{bmatrix} \quad (27)$$

which leads to $T_D = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$.

The design parameters are determined as follow:

$$\lambda_1 = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}, \lambda_2 = \begin{bmatrix} 40 & 0 \\ 0 & 40 \end{bmatrix} \quad (28)$$

Values of φ and η are selected as $\varphi = 0.167$ and $\eta = [0.1 \ 0.1]^T$. Moreover, the factors N and N_v are selected as:

$$N = \begin{bmatrix} 50 & 0 \\ 0 & 5 \end{bmatrix}, N_v = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \quad (29)$$

In order to show the improvement due to the proposed method, the simulation results of applying this method are compared with the related results of the conventional SMC. The tracking error and control law in the case of conventional SMC have been shown in Fig. 3 and Fig. 4, respectively. The corresponding graphs for the case of applying fuzzy SMC-PID are also provided in Fig. 5 and 6.

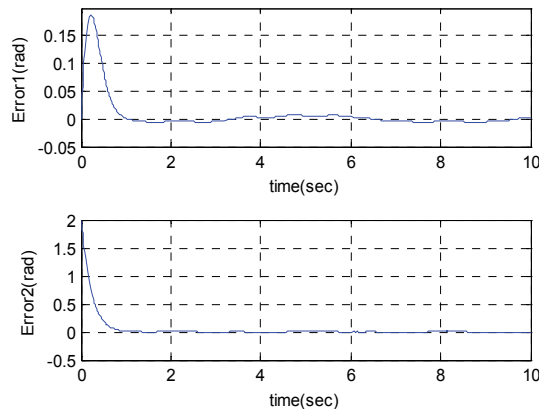


Fig. 3. The tracking errors in the case of using conventional SMC

As it can be seen from these figures, the proposed fuzzy SMC-PID has faster response and less tracking error in comparison with conventional SMC. In order to show more clearly the difference between the tracking errors in two cases, the enlarged graphs have been provided in Fig. 7 and 8.

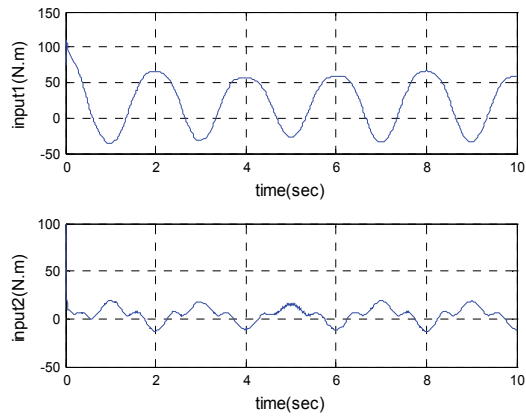


Fig. 4. The control inputs in the case of using conventional SMC

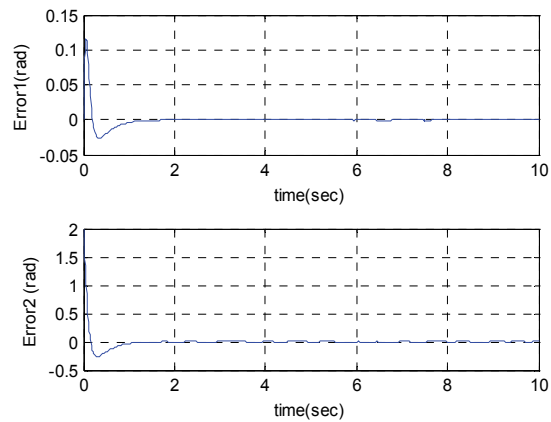


Fig. 5. The tracking errors in the case of using Fuzzy SMC-PID

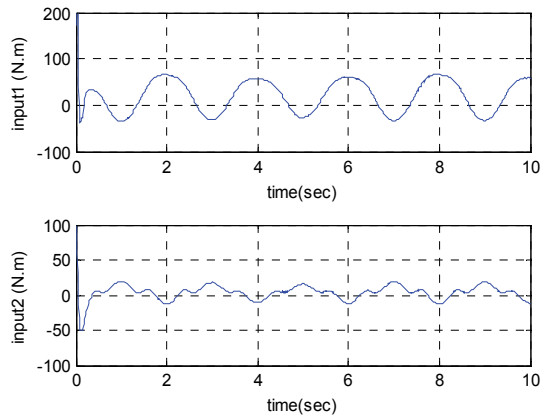


Fig. 6. The control inputs in the case of using Fuzzy SMC-PID

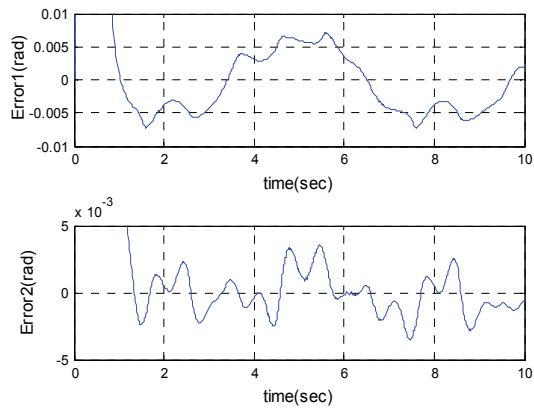


Fig. 7. The enlargement of the tracking errors in the case of using conventional SMC

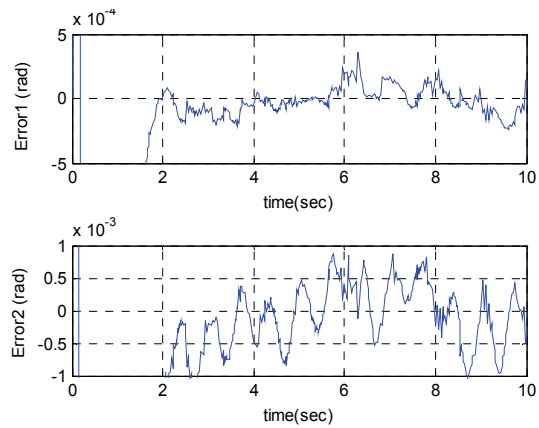


Fig. 8. The enlargement of the tracking errors in the case of using Fuzzy SMC-PID

2.2 Incorporating sliding mode and fuzzy control

In this section, a combined controller includes SMC term and fuzzy term is proposed for set-point tracking of robot manipulators. Some practical issues, such as existence of joint frictions, restriction on input torque magnitude due to saturation of actuators, and modeling uncertainties have been considered here. Design procedure contains two steps. First, SMC design is accomplished and system stability in this case is provided by Lyapunov direct method. When the tracking error would be less than predefined value then a sectorial fuzzy controller (SFC), (Calcev, 1998), is responsible for control action. Designing of this kind of fuzzy controller is exactly the same as in which has performed in (Santibanez et al., 2005).

This proposed controller has following advantages. 1) There are less tracking errors versus traditional SMC in condition that the control input is limited, 2) the chattering is avoided, 3) convergence of tracking error is more rapid than fuzzy controller designed in (Santibanez et al., 2005) and modeling uncertainty is considered here (Shafiei & Sepasi, 2010).

2.2.1 Mathematical model and problem formulation

This time the friction of joint is considered and is added to dynamical equation (1) as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}, \tau) = \tau \quad (30)$$

where $F(\dot{q}, \tau) \in R^n$ stands for the friction vector which is as follows (Cai & Song, 1994):

$$f_i(\dot{q}, \tau_i) = b_i \dot{q}_i + f_{ci} \operatorname{sgn}(\dot{q}_i) + [1 - |\operatorname{sgn}(\dot{q}_i)|] \operatorname{sat}(\tau_i; f_{si}) \quad (31)$$

where $f_i(\dot{q}, \tau_i)$, $i = 1, 2, \dots, n$, denotes the i -th element of $F(\dot{q}, \tau)$ vector. b_i , f_{ci} and f_{si} are the viscous, Coulomb and static friction, respectively. The $\operatorname{sat}(\cdot; \cdot)$ indicates saturation function with following equation.

$$\operatorname{sat}(x; r) = \begin{cases} r & \text{if } x > r \\ x & \text{if } -r \leq x \leq r \\ -r & \text{if } x < -r \end{cases}$$

In the following, $M(q)$, $C(q, \dot{q})$ and $G(q)$ might be shown by M , C , and G , respectively in where it would be requisite.

Now, the boundedness properties are defined as below:

$$\sup_{q \in R^n} \{|g_i(q)|\} \leq \bar{g}_i, \quad i = 1, \dots, n \quad (32)$$

where g_i stands for the i -th element of $G(q)$ and \bar{g}_i is finite nonnegative constant. Assume that the maximum torque that joint actuator can supply is τ^{\max} . Therefore:

$$|\tau_i| \leq \tau_i^{\max}, \quad i = 1, \dots, n \quad (33)$$

and each actuator satisfies the following condition:

$$\tau_i^{\max} > \bar{g}_i + f_{si} \quad (34)$$

In robot modeling, one can well determine the terms $M(q)$ and $G(q)$ but it is difficult in most cases obtaining the parameters of $C(q, \dot{q})$ and $F(\dot{q}, \tau)$ exactly. So, in present section, the matrix C is considered as follows:

$$C = \hat{C} + \Delta C \quad (35)$$

where \hat{C} denotes estimation of C , and ΔC is bounded estimation error which has the following relation:

$$|\Delta C_{i,j}| \leq 0.1 |C_{i,j}| \quad (36)$$

where $C_{i,j}$ stands for elements of the matrix C . Also the vector F is supposed as an external disturbance with the following unknown upper bound:

$$\|F\| \leq F_{up} \quad (37)$$

where the operator $\|\cdot\|$ denotes Euclidean norm.

If one considers the desired point which joint position must be held on it as q_d , then the position error could be defined as:

$$\tilde{q} = q_d - q \quad (38)$$

Here, the set-point tracking problem refers to define the control law such that error e would be driven toward the inside of an arbitrary small region around zero with maintaining the torques within the constraints (33). In succeeding subsections, this aim will be attained.

2.2.2 Sliding mode controller design

The following sliding surface is considered for designing SMC controller.

$$s = \dot{e} + \lambda e \quad (39)$$

where $e = -\tilde{q} = q - q_d$ is error vector and λ is supposed symmetric positive definite matrix such that $s=0$ would become a stable surface. The reference velocity vector " \dot{q}_r " is defined as in (Slotin & Li, 1991):

$$\dot{q}_r = \dot{q}_d - \lambda e \quad (40)$$

Thus, one can interpret sliding surface as:

$$s = \dot{q} - \dot{q}_r \quad (41)$$

Here, the SMC controller design is expressed by lemma 2.2.

Lemma 2.2. Consider the system with dynamic equation (30) and sliding surface and reference velocity defined by (39) and (40), respectively. If one chooses the control law below,

$$\tau = \hat{\tau} - K \operatorname{sgn}(s) \quad (42)$$

such that

$$\hat{\tau} = M\ddot{q}_r + \hat{C}\dot{q}_r + G \quad (43)$$

and

$$K_i \geq \|\Delta C\dot{q}_r\| + \Gamma_i \quad (44)$$

then the sliding condition (10) is satisfied. In the last inequality, K_i denotes the element of sliding gain vector K and Γ is design parameter vector which must be selected such that $\Gamma_i \geq F_{ip} + \eta_i$.

Proof: Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}s^T Ms \quad (45)$$

Since M is positive definite, for $s \neq 0$ we have $V > 0$ and by taking time derivative of the relation (45) and regarding the symmetric property of M , it can be written:

$$\dot{V} = s^T M\dot{s} + \frac{1}{2}s^T \dot{M}s \quad (46)$$

from (40), gives:

$$\dot{V} = s^T (M\ddot{q} - M\ddot{q}_r) + \frac{1}{2}s^T \dot{M}s \quad (47)$$

By substituting (30) in (47) and considering asymmetry property $s^T (\dot{M} - 2C)s = 0$, we have:

$$\dot{V} = s^T (\tau - C\dot{q}_r - G - F - M\ddot{q}_r) \quad (48)$$

Now, applying (42) and (43) yields:

$$\dot{V} = s^T (\Delta C\dot{q}_r + F) - \sum_{i=1}^n K_i |s_i| \quad (49)$$

Finally, from relation (44) it can be concluded that:

$$\dot{V} \leq -\sum_{i=1}^n \eta_i |s_i| \quad (50)$$

This indicates that V is a Lyapunov function and the sliding condition (10) has been satisfied.

Note that, in general, the sign function is replaced by saturation function as $\text{sat}(s/\varphi)$, where φ denotes boundary layer thickness.

2.2.3 Fuzzy controller design

In this section, the SFC class of fuzzy controller studied in (Santibanez et al., 2005) is considered which has two-input one-output rules used in the formulation of the knowledge base. These IF-THEN rules have following form:

$$\text{IF } x_1 \text{ is } A_1^{l_1} \text{ and } x_2 \text{ is } A_2^{l_2} \text{ THEN } y \text{ is } B^{l_1 l_2} \tag{51}$$

where $x = [x_1 \ x_2]^T \in U = U_1 \times U_2 \subset \mathfrak{R}^2$ and $y \in V \subset \mathfrak{R}$. For each input fuzzy set $A_j^{l_j}$ in $x_j \in U_j$ and output fuzzy set $B^{l_1 l_2}$ in $y \in V$, exist an input membership function $\mu_{A_j^{l_j}}(x_j)$ and output membership function $\mu_{B^{l_1 l_2}}(y)$ shown in Fig. 10 and Fig. 11, respectively.

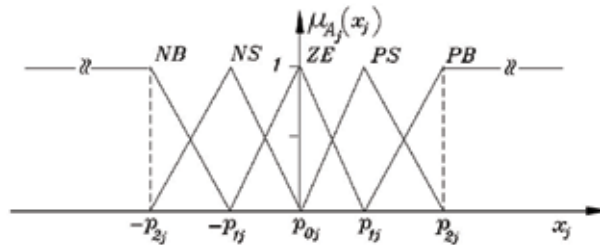


Fig. 9. Input membership functions

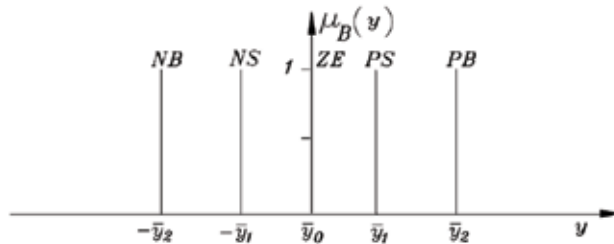


Fig. 10. Output membership functions

The fuzzy system considered here has following specifications: Singleton fuzzifier, triangular membership functions for each inputs, singleton membership functions for the output, rule base defined by (51), (see Table. 2), product inference and center average defuzzifier.

$x_2 \backslash x_1$	NB	NS	ZE	PS	PB
NB	NB	NB	NS	ZE	ZE
NS	NB	NB	NS	ZE	ZE
ZE	NS	NS	ZE	PS	PS
PS	ZE	ZE	PS	PB	PB
PB	ZE	ZE	PS	PB	PB

Table 2. The fuzzy rule base for obtaining output y

Thus, one can compute the output y in terms of inputs as follows (Wang, 1997):

$$y(x) = \varphi(x_1, x_2) = \frac{\sum_{l_1} \sum_{l_2} \bar{y}^{l_1 l_2} \left(\prod_{j=1}^2 \mu_{A_j^{l_j}}(x_j) \right)}{\sum_{l_1} \sum_{l_2} \left(\prod_{j=1}^2 \mu_{A_j^{l_j}}(x_j) \right)} \tag{52}$$

Special properties of this input-output mapping $y(x)$ for x_1, x_2 are given in (Santibanez et al., 2005).

Lemma 2.3. For the system with dynamical equation (30), if one chooses the following control law,

$$\tau = \varphi(\tilde{q}, \dot{\tilde{q}}) + G(q) \quad (53)$$

where \tilde{q} is defined as (38) and $\dot{\tilde{q}} = \dot{q}_d - \dot{q}$ is velocity error vector, then the closed-loop system shown in Fig. 11 becomes stable.

Proof: the stability analysis is based on the study performed in (Calcev 1998) and is fully discussed in (Santibanez et al., 2005), so it is omitted here. Note that for constant set-point we have $\dot{q}_d = 0$, hence $\dot{\tilde{q}} = -\dot{q}$.

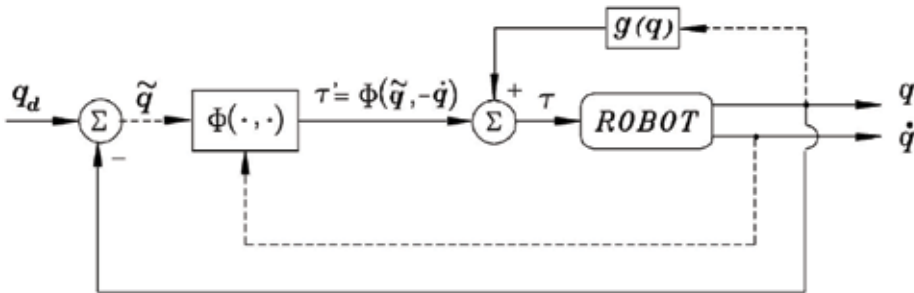


Fig. 11. Closed-loop system in the case of fuzzy controller (Santibanez et al., 2005)

2.2.4 Incorporating SMC and SFC

Each of the two controllers explained in last two subsections drives the robot joint angles to desired set-point in finite time and according to the Lemma 2.2 and 2.3 the closed-loop system is stable in both cases. In this section, for utilizing advantages of both sliding mode control and sectorial fuzzy control, and also minimizing the drawbacks of both of them, the following control law is proposed:

$$\tau = \begin{cases} \hat{\tau} - K \operatorname{sgn}(s) & \text{when } |q_e| \geq \alpha \\ y(q_e, \dot{q}_e) + G(q) & \text{when } |q_e| < \alpha \end{cases} \quad (54)$$

where α is strictly positive small parameter which can be determined adaptively or set to a constant value. So, while the magnitude of error is greater than or equal to α , SMC drives the system states, errors in our case, toward sliding surface and as soon as the magnitude of error becomes less than α , then the SFC which is designed independent of initial conditions, controls the system. Since the SMC shows faster transient response, the response of the system controlled by (54) is faster than the case of SFC. Additionally, in spite of the torque boundedness, since the SFC controls the system in the steady state, the proposed controller (54) has less set-point tracking error. Also, since near the sliding surface the proposed controller switch from SMC to SFC, therefore, the chattering is avoided here.

Simulation example 2.2. In order to show the effectiveness of the proposed control law, it is applied to a two-link direct drive robot arm with the following parameters (Santibanez et al., 2005):

$$\begin{aligned}
 M(q) &= \begin{bmatrix} 2.351 + 0.168 \cos(q_2) & 0.102 + 0.084 \cos(q_2) \\ 0.102 + 0.084 \cos(q_2) & 0.102 \end{bmatrix} \\
 \hat{C}(q, \dot{q}) &= \begin{bmatrix} -0.084 \sin(q_2) \dot{q}_2 & -0.084 \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ 0.084 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \\
 G(q) &= 9.81 \begin{bmatrix} 3.921 \sin(q_1) + 0.186 \sin(q_1 + q_2) \\ 0.186 \sin(q_1 + q_2) \end{bmatrix} \\
 F(\dot{q}) &= \begin{bmatrix} 2.288 \dot{q}_1 + 8.049 \operatorname{sgn}(\dot{q}_1) + [1 - |\operatorname{sgn}(\dot{q}_1)|] \operatorname{sat}(\tau_1; 9.7) \\ 0.186 \dot{q}_2 + 1.734 \operatorname{sgn}(\dot{q}_2) + [1 - |\operatorname{sgn}(\dot{q}_2)|] \operatorname{sat}(\tau_2; 1.87) \end{bmatrix} \\
 C &= \hat{C} + \Delta C
 \end{aligned} \tag{55}$$

According to the actuators manufacturer, the direct drive motors are able to supply torques within the following bounds:

$$\begin{aligned}
 |\tau_1| &\leq \tau_1^{\max} = 150 [\text{Nm}] \\
 |\tau_2| &\leq \tau_2^{\max} = 15 [\text{Nm}]
 \end{aligned} \tag{56}$$

The desired set-point is,

$$q_d = [\pi \quad -\pi]^T \tag{57}$$

which is applied as a step function at time zero. The SMC design parameters are as below:

$$\lambda = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \Gamma = \begin{bmatrix} 140 \\ 8 \end{bmatrix} \text{ and } \phi = 5 \tag{58}$$

For SFC case, according to Fig. 9 and Fig. 11, $p_{x_j} = \{-p_{2j}, -p_{1j}, p_{0j}, p_{1j}, p_{2j}\}$ is fuzzy partition of the input universe of discourse and $p_y = \{-\bar{y}_2, -\bar{y}_1, \bar{y}_0, \bar{y}_1, \bar{y}_2\}$ is for output universe of discourse. Now, SFC design parameters are given by following equations (Santibanez et al., 2005):

$$\begin{aligned}
 p_{\dot{q}_1} &= \{-180, -4, 0, 4, 180\} \\
 p_{\dot{q}_2} &= \{-180, -2, 0, 2, 180\} \\
 p_{\ddot{q}_1} &= \{-360, -270, 0, 270, 360\} \\
 p_{\ddot{q}_2} &= \{-360, -270, 0, 270, 360\} \\
 p_{y_1} &= \{-109, -90, 0, 90, 109\} \\
 p_{y_2} &= \{-13, -9, 0, 9, 13\}
 \end{aligned} \tag{59}$$

For our proposed controller (54), the constant $\alpha = 0.3$ is supposed. Additionally, to show the improvement achieved from applying the proposed method of this section (incorporating

SMC and SFC), the simulation results of applying this method are compared with the related results of the SMC case and SFC case, separately. The error vector and control law in the case of conventional SMC have been shown in Fig. 12 and Fig. 13, respectively.

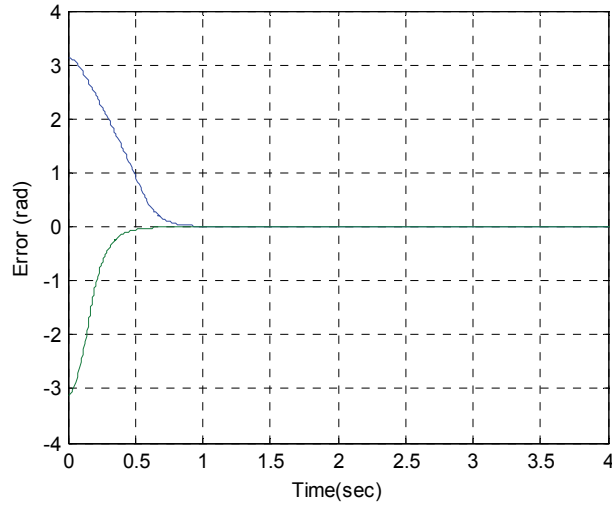


Fig. 12. Error vector in the case of SMC

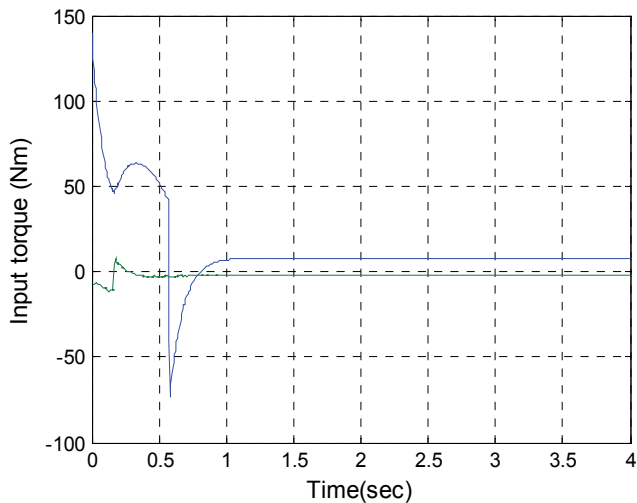


Fig. 13. The control torques in the case of SMC

The tracking error in this case is about 0.1(rad) and when one choose the thinner boundary layer to decrease this error, chattering will be occurred. The corresponding graphs for the case of applying SFC are also provided in Fig. 14, and Fig. 15.

In the case of control law proposed in the present section, Fig. 16 and Fig. 17 illustrate the error vector and control law, respectively. The tracking error is about 0.002 in this state of affairs.

As it can be seen from these results, the proposed incorporating SMC and SFC controller has faster response and less tracking error in comparison with SMC and also the error vector converges toward zero faster than SFC.

In order to show the robustness of the proposed method, the inertia and torque perturbations are considered as following. The elements of inertia matrix are supposed to increase fifty percent after 2 sec. It can be a weight that added to the mass of 2nd link. Also, disturbance torque is considered with the following equation.

$$\tau_d = [3 \sin 2\pi t \quad 3 \sin 2\pi]^T \quad (60)$$

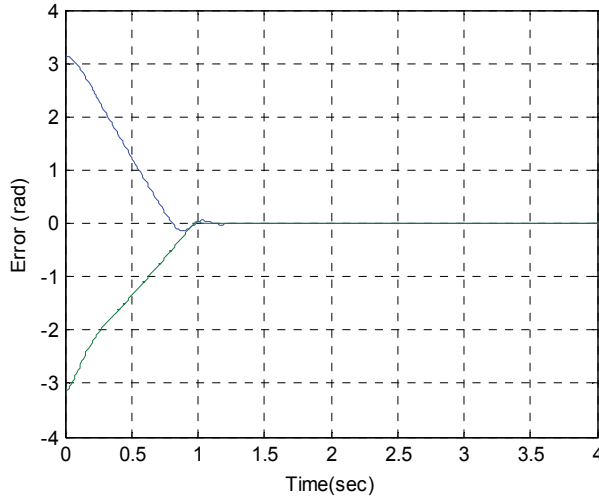


Fig. 14. Error vector in the case of SFC

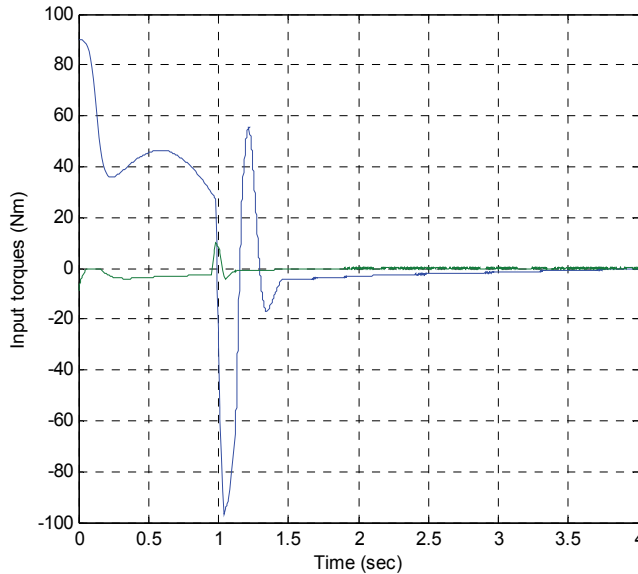


Fig. 15. The control torques in the case of SFC

In this case, the vector of joint errors is shown in Fig. 18. The errors are as good as previous case. Fig. 19 illustrates the control torques which are not change significantly, and because of existing perturbations, they alter trivially after 2 sec. these two recent results verify the robustness of the presented approach.

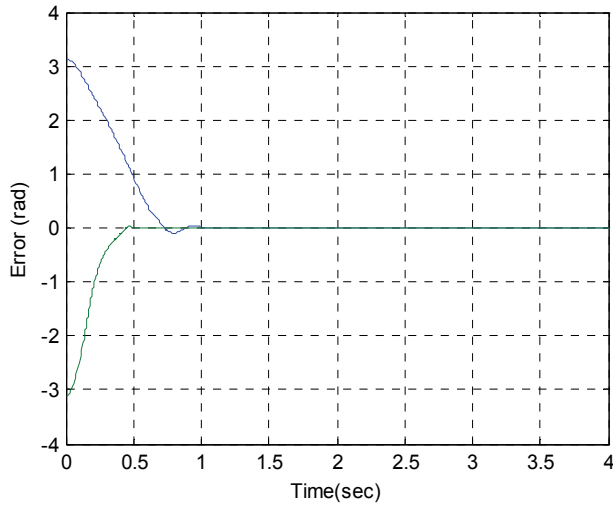


Fig. 16. Error vector in the case of incorporating SMC and SFC

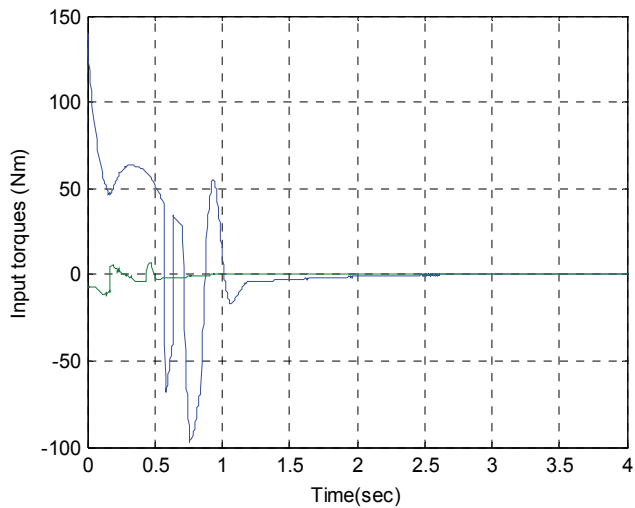


Fig. 17. The control torques in the case of incorporating SMC and SFC

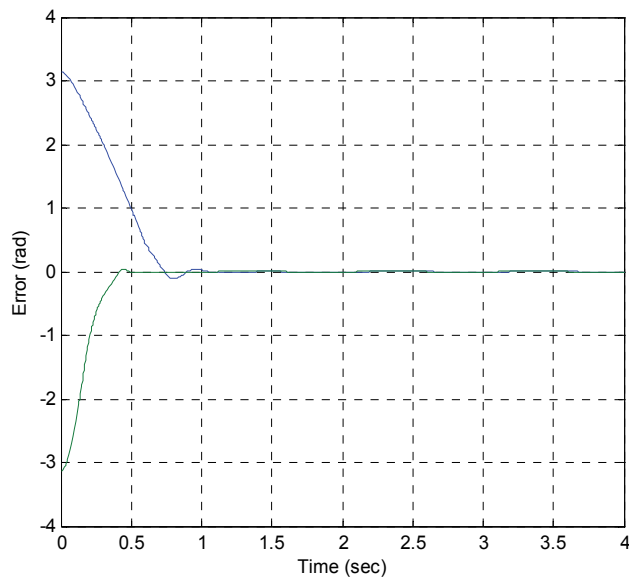


Fig. 18. Error vector in the case of torque and inertia perturbations

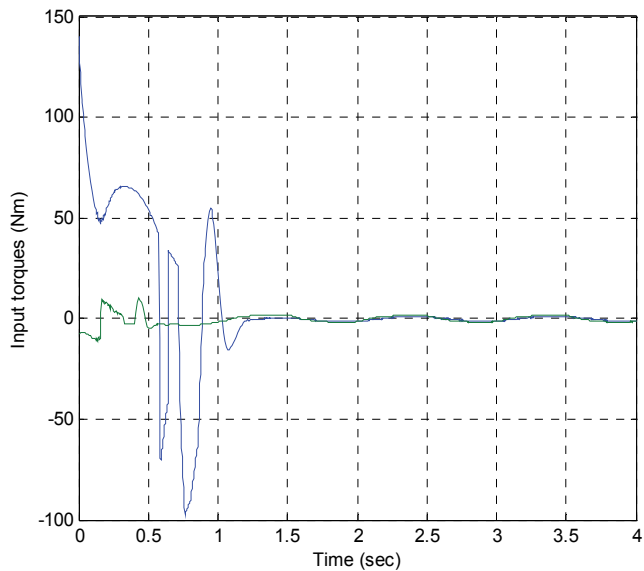


Fig. 19. The control torques in the case of torque and inertia perturbations

3. Sliding mode control using neural network approach

Sliding-Mode-PID control for robot manipulator was explored by (Ataei & Shafiei, 2008). In their study, although, the uncertainties are considered but controller design is extremely model-dependent. Also, control command starts with high gain and actuator dynamics is neglected. Moreover, stability analysis is not investigated after incorporating fuzzy tuning

system. A robust neural-fuzzy-network controller was designed in (Wai & Chen 2006) for the position control of an n-link robot manipulator including actuator dynamics. Although, their control scheme does not require compensating auxiliary control design, but the employed network is more complicated and uses excess number of neurons. In addition, the second derivative of position angle is required as a part of controller inputs. Capisani et al., (Capisani et al., 2009) presented an inverse dynamic-based second-order sliding mode controller to perform motion control of robot manipulators, but this method involves the higher order derivatives of the state variables.

In this section, the motion tracking control of multiple-link robot manipulators actuated by permanent magnet DC motors is addressed. Sliding-mode-PID tracking controller is designed such that all the states and signals of the closed loop system remain bounded in the presence of unknown parameters and uncertainties. Also, neural network universal approximation property is employed for compensating uncertainties. Furthermore, the proposed controller contains an outer PID-loop that enhances the approximation performance during the initial period of weight adaptations, and provides designing a simple NN with lower amount of layers and neurons. Adaptation laws are applied to adjust the NN weights on-line. In order to avoid high gain control, the gain factor of robustifying term is designed adaptively (Shafiei & Soltanpour, 2010).

3.1 Actuated robot dynamics

The mathematical equations describing electrical and mechanical dynamics of a permanent magnet DC motor are as follows (Spong & Vidiyasagar, 1989):

$$V = Ri + L \frac{di}{dt} + K_b \frac{d\theta}{dt} \quad (61)$$

$$J_m \ddot{\theta} + B_m \dot{\theta} + \tau_m = \tau \quad (62)$$

$$\tau = K_m i \quad (63)$$

where V is the armature voltage of the motor, R and L are armature equivalent resistance and inductance, respectively, K_b is the back electromotive force constant, i is the armature current and θ denotes the rotor position, J_m is the total moment of inertia, B_m is the damping coefficient, τ_m and τ represent the generated motor torque and the load torque, respectively, and K_m is the diagonal matrix of motor torque constant.

The dynamical equation of an n-link robot manipulator is in the standard form of (30) and is rewritten here.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \quad (64)$$

Here, $F(\dot{q}) \in R^n$ is the dynamic friction vector, $\tau_d \in R^n$ denotes the vector of disturbance and un-modeled dynamics, and τ is the torque vector.

With the purpose of increasing motion speed of the manipulators, motors are equipped with the high reduction gears as follows:

$$q = g_r \theta \quad (65)$$

and

$$\tau_m = g_r \tau \quad (66)$$

where g_r is the diagonal matrix of reduction ratio. In the following a practical constraint is considered.

Constraint 3.1. The maximum voltage that joint actuator can supply is V^{\max} . So, we have:

$$|V_i| \leq V_i^{\max}, \quad i = 1, \dots, n$$

It should be noted that, the applicable control input for driving robot arm is the armature voltage of the motors, here. So, by using equations (61)-(66) and neglecting the inductance L , because of its tiny amount, the following equation is achieved.

$$V = RK_m^{-1} \{ [J_m g_r^{-1} + g_r M] \ddot{q} + (B_m g_r^{-1} + g_r C + K_m R^{-1} K_b g_r^{-1}) \dot{q} + g_r G + g_r F(\dot{q}) + g_r \tau_d \} \quad (67)$$

The previous equation can be expressed in a compact form as:

$$U = D \ddot{q} + H + d \quad (68)$$

with $U = V$ is the control command and the other parameters are

$$D = RK_m^{-1} (J_m g_r^{-1} + g_r M) \quad (69)$$

$$H = V_m + RK_m^{-1} [(B_m g_r^{-1} + K_m R^{-1} K_b g_r^{-1}) \dot{q} + g_r G(q)] \quad (70)$$

$$V_m = RK_m^{-1} g_r C(q, \dot{q}) \quad (71)$$

$$d = RK_m^{-1} g_r (F(\dot{q}) + \tau_d) \quad (72)$$

Remark 3.1. By noting that the parameters, R , K_m , J_m and g_r are positive definite diagonal matrices, the matrix D is symmetric and positive definite.

Remark 3.2. From relations (69) and (71), and property 2.2, the matrix $(\dot{D} - 2V_m)$ is skew-symmetric too.

3.2 SMC- PID design and NN description

The tracking error could be defined as before as:

$$e = q_d - q \quad (73)$$

A key step in designing sliding mode controller is to introduce a proper sliding surface so that tracking errors and output deviations can be reduced to a satisfactory level (Eker, 2006). Accordingly, the sliding surface is considered as (74), containing the integral part in addition to the derivative term.

$$s = \dot{e} + \lambda_1 e + \lambda_2 \int_0^t e dt \quad (74)$$

where λ_i is diagonal positive definite matrix. Hence, $s = 0$ is a stable sliding surface and $e \rightarrow 0$ as $t \rightarrow \infty$. Only defining the sliding surface as (74) is not adequate to claim that SMC-PID is designed, but the control effort must contain the independent PID part. For this purpose, the robot dynamic equations can be rewritten based on the sliding surface (in term of filtered error) as follows:

$$D\dot{s} = -V_m s + f - U \quad (75)$$

where

$$f(x) = D(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) + V_m s + H + d \quad (76)$$

where D , V_m and d are given by (69), (71) and (72) respectively, and

$$x = \begin{bmatrix} \ddot{q}_d^T & s^T & \dot{q}^T \end{bmatrix}^T \quad (77)$$

Note that the input vector of s includes linear combination of e and \dot{e} , (i.e. $\dot{e} + \lambda_1 e$) which they comprise q_d , q and \dot{q}_d , \dot{q} , too, respectively. The input dimension of the two-layer NN designed here is less than that of given by (Lewis et al., 1996), and thus the proposed method is more desirable from an implementation point of view. Sliding mode control strategy consists of designing a two-part controller.

$$U_{SMC} = U_{eq} + U_s \quad (78)$$

with U_{eq} is equivalent control part which is applied to cancel the uncertain nonlinear function f , and U_s specifies robust control term. Considering unknown parameter, uncertainties and disturbances indicates that the function f is not accessible. Briefly speaking, neural networks incorporate to reconstruct the U_{eq} part by approximating the function f , here. According to universal approximation property of neural networks (Lewis et al., 1998), there is a two-layer NN with sufficient number of neurons, and sigmoid or RBF activation function for hidden layer and linear activation function for output layer (see Fig. 20) such that:

$$f(x) = W^T \sigma(V^T x) + \varepsilon \quad (79)$$

where $x \in R^{N_2}$ is the input vector computed by (77), $V \in R^{N_2 \times N_2}$ and $W \in R^{N_2 \times N_2}$ represents the NN weights for hidden and output layers, respectively, $\sigma(\cdot)$ denotes activation function of the hidden layer and ε is NN approximation error. Choosing activation function is arbitrary provided that the function satisfies an approximation property and it and its derivative are bounded (Lewis et al., 1998), consequently the sigmoid activation function is considered, here.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (80)$$

Succeeding section explains complete controller design and investigates stability content.

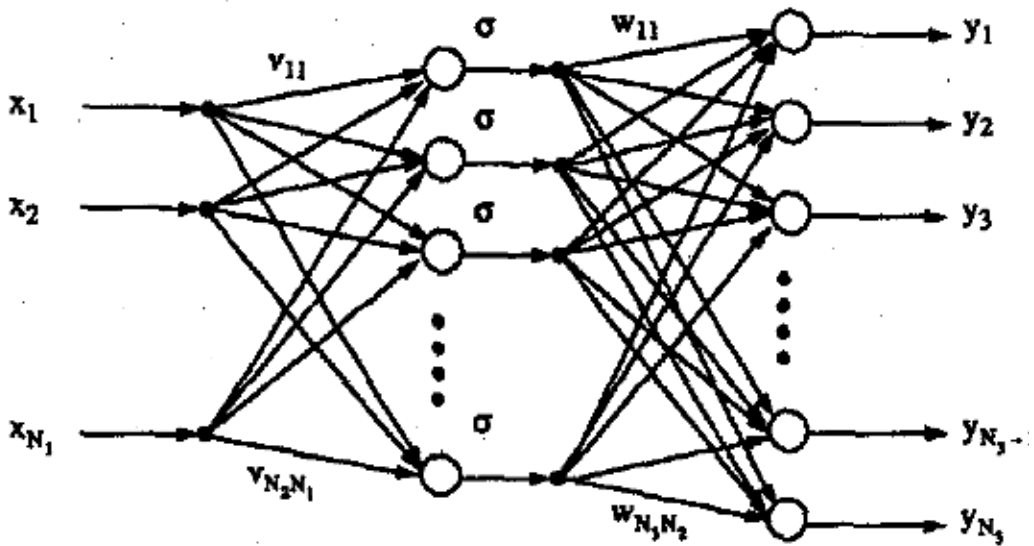


Fig. 20. Two-layer NN structure

3.3 Sliding mode control using adaptive neural network

Note that the utilized weights in (79) are optimum and $f(x)$ is approximated ideally, over there. Estimation of f is accomplished by the estimated weights \hat{W} and \hat{V} , respectively. So, the NN controller is designed as:

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \quad (81)$$

here $\hat{f}(x)$ is estimation of $f(x)$ and \hat{W} and \hat{V} are updated adaptively. The estimation errors are defined as follows:

$$\tilde{V} = V - \hat{V}, \quad \tilde{W} = W - \hat{W} \quad (82)$$

also, the hidden layer output error for a given input x is

$$\tilde{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x) = \sigma - \hat{\sigma} \quad (83)$$

Consider the $\sigma(V^T x)$ as its Taylor series expansion as

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x) \tilde{V}^T x + O_h(\tilde{V}^T x) \quad (84)$$

where $O_h(\cdot)$ denotes higher order terms in Taylor series and

$$\sigma'(z) \equiv \left. \frac{d\sigma(z)}{dz} \right|_{z=z} \quad (85)$$

From (83) and (84), we have:

$$\tilde{\sigma} = \sigma'(\hat{V}^T x) \tilde{V}^T x + O_h(\tilde{V}^T x) = \hat{\sigma}' \tilde{V}^T x + O_h \quad (86)$$

Now, one can obtain overall error between optimum function f and its estimation \hat{f} as:

$$\begin{aligned} f - \hat{f} &= W^T \sigma(V^T x) + \varepsilon - \hat{W}^T \sigma(\hat{V}^T x) = \tilde{W}^T \sigma(V^T x) + \hat{W}^T \tilde{\sigma}(\hat{V}^T x) + \varepsilon \\ &= \tilde{W}^T [\sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x) \tilde{V}^T x + O_h] + \hat{W}^T [\sigma'(\hat{V}^T x) \tilde{V}^T x + O_h] + \varepsilon \\ &= \tilde{W}^T \sigma(\hat{V}^T x) - \tilde{W}^T \sigma'(\hat{V}^T x) \hat{V}^T x + \hat{W}^T \hat{\sigma}' \tilde{V}^T x + \tilde{W}^T \hat{\sigma}' V^T x + W^T O_h + \varepsilon \\ &= \tilde{W}^T \sigma(\hat{V}^T x) - \tilde{W}^T \sigma'(\hat{V}^T x) \hat{V}^T x + \hat{W}^T \hat{\sigma}' \tilde{V}^T x + \varepsilon_N \end{aligned} \quad (87)$$

where

$$\varepsilon_N = \tilde{W}^T \hat{\sigma}' V^T x + W^T O_h + \varepsilon \quad (88)$$

is the uncertain term and is supposed to be bounded by K as demonstrated in (89).

$$\|\varepsilon_N\| \leq \|\tilde{W}^T \hat{\sigma}' V^T x\| + \|W^T O_h\| + \|\varepsilon\| < K \quad (89)$$

Design of the control system is provided in the following theorem and is illustrated in Fig. 21 schematically.

Theorem 3.1. Robot manipulator including actuator dynamics represented by equation (68) is considered, and the sliding surface is defined by (74). If the control input U is designed as (90) together with adaptation laws of NN controller as (91)-(93), then the asymptotic stability of the dynamical system is guaranteed.

$$U = K_v s + \hat{f} + \hat{K} \operatorname{sgn}(s) \quad (90)$$

$$\dot{\hat{W}} = \alpha \sigma(\hat{V}^T x) s^T - \alpha \hat{\sigma}' \hat{V}^T x s^T \quad (91)$$

$$\dot{\hat{V}} = \beta x s^T \hat{W}^T \hat{\sigma}' \quad (92)$$

$$\dot{\hat{K}} = \gamma s^T \operatorname{sgn}(s) \quad (93)$$

where K_v is a positive definite diagonal matrix, \hat{K} is the estimated value of K . Also, α , β and γ are positive constants and $\operatorname{sgn}(\cdot)$ denotes sign function.

Proof: consider the following Lyapunov function candidate

$$V_L = \frac{1}{2} s^T D s + \frac{1}{2\alpha} \operatorname{tr}(\tilde{W}^T \tilde{W}) + \frac{1}{2\beta} \operatorname{tr}(\tilde{V}^T \tilde{V}) + \frac{1}{2\gamma} \tilde{K}^T \tilde{K} \quad (94)$$

where $\operatorname{tr}(\cdot)$ denotes the trace operator and $\tilde{K} = K - \hat{K}$. Differentiating of the relation (94) gives

$$\dot{V}_L = s^T D \dot{s} + \frac{1}{2} s^T \dot{D} s + \frac{1}{\alpha} \operatorname{tr}(\tilde{W}^T \dot{\tilde{W}}) + \frac{1}{\beta} \operatorname{tr}(\tilde{V}^T \dot{\tilde{V}}) + \frac{1}{\gamma} \tilde{K}^T \dot{\tilde{K}} \quad (95)$$

By substituting (90) in to the first part of (95) and by using (87) one can obtain

$$\begin{aligned} S^T D \dot{s} &= s^T [-V_m s + f - U] = s^T [-V_m s + f - K_v s - \hat{f} - \hat{K} \operatorname{sgn}(s)] \\ &= s^T [-V_m s - K_v s + \tilde{W}^T \hat{\sigma} - \tilde{W}^T \hat{\sigma}' \hat{V}^T x + \hat{W}^T \hat{\sigma}' \tilde{V}^T x + \varepsilon_N - \hat{K} \operatorname{sgn}(s)] \end{aligned} \quad (96)$$

Some useful relations for manipulating last tow equations are provided in the following.

$$\begin{cases} s^T \tilde{W}^T \hat{\sigma} = \operatorname{tr}(\tilde{W}^T \hat{\sigma} s^T) \\ s^T \tilde{W}^T \hat{\sigma}' \hat{V}^T x = \operatorname{tr}(\tilde{W}^T \hat{\sigma}' \hat{V}^T x s^T) \\ s^T \hat{W}^T \hat{\sigma}' \tilde{V}^T x = \operatorname{tr}(\tilde{V}^T x s^T \hat{W}^T \hat{\sigma}') \end{cases}$$

Replacing (96) in (95) and using above relations, produce

$$\begin{aligned} \dot{V}_L &= -s^T K_v s + \frac{1}{2} s^T (\dot{D} - 2V_m) s + \operatorname{tr} \left[\tilde{W}^T \left(\frac{1}{\alpha} \dot{\tilde{W}} + \hat{\sigma} s^T - \hat{\sigma}' \hat{V}^T x s^T \right) \right] \\ &\quad + \operatorname{tr} \left[\tilde{V}^T \left(\frac{1}{\beta} \dot{\tilde{V}} + x s^T \hat{W}^T \hat{\sigma}' \right) \right] + s^T \varepsilon_N - \hat{K} s^T \operatorname{sgn}(s) + \frac{1}{\gamma} \dot{\hat{K}} \end{aligned} \quad (97)$$

Note that $\dot{\tilde{W}} = -\dot{\hat{W}}$, $\dot{\tilde{V}} = -\dot{\hat{V}}$, $\dot{\hat{K}} = -\dot{\tilde{K}}$, and Remark 3.2 yields $s^T (\dot{D} - 2V_m) s = 0$. Also, if adaptive laws (91) and (92) are taken in to account, then we have

$$\dot{V}_L = -s^T K_v s + s^T \varepsilon_N - \hat{K} s^T \operatorname{sgn}(s) - \frac{1}{\gamma} (K - \hat{K}) \dot{\hat{K}} = -s^T K_v s + s^T \varepsilon_N - K s^T \operatorname{sgn}(s) \quad (98)$$

substituting (93) in (98) and adopting (99), yields

$$\dot{V}_L \leq -K_{v_{\min}} \|s\|^2 + \|\varepsilon_N\| (\|s_1\| + \|s_2\| + \dots + \|s_m\|) - K \sum_{i=1}^m \|s_i\| \leq -K_{v_{\min}} \|s\|^2 \leq 0 \quad (99)$$

where $K_{v_{\min}}$ is minimum singular value of K_v . Since $\dot{V}_L \leq 0$, the stability in the sense of Lyapunov is guaranteed which implies that the parameters s , \tilde{W} , \tilde{V} and \tilde{K} (and consequently \hat{W} , \hat{V} , \hat{K}) are bounded. In addition, $\lim_{t \rightarrow \infty} \int_0^t -\dot{V}_L d\tau < \infty$ and $-\dot{V}_L$ is bounded, hence Barbalat's Lemma (Khalil, 2001) indicates that $\lim_{t \rightarrow \infty} (-\dot{V}_L) = 0$. Note that $(-\dot{V}_L) \geq K_{v_{\min}} \|s\|^2 \geq 0$, as a result $s \rightarrow 0$ as $t \rightarrow \infty$. Therefore, the proposed control system is asymptotically stable.

Remark 3.3. The PID term in the above control effort, makes Lyapunov derivative more negative, so it makes the transient response faster and also ensures the performance efficiency during the initial period of weights adaptations.

Remark 3.4. In practical systems, however, it is impossible to achieve infinitely fast switching control, because of finite time delays for the control computation and limitation of physical actuators. For that reason, the sign function is replaced by saturation function here, and the stability matter is investigated analytically.

The saturation function is selected as

$$\text{sat}\left(\frac{s}{\varphi}\right) = \begin{cases} \frac{s}{\varphi} & \|s\| \leq \varphi \\ \text{sgn}\left(\frac{s}{\varphi}\right) & \|s\| \geq \varphi \end{cases} \quad (100)$$

where φ is a thin boundary layer such that $0 < \varphi \leq 1$. The adaptive law (93) must be replaced by $\dot{\hat{K}} = \gamma s^T \text{sat}(s/\varphi)$; So, the equation (98) is changed to

$$\dot{V}_L = -s^T K_\rho s + s^T \varepsilon_N - K s^T \text{sat}(s/\varphi) \quad (101)$$

Now, there are two situations;

a. if $\|s\| > \varphi$, then

$$\dot{V}_L \leq -K_{v_{\min}} \|s\|^2 + \|\varepsilon_N\| (\|s_1\| + \|s_2\| + \dots + \|s_m\|) - K \sum_{i=1}^m \|s_i\| < 0 \quad (102)$$

b. if $\|s\| \leq \varphi$, then

$$\dot{V}_L \leq -K_{v_{\min}} \|s\|^2 + \|\varepsilon_N\| (\|s_1\| + \|s_2\| + \dots + \|s_m\|) - \frac{K}{\phi} \sum_{i=1}^m \|s_i\| < 0 \quad (103)$$

Note that, since $0 < \varphi < 1$, therefore $\frac{K}{\phi} \geq K > \|\varepsilon_N\|$. Both situations imply that $\dot{V}_L < 0$, and consequently, the control system remains stable after replacing saturation function.

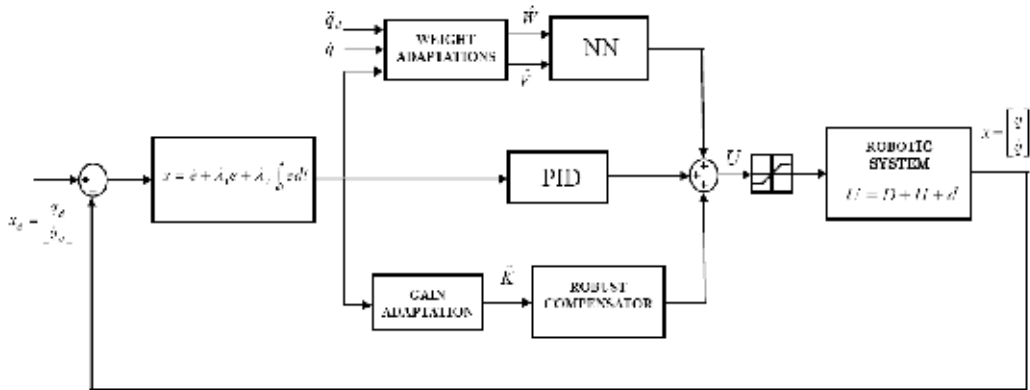


Fig. 21. Block diagram of the control system structure

Remark 3.5. The sliding gain \hat{K} is chosen dynamically and its dynamic depends on sliding surface. When the states go far from the sliding manifold, the absolute value of \hat{K} increases to force them back to sliding manifold, and when the states are close to the sliding manifold, the absolute value of \hat{K} decreases accordingly. This feature beside the replacing saturation function, act as what is heuristically designed by fuzzy system in (Ataei & Shafiei, 2008). Furthermore, the system stability is addressed here.

Simulation example 3.1. In order to show the effectiveness of the proposed control method, it is applied to a two-link elbow robot driven by permanent magnet DC motors with the following parameters:

$$\begin{aligned}
 M(q) &= \begin{bmatrix} m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) & m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) \\ m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) & m_2 l_{c2}^2 \end{bmatrix} \\
 C(q, \dot{q}) &= \begin{bmatrix} -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2 & -m_2 l_1 l_{c2} \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \\
 G(q) &= \begin{bmatrix} m_1 g l_{c1} \cos q_1 + m_2 g l_1 \cos q_1 + m_2 g l_{c2} \cos(q_1 + q_2) \\ m_2 g l_{c2} \cos(q_1 + q_2) \end{bmatrix}
 \end{aligned} \tag{104}$$

where q_i is the angle of joint i , m_i is the mass of link i , l_i is the total length of link i , l_{ci} is center-of-gravity length of link i , and $g = 9.8 \text{ m/s}^2$ is gravity acceleration. The detailed parameters of this robot manipulator and permanent magnet DC motor actuators are provided in Table 3 (Wai & Chen, 2006). According to the actuator manufacturer, the DC motors are able to accept input voltages within the following bounds:

$$V_1 \leq V_1^{\max} = 12 [\text{volt}], \quad V_2 \leq V_2^{\max} = 12 [\text{volt}] \tag{105}$$

For example, one can use 12V DC servo motors for actuating joints. In practice, also, a servo control card is required which should include multi-channels of digital/analog (D/A) and encoder interface circuits.

Two-link elbow robot		Permanent-magnet DC motors	
$m_1 = 3.55 \text{ kg}$	$m_2 = 0.75 \text{ kg}$	$J_{m1} = 3.7 \times 10^{-5} \text{ kg.m}^2$	$J_{m2} = 1.47 \times 10^{-4} \text{ kg.m}^2$
$l_1 = 205 \text{ mm}$	$l_2 = 210 \text{ mm}$	$B_{m1} = 1.3 \times 10^{-5} \text{ N.m/s}$	$B_{m2} = 2 \times 10^{-5} \text{ N.m/s}$
$l_{c1} = 154.8 \text{ mm}$	$l_{c2} = 105 \text{ mm}$	$R_1 = 2.8 \ \Omega$	$R_2 = 4.8 \ \Omega$
$K_{m1} = 0.21 \text{ Nm/A}$	$K_{m2} = 0.23 \text{ Nm/A}$	$L_1 = 3 \text{ mH}$	$L_2 = 2.4 \text{ mH}$
$g_{r1} = 1/60$	$g_{r2} = 1/30$	$K_{b1} = 2.42 \times 10^{-4} \text{ s/rad.V}$	$K_{b2} = 2.18 \times 10^{-4} \text{ s/rad.V}$

Table 3. Parameters of two-link elbow robot and actuators

The external disturbances can be considered as external forces injected into the robotic system, and are supposed to have following expression.

$$\tau_d = [\sin 4t \quad \sin 4t]^T \tag{106}$$

Also, the friction term is considered here as (Wai & Chen, 2006):

$$F(\dot{q}) = [20\dot{q}_1 + 0.8 \text{sgn}(\dot{q}_1) \quad 4\dot{q}_2 + 0.16 \text{sgn}(\dot{q}_2)]^T \tag{107}$$

In order to show the effectiveness of proposed controller in tracking of desired trajectory, it is assumed to have the sinusoidal shape in this simulation.

$$q_d = [\sin t \quad \sin t]^T \tag{108}$$

The design parameters are given in Table 4. The gain matrices λ_1 and λ_2 are selected such that the roots of the characteristic polynomial $\ddot{e} + \lambda_1 \dot{e} + \lambda_2 e = 0$ lie strictly in the open left half

of the complex plane when the system is in sliding mode ($\dot{s} = 0$). The neural network designed here has four neurons as hidden layer and two neurons as output layer, and its weights are totally initialized at zero.

Remark 3.6. For a two-layer NN designed here with the input vector given by (77), we have $N_1 = 6$, $N_2 = 4$ and $N_3 = 2$, for a two-link manipulator. Accordingly, the numbers of adaptive weights are 24 and 8 for input-to-hidden layer weights and output layer weights, respectively. So, only 32 weight parameters must be adaptively updated here while using the NN given in (Lewis et al., 1996), with $N_1 = 10$, $N_2 = 10$ and $N_3 = 2$, this number increases to 120. If the network size is chosen to large, the improvement of control performance is limited and the computation burden for the CPU is significantly increased.

The gain matrix K_v which acts as the gain of the PID term is determined large enough to improve transient response in the initial period of weight adaptations. On the other hand, choosing K_v to a large extent increases the overall controller gain and may exceed the permissible voltages of the actuators that are regarded in constraint 3.1. So, there is a trade off between fast response and practical limitations.

$\lambda_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	$\lambda_2 = \begin{bmatrix} 24 & 0 \\ 0 & 24 \end{bmatrix}$	$g_r = \begin{bmatrix} 1/60 & 0 \\ 0 & 1/30 \end{bmatrix}$	$K_v = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$
$\alpha = 5$	$\beta = 5$	$\gamma = 2$	$\varphi = 0.05$

Table 4. Design parameters

The mass variation of second link, the external disturbance and the friction are the major factors that affect the control performance of the robotic system. In the reminder of this section, two simulation cases are carried out to show the improvement due to the NNSM_PID control method proposed in this section. In both cases, the simulation results of applying presented method are compared with the related results of the fuzzy sliding mode_PID (FSM_PID) control method proposed in (Ataei & Shafiei, 2008). In the first case, the disturbance (106) and mass variation are injected and in the second case, the friction term is exerted too. The mass variation condition is that 1 kg weight is added to the mass of 2nd link (i.e. $m_2 = 1.75$ kg). For the FSM_PID case, the control law is as following (Ataei & Shafiei, 2008):

$$U_f = K_{vf}s + \hat{f}_f + K_f \operatorname{sgn}(s) \quad (109)$$

$$K_{vf} = N_{vf} K_{fuzzy} \quad (110)$$

$$K_f = N_f K_{fuzzy} \quad (111)$$

where, U_f is the control input, K_{fuzzy} is of fuzzy system output and N_{vf} and N_f are the scaling gain of the fuzzy system output. Here, it is assumed that only manipulator parameters could be estimated and actuator parameters are still unknown. So, f_f is chosen as (Ataei & Shafiei, 2008):

$$\hat{f}_f = \hat{M}(\ddot{q}_d + \lambda_1 \dot{e} + \lambda_2 e) + \hat{C}(\dot{q}_d + \lambda_1 e + \lambda_2 \int_0^t e dt) + \hat{G} \quad (112)$$

where \hat{M} , \hat{C} and \hat{G} are achieved from nominal value of manipulator parameters. However, all of the manipulator parameters are considered with 10% uncertainty. The design parameters of the FSM_PID controller are

$$N_{vf} = \begin{bmatrix} 3.2 & 0 \\ 0 & 3.5 \end{bmatrix}, N_f = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.7 \end{bmatrix} \quad (113)$$

Simulation 1— In this case, the friction term is neglected, mass variation occurs at 3 sec and external disturbance is injected at 6 sec. The desired trajectory is depicted in Fig. 22. The vectors of tracking errors of FSM_PID and NNSM_PID are shown in Fig. 23 (a) and (b), respectively. Both diagrams of Fig. 23 are plotted in the same scaled axes to achieve fairly comparison. The FSM_PID controller does not meet the tracking purpose in the unknown actuator parameters and mass variation conditions. On the contrary, the method proposed in this section provides swift and precise tracking responses. Fig. 24 displays the control efforts (i.e. input armature voltages of motors). The FSM_PID associated control commands are jagged to some extent, while, the NNSM_PID case produces smooth control commands with slowly variation and lower voltage amplitude. Lower voltage commands are more protected toward actuator saturations. The NN outputs are shown in Fig. 25 and it indicates that the designed neural network can approximate nonlinear terms with unknown parameters, smoothly and boundedly.

Simulation 2— With the purpose of showing robustness of our designed controller against uncertainties and un-modeled dynamics, the friction term (107) is added here. The vectors of tracking errors of FSM_PID and NNSM_PID are shown in Fig. 26 (a) and (b), respectively. However, the response of the FSM_PID case is further undesirable in this condition, on the other hand, the NNSM_PID control remains robust and its response is satisfactory, as well as previous simulation case. Control efforts of this case are demonstrated in Fig. 27. Because of exerting friction term, the input voltage commands are higher than previous case but the NNSM_PID control commands are still smooth and vary slowly. The NN output is shown in Fig. 28. Finally, as can be seen from Fig. 29, matrix norm of the adaptive weights, $\|\hat{W}\|$ and $\|\hat{V}\|$, have bounded value, less than 3, that it verifies what was claimed in the Theorem 3.1 about boundedness of these signals.

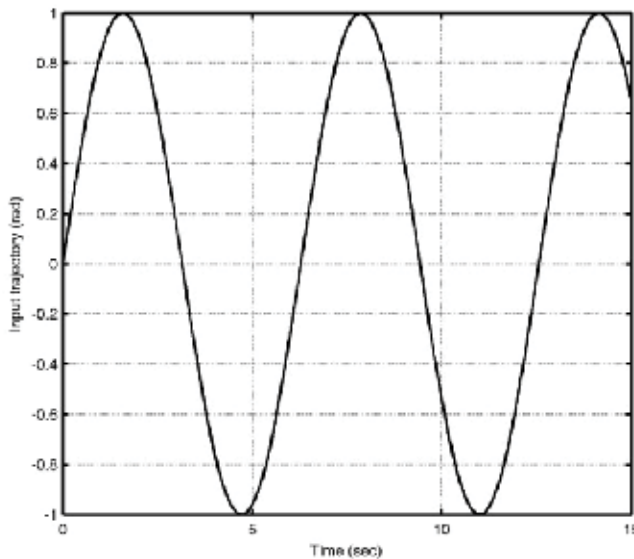
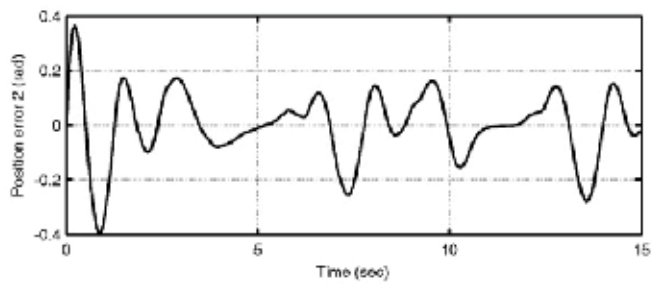
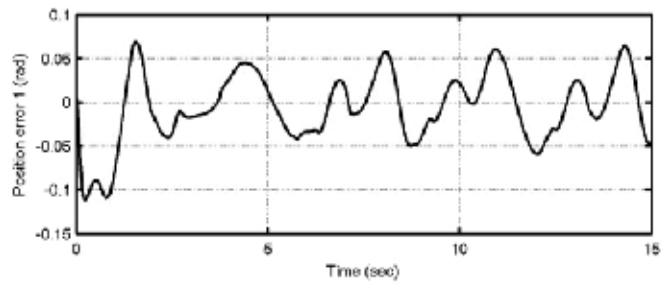
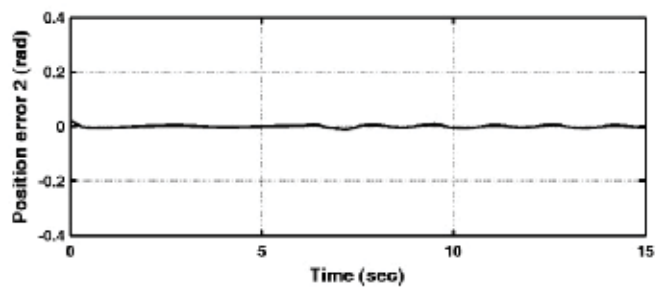
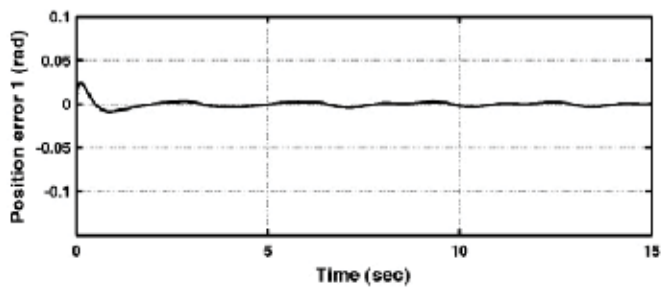


Fig. 22. Desired input trajectory q_d

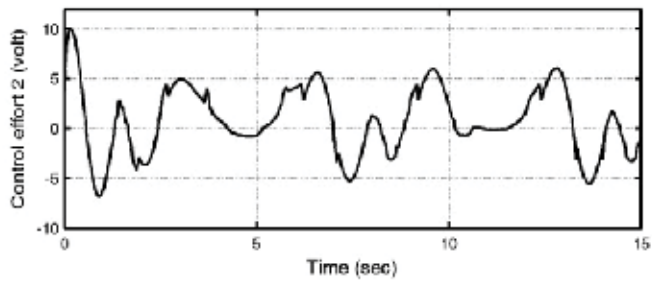
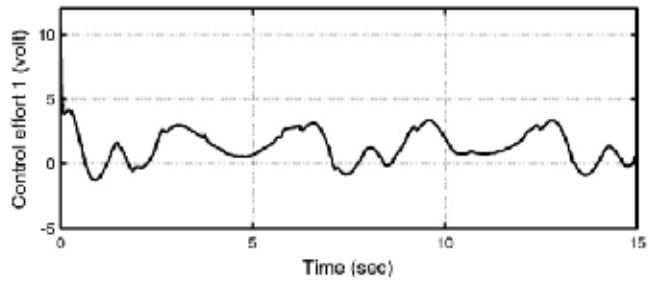


(a)

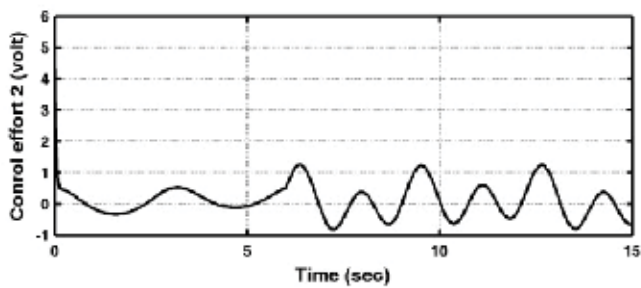
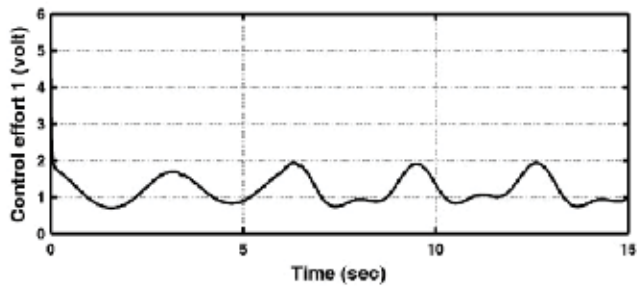


(b)

Fig. 23. (sim1) Tracking error of joints, (a) FSM_PID (b) NNSM_PID



(a)



(b)

Fig. 24. (sim1) Control commands (a) FSM_PID (b) NNSM_PID

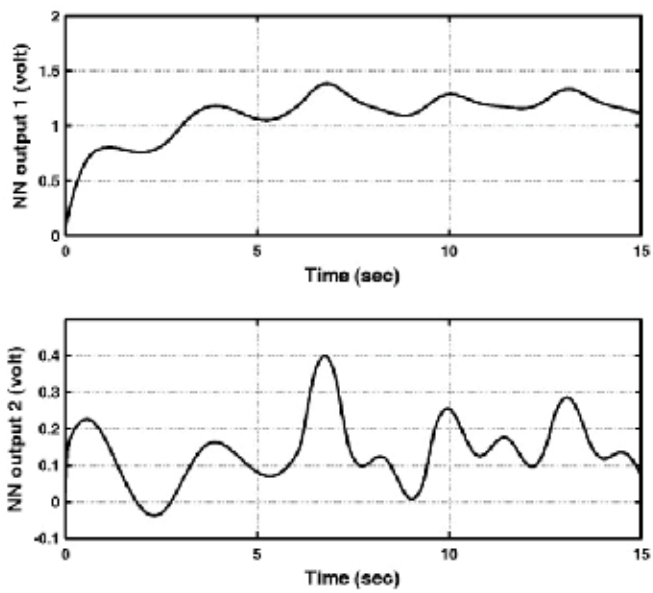
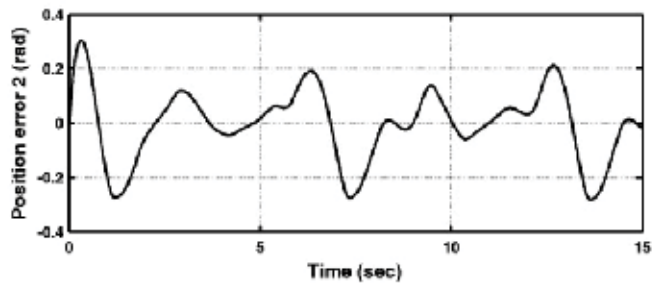
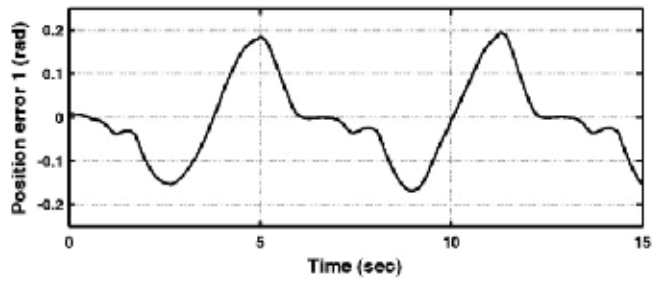
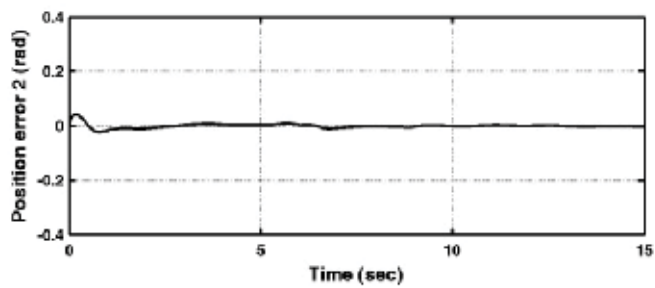
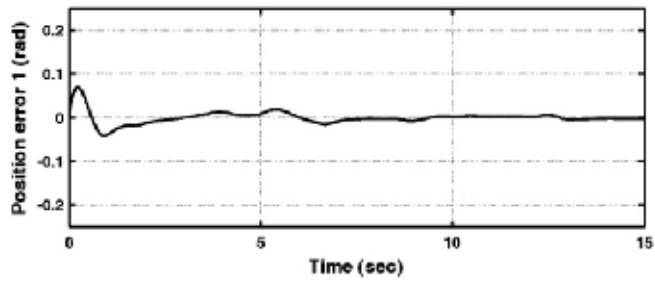


Fig. 25. (sim1) NN control effort

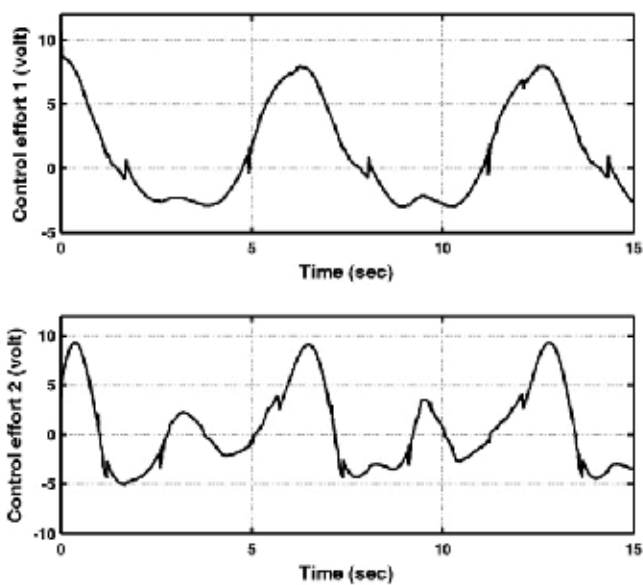


(a)

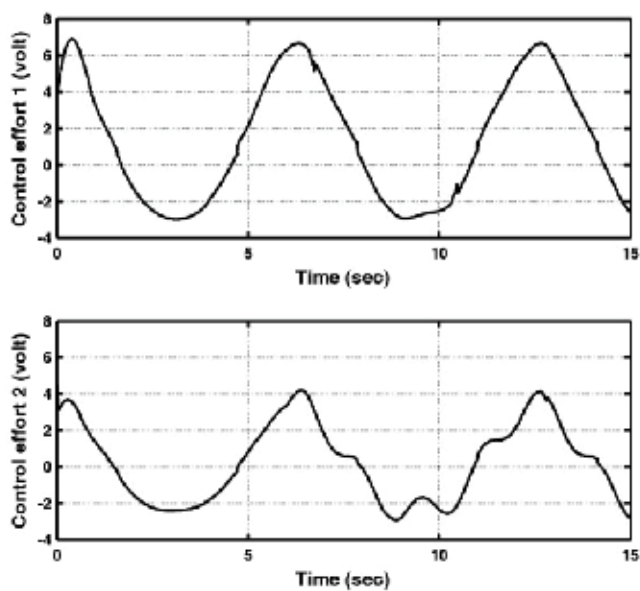


(b)

Fig. 26. (sim2) Tracking error of joints (a) FSM_PID (b) NNSM_PID



(a)



(b)

Fig. 27. (sim2) Control commands (a) FSM_PID (b) NNSM_PID

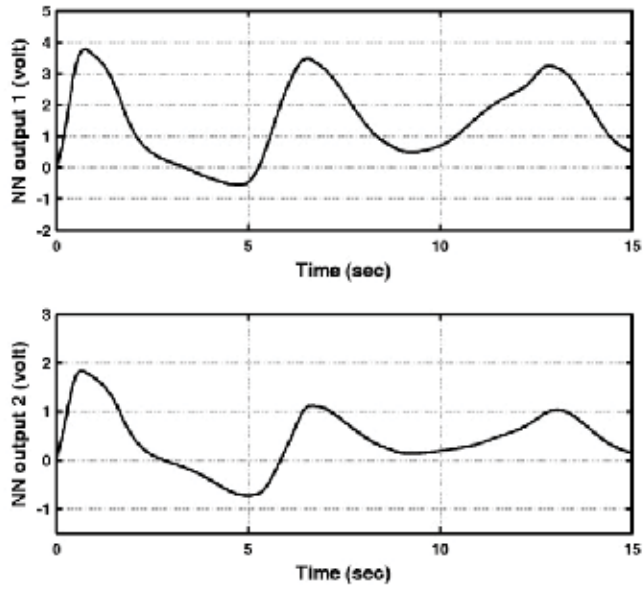


Fig. 28. (sim2) NN control effort

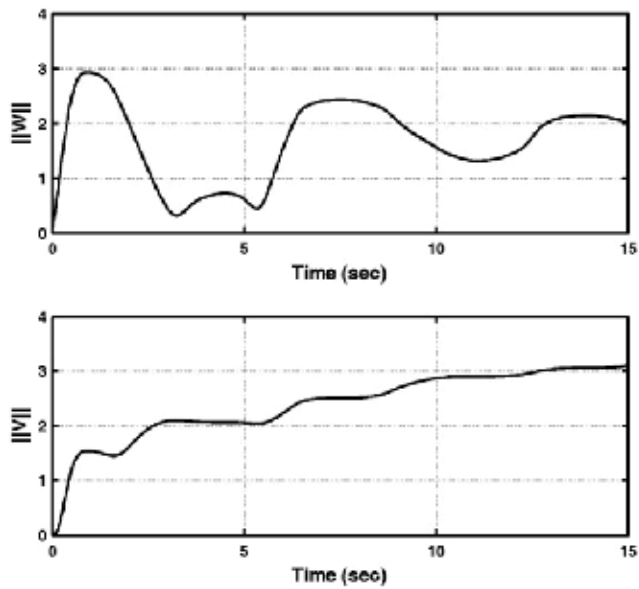


Fig. 29. (sim2) Matrix norm of adaptive weights $\|W\|$ and $\|V\|$

4. Conclusion

This chapter addressed sliding mode control (SMC) of n-link robot manipulators by using of intelligent methods including fuzzy logic and neural network strategies. In this regard, three control strategies were investigated. In the first case, design of a sliding mode control with a PID loop for robot manipulator was presented in which the gain of both SMC and PID was tuned on-line by using fuzzy approach. The proposed methodology in fact tries to use the advantages of the SMC, PID and Fuzzy controllers simultaneously, i. e., the robustness against the model uncertainty and external disturbances, quick response, and on-line automatic gain tuning, respectively. Finally, the simulation results of applying the proposed methodology to a two-link robot were provided and compared with corresponding results of the conventional SMC which show the improvements of results in the case of using the proposed method. In the second case, a new combination of sliding mode control and fuzzy control is proposed which is called incorporating sliding mode and fuzzy controller. Three practical aspects of robot manipulator control are considered there, such as restriction on input torque magnitude due to saturation of actuators, friction and modeling uncertainty. In spite of these features, the designed controller can improve the sliding mode and fuzzy controller performance in the tracking error and faster transient points of view, respectively. As previous case, the simulation results of applying the proposed methodology and other two methodologies to a two-link direct drive robot arm were provided. Comparing these results demonstrate the success of the proposed method.

Whenever, fast and high-precision position control is required for a system which has high nonlinearity and unknown parameters, and also, suffers from uncertainties and disturbances, such as robot manipulators, in that case, necessity of designing a developed controller that is robust and has self-learning ability is appeared. For this purpose, an efficient combination of sliding mode control, PID control and neural network control for position tracking of robot manipulators driven by permanent magnet DC motors was addressed in the third case. SMC is robust against uncertainties, but it is extremely dependent on model and uses unnecessary high control gain; So, NN control approach is employed to approximate major part of the model. A PID part was added to make the response faster, and to assure the reaching of sliding surface during initial period of weight adaptations. Moreover, four practical aspects of robot manipulator control such as actuator dynamics, restriction on input armature voltage of actuators due to saturation of them, friction and uncertainties were considered. In spite of these features, the controller was designed based on Lyapunov stability theory and it could carry out the position control with fast transient and high-precision response, successfully. Finally, two-step simulation was performed and its results confirmed the success of presented approach. However, the presented design was performed in the joint space of robot manipulator and kinematic uncertainty was not considered. For the future work, one can expand this method to work space design with uncertain kinematics.

5. References

- Ataei, M. & Shafiei, S. E. (2008). Sliding Mode PID Controller Design for Robot Manipulators by Using Fuzzy Tuning Approach, *Proceedings of the 27th Chinese Control Conference*, July 16-18 2008, Kunming, Yunnan, China, pp. 170-174.
- Cai, L. & Song, G. (1994). Joint Stick-Slip Friction Compensation of Robot Manipulators by using Smooth Robust Controllers, *Journal of Robotic Systems*, Vol. 11, No. 6, pp. 451-470.

- Calcev, G. (1998). Some Remarks on the Stability of Mamdani Fuzzy Control Systems, *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 4., pp. 436-442.
- Capisani, L. M.; Ferrara, A. & Magnani, L. (2009). Design and experimental validation of a second-order sliding-mode motion controller for robot manipulators, *International Journal of Control*, vol. 82, no. 2, pp. 365-377.
- Chang, Y. C.; Yen, H. M. & Wu, M. F. (2008). An intelligent robust tracking control for electrically driven robot systems, *International Journal of Systems Science*, vol. 39, no. 5, pp. 497-511.
- Chang, Y. C. & Yen, H. M. (2009). Robust tracking control for a class of uncertain electrically driven robots, *IET Control Theory and Applications*, vol. 3, no. 5, pp. 519-532.
- Craig, J. J. (1986). *Introduction to Robotics*, Addison & Wesley, Inc.
- Eker, I. (2006). Sliding mode control with PID sliding surface and experimental application to an electromechanical plant, *ISA Transaction.*, vol. 45, no. 1, pp. 109-118.
- Hung, J. Y.; Gao, W. & Hung, J. C. (1993). Variable structure control: A survey, *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 2-21.
- Kaynak, O.; Erbatur, K. & Ertuğrul, M. (2001). The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control: A Survey, *IEEE Transactions on Industrial Electronics*, vol. 48, no. 1, pp. 4-17.
- Khalil, K. H. (2001). *Nonlinear Systems*, Third edition, Prentice Hall Inc, New York, USA.
- Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I and II, *IEEE Transaction on System, Man and Cybernetics*, Vol. 20, No. 2, 404-435.
- Lewis, F. L.; Yesidirek, A. & Liu, K. (1996). Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance, *IEEE Transactions on Neural Networks*, vol. 7, no. 2.
- Lewis, F. L.; Jagannathan, S. & Yesildirek, A. (1998). *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis.
- Santibanez, V.; Kelly, R. & Liama, L.A. (2005). A Novel Global Asymptotic Stable Set-Point Fuzzy Controller with Bounded Torques for Robot Manipulators, *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 3, pp. 362-372.
- Shafiei, S. E. & Sepasi, S. (2010). Incorporating Sliding Mode and Fuzzy Controller with Bounded Torques for Set-Point Tracking of Robot Manipulators, Scheduled for publishing in the *Journal of Electronics and Electrical Engineering, T125 Automation, Robotics*, No. 10(106).
- Shafiei, S. E. & Soltanpour, M. R. (2010). Neural Network Sliding-Model-PID Controller Design for Electrically Driven Robot Manipulators, Scheduled for publishing in the *International journal of Innovative Computing, Information and Control*, vol. 6, No. 12.
- Slotin, J. J. E. & Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, New York, USA.
- Spong, M. W. & Vidyasagar, M. (1989) *Robot Dynamics and Control*, Wiley, New York, USA.
- Utkin, V. I. (1978). *Sliding Modes and their Application in Variable Structure Systems*, MIR Publishers, Moscow.
- Wai, R. J. & Chen, P. C. (2006). Robust Neural-Fuzzy-Network Control for Robot Manipulator Including Actuator Dynamics, *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1328-1349.
- Wang, L. X. (1997). *A Course in Fuzzy Systems and Control*, Prentice Hall, NJ, New York, USA.
- Zhang, M.; Yu, Z.; Huan, H. & Zhou, Y. (2008). The Sliding Mode Variable Structure Control Based on Composite Reaching Law of Active Magnetic Bearing, *ICIC Express Letters*, vol.2, no.1, pp.59-63.

Supervision and Control Strategies of a 6 DOF Parallel Manipulator Using a Mechatronic Approach

João Mauricio Rosário¹, Didier Dumur², Mariana Moretti¹,
Fabian Lara¹ and Alvaro Uribe¹
¹UNICAMP, Campinas, SP,
²SUPELEC, Gif-sur-Yvette,
¹Brazil
²France

1. Introduction

Currently, the Stewart Platform is used in different engineering applications (machine tool technology, underwater research, entertainment, medical applications surgery, and others) due to its low mechatronic cost implementation as an alternative to conventional robots. The current trend of using parallel manipulators has created the need for developing open supervision and control architectures. This chapter presents the mathematical analysis, simulation, supervision and control implementation of a six degree of freedom (DOF) parallel manipulator known as the Stewart platform. The related studies are critically examined to ascertain the research trends in the field. An analytical study of the kinematics, dynamics and control of this manipulator covers the derivation of closed form expressions for the inverse Jacobian matrix of the mechanism and its time derivative, the evaluation of a numerical iterative scheme for forward kinematics on-line solving, the effects of various configurations of the unpowered joints due to angular velocities and accelerations of the links, and finally the Newton-Euler formulation for deriving the rigid body dynamic equations.

The contents of this chapter are organized as follows:

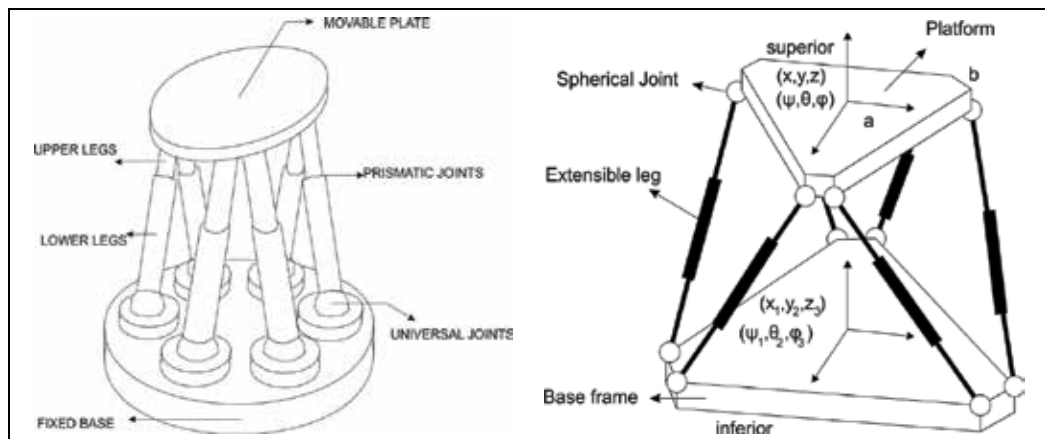
- Section II presents the features of a Stewart Platform manipulator, describing its spatial motion and applications.
- Section III covers the mathematical description, with the kinematics and dynamics modelling, and the actuator control using a mechatronic prototyping approach.
- Section IV details the control structure, and compares two different control strategies: the PID joint control structure and the Generalized Predictive Control (GPC). Both controllers structured in the polynomial RST form, as a generic framework for numerical control laws satisfying open architecture requirements.
- Section V describes the supervision and control architecture, particularly the spatial tracking error is analyzed for both controllers.

- Section VI provides time domain simulation results and performance comparison for several scenarios (linear and circular displacements, translational or rotational movements), using reconfigurable computing applied to a Stewart-Gough platform.
- Section VII presents the supervisory control and hardware interface implemented in a Labview™ environment.
- Finally, section VII presents the conclusions and contributions.

2. Stewart platform manipulator

The Stewart platform is a 6 DOF mechanism with two bodies connected by six extendable legs. The manipulation device is obtained from the generalisation of the proposed mechanism of a flight simulator presented in (Stewart, 1965)(Gough & Whitehall, 1962)(Karger, 2003)(Cappel, 1967). Its legs are connected through spherical joints at both ends, or a spherical joint at one end, and a universal joint at the other. The structure with spherical joints at both ends is the 6-SPS (spherical-prismatic-spherical) Stewart platform (Fig. 1), while the one, with an universal joint at the base and a spherical joint at the top is the 6-UPS (universal-prismatic-spherical) Stewart platform (Dasgupta, 1998)(Bessala, Philippe & Oueddou, 1996).

The spatial movements of the six-axis parallel manipulator provide three translational and three rotational DOF of the movable plate, allowing position accuracy, stiffness and payload-to-weight ratio to exceed conventional serial manipulators performances. Due to these mechanical advantages, the Stewart platform manipulator is used in many applications such as flight simulators, parallel machine-tools, biped locomotion systems and surgery manipulators (Sugahara et al., 2005)(Wapler et al., 2003)(Wentlandt & Sastry, 1994).



a) Mathworks™ description

b) The 6-UPS Stewart Platforms

Fig. 1. Schematic Representation of the Stewart-Gough Platform.

3. Mathematical description

The mathematical model has to respond to a desired trajectory by actuating forces in order to properly move the mobile plate to the targeted position and orientation. For obtaining the

mathematical representation, a reference coordinated system for analyzing the manipulator is presented in Fig. 1.

3.1 Geometric model

Given the accomplishment of numerous tasks due to its configuration, the platform legs are identical kinematics chains whose motion varies accordingly to the tip of the joint used (Fasse & Gosselin, 1998)(Boney, 2003). Typically, the legs are designed with an upper and lower adjustable body, so each one has a variable length (Fig. 1). The geometrical model of a platform is expressed by its (X, Y, Z) position and the (ψ, θ, ϕ) orientation due to a fixed coordinate system linked at the base of the platform. The obtained function of this generalized coordinates (joints linear movements), is presented in (1).

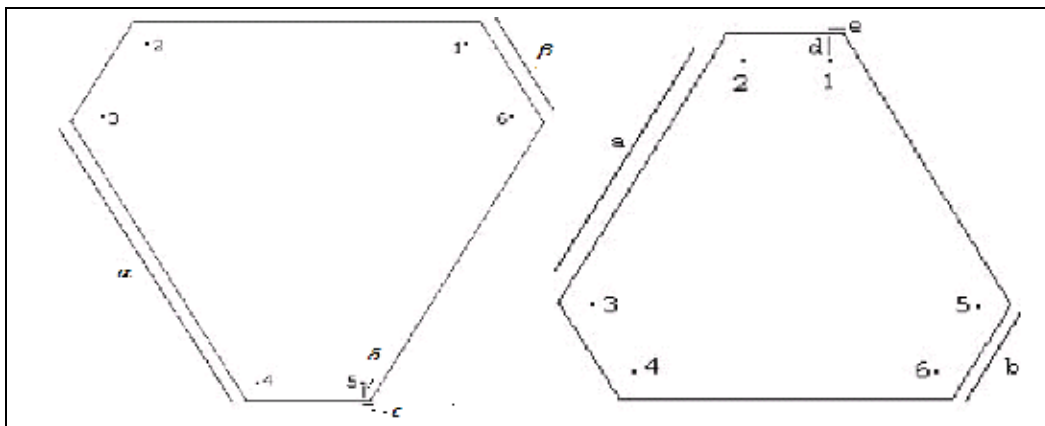
$$X_i = f(L_i) \quad (1)$$

where $L_i = (L_1 \ L_2 \ \dots \ L_6)$ are each joint linear position, $X_i = (X \ Y \ Z \ \psi \ \theta \ \phi)$ the position-orientation vector of a point of the platform. Then the transformation matrix for rotations can be organised as Shown in (2), where, $c\psi: \cos \psi$, $s\psi: \sin \psi$

$$T(\psi, \theta, \phi) = \text{rot}(x, \phi) \text{rot}(y, \theta) \text{rot}(z, \psi) = \begin{bmatrix} c\phi c\theta & -c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & -s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} \quad (2)$$

where, $\theta = \text{ATAN} 2 \left[\frac{-n_z}{c\phi n_x + s\phi n_y} \right]$, $\phi = \text{ATAN} 2 \left[\frac{n_y}{n_x} \right]$, $\psi = \text{ATAN} 2 \left[\frac{s\phi a_x - c\phi a_y}{-s\phi s_x + c\phi s_y} \right]$ and

$\mathbf{n} = [n_x \ n_y \ n_z]$, $\mathbf{s} = [s_x \ s_y \ s_z]$, $\mathbf{a} = [a_x \ a_y \ a_z]$: are the orthonormal vectors that describe the platform's orientation.



a) Inferior base

b) Superior base

Fig. 2. Platform Geometric Model - Actuators reference points.

This transformation matrix allows changing each actuator's position into a new configuration in order to define the kinematics model as shown in Fig.2 (Kim, Chung & Youmt, 1997)(Li & Salcudean, 1997).

The points that define the upper base motion are located at the extremities of the six linear actuators fixed at the lower base of the platform. When assuming that the actuators have reached their final position and orientation, the problem is calculating the coordinates of the center of mass on the superior base, and the RPY orientation angles (roll, pitch and yaw).

The relative positions can be calculated from the position and orientation analysis (using the transformation matrix), leading to new ones within the platform's workspace.

The position vector for the actuator of the upper/lower base, P_i, P_s , is determined in relation to the fixed reference system at the center of mass of the inferior part as described in (3). The parameters $\alpha, \beta, \delta, \varepsilon, a, b, d, e$ are reported in Fig.2, where h represents the position of the center of mass of the upper base in its initial configuration, and each line of P_i, P_s represents the lower ($A_1 \dots A_6$) and superior ($B_1 \dots B_6$) coordinated extremities of the actuators.

$$P_i = \begin{bmatrix} A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon + C_i & D_i - \delta + C_i & 0 \\ -B_i + \varepsilon & D_i + \delta & 0 \\ B_i + \varepsilon & -D_i + \delta & 0 \\ A_i - \varepsilon - C_i & D_i - \delta + C_i & 0 \end{bmatrix} \quad P_s = \begin{bmatrix} A_s + e & D_s - d & h \\ -A_s + e & D_s - d & h \\ -A_s + e + C_s & D_s - d + C_s & h \\ -B_s + e & D_s + d & h \\ B_s + e & -D_s + d & h \\ A_s - e - C_s & D_s - d + C_s & h \end{bmatrix} \quad (3)$$

where, $A_i = 0.5\alpha$, $A_s = 0.5b$, $B_i = 0.5\beta$, $B_s = 0.5a$, $C_i = 2(\varepsilon - B_i)\cos(t)$, $C_s = 2(e - B_s)\cos(t)$, $D_i = (A_i + B_i)\cos(t)$, $D_s = (A_s + B_s)\cos(t)$

Each actuator is associated to a position vector X_i considering its inferior end and the value of the distension associated with the i th actuator. With the transformation matrix, X_i^T is the new associated position vector for each upper position i th, obtained in (4).

$$\underline{X}_i = T(\psi, \theta, \varphi) \underline{X}_i^T \quad (4)$$

From the known position of the upper base, the coordinates of its extremities are calculated using the previous equations resulting in new ones, whose norm corresponds to the new size of the actuator. If X_0 is the reference point, then the difference between the current sizes and the target ones is the distension that must be imposed to each actuator in order to reach its new position as presented in (5)

$$\Delta L = \left| \underline{X}_i^T - \underline{X}_0 \right| - \left| \underline{X}_i - \underline{X}_0 \right| \quad (5)$$

Thus, the distance between the extremities is calculated using the transformation matrix and the known coordinates. The kinematic model of the platform receives the translation information in vector form and the rotation from a matrix with the RPY angles.

This analysis allows calculating each axes lengths so that the platform moves to the target position, so the required of each linear actuator k connected to the upper mobile base before and after movement is described in Eqs. 6 and 7.

$$L = \sqrt{\sum_{j=1}^3 (P_s^{kj} - P_i^{kj})^2} \quad \text{with } k=1, \dots, 6 \quad (6)$$

$$L + \Delta L = \sqrt{\sum_{j=1}^3 (T_j^{-1}(\psi, \theta, \phi) P_s^{kj} - P_i^{kj})^2} \quad (7)$$

The links of the platform are defined by:

$$A_i = [r_p \cos(\alpha_i), r_p \sin(\alpha_i), 0]^T = [A_{ix}, A_{iy}, A_{iz}]^T \quad \alpha_i = \frac{i\pi - a_p}{2} \quad \text{for } i=1,3,5 \quad \alpha_i = \alpha_{i-1} + a_p \quad \text{for } i=2,4,6 \quad (8)$$

And the links of the base by:

$$B_i = [r_b \cos(\beta_i), r_b \sin(\beta_i), 0]^T = [B_{ix}, B_{iy}, B_{iz}]^T \quad \beta_i = \frac{i\pi - a_b}{2} \quad \text{for } i=1,3,5 \quad \beta_i = \beta_{i-1} + a_b \quad \text{for } i=2,4,6 \quad (9)$$

Where r_p : radius of platform; r_b : radius of base; a_p : angle of platform and a_b : angle of base

3.2 Kinematic model

The Stewart Platform Manipulator changes its position and orientation as a function of its linear actuator's length. Fig. 3 shows the corresponding geometric model viewed from the top, where the bottom base geometry is formed by the B1 to B6 points, and the upper one by A1 to A6 points.

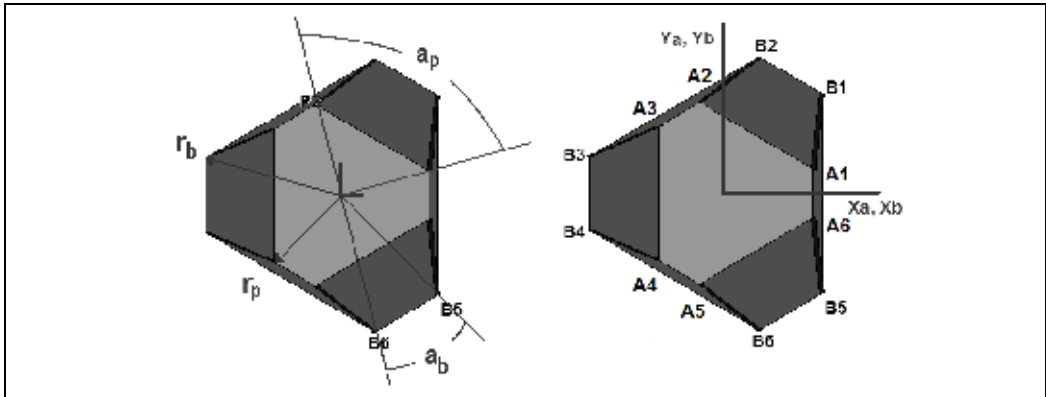


Fig. 3. Stewart Platform geometric model

3.3 Inverse kinematics

The inverse kinematics model of the manipulator expresses the joint linear motion as a position and orientation function due to the fixed coordinate system at the base of the platform (Wang, Gosselin & Cheng, 2002)(Zhang & Chen, 2007), as presented in Eq. 10:

$$l=f(x) \quad (10)$$

Where, $l=(l_1, l_2, l_3, l_4, l_5, l_6)$ is the linear position of the joints, $x=(X, Y, Z, \psi, \theta, \phi)$ is the position vector of the platform, X, Y, Z the cartesian position and ψ, θ, ϕ represents the orientation of

the platform. The reference systems are fixed to $A(u,v,w)$ and $B(x,y,z)$ at the base, as shown in Fig. 4.

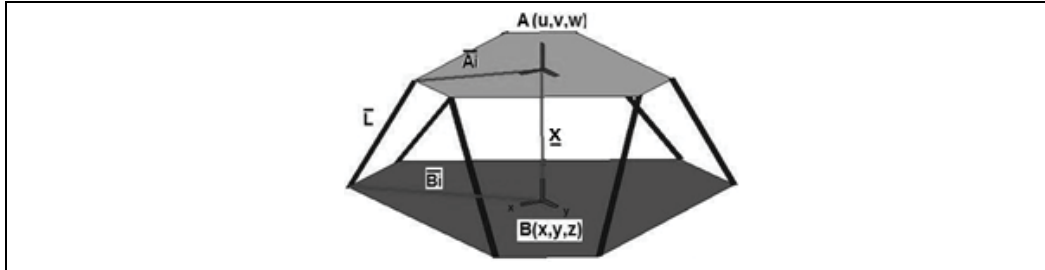


Fig. 4. Vector representation of the manipulator.

The transformation for the mobile platform's centroid to the base, is described by the position vector x and the rotation matrix ${}^B R_A$, where,

$${}^B R_A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (11)$$

The angular motions are expressed as Euler angle rotations in respect to x-axis, y-axis, and z-axis, i.e. roll, pitch and yaw, in sequence.

$${}^B R_A = \begin{bmatrix} c\psi c\phi & c\psi s\phi s\theta - s\psi c\theta & c\psi s\phi c\theta + s\psi s\phi c\theta \\ c\psi c\phi & s\psi s\phi s\theta + c\psi c\theta & s\psi s\phi c\theta - c\psi s\theta \\ -s\phi & c\phi c\theta & c\phi c\theta \end{bmatrix} \quad (12)$$

The vector-loop equation for the i th actuator of the manipulator is as follows:

$$l_i = {}^A R_B A_i + x - B_i \quad (13)$$

By substituting the terms for each actuator, (14) describes the platform motion in relation to its base.

$$l_i^2 = X^2 + Y^2 + Z^2 + r_p^2 + r_b^2 + 2(r_{11}A_{ix} + r_{12}A_{iy})(X - B_{ix}) + 2(r_{21}A_{ix} + r_{22}A_{iy})(Y - B_{iy}) + 2(r_{31}A_{ix} + r_{23}A_{iy})(Y - B_{iy}) + 2Z(r_{31}A_{ix} + r_{23}A_{iy}) - 2(XB_{ix} + YB_{iy}) \quad (14)$$

3.4 Dynamics study

The dynamic equations are derived for the Stewart Platform with a universal joint at the base and a spherical joint at the top of each leg. For this study, it is assumed that there is no rotation allowed on any leg about its own axis, so the kinematics and dynamics for each one considers and calculates the constraining force over the spherical joint at its top.

Finally, the kinematics and dynamics of the platform are considered so the spherical joint forces from all the six legs complete the dynamic equations.

The motion control can be implemented on every joint considering the movements of each actuator (Guo & Li, 2006). Considering the coupling effects and to solve the trajectory

problem, the dynamic control takes the inputs of the system so the drive of each joint moves its links to the target position with the required speed.

The dynamic model of a 6-DOF platform can be calculated with the Euler-Lagrange formulation that expresses the generalized torque (Jaramillo et al, 2006)(Liu, Li and Li, 2000).The dynamic model is described by a set of differential equations called dynamic equations of motion as shown in (15).

$$\tau_i = J_i \ddot{L}_i + F_i \dot{L}_i + \Gamma_i \quad i = 1, \dots, 6 \quad (15)$$

where $\tau_i(t)$ is the generalized torque vector, $L_i(t)$ the generalized frame vector (linear joints), $J_i(t)$ the inertial matrix, $F_i(t)$ the non-linear forces (for example centrifugal) matrix, Γ_i the gravity force matrix.

3.5 Actuator model

Each joint is composed of a motor, a transmission system and an encoder and by considering DC motor (Ollero, Boverie & Goodal, 2005), its three classic equations are presented in Eq. 16

$$\begin{aligned} u(t) &= L_{mot} \frac{di(t)}{dt} + R_{mot} i(t) + K_E \frac{d\theta_m(t)}{dt} \\ T_m(t) &= J_{eq} \frac{d^2 \theta_m(t)}{dt^2} + B_{eq} \frac{d\theta_m(t)}{dt} = K_T i(t) \end{aligned} \quad (16)$$

where $T_m(t)$ is the torque, $\theta_m(t)$ the angular position of the motor axis, $i(t)$ the current, L_{mot}, R_{mot} respectively the inductance, resistance, J_{eq}, B_{eq} the inertia, friction of the axis load calculated on the motor side.

4. Control structure

A simulation environment allows implementing and testing advanced axis control strategies, such as Predictive Control, which is a well known structure for providing improved tracking performance. The purpose of the control structure is to obtain a model of the system that predicts the future system's behaviour, calculates the minimization of a quadratic cost function over a finite future horizon using future predicted errors. It also elaborates a sequence of future control values; only the first value is applied both on the system and the model, finally the repetition of the whole procedure at the next sampling period happens accordingly to the preceding horizon strategy (Li & Salcudean, 1997) (Nadimi, Bak & Izadi, 2006)(Remillard & Boukas, 2007)(Su et al, 2004).

4.1 Model

The Controlled Autoregressive Integrated Moving Average Model (CARIMA) form is used as numerical model for the system so the steady state error is cancelled due to a step input or disturbance by introducing an integral term in the controller (Clarke, Mohtadi & Tuffs,

1987). The predictive control law uses an external input-output representation form, given by the polynomial relation:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \frac{\xi(k)}{\Delta(q^{-1})} \quad (17)$$

where u is the control signal applied to the system, y the output of the system, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, A and B polynomials in the backward shift operator q^{-1} , of respective order n_a and n_b , ξ an uncorrelated zero-mean white noise.

4.2 Predictive equation

The predictive method requires the definition of an optimal j -step ahead predictor which is able to anticipate the behaviour of the process in the future over a finite horizon. From the input-output model, the polynomial predictor is designed under the following form:

$$\hat{y}(k+j) = \underbrace{F_j(q^{-1})y(k) + H_j(q^{-1})\Delta u(k-1)}_{\text{free response}} + \underbrace{G_j(q^{-1})\Delta u(k+j-1) + J_j(q^{-1})\xi(k+j)}_{\text{forced response}} \quad (18)$$

where F_j , G_j , H_j and J_j , unknown polynomials, corresponding to the expression of the past and of the future, are derived solving Diophantine equations, with unique solutions controller (Clarke, Mohtadi & Tuffs, 1987).

4.3 Cost function

The GPC strategy minimizes the weighted sum of the square predicted future errors and the square control signal increments:

$$J = \sum_{j=N_1}^{N_2} (\hat{y}(k+j) - w(k+j))^2 + \lambda \sum_{j=1}^{N_u} \Delta u(k+j-1)^2 \quad (19)$$

Assuming that $\Delta u(t+j) = 0$ for $j \geq N_u$. Four tuning parameters are required: N_1 , the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control-weighting factor.

4.4 Cost function minimization

The optimal j -step ahead predictor (20) is rewritten in matrix form:

$$\hat{\mathbf{y}} = \mathbf{G} \tilde{\mathbf{u}} + \mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1})\Delta u(t-1) \quad (20)$$

with:

$$\begin{aligned} \mathbf{if}(q^{-1}) &= \begin{bmatrix} F_{N_1}(q^{-1}) & \cdots & F_{N_2}(q^{-1}) \end{bmatrix}' & \tilde{\mathbf{u}} &= \begin{bmatrix} \Delta u(t) & \cdots & \Delta u(t+N_u-1) \end{bmatrix}' \\ \mathbf{ih}(q^{-1}) &= \begin{bmatrix} H_{N_1}(q^{-1}) & \cdots & H_{N_2}(q^{-1}) \end{bmatrix}' & \hat{\mathbf{y}} &= \begin{bmatrix} \hat{y}(t+N_1) & \cdots & \hat{y}(t+N_2) \end{bmatrix} \end{aligned} \quad (21)$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \cdots & \cdots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \cdots & g_{N_2-N_u+1}^{N_2} \end{bmatrix} \quad (22)$$

The future control sequence is then obtained by minimizing the criterion (23) (Clarke, Mohtadi and Tuffs, 1987):

$$\tilde{\mathbf{u}} = \mathbf{M} \left[\mathbf{w} - \mathbf{if}(q^{-1})y(t) - \mathbf{ih}(q^{-1})\Delta u(t-1) \right] \quad (23)$$

with:

$$\mathbf{M} = \mathbf{Q} \mathbf{G}^T, N_u \times (N_2 - N_1 + 1), \mathbf{Q} = \left[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}_{N_u} \right]^{-1}, \mathbf{w} = [w(t+N_1) \cdots w(t+N_2)] \quad (24)$$

4.5 RST form controller

The minimization of the previous cost function (Clarke, Mohtadi & Tuffs, 1987), results in the predictive controller derived in the RST form according to Fig. 5 and implemented through a differential equation in (25).

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q)w(t) \quad (25)$$

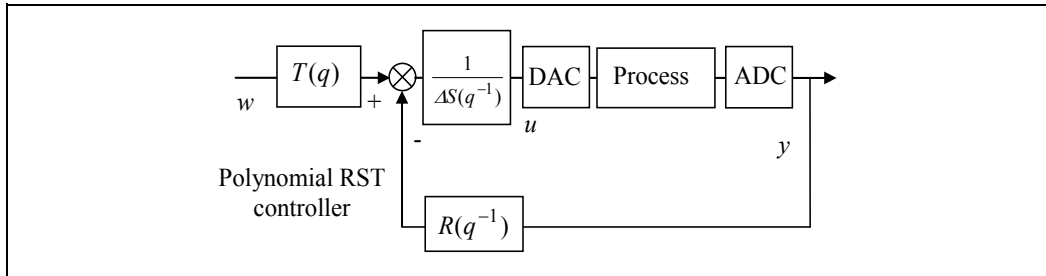


Fig. 5. GPC in a RST form.

The main feature of this RST controller is the non-causal form of the T polynomial, creating the anticipative effect of this control law.

4.6 Complete model implementation

Taking the x_r as the system's input trajectory the objective is to calculate the actuator's length l_r for each sampled position. Mechanism and actuator controller dynamic effects are considered over the six legs having as outputs their δ_{ld} and previous position x_{i-1} , this is done in order to calculate the current manipulator position x_o, x_f is determined by the length of the actuator l_o . Then these values are compared with the target position in order to estimate the error δ_l between the reference position x_r and the manipulator's position x_o after all the dynamics effects have been considered (Fig. 6 and Fig. 7) (Hunt, 1978)(Jaramillo et al, 2006) (Ghobakhloo, Eghtesad & Azadi, 2006).

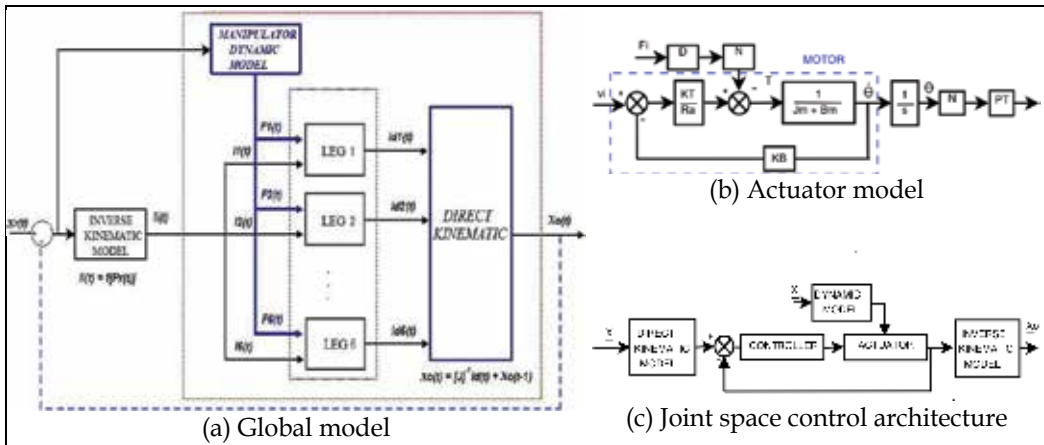


Fig. 6. Total system Model

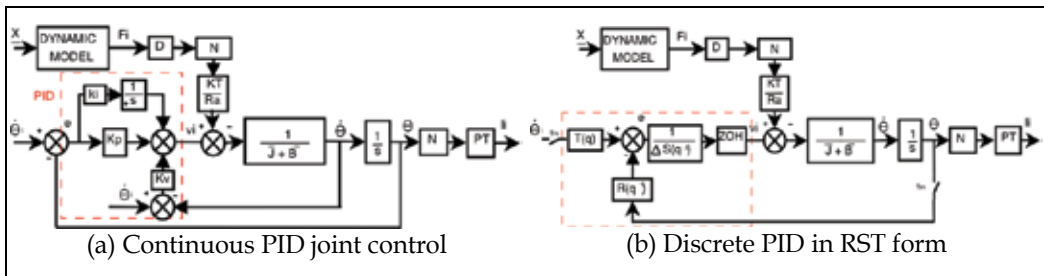


Fig. 7. Continuous and Discrete PID Controller.

$$\begin{aligned}
 \text{degree}(R(q^{-1})) &= \text{degree}(R(q^{-1})) \\
 \text{degree}(S(q^{-1})) &= \text{degree}(B(q^{-1})) \\
 \text{degree}(T(q)) &= N_2
 \end{aligned}
 \tag{44}$$

The GPC has shown to be an effective strategy in many fields of applications, with good time-domain and frequency properties (small overshoot, improved tracking accuracy and disturbance rejection ability, good stability and robustness margins), is able to cope with important parameters variations.

5. Simulation

The modelling of the Parallel Manipulator leads to the design of a simulator adopting electric and mechanical libraries blocks using Simulink (Gosselin, Lavoie & Toutant, 1992). The main elements of the robotics joints are brushless DC motor drives, axis inertia, gears and control blocks. Other elements of the manipulator (including loads) are represented by three nonlinear models, one for each motor drive. The control system itself consists, essentially, in a cascade of control loops (for each axis). The inner speed and torque control loops are part of the drive model where only the position loop is explicitly modelled. In fact,

the position control of the manipulator can be implemented through the control feedback of each isolated joint (Cappel, 1967).

The developed simulator also includes a path generation module, providing the joints with axis trajectories as reference signal for controlling each of the parts (Jaramillo et al, 2006). Finally, a graphic interface is developed, showing the results of joint motion obtained through typical trajectories. The simulation software was implemented using Matlab® and programmed with the equations of the Stewart Platform manipulator. This interface allows the input of the dynamic simulation parameters: mass and inertia of the mobile platform, actuator parameters and the gains of the PID controller. Fig. 8 shows a screen capture of the developed interface.

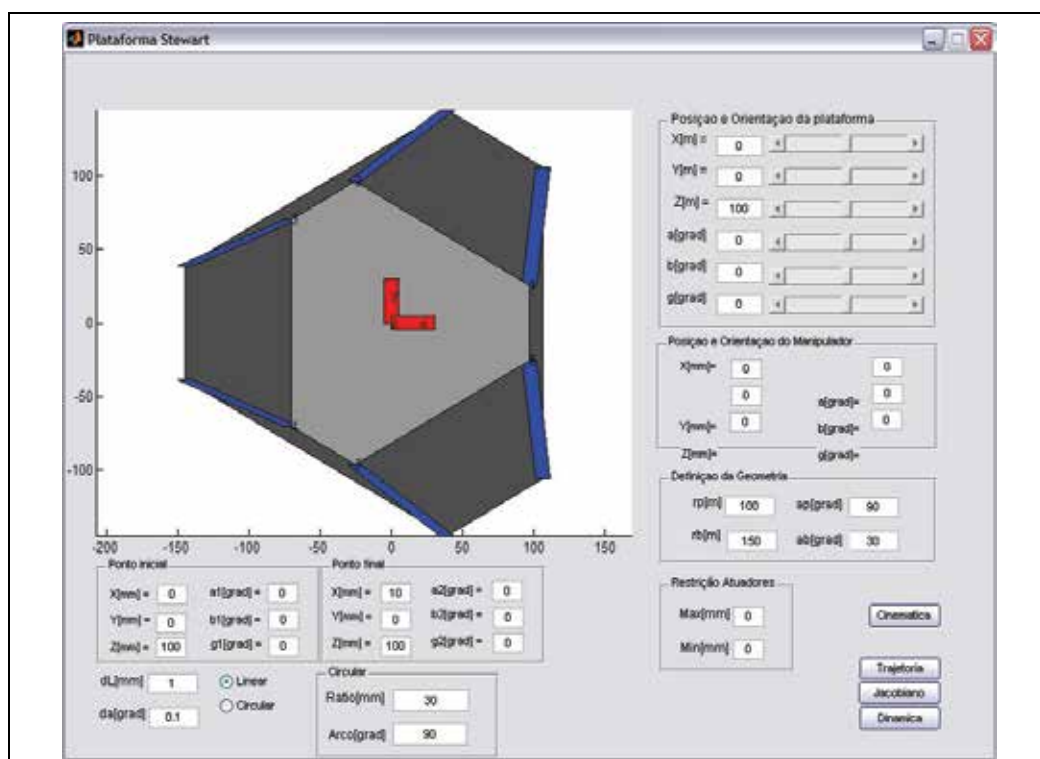


Fig. 8. Implemented simulation environment

In Fig. 9 the overall block diagram with the dynamic and control model (Fig. 3) implemented in Simulink is presented

The considered system used for supervision and control implementation includes 3 DC motors, a 1:100 gear box (N), a ball screw transmission (for joint 1 only) and incremental encoders (Table 1). The joint controllers are designed independently, resulting in three RST parameters, considering the same axis motor but with different inertia on the motor side due to different geometrical features for each one.

Four tuning parameters are required: N_1 the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control weighting factor. These are given in Table 2 have been chosen to provide good stability and robustness margins (Clarke, Mohtadi, & Tuffs, 1998).

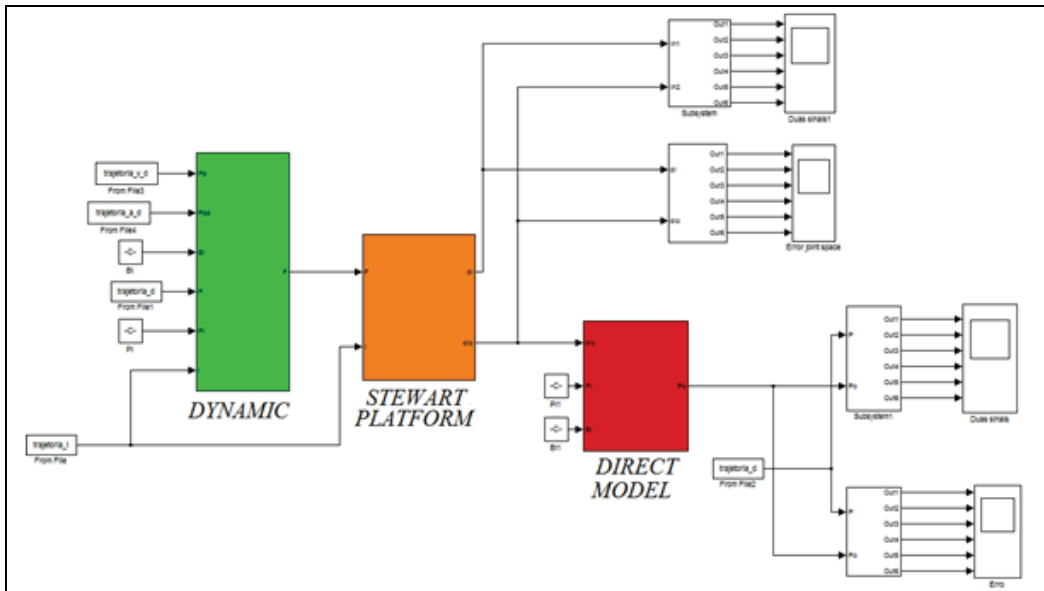


Fig. 9. Simulink Dynamic and control Model

Jm - Inertia (kgm ²)	0.71 10 ⁻³
Weight (kg)	8
Mechanical time constant (ms)	1.94
Voltage constant (V/rad/s)	0.807
Torque constant (Nm/A)	1.33
L - Inductance (mH)	14.7
R - Resistance (Ω)	1.44

Table 1. Motor Parameters.

Joint	N ₁	N ₂	N _u	λ
1	1	8	1	92
2	1	8	1	107.3
3	1	8	1	126

Table 2. GPC tuning parameters for each joint.

5.1 Manipulator geometry variation: case study

The manipulator workspace and behaviour can be studied from the variation and simulation of various upper and bottom plate geometries, these configurations are presented in Fig. 10 with their corresponding geometry parameters. Once the geometry of each plate is chosen, motion to target positions can be simulated using the implemented path generator, Fig. 11 presents a circular path over a xy plane.

An initial point of the circular trajectory on the xy plane is presented in Fig. 11.

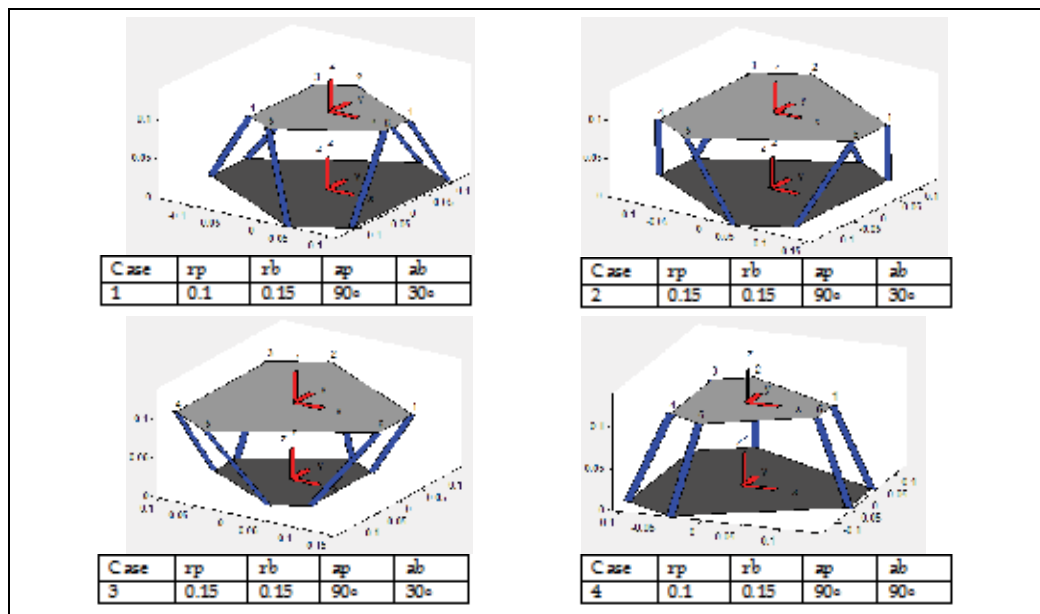


Fig. 10. Top and bottom implemented base geometries and parameters

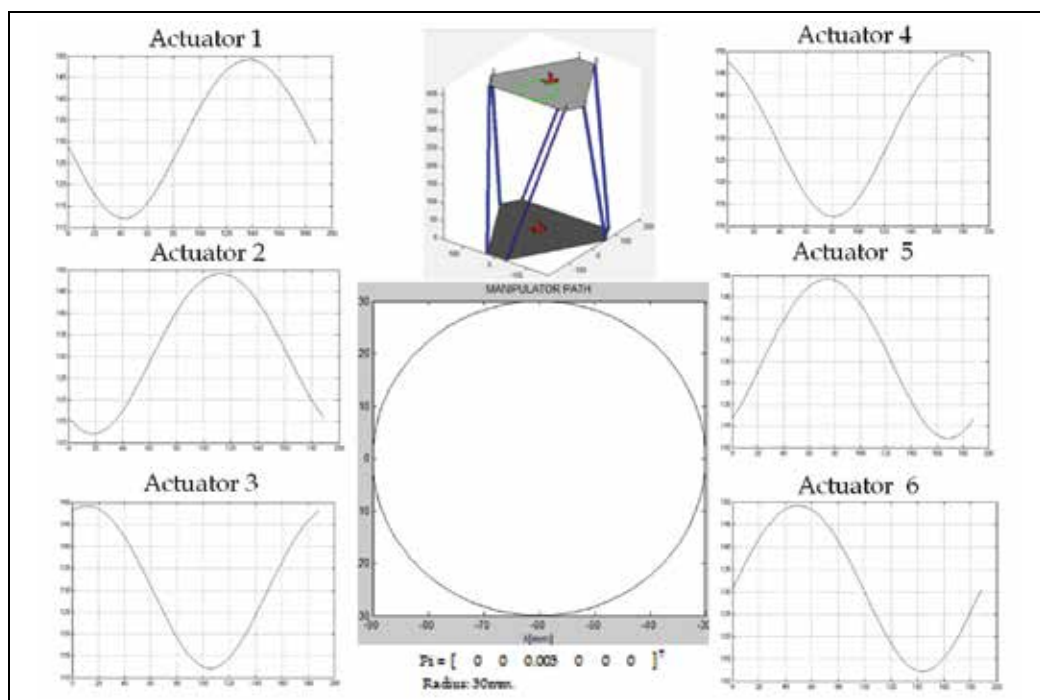


Fig. 11. Path Generator Results

The maximum velocity for this workspace trajectory is 2mm/s and the maximum acceleration is 0.1 mm/s² (Fig. 12).

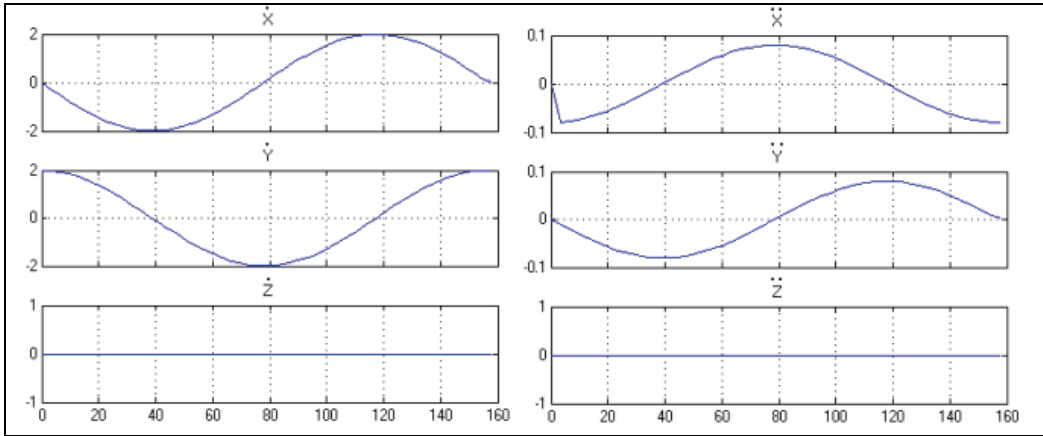


Fig. 12. a. workspace velocity b. workspace acceleration

The joint space trajectory corresponding to the target workspace path for each actuator $l_i(t)$ is obtained through the inverse kinematics model obtaining the trajectories trajectories presented in Fig. 13.

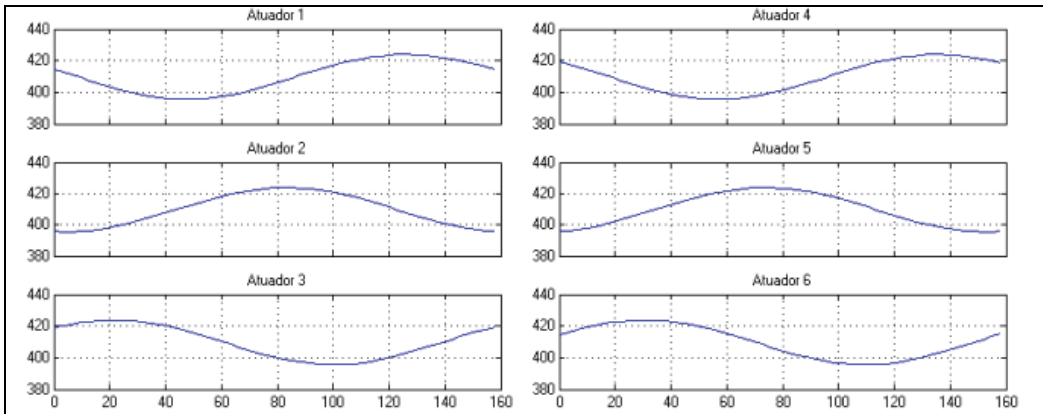


Fig. 13. Joint space trajectory

The singular configurations with various conditions along the trajectory are analyzed (Fig. 14), in this case, the variation of the singular number respect to the initial condition is 2.52%; with this small variation singular configurations are avoided, while in other cases changing the number of conditions results in higher singularities.

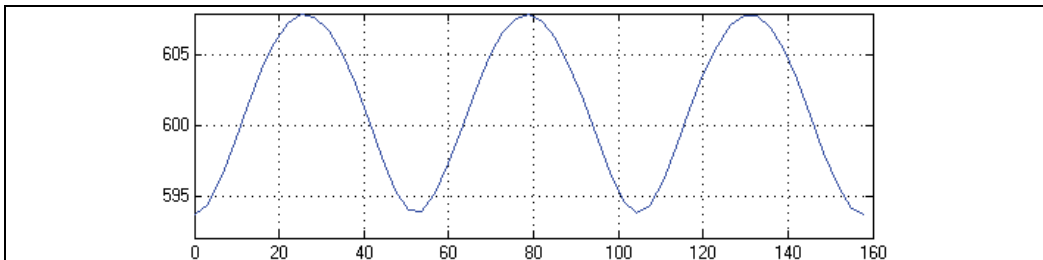


Fig. 14. Singular Analyses.

The constant workspace volume of the manipulator is also evaluated (Fig. 15). This useful characteristic helps to plan new workspace trajectories with constant orientation.

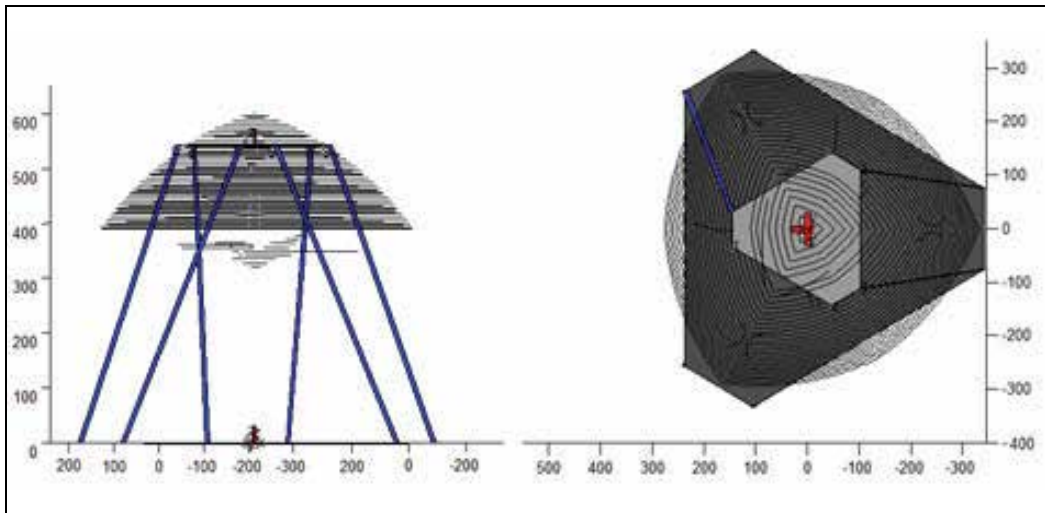


Fig. 15. Work space volume

5.2 Dynamics study

In order to perform the dynamic study and analyze the associated effects of various forces over the platform, the entire system has been modelled in a Simulink environment. The obtained model is composed of one equation and four integration blocks for calculating velocity and position from the known acceleration. In addition, to simplify the implementation of the simulation, the platform initial conditions are declared through an initialization button. Finally a graph button allows the visualization of the results after simulation.

The simulation tests were performed using values defined in subsection 5.1, and also the initial position of the center of gravity of the following platform:

$$T_o = [\quad 0.1 \quad 0 \quad 0.395 \quad] \quad \text{m}$$

$$\theta_o = [\quad 0.1 \quad 0 \quad -0.2 \quad] \quad \text{rad}$$

The initial position of the 3D platform is presented in Fig. 16a, where a 50N constant force is applied on each arm for 0.5 s resulting in the position of the platform shown in Fig. 16b. During this movement the change of position, linear and angular velocities of the center of gravity is calculated and presented in Fig.17.

5.3 Control analysis

For joint space position control the PID controller is tuned with the following gains:

$K_p=100$, $K_i=1$ and $k_d=1$. the $li(t)$ input and $ld(t)$ output joint space trajectory is presented in Fig. 18.

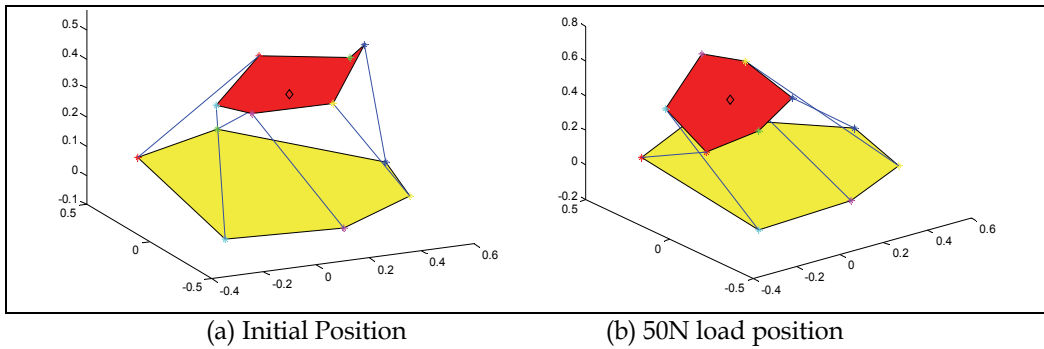


Fig. 16. Stewart Platform Positions

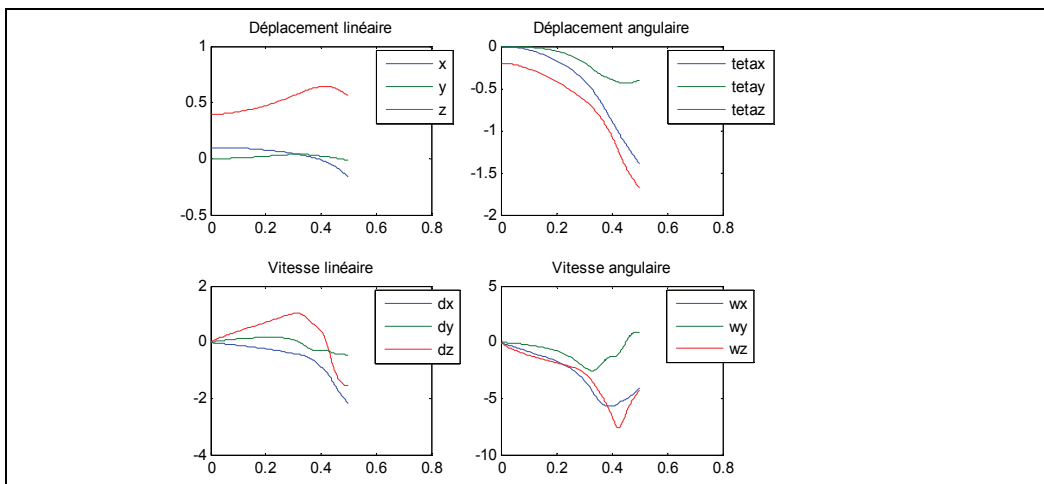


Fig. 17. Linear and angular displacement of of the center of gravity of the Stewart Platform.

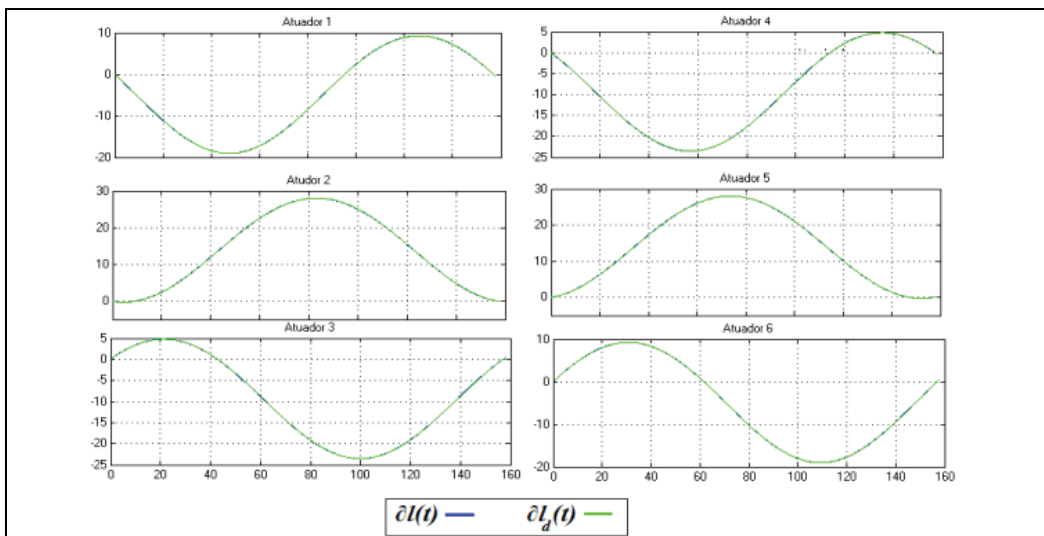


Fig. 18. Input and output joint space trajectory.

The maximum joint space error is 0.1 mm in all the actuators resulting in a maximum motion of 30 mm or 0.333%.

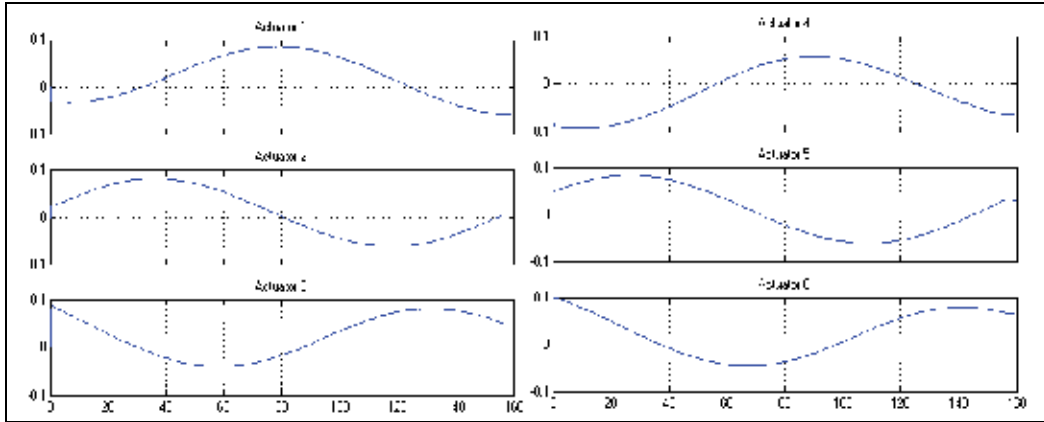


Fig. 19. Joint Space error

Applying the forward kinematics for calculating and comparing the workspace output x_0 is calculated and compared with the workspace input x_r .

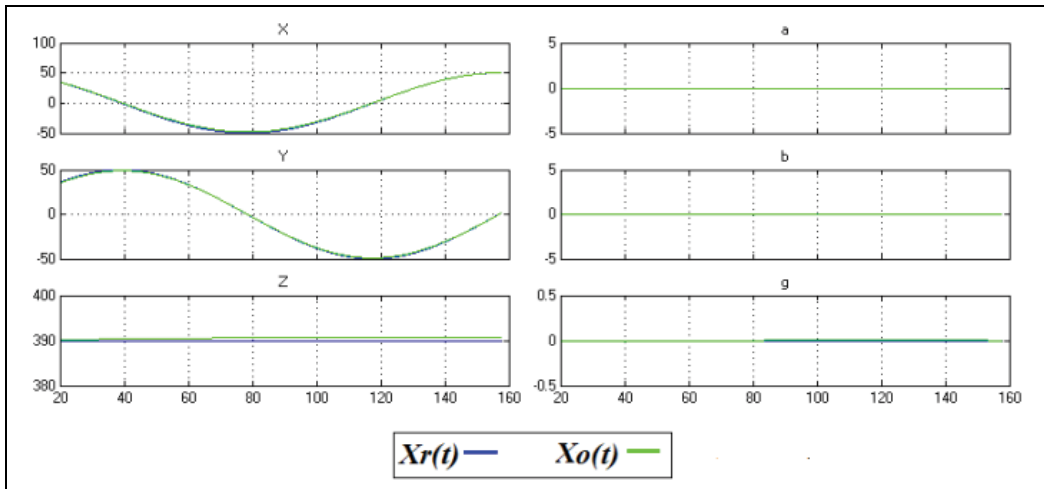


Fig. 20. Output and input workspace trajectory

The maximum work space error is 0.5 mm for a maximum linear motion of 390 mm in the z axis (Fig. 21).

6. Supervision and control architecture

The purpose of implementing a supervisory system over the platform is to permit an easy, fast adaptation and expansion of the system due to current technological trends, resulting in better portability and scalability of the system. Through the structure division in functional blocks, with very specific dedicated interfaces, the project implementation becomes more efficient. The rapid prototyping tools allow designing integrated environments for

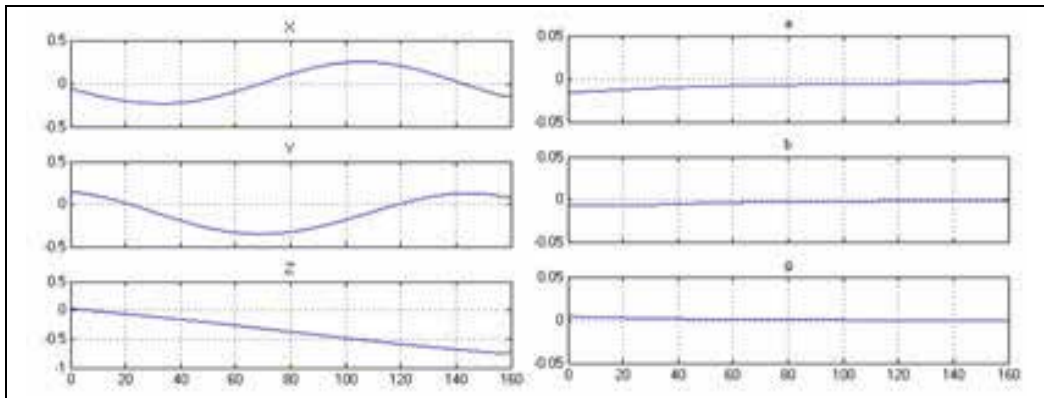
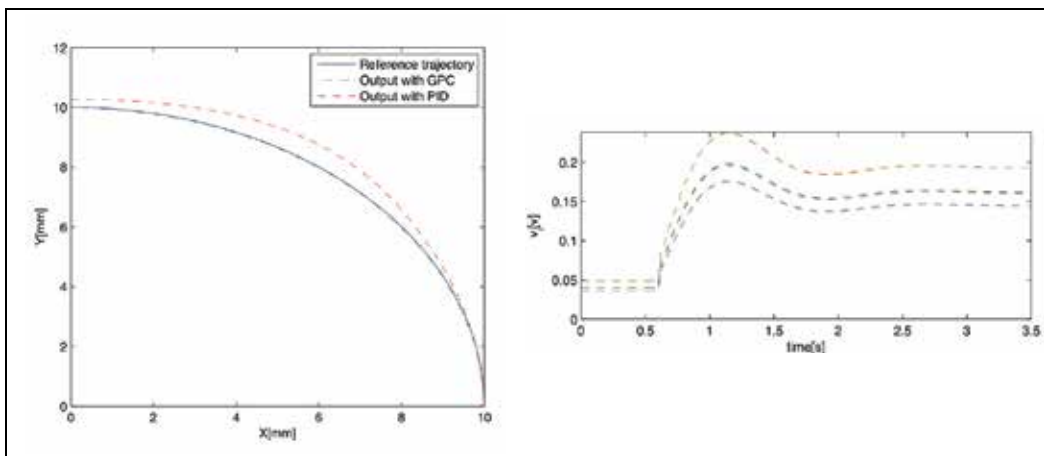


Fig. 21. Work space error



a) Tracking error

b) Disturbance reaction, PID

Fig. 22. Time domain simulation results, tracking error and disturbance reaction

modelling, simulating, and testing algorithm development, through components that simulates the dynamic models of the mechatronic systems; performs complex simulation of the overall system and environment; generates programming code for embedded robot control, and communicates with the platform for controlling it locally or remotely (McCallion, 1977).

The proposed control architecture is a set of implemented hardware and software modules emphasizing on rapid prototyping systems integrated to support the development of the platform tasks.

6.1 Control levels

In the supervisory control level, the supervision of a generic platform task can be achieved through the execution of global control strategies. This level also allows correcting the task execution according to the data obtained through the sensors. The embedded control level is dedicated for executing control strategies allowing locally decision making, with occasional corrections from the supervisory control level. The local control is restricted to local

strategies associated with the sensors and actuators data. The strategies in this level can be implemented under a rapid prototyping framework like FPGA, as described in Fig. 23.

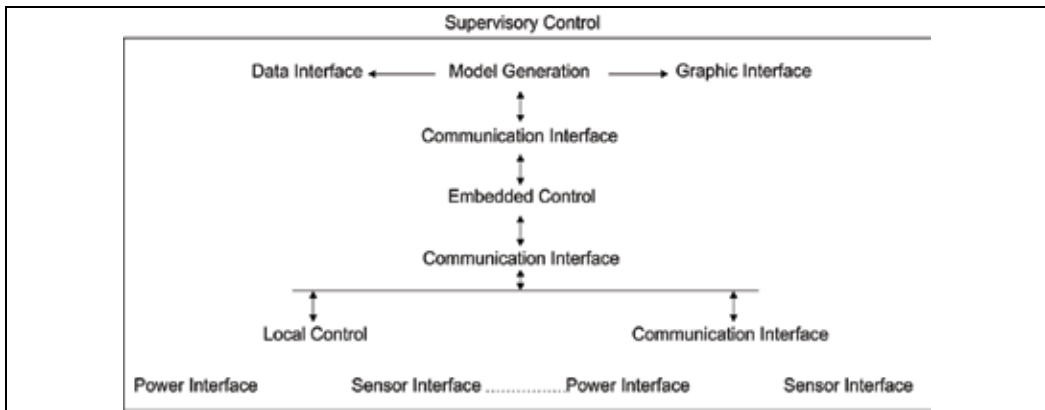


Fig. 23. Stewart-Gough Platform – Control Architecture.

6.2 Embedded level

At the embedded control level two main tasks are implemented: the command decoder and the logic control. The first task decodes commands received by the embedded communication interface (from supervisory control), allowing different actions to be executed according to the received data. The second task generates control signals to actuators' interfaces and receives signals from sensors' interfaces, both located at local control level, so control strategies are implemented in the logic control block.

The prototype uses a FPGA from Altera (Stratix II EP2S60) (Altera, 2008), the configware blocks were implemented in VHDL or Graphic language in the Altera's development platform Quartus II. The embedded control strategies in the logic control block were development using C++ language, in a system-on-a-programmable-chip (SOPC) environment or through the use of blocks implemented in reconfigurable hardware.

6.3 Position control using FPGA

The objective of the proposed controller is to control the linear actuators of the platform. It is able to process the digital signals from the encoders coupled to each linear actuator and the digital signals of the target trajectory. For example, a PID digital controller written in a RST form can be implemented in PLD, with the fitted gain parameters through external programming. The controller's output is a digital signal for the PWM power block. Various implementations of the digital PID (Proportional-Integrative-Derivative) controllers are implemented, and, consequently tuning parameters are necessary for fulfilling the different performance requirements, or to endure different levels of operating noise. A typical implementation of a PID controller can be achieved using a set of differential equations, as follows:

$$U[n] = P[n] + I[n] + D[n], \quad (45)$$

$$P[n] = K_p \cdot e[n], \quad (46)$$

$$I[n] = I[n-1] + \frac{K_p \cdot T_s}{2 \cdot T_i} (e[n] - e[n-1]) \quad (47)$$

$$D[n] = \frac{(pT_s - 2)}{(pT_s + 2)} \cdot D[n-1] + \frac{2 \cdot K_p \cdot T_d}{T_s \cdot (pT_s + 2)} \cdot (e[n] - e[n-1]) \quad (48)$$

where: $U[n]$ is the current control signal resultant, $P[n]$ the current proportional control signal, $I[n]$ the current integral control signal, $D[n]$ is composed of the proportional, derivative and integral parameters (K_p , T_d and T_i) where T_s is the sampling time, respectively. Also, $e[n]$ the current error sample, and finally, $e[n-1]$ the previous error sample. A register error block stores values of $e[n]$ and $e[n-1]$, and makes shift operations ($e[n-1] = e[n]$ and $u[n-1] = u[n]$). An output register block stores $u[n]$ and $u[n-1]$.

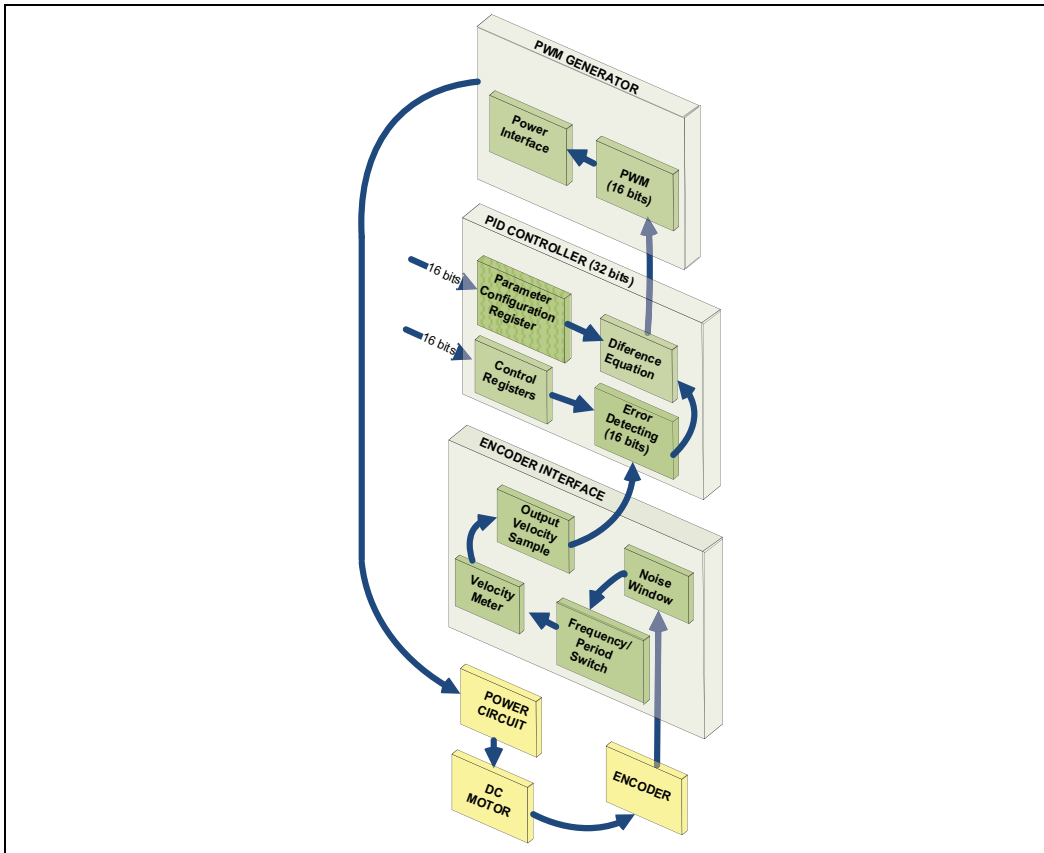


Fig. 24. Embedded DC motor control blocks.

Some of the blocks are described as follows: **Error Detecting** is used for comparing of the reference input and output velocity signals, allowing the generation of a proportional binary word to the error among the periods of the signs. The obtained output of this block is $U[n]$, **Difference Equation** implements the PID digital controller, using the gain parameters (K_p , T_s and T_i) contained in the control input registers. **Control Register** implements the control registers, responsible for the programming of several operational parameters, including the

gain parameters. **PWM and Power Interface** converts the binary word supplied by PID controller in a pattern of digital signs to control the PWM potency block.

The considered reconfiguration in the interface and logical block design eases testing, implementation and future updating, due to this, the development of systems based on reconfigurable computing present well-suited features for developing this kind of problem.

The synchronized control of the actuator system can be easily achieved through the same PLD.

6.4 Prototyping environment

A simulation tool was developed for the 6 DOF parallel manipulator, including motor drives, gear boxes, kinematic and dynamic models, and design of the control system for three axes. Simulations described below consider trajectories issued from the path generation module. The model was tested first in Matlab-Simulink language and the final control hardware implementation was performed in visual programming using LabVIEW™ software (Fig. 25). This last one is used for communication purposes between the program and the control hardware of the prototype.

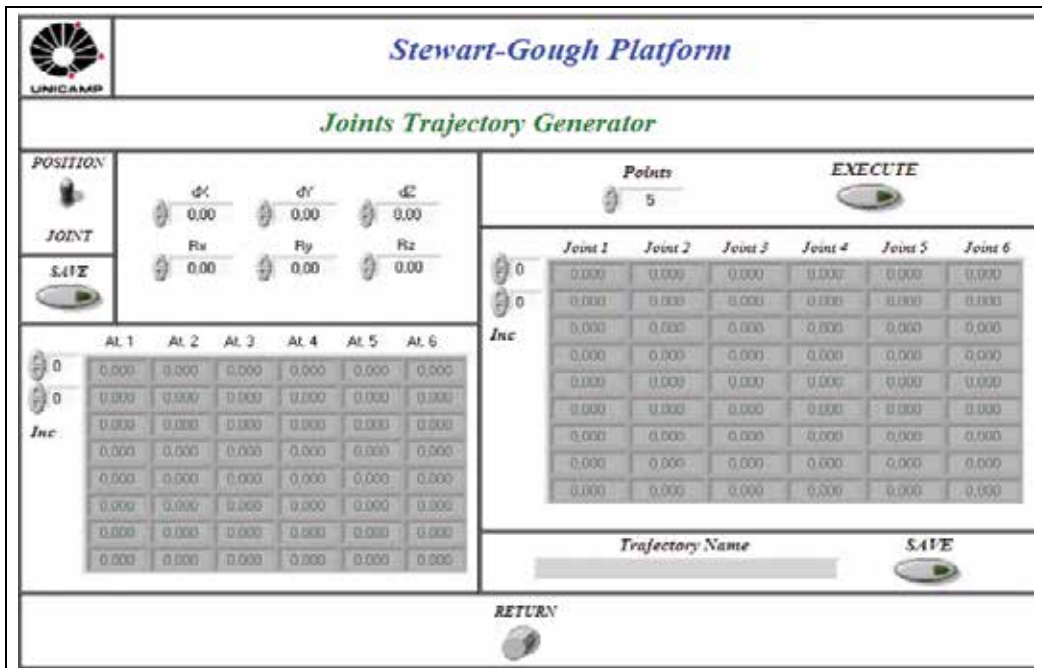
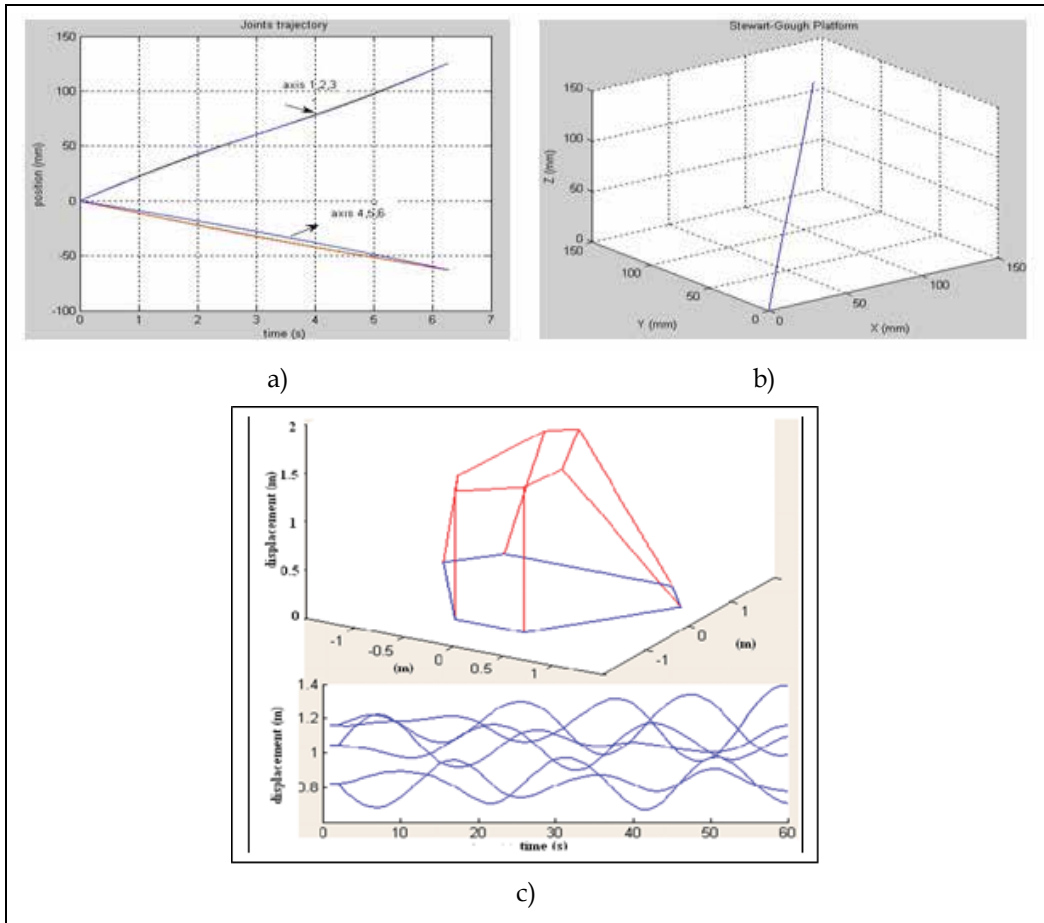


Fig. 25. Model implemented in LabVIEW™.

6.5 Experimental results

The development of a numerical algorithm [8], allows calculating the linear positions for a task defined with respect of the platform center in the Cartesian Space, contains the solution of the inverse kinematics through the use of recursive numerical methods based on the calculation of the kinematics model and of the inverse Jacobian matrix of the manipulator. This algorithm has been validated through different simulations, assessing the behavior of

the trajectory (joint coordinate). For this purpose the kinematics model of the platform was used with six linear joints. Fig. 26a shows the joints movements of each linear actuator and their displacement (45 degrees, approximately) of one point of the upper base of this platform obtained through the inverse kinematics model (Fig. 26b). Fig. 26c shows results of the proposed simulation, obtained with PID axis controllers implemented through FPGA, considering general sea movements and LABVIEW™ experimental platform.



a) Joints evolutions. b) Trajectory description. c) Joint motion

Fig. 26. Kinematics model - Simulation results.

7. Conclusions

This chapter presents the study of kinematics, dynamics and supervision and control of a Stewart-Gough platform, under a reconfigurable architecture concept, considering the division of the system in small functional blocks. This implementation consisted in merging knowledge acquired in multiple areas, and appears as a very promising design strategy for a better reconfiguration capability and portability.

This platform also becomes a powerful benchmark for many research activities, such as the validation of controllers and supervision strategies, model generation and data transmission protocols, among others. For example, the implementation of predictive controllers on this prototype may enable the test of this advanced control strategy under severe conditions of use.

To simplify tests, implementation and future modifications, the use of rapid prototyping functions in the implementation of the interfaces and other logical blocks is emphasized in the proposed prototype. The control block, for example, can benefit of the characteristics of low consumption, high-speed operations, integration capacity, flexibility and simple programming. Some promising aspects of this architecture are:

- Flexibility, as there is a large variety of possible configurations in the implementation of solutions for several problems,
- It is a powerful tool for prototype design, allowing simple solution to control the several sensors and actuators usually present in this kind of projects,
- Possibility of modification of control strategies during operation of the platform,
- The open architecture of this platform enables the use for educational and researches activities.

8. References

- Altera Corporation. <http://www.altera.com>, 2008.
- Bessala, J.; Philippe, B. & Ben Ouezdou, F. (1996). Analytical Study of Stewart Platforms Workspaces. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*. pp. 3179-3184.
- Cappel, K. (1967). Motion simulator. Patent No. 3,295,224 , US Patent No. 3,295,224.
- Clarke D. W., Mohtadi C., Tuffs P.S. (1987) Generalized Predictive Control. Part I. The Basic Algorithm. Part II. *Extensions and Interpretation, Automatica*. Vol.23(2). pp. 137-160.
- Dasgupta, B. & Mruthyunjaya, T. (1998). Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator. *Mechanism and Machine Theory* Vol. 33(8). pp. 135-1152.
- Fasse, E. & Gosselin, C. (1998). On the spatial impedance control of Gough-Stewart platforms. *IEEE International Conference on Robotics and Automation*. pp. 1749-1754.
- Ghobakhloo, A.; Eghtesad, M. & Azadi, M. (2006). Position Control of a Stewart-Gough Platform using Inverse dynamics Method with full dynamics. *International Workshop on Advanced Motion Control, AMC*. Vol. 1. pp. 50-55.
- Gosselin C.; Lavoie, E. & Toutant, P. (1992). An efficient algorithm for the graphical representation of the three-dimensional workspace of parallel manipulators. *Proceeding of the 22nd ASME Mechanisms Conference*. pp. 323-328.
- Gosselin, C. M.; Perreault, L. & Vaillancourt, C. (1999). Simulation and Computer-Aided Kinematic Desing of Tree-Degree-of-Freedom Spherical Parallel Manipulators. *Journal of Robotic Systems*. Vol.12(12). pp. 857-869.
- Gough, V.E. e Whitehall, S. (1962). Universal tyre test machine. *Proceedings of the FISITA Ninth International Technical Congress*. pp. 117-137.
- Guo, H. B. & Li, H. R. (2006). Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator. *Proceedings of the Institution of Mechanical Engineers. Part C, Journal of Mechanical Engineering Science*. Vol. 220(1). pp. 61-72.

- Jaramillo-Botero, A.; Matta-Gomez, A.; Correa-Cacedo, J. F. & Perea-Castro, W. (2006). Robotics Modeling and Simulation Platform. *Robotics & Automation Magazine*. IEEE Vol. 13. pp. 62-73.
- Karger, A. (2003) Architecture singular planar parallel manipulators. *Mechanism and Machine Theory*, Vol. 38, pp. 1149-1164.
- Kim, D. I.; Chung, W. K. & Youmt, Y. (1997). Geometrical Approach for the Workspace of 6-DOF Parallel Manipulators. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. pp. 2986-2991.
- Li, D. & Salcudean, S. E. (1997). Modeling, Simulation, and Control of a Hydraulic Stewart Platform. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. Vol. 4. pp. 3360-3366.
- Liu, M.; Li, C. & Li, C. (2000). Dynamics Analysis of the Gough-Stewart Platform Manipulator. *IEEE Transactions on Robotics and Automation*. Vol.16(1). pp. 94-98.
- MacCallion, H. e. D. P. (1979). The analysis of six degrees of freedom work and station for mechanized assembly. *5th Congress on theory of machines and mechanisms*. pp. 616.
- Nadimi, E. S.; Bak, T. & Izadi-Zamanabadi, R. (2006). Model Predictive Controller Combined with LQG Controller and Velocity Feedback to Control the Stewart Platform. *Advanced Motion Control, 2006. 9th IEEE International Workshop* Vol. 1(0.1109/AMC.2006.1631630). pp. 44-49.
- Ollero, S. Boverie, R. Goodal. Mechatronics. (2005). *Robotics and Components for Automation and Control, Annual Reviews in Control IFAC Journal*. Vol. 25. pp. 203-228.
- Remillard, V. & Boukas, E. (2007). Gough-Stewart Platform Control: A Fuzzy control approach. *Annual meeting of the North American Fuzzy Information Processing Society*. Vol. 1. pp. 108 - 113.
- Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the IMechE*. 180(15). pp. 371-385.
- Su, Y. X.; Duan, B. Y.; Zheng, C. H.; Zhang, Y. F.; Chen, G. D. & Mi, J. W. (2004), 'Disturbance-Rejection High-Precision Motion Control of a Stewart Platform'(3)'IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY'.
- Sugahara, Y.; Ohta, A.; Hashimoto, K.; Sunazuka, H.; Kawase, M.; Tanaka, C.; Lim, H. & Takanishi, A. (2005), 'Walking Up and Down Stairs Carrying a Human by a Biped Locomotor with Parallel Mechanism', *Intelligent Robots and Systems International Conference on Volume*, 1489 - 1494.
- Wang, J.; Gosselin, C. & Cheng, L. (2002). Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach. *Multibody System Dynamics*. Vol. 7(2). pp. 145-170.
- Wendlandt, J. M. & Sastry, S. S. (1994). Design and Control of a Simplified Stewart Platform for Endoscopy. *Proceedings of the 33rd conference on Decision and Control*. Vol.1. pp. 357-362.
- Zhang, Z. & Chen, T. (2007). Modeling and Movement Simulation of a Manipulator of 6-DOF Based on Stewart Platform with Pro/E. *10th IEEE International Conference on Computer-Aided Design and Computer Graphics*. pp. 533 - 536.

Collision Detection and Control of Parallel-Structured Flexible Manipulators Based on Unscented Kalman Filter

Yuichi Sawada, Yusuke Watanabe and Junki Kondo
Kyoto Institute of Technology
Department of Mechanical and System Engineering
Japan

1. Introduction

Flexible manipulators that are lightweight and mechanically flexible are useful for operations in various fields, e.g. space development programs, robotic assistants to humans and so forth. However, the derivation of their exact mathematical model and synthesis of the accurate positioning controller is exceptionally difficult because of the mechanical flexibility. On the other hand, the mechanical flexibility is conducive to the safety in collision between the manipulator and the obstacles. However, it is not positive safety measures that reduction of the influence due to the impact force of collision for the flexible manipulator depends only on its mechanical flexibility. In order to develop flexible manipulators so that they work safely with persons cooperatively, we need to introduce the active collision detection and suspension control algorithms to the flexible manipulators.

The functional requirements of safety in the operations of flexible manipulators are as follows: i) to avoid collisions with obstacles placed or moving in the work space; ii) to detect collisions when unlooked-for obstacles contact with the flexible arm of the manipulator and to suspend the motion as immediately as possible; iii) to plan a new path so as to avoid the place of obstacles.

There are several researches on collision detection methods without extra sensors (A. Garcia & Somolinos, 2003), (M. Kaneko & Tsuji, 1998), (T. Matsumoto & Kosuge, 2000). Moorehead and Wang proposed (Moorehead & Wang, 1996) a collision detection method using strain gauges to determine the intensity and position of external force due to collision with a flexible cantilevered beam. The estimation of the contact position in their approach was achieved by the mechanical relation between positions of the two strain gauges and the bending moments measured by the sensors. Payo et al. (I. Payo & Cortazar, 2009) is produced the method of collision detection and suspend control of the very lightweight single-link flexible arm based on coupling torque feedback. They used the variation of the control torque.

The authors have focused our attention on the second item mentioned above. We already developed a method of collision detection for the single-link flexible manipulator using the innovation process of the Kalman filter (Sawada, 2002a), (Sawada, 2002b), (Sawada, 2002c), (Sawada, 2004 (in Japanese)), (Kondo & Sawada, 2008). Our approach requires no particular

sensors for measuring the contact events between the flexible arm and the obstacles. This collision detection method is based on the observation data for vibration control of the flexible manipulator. The mathematical model of the flexible manipulators is expressed by nonlinear partial differential equations and ordinary differential equations, which is regarded as the infinite-dimensional system. The Kalman filter is constructed for the linearized finite-dimensional model corresponding to the mathematical model of the manipulator.

This chapter describes a method of collision detection and suspend control for parallel-structured flexible manipulators subject to random disturbance using unscented Kalman filter (UKF), which is one of the nonlinear filters. The features of the parallel-structured flexible manipulator are that it holds sufficient rigidity along the vertical axis and mechanical flexibility along the displacement axis of the arm (Sawada & Watanabe, 2007). The exact mathematical model of the parallel-structured flexible manipulator is described by quite complex nonlinear partial and ordinary differential equations, because the manipulator consists of two flexible beams which are disposed parallel. In this chapter, the parallel-structured flexible manipulators are approximately modeled by a flexible arm consisting of a flexible beam with the same boundary conditions as the parallel-structured one.

The approximated model of the flexible manipulator is also a nonlinear system. In order to construct the state estimate for the flexible manipulator, we employed the unscented Kalman filter as the nonlinear state estimator (S. Julier & Durrant-Whyte, 2000), (S.J. Julier & Durrant-Whyte, 1997), (Y.-S. Chen & Wakui, 1989). The UKFs are based on the Monte Carlo method, which have been developed by Simon Julier (S. Julier & Durrant-Whyte, 2000), (S.J. Julier & Durrant-Whyte, 1997) in order to improve the accuracy of the extended Kalman filters. The UKF generates a population of so-called sigma-points on the basis of the current mean and covariance of the state vector. The mean and covariance of the state are calculated using these sigma-points, which means that the algorithm is not necessary to evaluate the Jacobians.

Collision between the parallel-structured flexible manipulator and an undesirable obstacle can be detected using the innovation process of the UKF based on the measured data of strain sensors pasted on the side of the manipulator. The UKF is constructed for the nonlinear state space model corresponding to the parallel-structured flexible manipulator without the impact force term due to the collision. The collision detection function is defined by the strength of the innovation process. The detection algorithm decides that the collision occurs if the collision detection function exceeds a preassigned threshold.

The controller for the manipulator has the following two objectives: i) to rotate the flexible arm from the initial position to the desired position; ii) to safely suspend the rotation of the arm when the collision is detected.

2. Mathematical model of parallel-structured single-link flexible manipulator

Consider a parallel-structured single-link flexible arm with collision illustrated in Fig.1. This arm consists of two uniform Euler-Bernoulli beams with their length ℓ . The end of each beam is clamped to an unit of hub and the other end is to a tip-mass.

Let OXY be the inertial Cartesian coordinate system; Ox_1y the rotating coordinate system around the servomotor shaft at the hub; O_1x_1y and O_2x_2y the rotating coordinate systems for Beam 1 and 2, respectively. $u_i(t, x_i)$ ($i = 1, 2$) denotes the transverse displacement of Beam i from the x_i -axis. Physical parameters of the beams are as follows: ρ the uniform mass

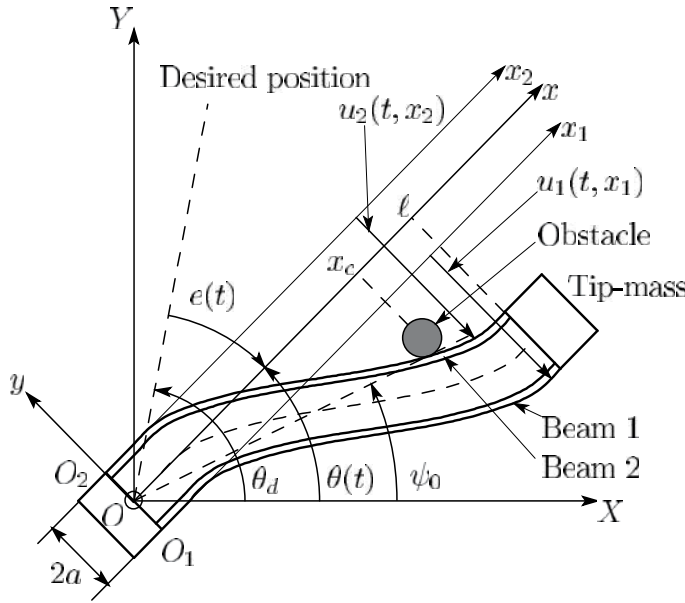


Fig. 1. Parallel-structured single-link flexible arm.

density; S the cross section; EI the uniform flexible rigidity (where E denotes the Young's modulus and I the second moment of cross sectional area); and c_D the coefficient of Kelvin-Voigt type damping. The unit of the hub has the moment of inertia J_0 . The tip-mass has its mass m and the moment of inertia J_1 . It is assumed that the obstacle collides at $x = x_c$ ($0 < x_c < l$) at $t = t_c$, where x_c and t_c are all unknown. Ψ_0 denotes the angle of position which is the contact point between the arm and the obstacle, where Ψ_0 is also unknown.

The exact mathematical model of the parallel-structured single-link flexible arm derived by the Hamilton's principle is highly complex. For the sake of simplicity, the parallel-structured single-link flexible arm consisting of two Euler-Bernoulli type beams is approximately modeled by a single-link flexible arm constructed by a flexible beam with the same boundary conditions as the parallel-structured one (see Fig.2). $u(t, x)$ denotes the transverse displacement of the approximated model from the equilibrium state of the beam. $\theta(t)$ is the angle of the tangential axis of the root of the arm from the X -axis; $e(t)$ the error of the rotation angle $\theta(t)$ from the desired position θ_d , i.e. $e(t) := \theta(t) - \theta_d$.

Now we derive the approximated mathematical model of the parallel-structured single-link flexible arm with the collision based on the simple-structured model using the Hamilton's principle. The position vectors of the arbitrary point of the beam, $r(t, x)$, and the mass center of the tip-mass, $p(t)$, are expressed by

$$r(t, x) = \begin{bmatrix} x \cos \theta(t) - \{u(t, x) - a\} \sin \theta(t) \\ x \sin \theta(t) + \{u(t, x) + a\} \sin \theta(t) \end{bmatrix} \quad (1)$$

$$p(t) = \begin{bmatrix} (\ell + h) \cos \theta(t) - u(t, \ell) \sin \theta(t) \\ (\ell + h) \sin \theta(t) + u(t, \ell) \sin \theta(t) \end{bmatrix}, \quad (2)$$

where $2h$ denotes the length of the tip-mass.

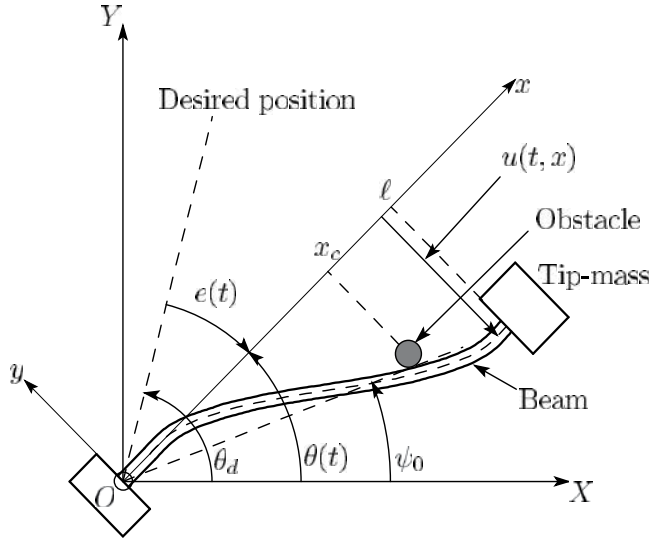


Fig. 2. Simplified structure of parallel-structured single-link flexible arm.

The kinetic energy of rigid part of the arm, $T_R(t)$, is given by the sum of the kinetic energies of the translation motion and rotation of the tip-mass and the rotation of the hub attached to the shaft of the servomotor:

$$T_R(t) = \frac{1}{2} m \|\dot{p}(t)\|^2 + \frac{1}{2} (J_0 + J_1) \dot{\theta}^2(t), \quad (3)$$

where $\|\cdot\|$ denotes the Euclid norm. Similarly, the kinetic energy of the flexible beam is given by

$$T_F(t) = \int_0^l \hat{T}(t, x) dx, \quad (4)$$

where $\hat{T}(t, x)$ represents the kinetic energy density of flexible part defined by

$$\hat{T}(t, x) := \frac{1}{2} \rho S \|\dot{r}(t, x)\|^2. \quad (5)$$

The total kinetic energy of the arm is expressed by the sum of the kinetic energies of the rigid and flexible parts, i.e.,

$$T(t) = T_R(t) + T_F(t). \quad (6)$$

The potential energy of the whole arm is expressed by

$$V(t) = \int_0^l \hat{V}(t, x) dx, \quad (7)$$

where $\hat{V}(t, x)$ is the density function of the potential energy of the flexible part given by

$$\hat{V}(t, x) = \frac{1}{2} EI \left\{ \frac{\partial^2 u(t, x)}{\partial x^2} \right\}. \quad (8)$$

The Hamilton's principle is described by the following equation:

$$\int_{t_1}^{t_2} \{ \delta T(t) - \delta V(t) + \delta W_{nc}(t) + s(t)\delta\psi(t) \} dt = 0, \quad (9)$$

where t_1 and t_2 are arbitrary times; $s(t)$ represents the Lagrange multiplier which is equivalent to the external force generated by the collision with the obstacle; and $\delta W_{nc}(t)$ denotes the virtual work due to the nonconservative forces, e.g. internal damping forces of the beams, the control torque and external disturbances. $\psi(t)$ describes the geometric constrained condition between the unlooked-for obstacle and the flexible beam, i.e.

$$\psi(t) = u(t, x_c) - x_c \tan\{\varphi_0 - \theta(t)\} \equiv 0. \quad (10)$$

Let us assume that $|\varphi_0 - \theta(t)|$ is sufficiently small. We can regard $\tan\{\varphi_0 - \theta(t)\} \cong \varphi_0 - \theta(t)$. Then, (10) can be rewritten into

$$\psi(t) = \int_0^\ell u(t, x)\delta(x - x_c)dx - x_c \{\varphi_0 - \theta(t)\} \equiv 0. \quad (11)$$

$\delta W_{nc}(t)$ is expressed by

$$\begin{aligned} \delta W_{nc}(t) = & -\mu\dot{\theta}(t)\delta\theta + \tau(t)\delta\theta + g_\theta\gamma_\theta(t)\delta\theta \\ & - \int_0^\ell c_D \left(\frac{\partial^3 u(t, x)}{\partial x^2 \partial t} \right) \delta u'' dx + \int_0^\ell g_f \gamma_f(t, x) \delta u dx_i, \end{aligned} \quad (12)$$

where the prime denotes the differentiation with respect to x ; μ denotes the damping coefficient corresponding to the damping force acting at the shaft of the servomotor; $\tau(t)$ the control torque; $\gamma_\theta(t)$ the random disturbance acting at the rotation of the arm; $\gamma_f(t, x)$ the distributed random disturbance along the beam due to the aerodynamic resistance; and g_θ and g_f are constants.

As the generalized coordinates, we consider the following variables: $\theta(t)$, $\dot{\theta}(t)$, $u(t, x)$, $\dot{u}(t, x)$, $u'(t, x)$, $u''(t, x)$, $u'''(t, x)$, $u(t, \ell)$, $\dot{u}(t, \ell)$. Substituting (7), (8) and (11) into (10); in addition, performing a large amount of calculations, we have the following nonlinear differential equations as the mathematical model of the approximated dynamical model corresponding to the parallel-structured single-link flexible manipulator:

$$\begin{aligned} \rho S \frac{\partial^2 u(t, x)}{\partial t^2} + c_D I \frac{\partial^5 u(t, x)}{\partial x^4 \partial t} + EI \frac{\partial^4 u(t, x)}{\partial x^4} = & -\rho S x \ddot{\theta}(t) + \rho S \dot{\theta}^2(t) u(t, x) + g_1 \gamma_f(t, x) \\ & + s(t)\delta(x - x_c) - \left\{ m(\ell + h)\ddot{\theta}(t) + m \frac{\partial^2 u(t, x)}{\partial t^2} - m\dot{\theta}^2(t)u(t, x) \right\} \delta(x - \ell), \end{aligned} \quad (13)$$

where $\gamma_f(t, x)$ the distributed random disturbance modeled by the white Gaussian noise; g_1 and h are constants; $\delta(\cdot)$ denotes the Dirac's delta function; and $s(t)$ the magnitude of collision input. Assuming that the collision occurs momentarily, the magnitude of collision is assumed to be expressed by $s(t) := s_0\delta(t - t_c)$, where s_0 and t_c are all unknown. The initial and boundary conditions of (13) are

$$I.C.: u(0, x) = \frac{\partial u(0, x)}{\partial t} = 0 \quad (14)$$

$$B.C.: u(t, 0) = \frac{\partial u(t, 0)}{\partial x} = \frac{\partial u(t, \ell)}{\partial x} = \frac{\partial^3 u(t, \ell)}{\partial x^3} = 0. \quad (15)$$

The initial condition of the error of rotation $e(t)$ is given by $e(0) = \theta_0 - \theta_i$, where θ_0 is initial angle position of the arm.

The dynamics of rotation is given by the following nonlinear differential equation:

$$\begin{aligned} & \left[J_0 + J_1 + mu^2(t, \ell) + mh(\ell + h) + \rho S \int_0^\ell u^2(t, x) dx \right] \ddot{e}(t) \\ & + \left[\mu_\theta + 2mu(t, \ell)\dot{u}(t, \ell) + \rho S \int_0^\ell 2u(t, x)\dot{u}(t, x) dx \right] \dot{e}(t) + mh\ddot{u}(t, \ell) \\ & - \left[\rho S \int_0^\ell u(t, x) dx - m\ell u(t, \ell) \right] \dot{e}^2(t) - \int_0^\ell x \left[c_D I \frac{\partial^5 u(t, x)}{\partial x^4 \partial t} + EI \frac{\partial^4 u(t, x)}{\partial x^4} \right] dx \\ & + \int_0^\ell g_1 x \gamma(t, x) dx - \tau(t) - g_\theta \gamma_\theta(t) = 0. \end{aligned} \quad (16)$$

The observation data is obtained by means of P strain sensors pasted at $x = \xi_j$, ($j = 1, \dots, P$) and a potentiometer installed at the shaft of the hub, i.e.

$$y_0(t) = c_0 e(t) + e_0 \beta_0(t) \quad (17)$$

$$y_j(t) = c_j \int_{\xi_j}^{\xi_j + b_j} \frac{\partial^2 u(t, x)}{\partial x^2} dx + e_j \beta_j(t), \quad (18)$$

where c_j and e_j are constants; and $\beta_j(t)$, ($j = 0, 1, \dots, P$) represents the observation noise which is modeled by the white Gaussian noise. In order to use the finite-dimensional controller and state estimator, the dynamics of the arm described by (13) and (16) are converted into the stochastic finite-dimensional state space model via the modal expansion technique,

$$u(t, x) = \sum_{k=1}^N u_k(t) \phi_k(x), \quad (19)$$

where $\{u_k(t)\}_{k=1, \dots, N}$ denote the modal displacements; N the large positive number; $\phi_k(x)$ the eigenfunction (mode function) of the following eigenvalue problem with respect to the operator $\mathcal{A} = \{(EI)/(\rho S)\}(d^4/dx^4)$:

$$\mathcal{A}\phi_k(x) = \lambda_k \phi_k(x). \quad (20)$$

Introducing the state vector defined by $v(t) = [u_1(t), \dots, u_N(t), \dot{u}_1(t), \dots, \dot{u}_N(t), e(t), \dot{e}(t)]^T$, the state space model of the approximated flexible arm can be described by the following stochastic differential equation:

$$\dot{v}(t) = A(v)v(t) + b(v)\tau(t) + G(v)\gamma(t) + g_c(v; x_c)s(t) \quad (21)$$

$$y(t) = Cv(t) + E\beta(t), \quad (22)$$

where $\gamma(t) = [\gamma_1(t), \dots, \gamma_N(t), \gamma_\theta(t)]^T$; $\gamma_k(t) = \int_0^\ell \gamma(t, x) \phi_k(x) dx$; $\beta(t) := [\beta_0(t), \beta_1(t), \dots, \beta_p(t)]^T$; $\mathcal{E}\{\gamma(t) \gamma^T(\tau) g\} = W\delta(t - \tau)$; $\mathcal{E}\{\beta(t) \beta^T(\tau)\} = V\delta(t - \tau)$ ($\mathcal{E}\{\cdot\}$: mathematical expectation).

3. Nonlinear state estimation using UKF

The state space model described by (21) and (22) is a stochastic nonlinear system with the collision input. In order to control the tip position and to reduce the random vibration of the whole flexible manipulator, the information of the state $v(t)$ is required. However, the collision input affects as an unknown disturbance. For achieving these purposes, the unscented Kalman filter (UKF) for the following collision-free system is employed:

$$\dot{v}_f(t) = A(v_f)v_f(t) + B(v_f)f(t) + G(v_f)\gamma(t), \quad (23)$$

where $v_f(t)$ denotes the state vector of collision-free system.

The UKF is constructed for the discrete-time nonlinear stochastic system given by equation (23) that is the continuous-time system. Equation (23) and (22) can be converted into the discretized version of the system. By using the Runge-Kutta method, (22) and (23) are rewritten into the following nonlinear discrete-time system:

$$v_f(k+1) = F(v_f(k), \tau(k), \gamma(k)) \quad (24)$$

$$y(k) = Cv(k) + E\beta(k), \quad (25)$$

where k denotes the time-step; Δt the time interval; and $F(\cdot)$ the nonlinear function defined by

$$F(v_f(k), \tau(k), \gamma(k)) = v_f(k) + \frac{\Delta t}{6} \{H_1(v_f(k), \tau(k), \gamma(k)) + 2H_2(v_f(k), \tau(k), \gamma(k)) + 2H_3(v_f(k), \tau(k), \gamma(k)) + H_4(v_f(k), \tau(k), \gamma(k))\} \quad (26)$$

$$H_1(v_f(k), \tau(k), \gamma(k)) = A(v_f(k))v_f(k) + B(v_f(k))\tau(k) + G(v_f(k))\gamma(k)$$

$$H_2(v_f(k), \tau(k), \gamma(k)) = A(v_f(k))\{v_f(k) + H_1(\cdot)\Delta t / 2\} + B(v_f(k))\tau(k) + G(v_f(k))\gamma(k)$$

$$H_3(v_f(k), \tau(k), \gamma(k)) = A(v_f(k))\{v_f(k) + H_2(\cdot)\Delta t / 2\} + B(v_f(k))\tau(k) + G(v_f(k))\gamma(k)$$

$$H_4(v_f(k), \tau(k), \gamma(k)) = A(v_f(k))\{v_f(k) + H_3(\cdot)\Delta t\} + B(v_f(k))\tau(k) + G(v_f(k))\gamma(k).$$

The algorithm of UKF is summarized the three steps as follows:

Step 1. The $(2N + 2)$ -dimensional random variable $v_f(k)$ is approximated by $2(2N + 2) + 1$ sigma points \mathcal{X}_i with weight coefficients W_i .

$$\mathcal{X}_0 = \hat{v}_f(k | k) \quad (27)$$

$$W_0 = \frac{\kappa}{2N + 2 + \kappa} \quad (28)$$

$$\mathcal{X}_i = \hat{v}_f(k|k) + \{\sqrt{(2N+2+\kappa)P(k|k)}\}_i \quad (29)$$

$$W_i = \frac{1}{2(2N+2+\kappa)} \quad (30)$$

$$\mathcal{X}_{i+2N+2} = \hat{v}_f(k|k) - \{\sqrt{(2N+2+\kappa)P(k|k)}\}_i \quad (31)$$

$$W_{i+2N+2} = \frac{1}{2(2N+2+\kappa)}, \quad (i = 1, \dots, 2N+2) \quad (32)$$

where κ denotes the integer scaling parameter; W_i the weight coefficient that is associated with the i -th point and $\{\sqrt{(2N+2+\kappa)P(k|k)}\}_i$ represents the i -th column of the matrix U satisfying $M = UU^T$ if $M := (2N+2+\kappa)P(k|k)$. In this paper, the matrix U is calculated via the incomplete Cholesky decomposition (Saad, 1996).

Step 2. Transform each point through the nonlinear function $F(\cdot)$, and the predicted mean, covariance and observation, the innovation covariance $P_{yy}(k+1|k)$ and the cross correlation matrix $P_{xy}(k+1|k)$ are computed as follows:

$$\mathcal{X}_i(k+1|k) = F(\mathcal{X}_i(k|k), \tau(k), 0) \quad (33)$$

$$\hat{v}_f(k+1|k) = \sum_{i=0}^{2(2N+2)} W_i \mathcal{X}_i(k+1|k) \quad (34)$$

$$P(k+1|k) = \sum_{i=0}^{2(2N+2)} W_i \{\mathcal{X}_i(k+1|k) - \hat{v}_f(k+1|k)\} \\ \times \{\mathcal{X}_i(k+1|k) - \hat{v}_f(k+1|k)\}^T + G W G^T \quad (35)$$

$$\mathcal{Y}_i(k+1|k) = C \mathcal{X}_i(k+1|k) \quad (36)$$

$$\hat{y}_i(k+1|k) = \sum_{i=0}^{2(2N+2)} W_i \mathcal{Y}_i(k+1|k) \quad (37)$$

$$P_{yy}(k+1|k) = \sum_{i=0}^{2(2N+2)} W_i \{\mathcal{Y}_i(k+1|k) - \hat{y}(k+1|k)\} \\ \times \{\mathcal{Y}_i(k+1|k) - \hat{y}(k+1|k)\}^T + E V E^T \quad (38)$$

$$P_{vy}(k+1|k) = \sum_{i=0}^{2(2N+2)} W_i \{\mathcal{X}_i(k+1|k) - \hat{v}_f(k+1|k)\} \\ \times \{\mathcal{Y}_i(k+1|k) - \hat{y}(k+1|k)\}^T. \quad (39)$$

Step 3. The state estimate and covariance are given through updating the prediction by the linear update rule which is specified by the weights chosen to minimize the mean squared error of the estimate. The update rule is

$$\hat{v}_f(k+1|k+1) = \hat{v}_f(k+1|k) + K(k+1)\{y(k+1) - \hat{y}(k+1|k)\} \quad (40)$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)P_{yy}(k+1|k)K^T(k+1|k), \quad (41)$$

where $K(k+1)$ is the Kalman filter gain given by

$$K(k+1) = P_{vy}(k+1|k)P_{yy}^{-1}(k+1|k). \quad (42)$$

4. Collision detection algorithm

It is an undesirable accident that the flexible manipulator collides with an unknown obstacle, because the collision input $s(t)$ affects the state of flexible manipulators as the disturbance. The problem of the collision detection is considered as a detection problem of the abrupt change from the collision-free system to the system with the collision. The change of the systems can be detected using the observation data measured by the piezoelectric sensors pasted at the root of the flexible arm. In other words, the collision is detected by making a decision whether the observation data $y(t)$ is provided from the collision included model or the collision-free model.

For this purpose, the intensity of the innovation process is used. The innovation process of the UKF, $\mu(k)$, is defined by the difference between the actual observation data and the estimated observation data measuring collision-free system, i.e.,

$$\mu(k) = y(k) - C\hat{v}_f(k|k), \quad (43)$$

where $\hat{v}_f(k|k)$ is the estimate of $v_f(k)$ which is calculated by the UKF mentioned in the previous Section. Substituting (25) into (43), we have

$$\mu(k) = Cz(k) + E\beta(k), \quad (44)$$

where $z(k)$ is the estimation error defined by $z(k) := v(k) - \hat{v}_f(k|k)$. If the collision does not occur, the state vector $v(k)$ is equal to $v_f(k)$. However, if the collision occurs, $v(k)$ is equal to the state vector of collision model. In this case, $z(k)$ becomes large because of the collision input. In order to detect the collision, the following scalar function (*collision detection function*) is introduced:

$$r(k) = \mu^T(k)\mu(k). \quad (45)$$

If the collision detection function $r(k)$ exceeds a threshold ε , then the collision has occurred. In fact, it is assumed that the estimation error when the collision has occurred is separated as

$$z(k) = \tilde{z}(k) + \alpha(k), \quad (46)$$

where $\tilde{z}(k)$ is the estimation error of the UKF based on the collision-free system; and $\alpha(k)$ the estimation error caused by the collision input. Substituting (46) into (44), the innovation process $\mu(k)$ is rewritten into

$$\mu(k) = C\tilde{z}(k) + C\alpha(k) + E\beta(k). \quad (47)$$

From equations (45) and (47), the mathematical expectation of the collision detection function $r(k)$ is evaluated by

$$\mathcal{E}\{r(k)\} = \text{tr}\{CP_z(k)C^T\} + \text{tr}\{EVE^T\} + \text{tr}\{C\alpha^T(k)\alpha(k)C^T\}, \quad (48)$$

where $P_z(k) := \mathcal{E}\{\tilde{z}^T(k)\tilde{z}(k)\}$. The first two terms in the right-hand side of (48) is the bias depending on the observation and the system noises. The third term is caused by the collision input which is the deterministic process. If the third term becomes sufficiently large, then $r(k)$ becomes also large at the time when the collision occurs.

5. Position and suspend control

The purpose of controller is to generate the control torque for the servomotor so that the tip position of the flexible manipulator follows the reference trajectory. In this work, the sliding mode controller is employed (Utkin, 1992). The flexible manipulator is controlled so that its state converges to the equilibrium sate.

The sliding mode controller is constructed for the following deterministic collision-free system:

$$\dot{v}_f(t) = A(v_f(t))v_f(t) + B(v_f(t))\tau(t) \quad (49)$$

which has no system noise term. If the flexible manipulator is well controlled, the error state vector $v_f(t)$ is assumed to sufficiently be small, i.e. $\|v_f(t)\| \ll 1$. In this work, we consider that the matrices $A(v_f(t))$ and $B(v_f(t))$ can be approximated as $A(v_f(t)) \cong A(0)$ and $B(v_f(t)) \cong B(0)$. Using these approximations, the error system is rewritten into the following equation:

$$\dot{v}_f(t) = A_e v_f(t) + B_e \tau(t), \quad (50)$$

where A_e and B_e are constant matrices defined by

$$A_e := A(0), \quad B_e := B(0). \quad (51)$$

The objectives of the sliding mode controller are to control the tip position, to suppress the random vibration of the whole manipulator, and to suspend the motion of the manipulator when a collision is detected. The control torque $\tau(t)$ is generated by the sliding mode controller. The output of the controller can be separated into two parts, i.e.:

$$\tau(t) = f_{eq}(t) + f_{nl}(t), \quad (52)$$

where $f_{eq}(t)$ represents the term of linear control input called the equivalent control input in the sliding mode; and $f_{nl}(t)$ the term of nonlinear control input in the reaching mode.

The switching function $\sigma(t)$ is given by

$$\sigma(t) = Sv_f(t), \quad (53)$$

where S represents the gradient of the switching plane. In the sliding mode, the switching function $\sigma(t)$ holds the following conditions:

$$\sigma(t) = 0 \quad (54)$$

$$\dot{\sigma}(t) = 0. \quad (55)$$

The equivalent control input is obtained using (50), (53) and (55) as:

$$f_{eq}(t) = -(SB_e)^{-1}SA_e v_f(t). \quad (56)$$

Substituting (56) into (50), the equivalent control system can be described by

$$\dot{v}_f(t) = \{A_e - B_e(SB_e)^{-1}SA_e\}v_f(t) \equiv \tilde{A}_e v_f(t), \quad (57)$$

where $\tilde{A}_e := A_e - B_e(SB_e)^{-1}SA_e$.

In order to find the switching plane, we consider the cost functional defined by

$$J = \int_0^t \left\{ \int_0^\ell [q_1 \{\dot{u}(s,x)\}^2 + q_2 \{u(s,x)\}^2 + q_3 \{\dot{u}''(s,x)\}^2 + q_4 \{u''(s,x)\}^2] dx + q_5 \dot{e}^2(s) + q_6 e^2(s) \right\} ds, \quad (58)$$

where q_i ($i = 1, \dots, 6$) are positive constants. Substituting the solution of the system described by the equation (13) given by (19) into this, it is rewritten into the following expression:

$$J = \int_0^T v_f^T(t) Q v_f(t) dt, \quad (59)$$

where $Q := \text{diag}\{\Theta_1, \Theta_2, q_5, q_6\}$; $\Theta_1 = q_1 I_N + q_3 \Psi$; $\Theta_2 = q_2 I_N + q_4 \Psi$; (I_N : N -dimensional unit matrix); and

$$\Psi = \begin{bmatrix} \int_0^\ell \phi_1''(x)\phi_1''(x)dx & \dots & \int_0^\ell \phi_1''(x)\phi_N''(x)dx \\ \vdots & & \vdots \\ \int_0^{\ell_i} \phi_N''(x)\phi_1''(x)dx & \dots & \int_0^\ell \phi_N''(x)\phi_N''(x)dx \end{bmatrix}. \quad (60)$$

The gradient of the switching plane S must be decided so that the eigenvalues of \tilde{A}_e becomes stable. There are a method to choose S as a feedback gain of the optimal control. Namely, S is determined as follows (Y.-S. Chen & Wakui (1989)):

$$S = B_e^T P \quad (61)$$

$$PA_e + A_e^T P - PB_e B_e^T P + Q = 0. \quad (62)$$

The nonlinear control input in reaching mode is considered. The sliding mode control is regarded as variable structure control as a required condition. So, using the switching function $\sigma(t)$, the term of nonlinear control input $f_{nl}(t)$ is defined by

$$f_{nl}(t) = -F \text{sgn}(\sigma(t)), \quad (63)$$

where F is the nonlinear controller gain; and $\text{sgn}(\cdot)$ the signum function. Therefore, the control input $f(t)$ is given by

$$\tau(t) = -(SB_e)^{-1} SA_e v_f(t) - F \operatorname{sgn}(\sigma(t)). \quad (64)$$

In order to achieve $\sigma(t) \rightarrow 0$ ($t \rightarrow \infty$), the Lyapunov function for the switching function is chosen as

$$V(t) = \frac{1}{2} \sigma^T(t) \sigma(t). \quad (65)$$

The time derivative of the the Lyapunov function defined by (65) is described with (50) and (64) by

$$\dot{V}(t) = -\sigma^T(t) SB_e F \operatorname{sgn}(\sigma(t)). \quad (66)$$

If $\dot{V}(t) < 0$, the switching function converges to zero. Hence, the nonlinear control gain F must satisfy the following condition:

$$F \begin{cases} > 0 & : \text{ if } SB_e > 0 \\ < 0 & : \text{ if } SB_e < 0. \end{cases} \quad (67)$$

At the neighborhood of the switching plane, the signum function raises the chattering. So, the signum function is approximated as follows:

$$\operatorname{sgn}(\sigma(t)) = \frac{\sigma(t)}{\|\sigma(t)\|} \approx \frac{\sigma(t)}{\|\sigma(t)\| + \delta}, \quad (68)$$

where δ is a positive constant. As a result, (64) is rewritten into

$$\tau(t) = -(SB_e)^{-1} SA_e v_f(t) - F \frac{\sigma(t)}{\|\sigma(t)\| + \delta}. \quad (69)$$

Because of using the unscented Kalman filter, it is necessary that the switching function $\sigma(t)$ and the controller input $\tau(t)$ given by the sliding mode controller with the UKF are converted into the discretized version given by

$$\sigma(k) = S \hat{v}_f(k) \quad (70)$$

$$\tau(k) = -(SB_e)^{-1} SA_e \hat{v}_f(k) - F \frac{\sigma(k)}{\|\sigma(k)\| + \delta}. \quad (71)$$

When the collision occurs, it is necessary that the flexible manipulator is suspended because of absorbing the impact of collision. The proposed flexible manipulator is controlled by tracking the reference trajectory using the sliding mode controller. For suspending the motion of the manipulator, we consider that the reference trajectory (position control) is changed into a steady position when the collision occurs. The angle position of flexible manipulator at the time t_c when the collision occurs is given by

$$\theta(t_c) = \theta_c, \quad (72)$$

After the collision occurred, the reference trajectory is changed into the trajectory given by the following equation:

$$\theta_d(t) \equiv \theta_c. \quad (73)$$

The desired position becomes the position that the flexible manipulator collides with the obstacle. Then, the flexible manipulator can be suspended at this position.

6. Simulation studies

In this section, several numerical results are presented. The flexible beam is assumed to be made with the phosphor bronze. The physical parameters and the coefficients of the flexible manipulator are listed in Table 1. The observation data is supposed to be measured by piezoelectric sensors with their length $b_s = 3 \times 10^{-2}$ [m] and width 1.2×10^{-2} [m] pasted at $\xi_1 = 3 \times 10^{-3}$ [m] and potentiometers installed at each hub. The parameters of the observation system were set as $c_0 = 10$, $c_1 = 1$, $e_0 = e_1 = 1$. The parameter for the UKF κ was set as $\kappa = 1$. The covariance matrices for the system and observation noises were given by $W = 10^{-5} \times I_{2N+2}$, $V = 10^{-8} \times I_2$. The number of modes of the system was set as $N = 2$. The time partition in numerical simulation was set as $\Delta t = 1 \times 10^{-3}$ [s].

The initial condition of state vector were set as $u(0, x) = 0$ [m], $\dot{u}(0, x) = 0$ [m/s], $\dot{\theta}(0) = 0$ [rad/s] and $\theta(0) = 0$ [rad]. The initial condition of the state vector of the control error system was also set as zero. The weight coefficients of cost functional for the hyperplane S were set as the values; $q_1 = 100$, $q_2 = 100$, $q_3 = 100$, $q_4 = 5$, $q_5 = 4500$, $q_6 = 200$. The nonlinear controller gain was $F = 4$, $\delta = 10$ and the simulation study was carried out for 5 seconds.

Parameters	Values
ℓ_1	0.3[m]
E	1.1×10^5 [MPa]
S	2×10^{-5} [m ²]
ρ	8.8×10^3 [kg/m ²]
c_D	4.84×10^7 [N·s/m ²]
J_0	5 [kg·m ²]
J_1	0.08 [kg·m ²]
m	0.61 [kg]
h	0.026 [m]
g_1	0.05
g_2	0.4

Table 1. Physical parameters of the flexible manipulator.

6.1 Position control

The simulation results of the position control in the collision-free case are shown in Figs.3-6. Fig.3 depicts the angle $\theta(t)$ and its estimate computed by the UKF $\hat{\theta}(t)$. The estimation error with respect to $\theta(t)$ sufficiently small. The controlled angle has converged at the desired position using the UKF based sliding mode control. Figs. 4 and 5 presents the observation

data of the strain measured by the piezoelectric sensor $y_1(t)$ and the displacement of the tip mass $u(t, \ell)$, respectively. Furthermore, Fig.5 also depicts the estimate of the tip-mass displacement $\hat{u}(t, \ell)$ which is calculated by $\hat{u}(t, \ell) = \sum_{i=1}^N \hat{u}_i(t) \phi_i(\ell)$. The estimation error of the tip-displacement based on the noisy observation data (see Figure 4) is adequately small for suppressing the vibration of the tip-mass. Fig.6 shows the response of the control torque $\tau(t)$.

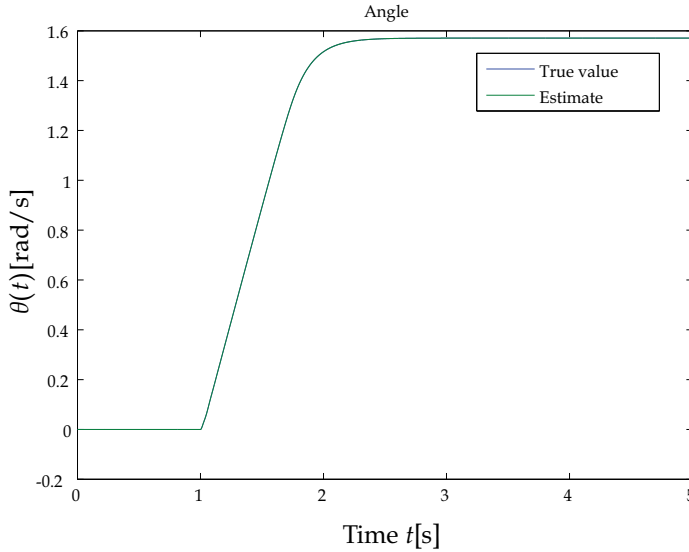


Fig. 3. Behavior of the rotational angle $\theta(t)$ and its estimate $\hat{\theta}(t)$ obtained using the UKF in the collision-free case. The solid line and the dashed line depict the true state of the angle $\theta(t)$ and its estimate $\hat{\theta}(t)$, respectively

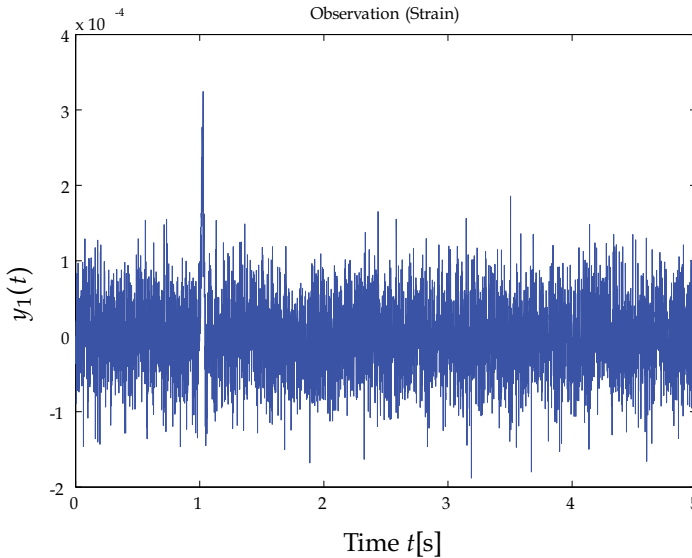


Fig. 4. Observation data of the strain measured by the piezoelectric sensor, $y_1(t)$ in the collision-free case.

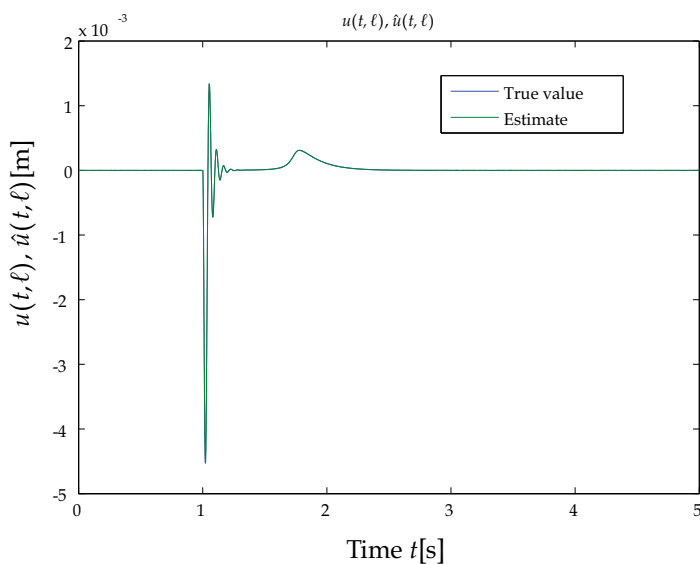


Fig. 5. Displacement of the tip mass $u(t, \ell)$ and its estimate $\hat{u}(t, \ell)$.

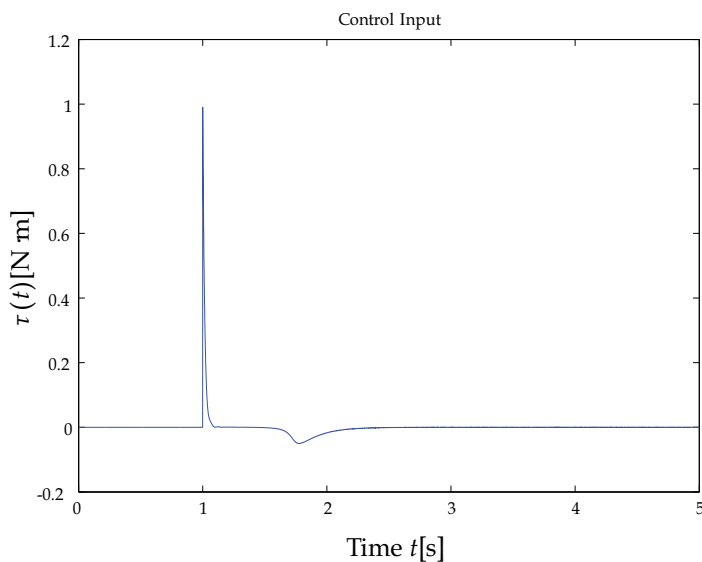


Fig. 6. Control torque $\tau(t)$ generated by the sliding mode controller in the collision-free case.

6.2 Collision detection and suspend control

The simulation results in the collision case are shown in Figs.7-11. In this case, the desired position θ_d was set as the same as in the collision-free case. Figure 7 shows the behavior of the collision detection function given by equation (45). We considered that the collision between the parallel-structured single-link flexible manipulator and the unlooked-for obstacle occurs at $t_c = 1.24[s]$. The value of $r(t)$ before the collision occurs is very small. When the collision occurs at $t = t_c$, the value of $r(t)$ abruptly increases. As seen in Fig. 8, the

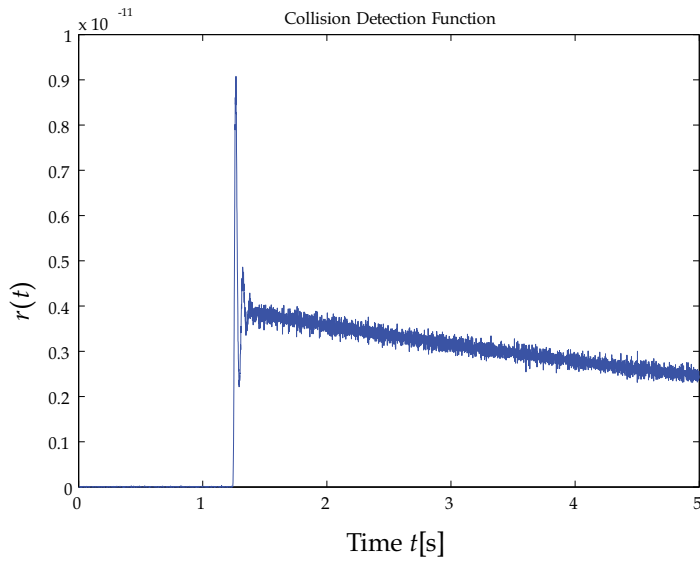


Fig. 7. Behavior of the collision detection function $r(t)$ generated by the innovation process of the UKF in the collision case. The collision occurs at $t_c = 1.24$ [s].

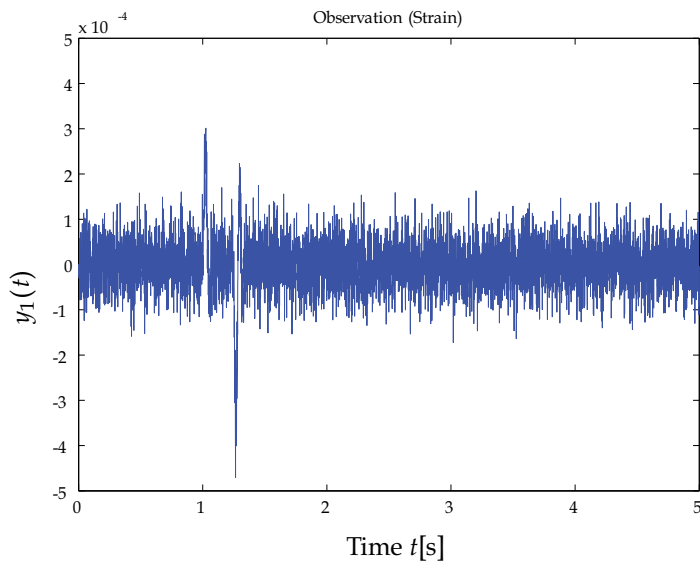


Fig. 8. Observation data of the strain measured by the piezoelectric sensor, $y_1(t)$ in the collision case.

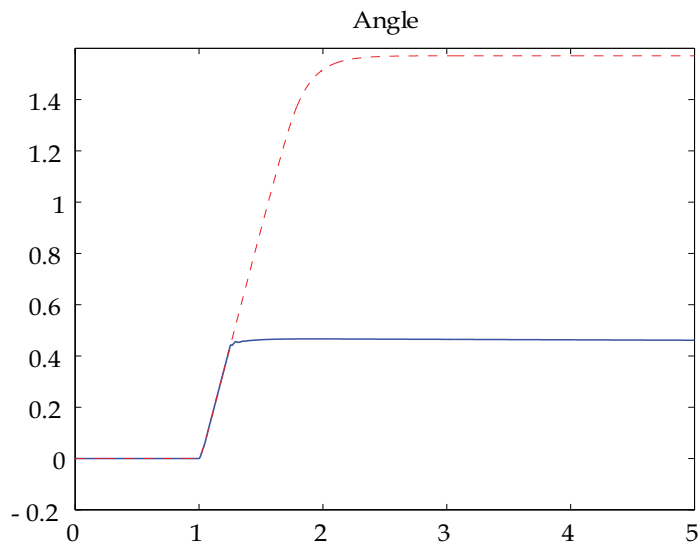


Fig. 9. Behavior of the rotational angle $\theta(t)$ obtained using the UKF in the collision-free and the collision cases. The solid line and the dashed line depict the angle $\theta(t)$ in the collision and collision-free cases, respectively.

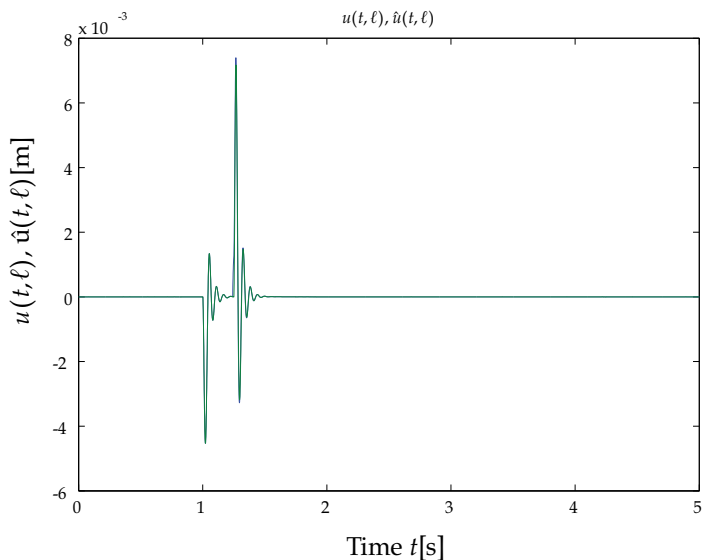


Fig. 10. Displacement of the tip mass $u(t, \ell)$ and its estimate $\hat{u}(t, \ell)$.

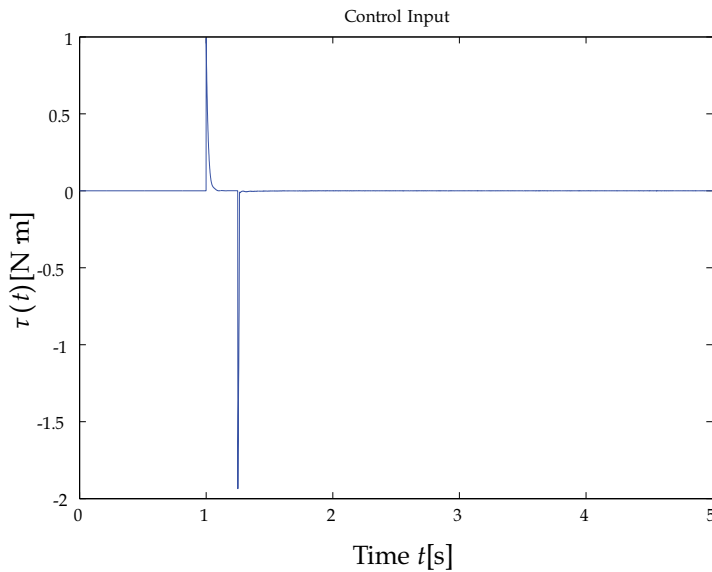


Fig. 11. Control torque $r(t)$ generated by the sliding mode controller in the collision case.

observation data of the strain is very noisy. From Figs. 7 and 8, we can see that the collision detection function can detect the weak collision based on the noisy observation data.

The results of the suspend control are shown in Figs. 9 and 10. The rotation of the manipulator was interrupted when the collision has been detected using the collision detection function (see Figure 9). After $r(t)$ exceeded the preassigned threshold $\varepsilon = 1 \times 10^{-12}$, the desired position has been changed from θ_d to $\theta(t_c)$. The displacement of the tip-mass $u(t, \ell)$ shows the vibration with large amplitude after the collision was detected (Fig. 10). The control torque is depicted in Fig. 11. This figure explains that the control torque when the motion of the manipulator was suspended requires the torque of large value.

7. Conclusions

This chapter has presented the new collision detection method and the suspend control of parallel-structured single-link flexible manipulators using the unscented Kalman filter and the sliding mode control. The main result is that the collision detection was achieved using the innovation process of the UKF which is one of the nonlinear filters. Furthermore, the high performance suspend control has been constructed using the sliding mode controller based on the UKF. The proposed approach brings an advantage that the system model requires no linearization. In our previous work, the linearized mathematical model was required because of using the Kalman filter and the LQG controller for collision detection

and control. The proposed collision detection method can be applied to the multi-link flexible manipulators, which have strong nonlinearity.

8. References

- García, A.; Feliu, V. & Somolions, J. A. (2003). Experimental testing of a gauge based collision detection mechanism for a new three-degree-of-freedom flexible robot, *Journal of Robotics Systems* Vol. 20(No. 6): 271–284.
- Payo, I.; Feiu V. & Cortazar, O. D. (2009). Force control of very lightweight single-link flexible arm based on coupling torque feedback, *Mechatronics* Vol. 19: 334–337.
- Kondo, J. & Sawada, Y. (2008). Collision detection and suspend control of parallel-structured single-link flexible arms, *Proceedings of SICE Annual Conference 2008, Tokyo* pp. 3250–3255.
- Kaneko, M.; Kanayama, N. & Tsuji, T. (1998). Active antenna for contact sensing, *IEEE Transactions on Robotics and Automation* Vol. 14(No. 2): 278–291.
- Moorehead, S. & Wang, D. (1996). Active antenna for contact sensing, *Proceedings of IEEE International Conference on Robotics and Automation* pp. 804–809.
- Julier, S.; Uhlmann, J. & Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators, *Transaction on Automatic Control* Vol. 45(No. 3): 477–482.
- Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company.
- Sawada, Y. (2002a). Collision detection for a flexible cantilever-beam subject to random disturbance based on innovation process, *Proceedins of IEEE International Conference on Control Applications* pp. 1171–1176.
- Sawada, Y. (2002b). Collision estimation for a flexible cantilevered-beam subject to random disturbance, *Proceedins of 34th ISCIE Int. Symp. on Stochastic Systems Theory and Its Application* pp. 183–188.
- Sawada, Y. (2002c). Detection of collisions for a flexible beam subject to random disturbance, *Proceedins of 41st SICE Annual Conference* pp. 268–273.
- Sawada, Y. (2004 (in Japanese)). On-line collision detection for flexible cantilevered beams using innovation process, *Transaction of The Institute of Systems, Control and Information Engineers* pp. 349–357.
- Sawada, Y. & Watanabe, T. (2007). Lqg control of a parallel-structured single-link flexible arm, *Proceedings of 51st Annual Conference of ISCIE* pp. 372–373.
- Julier, S.; Uhlmann, J. & Durrant-Whyte, H. F. (1997). A new extension of the kalman filter to nonlinear systems, *Proceedings of AeroSense: 11th International Symposium Aerospace/Defense Sensing, Simulation Control*.
- Matsumoto, T. & Kosuge, K. (2000). Collision detection of manipulator based on adaptive control law, *Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechanics* pp. 177–182.
- Utkin, V. I. (1992). *Sliding modes in optimization and control problems*, Springer, New York.

Chen, Y.-S.; Ikeda, H.; Mita, T. & Wakui, S. (1989). Trajectory control of robot arm using sliding mode control and experimented results, *Journal of the Robotics Society of Japan* Vol. 7(No. 6): 706-711.

On Saturated PID Controllers for Industrial Robots: The PA10 Robot Arm as Case of Study

Jorge Orrante-Sakanassi, Víctor Santibáñez and Ricardo Campa
*Instituto Tecnológico de la Laguna
Mexico*

1. Introduction

Industrial robots are naturally equipped with classical PID controllers, which theoretically assure semi-global asymptotic stability of the closed-loop system equilibrium for the regulation case (see, e.g., Arimoto & Miyazaki (1984), Arimoto et al., (1990), Kelly (1995b), Ortega et al., (1995), Alvarez-Ramirez et al., (2000), Kelly et al., (2005), Meza et al., (2007)). Uniform ultimate boundedness of the closed-loop solutions can be concluded when the desired position is a function of time (some stability analyzes for this case can be found in the works of Kawamura et al. (1988), Wen & Murphy (1990), Qu & Dorsey (1991), Rocco (1996), Cervantes & Alvarez-Ramirez (2001), Choi & Chung (2004), and Camarillo et al., (2008)), but to the authors' knowledge, so far there is not a proof of global regulation for such controller.

In the search of a practical globally stable PID regulator, some nonlinear control structures based on the classical PID controller, which assure global asymptotic stability of the closed-loop system, have emerged. Some works that deal with global nonlinear PID regulators based on Lyapunov theory and passivity theory have been reported by Arimoto (1995), Kelly (1998), Santibáñez & Kelly (1998a), and Meza & Santibáñez (1999). Recently, a particular case of the class of nonlinear PID global regulators originally proposed in (Santibáñez & Kelly, 1998a) was presented by Sun et al., (2009).

On the other hand, it is well known that saturation phenomena in robot control systems are intrinsically present when the actuators are driven by sufficiently large control signals. If these physical constraints are not considered in the controller design they may lead to a lack of the stability properties.

Even though no one of the controllers mentioned above considers the influence of the saturation phenomena, there are some works that have been reported to solve this saturation problem in PD-like controllers for the case of regulation tasks (Kelly & Santibáñez, 1996; Colbaugh et al., 1997a; Loria et al., 1997; Santibáñez & Kelly, 1997; 1998b). Solutions without considering velocity measurements and with gravity compensation are treated in (Loria et al., 1997). A full-state (position and velocity) feedback solution with adaptive gravity compensation is presented in (Zergeroglu et al., 2000). More recently, new schemes dealing with this regulation problem of robot manipulators with bounded inputs have been presented by Zavala & Santibáñez (2006), Zavala & Santibáñez (2007), Dixon (2007), Alvarez-Ramirez et al., (2003), and Alvarez-Ramirez et al., (2008). An adaptive

approach involving task-space coordinates, and considering the uncertainties of the kinematic model of the robot manipulator is proposed in Dixon (2007). Also, for the bounded input tracking case, the following works have appeared in the control literature: Loria & Nijmeijer (1998), Dixon et al., (1999), Santibañez & Kelly (2001), Moreno et al., (2008a), Moreno et al., (2008b), Aguinaga-Ruiz et al., (2009), Zavala-Rio et al., (2010).

Few saturated PID controllers (that is, bounded PID controllers taking into account the actuator torque constraints) have been reported: for the case of semiglobal asymptotic stability, a saturated linear PID controller was presented in (Alvarez-Ramirez et al., 2003) and (Alvarez-Ramirez et al., 2008); for the case of global asymptotic stability, saturated nonlinear PID controllers were introduced in (Gorez, 1999; Meza et al., 2005; Santibañez et al., 2008). The work introduced by Gorez (1999) was the first bounded PID-like controller in assuring global regulation; the latter works, introduced in (Meza et al., 2005) and (Santibañez et al., 2008), also guarantee global regulation, but with the advantage of a controller structure which is simpler than that presented in Gorez (1999). A local adaptive bounded regulator was presented by Laib (2000).

Most of nonlinear PID global regulators for robot manipulators are based on the energy-shaping methodology. There are two approaches: those controllers which do not take into account the effects of actuator saturations, and those which consider the saturation phenomena introduced only by the actuators. However, the actuators are not the only components of the closed-loop system that produce saturation constraints; there exist other devices, such as the servo-drivers and the output electronics of the control computer, presenting saturation effects.

In the practice, industrial robots are equipped with a position control computer which produces the commands of desired joint velocities to the joint actuator servo-drivers. In such a sense, Santibañez et al. (2010) recently proposed a new saturated nonlinear PID regulator for robot manipulators that considers the saturation phenomena of both the control computer, the velocity servo-drivers and the torque constraints of the actuators. The structure of this controller is closer to the structure of the practical PID controllers used in the industry. Fig. 1 shows the scheme that was considered to design such saturated nonlinear PID controller; in this figure the constraints over the input and output commands of the servo driver and the torque constraints of the actuators are clearly shown. Notice that because a cascade connection of two saturation blocks can be reduced to only one saturation function, and for simplicity, the saturation of the velocity PI loop and the saturation of the actuators, are both represented by one saturation block in Fig. 1; also, the driver is assumed to have an ideal inner torque controller. In such a work a proportional outer position loop and a PI inner velocity loop constitute the main structure of the controller, which is intrinsic to the industrial robots if we consider the typical low-level controllers in the actuator servo-drivers.

The contribution of this chapter is twofold: first, we present a variant of the work presented by Santibañez et al. (2010), where now the controller is composed by a saturated velocity proportional (P) inner loop, provided by the servo-driver, and a saturated position proportional-integral (PI) outer loop, supplied by the control computer (see Fig. 2). Such a controller also has a structure that naturally matches that of the practical industrial robots. Secondly, we present an experimental evaluation on the PA10-7C robot arm, comparing the nonlinear PID regulator previously reported in Santibañez et al. (2010) and the controller proposed in this chapter.

By following similar steps as those given in Santibañez et al. (2010) we employ the singular perturbation theory to analyze the exponential stability of the equilibrium of the closed-

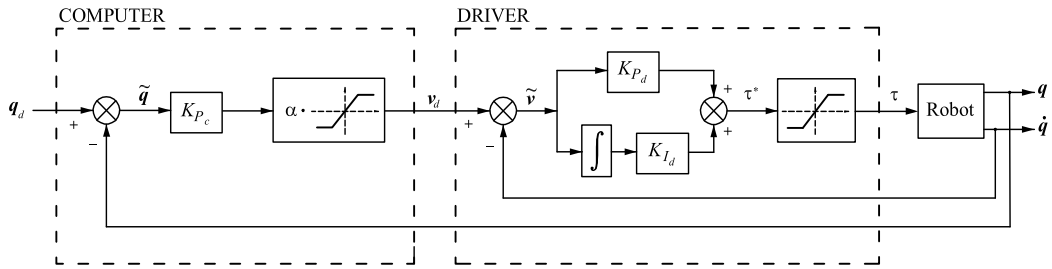


Fig. 1. Practical nonlinear PID controller with bounded torques for robot manipulators.

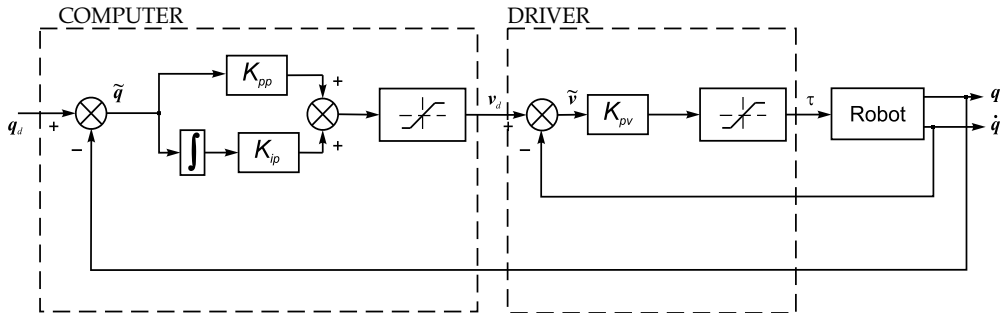


Fig. 2. Variant of the practical PID controller with bounded torques for robot manipulators.

loop system. This result guarantees that exponential stability of the classical PID linear regulator in industrial robots is preserved even though the saturation phenomena due to the electronic devices and/or the actuators are present.

The remainder of this chapter is organized as follows: Section 2 states the dynamic model of a serial n -link rigid robot manipulator in open-loop, some of its properties, as well as some considerations, assumptions and definitions that are useful throughout the analysis. The proposed control scheme is presented in Section 3. Section 4 shows the singularly perturbed system to analyze. Section 5 states the stability analysis and proves that the control objective is achieved. Section 6 is devoted to the real-time experimental evaluation carried out on the PA-10 robot arm. The conclusions of the work are presented in Section 7.

Throughout this chapter, we use the notation $\lambda_{\min}\{A(x)\}$ and $\lambda_{\max}\{A(x)\}$ to indicate the smallest and largest eigenvalues, respectively, of a symmetric positive definite bounded matrix $A(x)$, for any $x \in \mathbb{R}^n$. Also, we define $\lambda_{\min}\{A\}$ as the greatest lower bound (infimum) of $\lambda_{\min}\{A(x)\}$, for all $x \in \mathbb{R}^n$, that is, $\lambda_{\min}\{A\} = \inf_{x \in \mathbb{R}^n} \lambda_{\min}\{A(x)\}$. Similarly, we define $\lambda_{\max}\{A\}$ as the least upper bound (supremum) of $\lambda_{\max}\{A(x)\}$, for all $x \in \mathbb{R}^n$, that is, $\lambda_{\max}\{A\} = \sup_{x \in \mathbb{R}^n} \lambda_{\max}\{A(x)\}$. The norm of vector x is defined as $\|x\| = \sqrt{x^T x}$ and that of matrix $A(x)$ is defined as the corresponding induced norm $\|A(x)\| = \sqrt{\lambda_{\max}\{A(x)^T A(x)\}}$.

2. Preliminaries

2.1 Robot dynamics

The dynamics of a serial n -link rigid robot, without the effect of friction, can be written as (Spong & Vidyasagar, 1989):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1)$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ are the vectors of joint positions, velocities and accelerations, respectively, $\tau \in \mathbb{R}^n$ is the vector of applied torques, $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive-definite inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the matrix of centripetal and Coriolis torques, and $g(q) \in \mathbb{R}^n$ is the vector of gravitational torques obtained as the gradient of the robot potential energy $\mathcal{U}(q)$, i.e.

$$g(q) = \frac{\partial \mathcal{U}(q)}{\partial q}. \quad (2)$$

We assume that all the joints of the robot are of the revolute type.

2.2 Properties of the robot dynamics

We recall two important properties of dynamics (1) which are useful in our paper:

Property 1. The matrix $C(q, \dot{q})$ and the time derivative $\dot{M}(q)$ of the inertia matrix satisfy (Koditschek, 1984; Ortega & Spong, 1989):

$$\dot{q}^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{q} = 0 \quad \forall q, \dot{q} \in \mathbb{R}^n.$$

◇

Property 2. The gravitational torque vector $g(q)$ is bounded for all $q \in \mathbb{R}^n$. This means that there exist finite constants $\gamma_i \geq 0$ such that (Craig, 1998):

$$\sup_{q \in \mathbb{R}^n} |g_i(q)| \leq \gamma_i \quad i = 1, 2, \dots, n, \quad (3)$$

where $g_i(q)$ stands for the i -th element of $g(q)$. Equivalently, there exists a constant k' such that $\|g(q)\| \leq k'$, for all $q \in \mathbb{R}^n$. Furthermore, there exists a positive constant k_g such that

$$\left\| \frac{\partial g(q)}{\partial q} \right\| \leq k_g,$$

for all $q \in \mathbb{R}^n$, and $\|g(x) - g(y)\| \leq k_g \|x - y\|$, for all $x, y \in \mathbb{R}^n$. Moreover, a simple way to compute k_g is:

$$k_g \geq n \left(\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right) \quad \text{where } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, n. \quad (4)$$

A less restrictive constant k_{g_i} can be computed by:

$$k_{g_i} \geq n \left(\max_{j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right) \quad \text{where } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, n. \quad (5)$$

◇

2.3 Useful theorems

Here, we recall two versions of the Mean-Value Theorem, which are key in finding the less restrictive constants k_{g_i} related with the gravitational torque vector.

Theorem 1. [Kelly et al., (2005), p. 384] Consider the continuous function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. If $f(z_1, z_2, \dots, z_n)$ has continuous partial derivatives then, for any constant vectors $x, y \in \mathbb{R}^n$, we have

$$f(x) - f(y) = \begin{bmatrix} \left. \frac{\partial f(z)}{\partial z_1} \right|_{z=\xi} \\ \left. \frac{\partial f(z)}{\partial z_2} \right|_{z=\xi} \\ \vdots \\ \left. \frac{\partial f(z)}{\partial z_n} \right|_{z=\xi} \end{bmatrix}^T [x-y] \tag{6}$$

where $\xi \in \mathbb{R}^n$ is a vector suitably chosen on the line segment which joins vectors x and y . ◇

Theorem 2. [Kelly et al. (2005), p.385] Consider the continuous vectorial function $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$. If $f_i(z_1, z_2, \dots, z_n)$ has continuous partial derivatives for $i = 1, \dots, m$, then, for each pair of vectors $x, y \in \mathbb{R}^n$ and each $\omega \in \mathbb{R}^m$ there exists $\xi \in \mathbb{R}^n$ such that:

$$[f(x) - f(y)]^T \omega = \omega^T \left. \frac{\partial f(z)}{\partial z} \right|_{z=\xi} (x-y), \tag{7}$$

where $\xi \in \mathbb{R}^n$ is a vector on the line segment that joins vectors x and y . ◇

2.4 Problem formulation

Before presenting the formulation of the control problem, we recall some useful definitions.

Definition 1. The hard saturation function is denoted by $\mathbf{sat}(x;k) \in \mathbb{R}^n$, where

$$\mathbf{sat}(x;k) = \begin{bmatrix} \mathbf{sat}(x_1;k_1) \\ \mathbf{sat}(x_2;k_2) \\ \vdots \\ \mathbf{sat}(x_n;k_n) \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad k = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix},$$

with k_i being the i -th saturation limit, $i = 1, 2, \dots, n$, and each element of $\mathbf{sat}(x;k)$ is defined as:

$$\mathbf{sat}(x_i;k_i) = \begin{cases} x_i & \text{if } |x_i| \leq k_i \\ k_i & \text{if } x_i > k_i \\ -k_i & \text{if } x_i < -k_i \end{cases}$$

◇

Furthermore, the control scheme proposed in this chapter involves special saturation functions which fit in the following definition.

Definition 2. [Zavala & Santibañez (2006)] Given positive constants l and m , with $l < m$, a function $\text{Sat}(x; l, m) : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \text{Sat}(x; l, m)$ is said to be a strictly increasing linear saturation function for (l, m) if it is locally Lipschitz, strictly increasing, C^2 differentiable and satisfies:

1. $\text{Sat}(x; l, m) = x$ when $|x| \leq l$
2. $|\text{Sat}(x; l, m)| < m$ for all $x \in \mathbb{R}$. ◇

For instance, the following saturation function is a special case of the linear saturation given in Definition 2:

$$\text{Sat}(x; l, m) = \begin{cases} -l + (m-l) \tanh\left(\frac{x+l}{m-l}\right) & \text{if } x < -l \\ x & \text{if } |x| \leq l \\ l + (m-l) \tanh\left(\frac{x-l}{m-l}\right) & \text{if } x > l \end{cases} \quad (8)$$

n saturation functions of the form (8) can be joined together in an $n \times 1$ saturation function vector denoted by $\mathbf{Sat}(x; l, m)$, i.e.,

$$\mathbf{Sat}(x; l, m) = \begin{bmatrix} \text{Sat}(x_1; l_1, m_1) \\ \text{Sat}(x_2; l_2, m_2) \\ \vdots \\ \text{Sat}(x_n; l_n, m_n) \end{bmatrix},$$

where $x, l, m \in \mathbb{R}^n$, that is,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad l = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{bmatrix}, \quad m = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}.$$

Consider the robot dynamic model (1). Assume that each joint actuator is able to supply a known maximum torque τ_i^{\max} so that:

$$|\tau_i| \leq \tau_i^{\max}, \quad i = 1, 2, \dots, n \quad (9)$$

where τ_i stands for the i -th entry of vector τ . In other words, if u_i represents the control signal (controller output) before the actuator, related to the i -th joint, then

$$\tau_i = \tau_i^{\max} \text{sat}\left(\frac{u_i}{\tau_i^{\max}}\right), \quad (10)$$

for $i = 1, \dots, n$, where $\text{sat}(\cdot)$ is the standard hard saturation function. We also assume:

Assumption 1. The maximum torque τ_i^{\max} of each actuator satisfies the following condition:

$$\tau_i^{\max} > \gamma_i, \tag{11}$$

where γ_i was defined in Property 2, with $i = 1, 2, \dots, n$. ◇

This assumption means that the robot actuators are able to supply torques in order to hold the robot at rest for all desired joint positions $q_d \in \mathbb{R}^n$.

The control problem is to design a controller to compute the torque $\tau \in \mathbb{R}^n$ applied to the joints, satisfying the constraints (9), such that the robot joint positions q tend asymptotically toward the constant desired joint positions q_d .

3. Proposed control scheme

In this section we present a nonlinear PID controller which can be seen as a practical version of the classical PID control of robot manipulators.

The proposed nonlinear PID controller has the form:

$$\tau = \text{Sat}[K_{pv}[\text{Sat}(K_{pp}\tilde{q} + w^*; l_{pi}^*, m_{pi}^*) - \dot{q}]; l_p, m_p] \tag{12}$$

$$w^* = K_{ip} \int_0^t \tilde{q} \, dr \tag{13}$$

where K_{pv} , K_{pp} and K_{ip} are diagonal positive definite matrices. This control law is formed by two loops: an outer joint-position proportional-integral PI loop and an inner joint-velocity proportional P loop, and considers the saturation effects existing in the output of the control stage (see Figure 2), where $\text{Sat}[K_{pv}[\text{Sat}(K_{pp}\tilde{q} + w^*, l_{pi}^*, m_{pi}^*) - \dot{q}]; l_p, m_p]$ is a vector where each element is a saturation function as in Definition 2 for some (l_p, m_p) , where l_p and m_p are vectors with elements l_{pi} and m_{pi} respectively, and $i = 1, 2, \dots, n$. The control law (12) can be rewritten as:

$$\tau = \text{Sat}[\text{Sat}(K_p\tilde{q} + w; l_{pi}, m_{pi}) - K_v\dot{q}; l_p, m_p] \tag{14}$$

$$w = K_i \int_0^t \tilde{q} \, dr \tag{15}$$

where

$$K_p = K_{pv}K_{pp}, K_i = K_{pv}K_{ip}, K_v = K_{pv}, l_{pi} = K_{pv}l_{pi}^*, m_{pi} = K_{pv}m_{pi}^*$$

and the following assumption is satisfied.

Assumption 2: The saturation limits of the PI and P loops satisfy:

$$\gamma_i < l_{pi} < m_{pi} \tag{16}$$

$$\gamma_i < l_{pi} < m_{pi} < \tau_i^{\max}. \tag{17}$$

◇

Remark: In practice, the saturation constraints of the electronic devices and the actuators are, in fact, hard saturations like those in Definition 1. However, with the end of carrying out the stability analysis, they can be approximated by linear saturation functions like those defined in Definition 2, with $l < m$, and l arbitrarily close to m .

In order to simplify the notation, henceforth, we will omit, in the argument, the limits of the saturation functions.

4. Singularly perturbed system

4.1 Closed-loop system

By substituting (14) into the robot dynamics (1), we obtain

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{q} \\ w \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [\text{Sat}[\text{Sat}(K_p \tilde{q} + w) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \\ K_i \tilde{q} \end{bmatrix} \quad (18)$$

which is an autonomous differential equation with a unique equilibrium point given by $[\tilde{q}^T \dot{q}^T w^T]^T = [0^T 0^T g(q_d)^T]^T \in \mathbb{R}^{3n}$, where we have used Assumption 2, and (3), to get that $\text{Sat}(\text{Sat}(w)) - g(q_d) = 0$ implies $w = g(q_d)$. In order to move the equilibrium point of (18) to the origin, we apply the change of variables $x = w - g(q_d)$. Now the new closed-loop system is given by:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{q} \\ x \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \\ K_i \tilde{q} \end{bmatrix}. \quad (19)$$

The previous closed-loop system can be studied as a singularly perturbed system. To this end, system (19) can be described as two first-order differential equations as follows:

$$\frac{d}{dt} x = K_i \tilde{q} \quad (20)$$

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \end{bmatrix}. \quad (21)$$

Moreover, by choosing the integral gain matrix as $K_i = \varepsilon K_i^*$, where K_i^* is a diagonal positive-definite matrix and $\varepsilon > 0$ is a small parameter, and letting $t' = \varepsilon t$ be a new time-scale (t' is a slow time compared to t), we can rewrite (20)–(21) as

$$\frac{d}{dt'} x = K_i^* \tilde{q} \quad (22)$$

$$\varepsilon \frac{d}{dt'} \begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \end{bmatrix} \quad (23)$$

where, in the forthcoming analysis, and in accordance with the singular perturbation theory, x in (23) will be treated as a fixed parameter, due to its slow variation.

4.2 Equilibrium points

For each fixed x representing the frozen variable as a fixed parameter in (23), the equilibrium points are the solutions of the nonlinear system:

$$\dot{q} = 0, \quad (24)$$

$$\text{Sat}[\text{Sat}[K_p \tilde{q} + x + g(q_d)]] - g(q) = 0. \quad (25)$$

According to Definition 2 and Assumption 2, (25) can be written as:

$$K_p \tilde{q} + x + g(q_d) - g(q) = 0. \quad (26)$$

Now, the Contraction Mapping Theorem (Kelly et al., 2005; Khalil, 2002), guarantees that (26) has a unique solution $\tilde{q} = h_1(x) \in \mathbb{R}^n$ provided that

$$k_{p_i} > k_{g_i} \quad (27)$$

is satisfied (see Appendix A).

Then we have that, for each $x \in \mathbb{R}^n$, the unique equilibrium point of (23) is:

$$\begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} h_1(x) \\ 0 \end{bmatrix} = h(x) \in \mathbb{R}^{2n}. \quad (28)$$

Consequently, we have that:

$$x = h^{-1}(\tilde{q}) = -K_p \tilde{q} - g(q_d) + g(q) \quad (29)$$

which we will use later on.

4.3 Overall singularly perturbed system

In order to proceed with the stability analysis, we shift the equilibrium point of (23) to the origin. To this end, we make the following change of variables:

$$\begin{bmatrix} y_1(t') \\ y_2(t') \end{bmatrix} = \begin{bmatrix} \tilde{q}(t') - h_1(x) \\ \dot{q}(t') \end{bmatrix} \quad (30)$$

which implies that $\tilde{q} = y_1 + h_1(x)$. Then, (22)–(23) can be now represented by the new variables as a singularly perturbed system given by

$$\frac{d}{dt'}x = K_i^*[y_1 + h_1(x)] \quad (31)$$

$$\varepsilon \frac{d}{dt'} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -y_2 - \varepsilon \left[\frac{\partial h_1(x)}{\partial x} \right] K_i^*[y_1 + h_1(x)] \\ M(q_{d-y_1-h_1(x)})^{-1} [\text{Sat}[\text{Sat}[K_p(y_1 - h_1(x)) + x + g(q_d)] - K_v y_2] \\ - C(q_{d-y_1-h_1(x)}, y_2) y_2 - g(q_{d-y_1-h_1(x)})] \end{bmatrix}. \quad (32)$$

5. Stability analysis

According to the theory of singularly perturbed systems (Khalil, 2002), the origin of (22)–(23) is asymptotically stable if and only if the origin of (31)–(32) is asymptotically stable. It is important to remember that x is a fixed parameter in (23) and (32), this is because t' and x are varying slowly since, in the t time scale, they are given by (Khalil, 2002):

$$t' = t_0 + \varepsilon t, \quad x = x(t_0 + \varepsilon t), \quad (33)$$

being t_0 the initial time. The setting of $\varepsilon = 0$ freezes these variables at $t' = t_0$ and $x = x(t_0)$ (initial conditions).

By simplicity, we divide the stability analysis in two parts:

- First, we will prove asymptotic stability and local exponential stability of the origin of a saturated PD controller with desired gravity compensation plus a constant vector x , which can be seen as a constant control input.
- Second, based on a theorem of singularly perturbed systems, we will prove that the origin of (22)–(23) is locally exponentially stable.

5.1 Stability analysis of a Saturated PD Controller with Desired Gravity Compensation plus a constant vector x

The control law that describes the proposed Saturated PD Controller with Desired Gravity Compensation plus a constant vector x is given by:

$$\tau = \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}]. \quad (34)$$

By substituting (34) into the robot dynamics (1), we obtain

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \end{bmatrix} \quad (35)$$

whose equilibrium points are the solutions of the nonlinear equations (24)–(25) and they have already been proven to have a unique solution $\begin{bmatrix} \tilde{q}^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} h_1(x)^T & 0^T \end{bmatrix}^T$, provided that $k_{p_i} > k_{g_i}$ is satisfied.

5.1.1 Asymptotic stability analysis

To carry out the stability analysis of the equilibrium of (35), we propose the following Lyapunov function candidate, which is inspired from one in (Alvarez-Ramirez et al., 2008):

$$W(\tilde{q}, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + W_1(\tilde{q}) \quad (36)$$

where

$$W_1(\tilde{q}) = \sum_{i=1}^n \int_0^{\tilde{q}_i} \text{Sat}[\text{Sat}(k_{p_i} r_i + x_i + g_i(q_d))] dr_i + \mathcal{U}(q_d - \tilde{q}) \\ - \sum_{i=1}^n \int_0^{h_1(x)} \text{Sat}[\text{Sat}(k_{p_i} r_i + x_i + g_i(q_d))] dr_i - \mathcal{U}(q_d - h_1(x)).$$

By following similar steps to those given by Zavala & Santibañez (2007) (see Appendix B) we prove that (36) is a positive definite and radially unbounded function, provided that $k_{p_i} > k_{g_i}$. The time derivative of $W(\tilde{q}, \dot{q})$ along the trajectories of (35), and after some algebraic simplifications, results in:

$$\dot{W}(\tilde{q}, \dot{q}) = \dot{q}^T \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - \dot{q}^T \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d))]. \quad (37)$$

Finally, by using the following property of linear saturation functions (Santibañez et al., 2010):

$$\dot{q}_i [\text{Sat}(z_i - \dot{q}_i) - \text{Sat}(z_i)] \leq -|\text{Sat}(z_i - \dot{q}_i) - \text{Sat}(z_i)|^2$$

we have that $\dot{W}(\tilde{q}, \dot{q})$ is upper bounded by:

$$\dot{W}(\tilde{q}, \dot{q}) \leq -\left\| \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d))] \right\|^2 \leq 0.$$

Thus $\dot{W}(\tilde{q}, \dot{q})$ is a negative semidefinite function and we can conclude stability of the equilibrium point $\begin{bmatrix} \tilde{q}^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} h_1(x)^T & 0^T \end{bmatrix}^T \in \mathbb{R}^{2n}$ of (35). We can use the LaSalle's Invariance Principle (Kelly et al., 2005) to conclude that the equilibrium point is, in fact, globally asymptotically stable. To this end, let us define Ω as:

$$\Omega = \{\tilde{q}, \dot{q} \in \mathbb{R}^n: \dot{W}(\tilde{q}, \dot{q}) = 0\} = \{\dot{q} = 0, \tilde{q} \in \mathbb{R}^n\}.$$

Notice that, from (35),

$$\dot{q}(t) \equiv 0 \Rightarrow \ddot{q}(t) \equiv 0 \Rightarrow \text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d))] - g(q_d - \tilde{q}) \equiv 0.$$

Furthermore, under the assumption (27) we can assure that

$$\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d))] - g(q_d - \tilde{q}) \equiv 0 \Rightarrow \tilde{q} \equiv h_1(x).$$

Therefore, from LaSalle's Invariance Principle we conclude that the equilibrium point

$$\begin{bmatrix} \tilde{q}^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} h_1(x)^T & 0^T \end{bmatrix}^T \in \mathbb{R}^{2n} \text{ of (35) is globally asymptotically stable.}$$

5.1.2 Local exponential stability analysis

Before proceeding with the stability analysis of this section, we recall a useful existing lemma presented in (Kelly, 1995a).

Lemma1. Consider the nonlinear system:

$$\dot{y} = A(y)y + B(y)f(y), \quad (38)$$

where $y \in \mathbb{R}^m$, $A(y)$ and $B(y)$ are $m \times m$ nonlinear functions of y , and $f(y)$ is a $m \times 1$ nonlinear function of y . Assume that $f(0) = 0$; hence, $y = 0 \in \mathbb{R}^m$ is an equilibrium point of the system (38). Then, the linearized system of (38) around the equilibrium $y = 0$ is given by:

$$\dot{y} = \left[A(0) + B(0) \frac{\partial f(0)}{\partial y} \right] y. \quad (39)$$

◇

In order to prove that the equilibrium point of the closed-loop system (35) is locally exponentially stable, we consider a local linearization of the closed-loop system around the equilibrium point $\begin{bmatrix} \tilde{q}^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} h_1(x)^T & 0^T \end{bmatrix}^T \in \mathbb{R}^{2n}$ (Khalil, 2002). In the neighborhood of this equilibrium point, the closed-loop system (35) can be represented by:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) - K_p \tilde{q} + K_v \dot{q} - x - g(q_d) = 0. \quad (40)$$

A local change of variables $y_1 = \tilde{q} - h_1(x)$, and $y_2 = \dot{q}$ leads to:

$$\begin{aligned} & M(q_d - y_1 - h_1(x)) \dot{y}_2 + C(q_d - y_1 - h_1(x), y_2) y_2 \\ & + g(q_d - y_1 - h_1(x)) - K_p [y_1 + h_1(x)] + K_v y_2 - x - g(q_d) = 0 \end{aligned}$$

whose unique equilibrium is the origin, provided that (27) is satisfied. The previous equation can be written as:

$$\dot{y} = A(y)y + B(y)f(y). \quad (41)$$

where,

$$\begin{aligned} \dot{y} &= \frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ A(y) &= \begin{bmatrix} 0 & -I \\ 0 & -M(q_d - y_1 - h_1(x))^{-1} [K_v + C(q_d - y_1 - h_1(x), y_2)] \end{bmatrix} \\ B(y) &= \begin{bmatrix} 0 & 0 \\ 0 & M(q_d - y_1 - h_1(x))^{-1} \end{bmatrix} \end{aligned}$$

$$f(y) = \begin{bmatrix} 0 \\ K_p[y_1 + h_1(x)] + x + g(q_d) - g(q_d - y_1 - h_1(x)) \end{bmatrix}.$$

According to Lemma 1, the linearized system from (41), around the equilibrium $y = 0$, has the form (39), with:

$$\begin{aligned} A(0) &= \begin{bmatrix} 0 & -I \\ 0 & -M(q_d - h_1(x))^{-1}K_v \end{bmatrix} \\ B(0) &= \begin{bmatrix} 0 & 0 \\ 0 & M(q_d - h_1(x))^{-1} \end{bmatrix} \\ \frac{\partial f(0)}{\partial y} &= \begin{bmatrix} 0 & 0 \\ K^* & 0 \end{bmatrix} \end{aligned}$$

which can be compacted in:

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -I \\ M(q_d - h_1(x))^{-1}K^* & -M(q_d - h_1(x))^{-1}K_v \end{bmatrix}}_J \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (42)$$

where K^* is given by:

$$K^* = K_p - \frac{\partial g(q_d - y_1 - h_1(x))}{\partial y_1}.$$

Notice that if (27) is satisfied then K^* is a positive definite matrix (Hernandez-Guzman et al., 2008). To analyze the stability of the origin of (42), we propose the Lyapunov function candidate:

$$W_L(y_1, y_2) = \frac{1}{2} y_2^T M(q_d - h_1(x)) y_2 + \frac{1}{2} y_1^T K^* y_1 \quad (43)$$

which is a positive definite function. The time derivative along the trajectories of (42) is:

$$\begin{aligned} \dot{W}(y_1, y_2) &= y_2^T M(q_d - h_1(x)) \dot{y}_2 + y_1^T K^* \dot{y}_1 \\ &= y_2^T [K^* y_1 - K_v y_2] - y_1^T K^* y_2 = -y_2^T K_v y_2 \end{aligned}$$

which is a negative semidefinite function. By using the LaSalle's Invariance Principle we can conclude global asymptotic stability of the closed-loop system (42). To this end, let us define Ω as:

$$\Omega = \{y_1, y_2 \in \mathbb{R}^n : \dot{W}(y_1, y_2) = 0\} = \{y_2 = 0, y_1 \in \mathbb{R}^n\}. \quad (44)$$

Notice that, from (42):

$$y_2(t) \equiv 0 \Rightarrow \dot{y}_2(t) \equiv 0 \Rightarrow M(q_d - h_1(x))^{-1} K^* y_1 \equiv 0. \quad (45)$$

Furthermore, under assumption (27) we can assure that

$$M(q_d - h_1(x))^{-1} K^* y_1 \equiv 0 \Rightarrow y_1 \equiv 0.$$

Therefore, from LaSalle's Invariance Principle we conclude that the origin of the linear system (42) is globally asymptotically stable. This implies that the eigenvalues of J in (42) are located in the left-hand side of the complex plane (see Theorem 4.5 in Khalil (2002)), and hence, the origin of the linear system (42) is exponentially stable (see e.g. Theorem 4.11 in Khalil (2002) that shows that, for linear systems, uniform asymptotic stability of the origin is equivalent to exponential stability). According to this, exponential stability of the origin for the linear system (42) implies the local exponential stability of the origin for the nonlinear system (41) (see e.g. Theorem 4.13 in Khalil (2002)).

Finally, we can conclude that the equilibrium point of the nonlinear system (35) is locally exponentially stable. So we have proven the following:

Proposition 1. Under Assumption 2, and (27), the control law (34) guarantees global asymptotic stability and local exponential stability of the closed-loop system (35) with

$$|\tau_i(t)| \leq \tau_i^{\max} \text{ for all } i = 1, 2, \dots, n \text{ and } t \geq 0.$$

5.2 Stability analysis of the singularly perturbed system.

To prove the exponential stability of the origin of (22)–(23), we recall an existing theorem:

Theorem 3 (Khalil, 2002): *Consider the singularly perturbed system*

$$\dot{x} = f(t', x, z, \varepsilon) \quad (46)$$

$$\varepsilon \dot{z} = g(t', x, z, \varepsilon). \quad (47)$$

Assume that the following are satisfied for all $(t', x, \varepsilon) \in [0, \infty) \times B_r \times [0, \varepsilon]$, with $B_r = \{x \in \mathbb{R}^n : \|x\| \leq r\}$:

- $f(t', 0, 0, \varepsilon) = 0$ and $g(t', 0, 0, \varepsilon) = 0$.
- The equation $0 = g(t', x, z, 0)$ has an isolated root $z = h(t', x)$ such that $h(t', 0) = 0$.
- The functions f , g , h and their partial derivatives up to the second order are bounded for $z - h(t', x) \in B_\rho$, with $B_\rho = \{y \in \mathbb{R}^{2n} : \|y\| \leq \rho\}$.
- The origin of the reduced system

$$\dot{x} = f(t', x, h(t', x), 0) \quad (48)$$

is exponentially stable.

- The origin of the boundary-layer system

$$\frac{dy}{dt} = g(t', x, y + h(t', x), 0) \quad (49)$$

is exponentially stable, uniformly in (t', x) .

Then, there exists $\varepsilon^* > 0$ such that, for all $\varepsilon < \varepsilon^*$, the origin of (46)–(47) is exponentially stable. \diamond

We are now ready to present our main contribution.

Proposition 2. Consider the robot dynamics (1) in closed-loop with the practical saturated PID control law (12). Under Assumption 2, and (27), the origin of the closed-loop system

(22)–(23) is locally exponentially stable, and therefore, the equilibrium point of (18) is locally exponentially stable. Besides $|\tau_i(t)| \leq \tau_i^{\max}$ for all $i = 1, 2, \dots, n$ and $t \geq 0$. \diamond

Proof. Notice that (46)–(47) correspond to (22)–(23), respectively, with

$$\begin{aligned} f(t', x, z, \varepsilon) &= K_i^* \tilde{q} \\ g(t', x, z, \varepsilon) &= \begin{bmatrix} -\dot{q} \\ M(q)^{-1}[\text{Sat}[\text{Sat}(K_p \tilde{q} + x + g(q_d)) - K_v \dot{q}] - C(q, \dot{q}) \dot{q} - g(q)] \end{bmatrix} \\ z &= \begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} \in \mathbb{R}^{2n}. \end{aligned}$$

In order to complete the stability analysis, we are going to check each item of the Theorem 3.

a) By substituting $x = \tilde{q} = \dot{q} = 0$ in (22)–(23), it is straightforward to verify this assumption.

b) This item is easily fulfilled by noting that the root of $g(t', x, z, \varepsilon)$ has been obtained in Section 4.2, where it was proven that, for each $x \in \mathbb{R}^n$, the unique root of (23) is $z = h(x) = [h_1(x)^T 0^T]^T \in \mathbb{R}^{2n}$, provided that (27) is satisfied. On the other hand, we know from (28) that $\tilde{q} = h_1(x)$, and therefore, when $x = 0$ we have that $\tilde{q} = h_1(0)$; then, from (29), $0 = h_1^{-1}(\tilde{q}) = -[K_p K_{pc} \tilde{q} + g(q_d) - g(q_d - \tilde{q})]$ which under assumption (27) has a unique solution $\tilde{q} = 0$. Hence, $h(0) = [h_1(0)^T 0^T]^T = [0^T 0^T]^T$ and assumption b) is verified.

c) This is straightforward given that the right-hand side of (22)–(23) is C^2 .

d) By substituting the isolated root $z = h(x)$ and $\varepsilon = 0$ in (22), that is $\tilde{q} = h_1(x)$ and $\dot{q} = 0$, we obtain the so-called reduced system, which is given by:

$$\frac{d}{dt'} x = K_i^* h_1(x) \tag{50}$$

whose unique equilibrium point results from $h_1(x) = 0$ and is given by $x = h_1^{-1}(0) = 0$ provided that (27) is satisfied. Comparing the reduced system (50) with the terms used in Theorem 3, we have $\dot{x} = f(t', x, h(t, x), 0) = K_i^* h_1(x)$.

On the other hand, to analyze the origin of the reduced system (50), let us define the quadratic Lyapunov function candidate

$$V(x) = \frac{1}{2} x^T (K_i^*)^{-1} x \tag{51}$$

which satisfies

$$\frac{1}{2} \lambda_{\max}\{(K_i^*)^{-1}\} \|x\|^2 \geq V(x) \geq \frac{1}{2} \lambda_{\min}\{(K_i^*)^{-1}\} \|x\|^2 \tag{52}$$

and hence, it is a positive definite and radially unbounded function. The time derivative along the trajectories of (50) is given by:

$$\dot{V}(x) = x^T (K_i^*)^{-1} \dot{x} = x^T h_1(x). \tag{53}$$

Consider (29) with $\tilde{q} = h_1(x)$:

$$\dot{x} = -K_p h_1(x) - g(q_d) + g(q_d - h_1(x)), \quad (54)$$

substituting in (53) we have

$$\begin{aligned} h_1^T \dot{x} &= h_1^T(x) [-K_p h_1(x) - g(q_d) + g(q_d - h_1(x))] \\ &= -h_1^T(x) K_p h_1(x) + h_1^T(x) [-g(q_d) + g(q_d - h_1(x))] \\ &\leq -h_1^T(x) \left[K_p + \left. \frac{\partial g(z)}{\partial z} \right|_{z=\xi} \right] h_1(x) \end{aligned}$$

where we use Theorem 2, and

$$K_p + \left. \frac{\partial g(z)}{\partial z} \right|_{z=\xi} \quad (55)$$

is a positive definite matrix provided that

$$k_{p_i} > \sum_{j=1}^n \max_q \left| \frac{\partial g_i(q)}{\partial q_j} \right| \quad \text{for } i = 1, \dots, n. \quad (56)$$

is satisfied (Hernandez-Guzman et al., 2008).

Note that (27) implies (56). Therefore

$$\dot{V}(x) \leq -h_1^T(x) \left[K_p + \left. \frac{\partial g(z)}{\partial z} \right|_{z=\xi} \right] h_1(x) \leq -\lambda_{\min} \left\{ K_p + \left. \frac{\partial g(z)}{\partial z} \right|_{z=\xi} \right\} \|h_1(x)\|^2 \quad (57)$$

Notice that, due to $h_1(0) = 0$, the time derivative (53) is a negative definite function and we can conclude global asymptotic stability of the origin of (50).

Moreover, we have that:

$$\begin{aligned} \|x\|^2 &= x^T x \\ &= [-K_p h_1(x) - g(q_d) + g(q_d - h_1(x))]^T [-K_p h_1(x) - g(q_d) + g(q_d - h_1(x))] \\ &= h_1^T(x) K_p^2 h_1(x) + 2h_1^T(x) K_p [-g(q_d) + g(q_d - h_1(x))] \\ &\quad + [-g(q_d) + g(q_d - h_1(x))]^T [-g(q_d) + g(q_d - h_1(x))] \\ &\leq [\lambda_{\max}\{K_p\}]^2 + 2k_g \lambda_{\max}\{K_p\} + k_g^2 \|h_1(x)\|^2 \\ &= [\lambda_{\max}\{K_p\} + k_g]^2 \|h_1(x)\|^2. \end{aligned}$$

Then

$$\|h_1(x)\|^2 \geq \frac{1}{[\lambda_{\max}\{K_p\} + k_g]^2} \|x\|^2, \quad (58)$$

and we have that

$$\dot{V}(x) \leq \frac{-\lambda_{\min} \left\{ K_p + \frac{\partial g(z)}{\partial z} \Big|_{z=\xi} \right\}}{[\lambda_{\max}\{K_p\} + k_g]^2} \|x\|^2. \tag{59}$$

Therefore, from (52) and (59), we can conclude that $x = 0$ is a globally exponentially stable equilibrium point for the reduced system (50) provided that (27) is satisfied (see Theorem 4.10, Khalil (2002)). So we have verified the assumption d) of Theorem 3.

e) By setting $\varepsilon = 0$ and considering that $\varepsilon \frac{dy}{dt'} = \frac{dy}{dt}$ in (32), we obtain the boundary-layer system:

$$\underbrace{\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}_{\frac{dy}{dt}} = \underbrace{\begin{bmatrix} -y_2 \\ M(q_d - y_1 - h_1(x))^{-1} \left[\text{Sat} \left[\text{Sat} \left(K_p (y_1 + h_1(x)) + g(q_d) + x \right) - K_v y_2 \right] \right. \\ \left. - C(q_d - y_1 - h_1(x), y_2) y_2 - g(q_d - y_1 - h_1(x)) \right] \end{bmatrix}}_{g(t,x,y+h(t,x),0)} \tag{60}$$

where, according to (33), x is frozen at $x = x(t_0)$, which corresponds to the robotic system under the Saturated PD Controller with Desired Gravity Compensation plus a constant vector x , whose unique equilibrium point is the origin, provided that (27) is satisfied. The stability analysis of (60) has already been carried out in the previous subsection, where we concluded, in accordance with Proposition 1, that the origin of (60) is asymptotically stable and locally exponentially stable, uniformly in x . The uniformity in x is given straightforward with the asymptotic stability of the origin of (60) because it is an autonomous system. This checks the assumption e). Finally, we conclude, in accordance with Theorem 3, that the equilibrium point of the closed-loop system (18) is locally exponentially stable for a sufficiently small ε . Under Assumption 2 the constraints (9) are trivially satisfied. This completes the proof. \diamond

6. Experimental results

6.1 The PA10 robot system

The Mitsubishi PA10 arm is an industrial robot manipulator which completely changes the vision of conventional industrial robots. Its name is an acronym of *Portable General-Purpose Intelligent Arm*. There exist two versions (Higuchi et al., 2003): the PA10-6C and the PA10-7C, where the suffix digit indicates the number of degrees of freedom of the arm. This work focuses on the study of the PA10-7CE model, which is the enhanced version of the PA10-7C. The PA10-7CE robot is a 7-dof redundant manipulator with revolute joints. Figure 3 shows a diagram of the PA10 arm, indicating the positive rotation direction and the respective names of each of the joints. The PA10 arm is an open architecture robot; it means that it possesses (Oonishi, 1999):

- A hierarchical structure with several control levels.
- Communication between levels, via standard interfaces.
- An open general-purpose interface in the higher level.

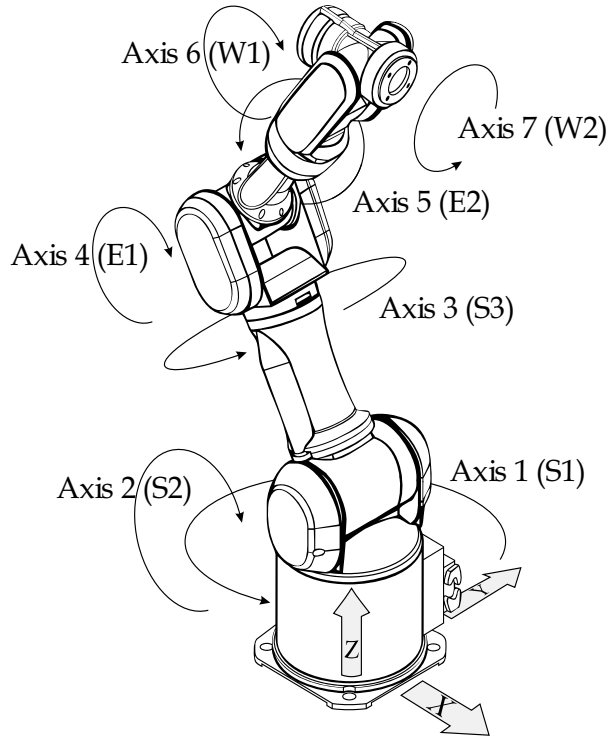


Fig. 3. Mitsubishi PA10-7CE robot

This scheme allows the user to focus on the programming of the tasks at the higher level of the PA10 system, without regarding on the operation of the lower levels. The control architecture of the PA10-7CE robot arm has been modified in order to have access to the low-level signals and configure it in both torque and velocity modes (Ramirez, 2008).

6.2 Numeric values of the parameters for the PA10-7CE.

The vector of gravitational torques for the PA10-7CE is (Ramirez, 2008):

$$g(q) = [g_1(q) \quad g_2(q) \quad \dots \quad g_n(q)]^T$$

where

$$g_1(q) = 0$$

$$g_2(q) = 9.81(-6.9472 \sin(q_2) - 3.1393(\cos(q_2)\cos(q_3)\sin(q_4) + \sin(q_2)\cos(q_4))) \\ - 0.004(((-\cos(q_2)\cos(q_3)\cos(q_4) + \sin(q_2)\sin(q_4))\cos(q_5) \\ + \cos(q_2)\sin(q_3)\sin(q_5))\sin(q_6) - (\cos(q_2)\cos(q_3)\sin(q_4) + \sin(q_2)\cos(q_4))\cos(q_6)))$$

$$g_3(q) = 9.81(3.1393\sin(q_2)\sin(q_3)\sin(q_4) - 0.004((\sin(q_2)\sin(q_3)\cos(q_4)\cos(q_5) + \sin(q_2)\cos(q_3)\sin(q_5))\sin(q_6) + \sin(q_2)\sin(q_3)\sin(q_4)\cos(q_6)))$$

$$g_4(q) = 9.81(-3.1393(\sin(q_2)\cos(q_3)\cos(q_4) + \cos(q_2)\sin(q_4)) - 0.004((\sin(q_2)\cos(q_3)\sin(q_4) - \cos(q_2)\cos(q_4))\cos(q_5)\sin(q_6) - (\sin(q_2)\cos(q_3)\cos(q_4) + \cos(q_2)\sin(q_4))\cos(q_6)))$$

$$g_5(q) = 9.81(-0.004(-\sin(q_5)(-\sin(q_2)\cos(q_3)\cos(q_4) - \cos(q_2)\sin(q_4)) + \sin(q_2)\sin(q_3)\cos(q_5))\sin(q_6))$$

$$g_6(q) = 9.81(-0.004(((\sin(q_2)\cos(q_3)\cos(q_4) - \cos(q_2)\sin(q_4))\cos(q_5) + \sin(q_2)\sin(q_3)\sin(q_5))\cos(q_6) + (\sin(q_2)\cos(q_3)\sin(q_4) - \cos(q_2)\cos(q_4))\sin(q_6)))$$

$$g_7(q) = 0$$

The following expressions recall how the parameters of interest can be found:

$$k_g \geq n \max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right|, \quad k_{g_i} \geq n \max_{j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right|,$$

$$\gamma_i \geq \sup_q |g_i(q)|, \quad k' \geq \sqrt{\gamma_1^2 + \gamma_2^2 + \dots + \gamma_n^2}.$$

The numerical values of the parameters for the PA10-7CE are shown in Table 1. The table also shows the torque and velocity saturation limits of each joint, which are employed to select the corresponding limits of the saturation functions in the controller.

Parameter	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Units
k_{g_i}	0	909.58	216.39	432.25	0.8240	1.3734	0	[Nm/rad]
γ_i	0	129.94	30.91	61.75	0.11772	0.1962	0	[Nm]
τ_i^{\max}	232	232	100	100	14.5	14.5	14.5	[Nm]
v_i^{\max}	1	1	2	2	2π	2π	2π	[rad/s]
k_g	909.58							[Nm/rad]
k'	147.1513							[Nm]

Table 1. Numerical values of the parameters for the PA10-7CE

In order to illustrate the stability results described in the previous pages, this section shows a real-time experiment essay on the PA10-7CE robot system, using the controller proposed in this chapter, given by equation (12) and labeled in this section as Sat(Sat(PI)+P)), and the controller presented in Santibañez et al. (2010), labeled Sat(Sat(P)+PI), whose equation is given by

$$\tau = \text{Sat} \left[K_{pd} \text{Sat} \left(K_{pc} \tilde{q}; l_p, m_p \right) - K_{pd} \dot{q} + w; l_{pi}, m_{pi} \right] \quad (61)$$

$$w = K_{id} \int_0^t \left[\text{Sat} \left(K_{pc} \tilde{q}(r); l_p, m_p \right) - \dot{q}(r) \right] dr$$

where $K_{pd}, K_{pc}, K_{id} \in \mathbb{R}^{n \times n}$ are diagonal positive definite matrices, and we take $\alpha = 1$ (see Fig. 1).

Each of the experiments consisted in taking the robot from the vertical home position (where $q = 0$) to the following desired position: $q_d = \left[-\frac{\pi}{2} \frac{\pi}{3} \frac{\pi}{2} \frac{\pi}{3} \frac{\pi}{2} -\frac{\pi}{2} \frac{\pi}{2} \right]^T$ rad.

6.3 Sat(Sat(PI)+P) scheme

Table 2 shows the values of the gains and the saturation limits for each joint of the proposed control scheme (12). It is easy to check that the assumptions (16), (17) and (27) are fulfilled. Figure 4 shows the evolution of the position error for each joint. It can be seen that transient responses are relatively fast (lower than 1 second for joints 4 to 7 and lower than 2 seconds for joints 1 to 3) without overshoot. The steady state error for each joint is lower than 0.4 degrees. Figure 5 shows the applied torque for each joint. The torques evolve inside of the prescribed limits. For the joints 4 to 7 the torques reach, sometimes, the permitted torque limits, confirming in this way the stability theoretical result.

Gain	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Units
K_{pp}	10.0	100.0	60.0	60.0	50.0	35.0	30.0	[1/s]
K_{ip}	0.01	0.01	0.3	0.01	0.5	0.01	0.01	[1/s ²]
K_{pv}	90.0	150.0	35.0	85.0	10.0	6.0	12.0	[Nm s/rad]
l_{pi}^*	0.95	0.95	1.75	1.75	5.5	5.5	5.5	[rad/s]
m_{pi}^*	1	1	1.9	1.9	6	6	6	[rad/s]
l_p	185	185	75	75	12	12	12	[Nm]
m_p	200	200	80	80	13	13	13	[Nm]

Table 2. Values of the control parameters selected for the Sat(Sat(PI)+P) scheme

6.4 Sat(Sat(P)+PI) scheme

Table 3 shows the values of the gains and the saturation limits for each joint of the control scheme (61). The parameters of the controller have been chosen in such a way that assumptions for the controller (61), given in (Santibañez et al., 2010), are satisfied. Figure 6 shows the position error for each joint. Slightly slower transient responses were obtained, but without overshoot. The steady state errors are similar to those obtained for the Sat(Sat(PI)+P) scheme. Figure 7 shows the evolution of the applied torques, which are more noisy than those of the proposed scheme.

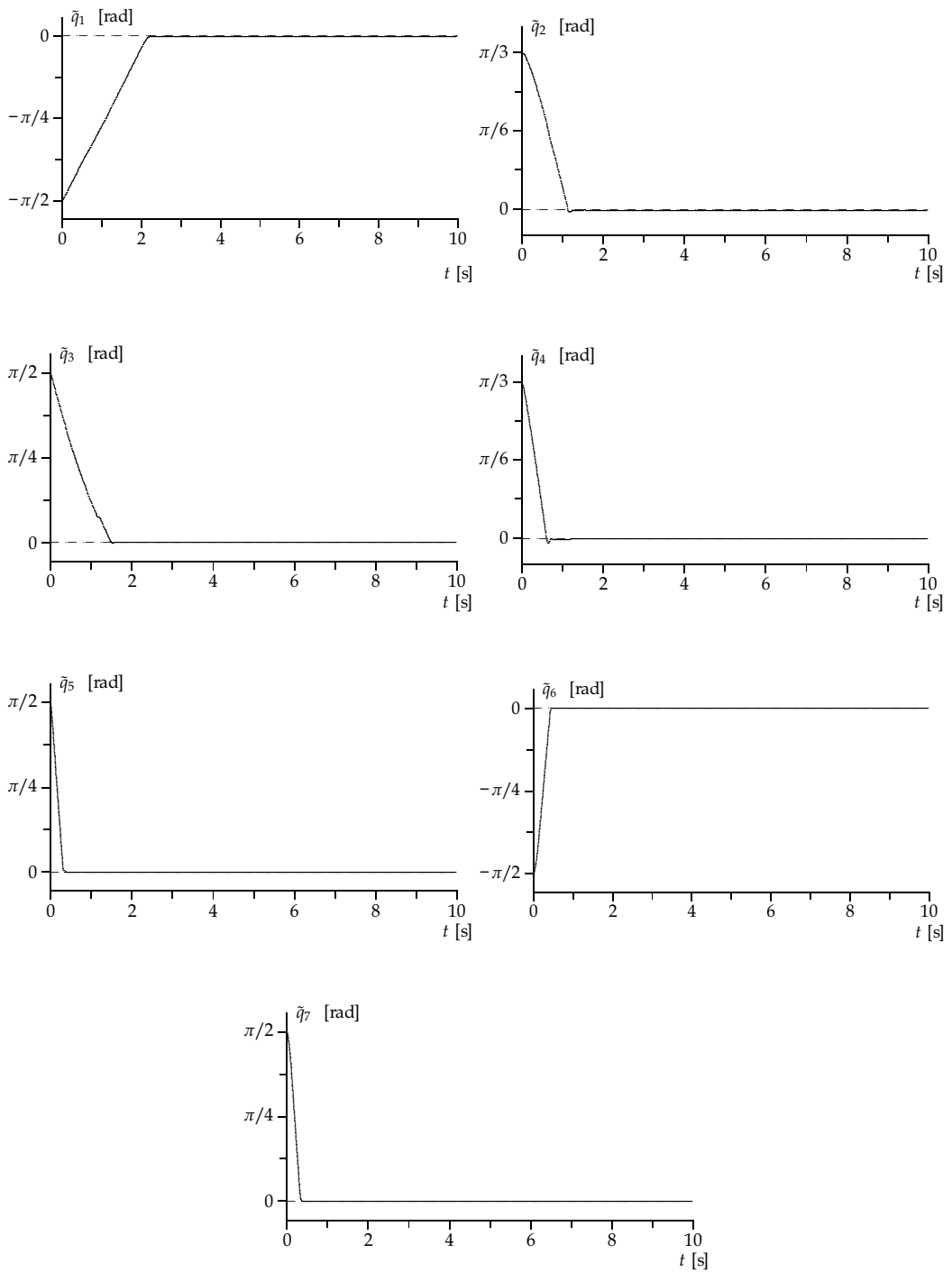


Fig. 4. Position errors for the (Sat(Sat(PI)+P)) scheme

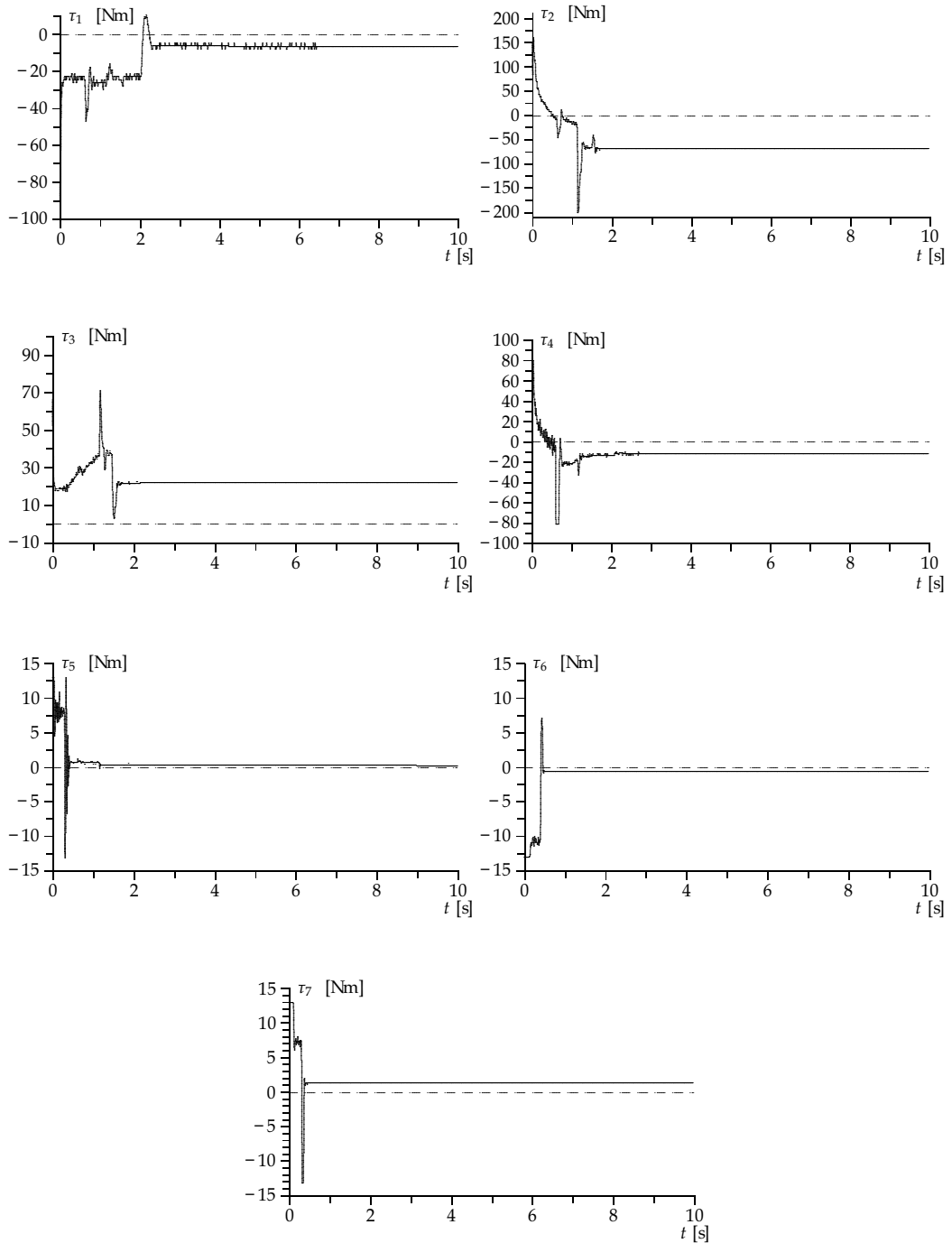


Fig. 5. Applied torques for the Sat(Sat(PI)+P) scheme

Gain	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Units
K_{pc}	3.0	15.0	8.0	8.0	1.2	2.25	1.0	[1/s]
K_{pd}	40.0	280.0	45.0	110.0	15.0	12.0	8.0	[Nm s/rad]
K_{id}	15.0	18.0	10.0	12.0	5.0	8.0	4.0	[Nm/rad]
l_p	0.95	0.95	1.75	1.75	5.5	5.5	5.5	[rad/s]
m_p	1	1	1.9	1.9	6	6	6	[rad/s]
l_{pi}	185	185	75	75	12	12	12	[Nm]
m_{pi}	200	200	80	80	13	13	13	[Nm]

Table 3. Values of the control parameters selected for the Sat(Sat(P)+PI) scheme

7. Conclusions

In this chapter we have proposed an alternative to the saturated nonlinear PID controller previously presented by Santibañez et al. (2010) which, also, results from the practical implementation of the classical PID controller, by considering the natural saturations of the electronics in the control computer, servo drivers, and actuators. The stability analysis of the closed-loop system is carried out by using the singular perturbation theory. Based on auxiliary Lyapunov functions, we prove local exponential stability of the equilibrium point of the closed-loop system. It is also guaranteed that, regardless of the initial conditions, the delivered actuator torques evolve inside the permitted limits. Experimental results confirm the proposed analysis. Furthermore, the theoretical result explains why the classical linear PID regulator used in industrial robot manipulators preserves the exponential stability in spite of entering the saturation zones inherent to the electronic control devices and the actuator torque constraints.

8. Acknowledgement

This work is partially supported by PROMEP, DGEST, and CONACYT (grant 60230), Mexico.

9. References

- Aguiñaga-Ruiz, E.; Zavala-Rio, A.; Santibañez, V. & Reyes, F. (2009). Global trajectory tracking through static feedback for robot manipulators with bounded inputs. *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 4, pp. 934-944.
- Alvarez-Ramirez, J.; Cervantes, I. & Kelly, R. (2000). PID regulation of robot manipulators: Stability and performance. *Systems and Control Letters*, Vol. 41, pp. 73-83.
- Alvarez-Ramirez, J.; Kelly, R. & Cervantes, I. (2003). Semiglobal stability of saturated linear PID control for robot manipulators. *Automatica*, Vol. 39, pp. 989-995.
- Alvarez-Ramirez, J.; Santibañez, V. & Campa, R. (2008). Stability of robot manipulators under saturated PID compensation. *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 6, pp. 1333-1341.
- Arimoto, S. (1995). Fundamental problems of robot control: Part I, Innovations in the realm of robot servo-loops. *Robotica*, Vol. 13, pp. 19-27.

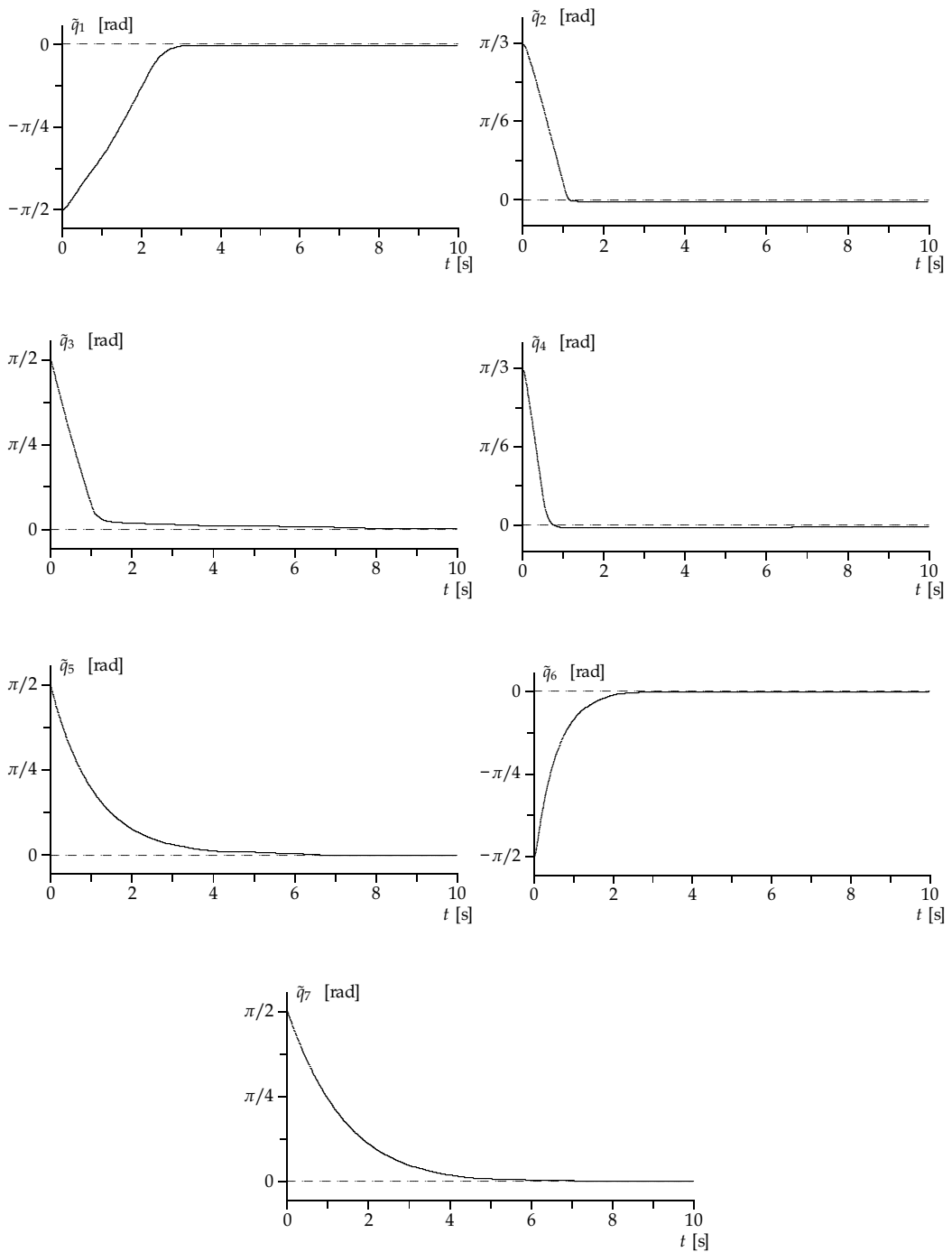


Fig. 6. Position errors for the Sat(Sat(P)+PI) scheme

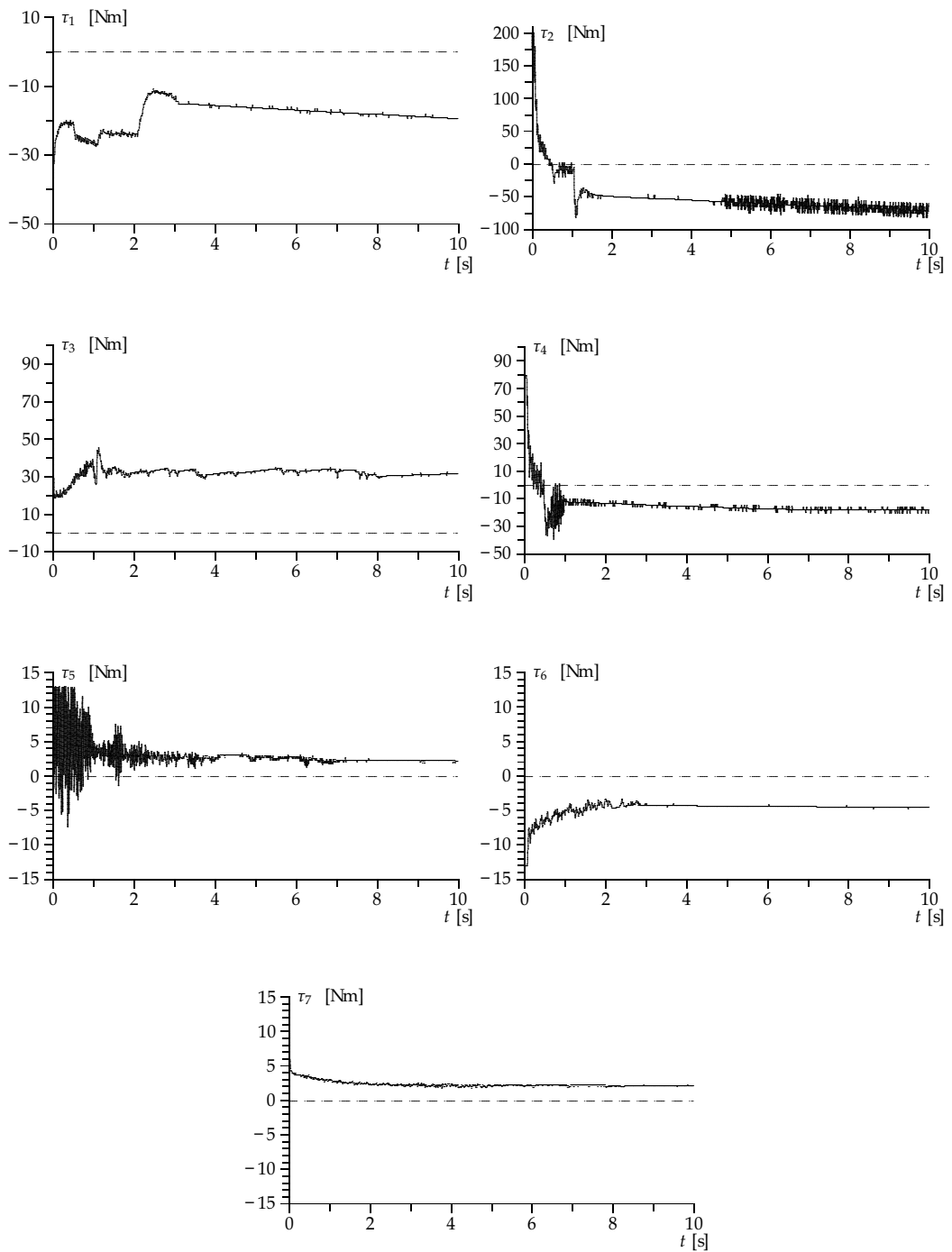


Fig. 7. Applied torques for the Sat(Sat(P)+PI) scheme

- Arimoto, S. & Miyazaki, F. (1984). Stability and robustness of PID feedback control for robot manipulators of sensory capability. In: *Robotics Researches: First International Symposium*, M. Brady and R.P. Paul (Eds.), pp. 783-799, MIT Press.
- Arimoto, S.; Naniwa, T. & Suzuki, H. (1990). Asymptotic stability and robustness of PID local feedback for position control of robot manipulators. *Proceedings of the International Conference on Automation Robotics and Computer Vision*, Singapore, June 1990.
- Camarillo, K.; Campa, R.; Santibañez, V. & Moreno J. (2008). Operational space control of industrial robots using their own joint velocity PI controllers: Stability analysis and experiments. *Robotica*, Vol 26, pp. 729-738.
- Cervantes, I. & Alvarez-Ramirez, J. (2001). On the PID tracking control of robot manipulators. *Systems and Control Letters*, Vol. 42, pp. 37-46.
- Choi, Y. & Chung, W. K. (2004). *PID trajectory tracking control for mechanical systems*. Springer-Verlag, 2004.
- Colbaugh, R.; Barany, E. & Glass, K. (1997) Global regulation of uncertain manipulators using bounded controls. *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- Craig, J. J. (1998). *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, 1998.
- Dixon, W. E. (2007). Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, Vol. 52, No. 3, pp. 488-493.
- Dixon, W. E.; de Queiroz, M. S.; Zhang, F. & Dawson, D. M. (1999). Tracking control of robot manipulators with bounded torque inputs. *Robotica*, Vol. 17, pp. 121-129.
- Gorez, R. (1999). Globally stable PID-like control of mechanical systems. *Systems and Control Letters*, Vol. 38, pp. 61-72.
- Hernandez-Guzman, V.; Santibañez V. & Silva-Ortigoza R. (2008). A new tuning procedure for PID control of rigid robots. *Advanced Robotics*, Vol. 22, pp. 1007-1023.
- Higuchi, M.; Kawamura, T.; Kaikogi, T.; Murata, T. & Kawaguchi, M. (2003). Mitsubishi clean room robot. Mitsubishi Heavy Industries, Ltd., Technical Review, Vol. 40, No. 5, 2003.
- Horn, R. A. & Johnson, C. R. (1985). *Matrix Analysis*, Cambridge University Press.
- Kawamura, S.; Miyasaki, F. & Arimoto, S. (1988). Is a local linear PD feedback control law effective for trajectory tracking of robot motion?. *Proceedings of the IEEE Conference on Robotics and Automation*, Philadelphia, PA., March 1988.
- Kelly, R. (1995a). Regulation of robotic manipulators: Stability analysis via the Lyapunov's first method. Technical report, CICESE, Ensenada, Mexico.
- Kelly R. (1995b). A tuning procedure for stable PID control of robot manipulators. *Robotica*, Vol. 13, No. 2, pp. 141-148.
- Kelly R. (1998). Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions. *IEEE Transactions on Automatic Control*, Vol. 43, No. 7, pp. 934-938.
- Kelly R. & Moreno, J. (2001). Learning PID structures in an introductory course of automatic control. *IEEE Transactions on Education*, Vol. 44, No. 4, pp. 373-376.
- Kelly, R. & Santibañez, V. (1996). A class of global regulators with bounded control actions for robot manipulators. *Proceedings of the IEEE Conference on Decision and Control*, Kobe, Japan, December 1996.
- Kelly, R.; Santibañez, V. & Loria, A. (2005) *Control of Robot Manipulators in Joint Space*, Springer-Verlag, 2005.
- Khalil, H. (2002). *Nonlinear Systems*, Prentice Hall, 2002.
- Koditschek, D. (1984). Natural motion for robot arms. *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, December 1984.

- Laib, A. (2000). Adaptive output regulation of robot manipulators under actuator constraints. *IEEE Transactions on Robotics and Automation*, Vol. 16, pp. 29-35.
- Loria, A.; Kelly, R.; Ortega, R. & Santibañez, V. (1997). On global output feedback regulation of Euler-Lagrange systems with bounded inputs. *IEEE Transactions on Automatic Control*, Vol. 42, pp. 1138-1143.
- Loria, A. & Nijmeijer, H. (1998). Bounded output feedback tracking control of fully actuated Euler-Lagrange systems. *Systems and Control Letters*, Vol. 33, pp. 151-161.
- Meza, J. L. & Santibañez, V. (1999). Analysis via passivity theory of a class of nonlinear PID global regulators for robot manipulators. *Proceedings of the IASTED International Conference on Robotics and Applications*, Santa Barbara, CA, October 1999.
- Meza, J. L.; Santibañez, V. & Campa, R. (2007). An estimate of the domain of attraction for the PID regulator of manipulators. *International Journal of Robotics and Automation*, Vol. 22, No. 3, pp. 187-195.
- Meza, J. L.; Santibañez, V. & Hernandez, V. (2005). Saturated nonlinear PID global regulator for robot manipulators: Passivity based analysis. *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- Moreno, J.; Santibañez, V. & Campa, R. (2008a). A class of OFT controllers for torque-saturated robot manipulators: Lyapunov stability and experimental evaluation. *Journal of Intelligent and Robotic Systems*, Vol. 51, pp. 65-88.
- Moreno, J.; Santibañez, V. & Campa, R. (2008b). An output feedback tracking control of robot manipulators with bounded torque input. *International Journal of Control, Automation, and Systems*, Vol. 6, No. 1, pp. 76-85.
- Oonishi, K. (1999). The open manipulator system of the MHIPA-10 robot. *Proceedings of the International Symposium on Robotics*, Tokio, Japan, October 1999.
- Ortega, R.; Loria, A. & Kelly, R. (1995). A semiglobally stable output feedback PID regulator for robot manipulators, *IEEE Transactions on Automatic Control*, Vol. 40, No. 8, pp. 1432-1436.
- Ortega, R. & Spong, M. (1989). Adaptive motion control of rigid robots: a tutorial. *Automatica*, Vol. 25, No. 6, pp. 877-888.
- Qu, Z. & Dorsey, J. (1991). Robust PID control of robots, *International Journal of Robotics and Automation*, Vol. 6, No. 4, pp. 228-235.
- Ramirez, C. (2008). Dynamic modeling and torque-mode control of the Mitsubishi PA10-7CE robot. Master's thesis (in Spanish). Instituto Tecnológico de la Laguna, Torreon, Mexico, December 2008.
- Reyes, R. & Kelly, R. (2001). Experimental evaluation of model-based controllers on a direct-drive robot arm. *Mechatronics*, Vol. 11, No. 3, pp. 267-282.
- Rocco, P. (1996). Stability of PID control for industrial robot arms, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, pp. 606-614.
- Santibañez, V.; Camarillo, K.; Moreno-Valenzuela, J. & Campa, R. (2010). A practical PID regulator with bounded torques for robot manipulators. *International Journal of Control Automation and Systems*, Vol. 8, No. 3, pp. 544-555.
- Santibañez, V. & Kelly, R. (1997). On global regulation of robot manipulators: Saturated linear state feedback and saturated linear output feedback, *European Journal of Control*, Vol. 3, pp. 104-113.
- Santibañez, V. & Kelly, R. (1998a). A class of nonlinear PID global regulators for robot manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.
- Santibañez, V. & Kelly, R. (1998b). A new set-point controller with bounded torques for robot manipulators, *IEEE Transactions on Industrial Electronics*, Vol. 45, pp. 126-133.

- Santibañez, V. & Kelly, R. (2001). Global asymptotic stability of bounded output feedback tracking control for robot manipulators, *Proceedings of the IEEE International Conference on Decision and Control*, Orlando, FL, December 2001.
- Santibañez, V.; Kelly, R.; Zavala-Rio, A. & Parada, P. (2008). A new saturated nonlinear PID global regulator for robot manipulators, *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.
- Spong, M. & Vidyasagar, M. (1989). *Robot Dynamics and Control*, John Wiley and Sons, 1989.
- Sun, D.; Hu, S.; Shao, X. & Liu, C. (2009), Global stability of a saturated nonlinear PID controller for robot manipulators. *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 4, pp. 892–899.
- Teel, A. R. (1992). Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems and Control Letters*, Vol. 18, No. 3, pp. 165–171.
- Wen, J. T. & Murphy, S. (1990). PID control for robot manipulators, *CIRSSE Document 54*, Rensselaer Polytechnic Institute, May 1990.
- Zavala-Rio, A.; Aguinaga-Ruiz, E. & Santibañez, V. (2010). Global trajectory tracking through output feedback for robot manipulators with bounded inputs. *Asian Journal of Control*. To appear.
- Zavala-Rio, A. & Santibañez, V. (2006) Simple extensions of the PD-with-gravity-compensation control law for robot manipulators with bounded inputs, *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 5, pp. 958–965.
- Zavala-Rio, A. & Santibañez, V. (2007) A natural saturating extension of the PD-with-desired-gravity compensation control law for robot manipulators with bounded inputs, *IEEE Transactions on Robotics*, Vol. 23, No.2, pp. 386–391.
- Zergeroglu, E.; Dixon, W.; Behal, A. & Dawson, D. (2000). Adaptive set-point control of robotic manipulators with amplitude-limited control inputs, *Robotica*, Vol. 18, pp. 171–181.

Appendix A

In this section we prove that (26) has a unique solution $\tilde{q} = h(x) \in \mathbb{R}^n$, provided that

$$k_{p_i} > k_{g_i} \geq n \left(\max_{q,j} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right) \quad \text{where } i=1,2,\dots,n \text{ and } j=1,2,\dots,n.$$

To this end, notice that we can rewrite (26) as

$$\tilde{q} = \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \vdots \\ \tilde{q}_n \end{bmatrix} = \begin{bmatrix} \frac{g_1(q) - g_1(q_d) - x_1}{k_{p_1}} \\ \frac{g_2(q) - g_2(q_d) - x_2}{k_{p_2}} \\ \vdots \\ \frac{g_n(q) - g_n(q_d) - x_n}{k_{p_n}} \end{bmatrix} = f(\tilde{q}, q_d). \quad (62)$$

If $f(\tilde{q}, q_d)$ satisfies the Contraction Mapping Theorem (Kelly et al., 2005; Khalil, 2002), then (62) has a unique solution \tilde{q}^* . Considering this, we have

$$\begin{aligned}
\|f(v, q_d) - f(w, q_d)\| &= \left\| \begin{array}{c} \frac{g_1(q_{d-v}) - g_1(q_d) - x_1 - g_1(q_{d-w}) + g_1(q_d) + x_1}{k_{p_1}} \\ \frac{g_2(q_{d-v}) - g_2(q_d) - x_2 - g_2(q_{d-w}) + g_2(q_d) + x_2}{k_{p_2}} \\ \vdots \\ \frac{g_n(q_{d-v}) - g_n(q_d) - x_n - g_n(q_{d-w}) + g_n(q_d) + x_n}{k_{p_n}} \end{array} \right\| \\
&= \left\| \begin{array}{c} \frac{g_1(q_{d-v}) - g_1(q_{d-w})}{k_{p_1}} \\ \frac{g_2(q_{d-v}) - g_2(q_{d-w})}{k_{p_2}} \\ \vdots \\ \frac{g_n(q_{d-v}) - g_n(q_{d-w})}{k_{p_n}} \end{array} \right\| \quad (63)
\end{aligned}$$

Using Theorem 1, we can rewrite $g_i(q_{d-v}) - g_i(q_{d-w})$ as

$$g_i(q_{d-v}) - g_i(q_{d-w}) = \frac{\partial g_i(z)}{\partial z_1} \Big|_{z=\xi_i} (w_1 - v_1) + \frac{\partial g_i(z)}{\partial z_2} \Big|_{z=\xi_i} (w_2 - v_2) + \dots + \frac{\partial g_i(z)}{\partial z_n} \Big|_{z=\xi_i} (w_n - v_n) \quad (64)$$

where ξ_i is a vector on the line segment that joins vectors w and v , and, by substituting in (63), we obtain

$$\begin{aligned}
\|f(v, q_d) - f(w, q_d)\| &= \left\| \begin{array}{c} \frac{g_1(q_{d-v}) - g_1(q_{d-w})}{k_{p_1}} \\ \frac{g_2(q_{d-v}) - g_2(q_{d-w})}{k_{p_2}} \\ \vdots \\ \frac{g_n(q_{d-v}) - g_n(q_{d-w})}{k_{p_n}} \end{array} \right\| \\
&= \left\| \begin{array}{c} \frac{\frac{\partial g_1(z)}{\partial z_1} \Big|_{z=\xi_1} (w_1 - v_1) + \frac{\partial g_1(z)}{\partial z_2} \Big|_{z=\xi_1} (w_2 - v_2) + \dots + \frac{\partial g_1(z)}{\partial z_n} \Big|_{z=\xi_1} (w_n - v_n)}{k_{p_1}} \\ \frac{\frac{\partial g_2(z)}{\partial z_1} \Big|_{z=\xi_2} (w_1 - v_1) + \frac{\partial g_2(z)}{\partial z_2} \Big|_{z=\xi_2} (w_2 - v_2) + \dots + \frac{\partial g_2(z)}{\partial z_n} \Big|_{z=\xi_2} (w_n - v_n)}{k_{p_2}} \\ \vdots \\ \frac{\frac{\partial g_n(z)}{\partial z_1} \Big|_{z=\xi_n} (w_1 - v_1) + \frac{\partial g_n(z)}{\partial z_2} \Big|_{z=\xi_n} (w_2 - v_2) + \dots + \frac{\partial g_n(z)}{\partial z_n} \Big|_{z=\xi_n} (w_n - v_n)}{k_{p_n}} \end{array} \right\|
\end{aligned}$$

$$\begin{aligned}
 &= \|A[w - v]\| \\
 &\leq \|A\| \|w - v\|
 \end{aligned}$$

where

$$A = \begin{bmatrix} \left. \frac{1}{k_{p_1}} \frac{\partial g_1(z)}{\partial z_1} \right|_{z=\xi_1} & \left. \frac{1}{k_{p_1}} \frac{\partial g_1(z)}{\partial z_2} \right|_{z=\xi_1} & \cdots & \left. \frac{1}{k_{p_1}} \frac{\partial g_1(z)}{\partial z_n} \right|_{z=\xi_1} \\ \left. \frac{1}{k_{p_2}} \frac{\partial g_2(z)}{\partial z_1} \right|_{z=\xi_2} & \left. \frac{1}{k_{p_2}} \frac{\partial g_2(z)}{\partial z_2} \right|_{z=\xi_2} & \cdots & \left. \frac{1}{k_{p_2}} \frac{\partial g_2(z)}{\partial z_n} \right|_{z=\xi_2} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{1}{k_{p_n}} \frac{\partial g_n(z)}{\partial z_1} \right|_{z=\xi_n} & \left. \frac{1}{k_{p_n}} \frac{\partial g_n(z)}{\partial z_2} \right|_{z=\xi_n} & \cdots & \left. \frac{1}{k_{p_n}} \frac{\partial g_n(z)}{\partial z_n} \right|_{z=\xi_n} \end{bmatrix} \quad (65)$$

If $\|A\| < 1$, then $f(\tilde{q}, q_d)$ fulfills the Contraction Mapping Theorem. Now notice that

$$\|A\| = \sqrt{\lambda_{\max}\{A^T A\}}. \quad (66)$$

By defining $\left. \frac{\partial g_i(z)}{\partial z_j} \right|_{z=\xi_i} = \Delta g_{ij}$, we have that

$$\begin{aligned}
 A^T A &= \begin{bmatrix} \frac{\Delta g_{11}}{k_{p_1}} & \frac{\Delta g_{21}}{k_{p_2}} & \cdots & \frac{\Delta g_{n1}}{k_{p_n}} \\ \frac{\Delta g_{12}}{k_{p_1}} & \frac{\Delta g_{22}}{k_{p_2}} & \cdots & \frac{\Delta g_{n2}}{k_{p_n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta g_{1n}}{k_{p_1}} & \frac{\Delta g_{2n}}{k_{p_2}} & \cdots & \frac{\Delta g_{nn}}{k_{p_n}} \end{bmatrix} \begin{bmatrix} \frac{\Delta g_{11}}{k_{p_1}} & \frac{\Delta g_{12}}{k_{p_1}} & \cdots & \frac{\Delta g_{1n}}{k_{p_1}} \\ \frac{\Delta g_{21}}{k_{p_2}} & \frac{\Delta g_{22}}{k_{p_2}} & \cdots & \frac{\Delta g_{2n}}{k_{p_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta g_{n1}}{k_{p_n}} & \frac{\Delta g_{n2}}{k_{p_n}} & \cdots & \frac{\Delta g_{nn}}{k_{p_n}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\Delta g_{11}^2}{k_{p_1}^2} + \frac{\Delta g_{21}^2}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n1}^2}{k_{p_n}^2} & \frac{\Delta g_{12}\Delta g_{11}}{k_{p_1}^2} + \frac{\Delta g_{21}\Delta g_{22}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n1}\Delta g_{n2}}{k_{p_n}^2} & \cdots & \frac{\Delta g_{11}\Delta g_{1n}}{k_{p_1}^2} + \frac{\Delta g_{21}\Delta g_{2n}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n1}\Delta g_{nn}}{k_{p_n}^2} \\ \frac{\Delta g_{12}\Delta g_{11}}{k_{p_1}^2} + \frac{\Delta g_{21}\Delta g_{22}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n1}\Delta g_{n2}}{k_{p_n}^2} & \frac{\Delta g_{12}^2}{k_{p_1}^2} + \frac{\Delta g_{22}^2}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n2}^2}{k_{p_n}^2} & \cdots & \frac{\Delta g_{12}\Delta g_{1n}}{k_{p_1}^2} + \frac{\Delta g_{22}\Delta g_{2n}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n2}\Delta g_{nn}}{k_{p_n}^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta g_{11}\Delta g_{1n}}{k_{p_1}^2} + \frac{\Delta g_{21}\Delta g_{2n}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n1}\Delta g_{nn}}{k_{p_n}^2} & \frac{\Delta g_{12}\Delta g_{1n}}{k_{p_1}^2} + \frac{\Delta g_{22}\Delta g_{2n}}{k_{p_2}^2} + \cdots + \frac{\Delta g_{n2}\Delta g_{nn}}{k_{p_n}^2} & \cdots & \frac{\Delta g_{1n}^2}{k_{p_1}^2} + \frac{\Delta g_{2n}^2}{k_{p_2}^2} + \cdots + \frac{\Delta g_{nn}^2}{k_{p_n}^2} \end{bmatrix}
 \end{aligned}$$

Considering (5) and (27), we have that each element in $A^T A$ fulfills

$$|A^T A(i, j)| < \frac{1}{n}. \quad (67)$$

Now, knowing that the eigenvalues of any matrix B, where b_{ij} denotes its ij -th element, fulfill (Horn & Johnson, 1985):

$$|\lambda_k| \leq n \left[\max_{i,j} \{ |b_{ij}| \} \right] \quad \forall k = 1, \dots, n$$

we obtain that

$$\lambda_k \{A^T A\} \leq \lambda_{\max} \{A^T A\} \leq n \left[\max_{i,j} \{ |A^T A(i,j)| \} \right] < n \left[\frac{1}{n} \right] = 1$$

and consequently we have that $\|A\| = \sqrt{\lambda_{\max} \{A^T A\}} < \sqrt{1} = 1$. Therefore, we get $\|f(v, q_d) - f(w, q_d)\| \leq \|A\| \|w - v\|$ where $\|A\|$ is strictly smaller than the unity. Hence, we have that (26) has a unique solution $\tilde{q} = h(x) \in \mathbb{R}^n$ provided that:

$$k_{p_i} > n \left(\max_{q,j} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right) \quad \text{where } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, n.$$

Appendix B

The positive definiteness and radial unboundedness analysis of $W(\tilde{q}, \dot{q})$ is dealt in this appendix. The Lyapunov function candidate $W(\tilde{q}, \dot{q})$ can be written as:

$$W(\tilde{q}, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + W_1(\tilde{q})$$

with

$$\begin{aligned} W_1(\tilde{q}) &= \sum_{i=1}^n \int_0^{\tilde{q}_i} \left[\text{Sat} \left[\text{Sat}(k_{p_i} r_i + x_i + g_i(q_d)) \right] - \bar{g}_i(r_i) \right] dr_i \\ &\quad - \sum_{i=1}^n \int_0^{h_i(x)} \left[\text{Sat} \left[\text{Sat}(k_{p_i} r_i + x_i + g_i(q_d)) \right] - \bar{g}_i(r_i) \right] dr_i. \end{aligned} \quad (68)$$

where

$$\begin{aligned} \bar{g}_1(r_1) &= g_1(q_{d_1} - r_1, q_{d_2}, \dots, q_{d_n}) \\ \bar{g}_2(r_2) &= g_2(q_{d_1} - \tilde{q}_1, q_{d_2} - r_2, \dots, q_{d_n}) \\ &\quad \vdots \\ \bar{g}_n(r_n) &= g_n(q_{d_1} - \tilde{q}_1, q_{d_2} - \tilde{q}_2, \dots, q_{d_n} - r_n) \end{aligned}$$

Notice that the positive definiteness and the radial unboundedness of $W_1(\tilde{q}, \dot{q})$ implies the positive definiteness and the radial unboundedness of $W(\tilde{q}, \dot{q})$.

Let us define a region β_1 where the saturation functions of the P and PI parts of the controller work in their linear section, such that:

$$\beta_1 = \{ \tilde{q}_i : |k_{p_i} r_i + x_i + g_i(q_d)| < l_{p_i} \text{ and } |k_{p_i} r_i + x_i + g_i(q_d)| < l_{p_i} \}.$$

Notice that, in this region we have that $\text{Sat}[\text{Sat}(k_{p_i} \tilde{q}_i + x_i + g_i(q_d))] = k_{p_i} k_{p_{c_i}} \tilde{q}_i + x_i + g_i(q_d)$. For this case, we will show that $W_1(\tilde{q})$ is a strictly convex function with a unique minimum

point at $\tilde{q} = h_1(x)$. To this end, we evaluate $W_1(\tilde{q})$ at $\tilde{q} = h_1(x)$ and obtain its gradient and Hessian:

a) $W_1(\tilde{q})|_{\tilde{q}=h_1(x)}$ can be written as:

$$\begin{aligned} W_1(\tilde{q})|_{\tilde{q}=h_1(x)} &= \sum_{i=1}^n \int_0^{\tilde{q}_i} \left[k_{p_i} r_i + x_i + g_i(q_d) - \bar{g}_i(r_i) \right] dr_i \Big|_{\tilde{q}=h_1(x)} \\ &\quad - \sum_{i=1}^n \int_0^{h_1(x)} \left[k_{p_i} r_i + x_i + g_i(q_d) - \bar{g}_i(r_i) \right] dr_i \\ &= 0 \end{aligned}$$

b) The gradient of $W_1(\tilde{q})$ with respect to \tilde{q} is given by:

$$\frac{\partial W_1(\tilde{q})}{\partial \tilde{q}} = K_p \tilde{q} + x + g(q_d) - g(q_d - \tilde{q}) = 0. \quad (69)$$

Under assumption (27), and by using the Contraction Mapping Theorem, (69) has a unique solution $\tilde{q} = h_1(x)$, that is, a unique critical point.

c) The Hessian of $W_1(\tilde{q})$ with respect to \tilde{q} is given by:

$$\frac{\partial^2 W_1(\tilde{q})}{\partial \tilde{q}^2} = K_p - \frac{\partial g(q_d - \tilde{q})}{\partial \tilde{q}} = 0$$

which is a positive definite function for all $\tilde{q} \in \mathbb{R}^n$ provided that (Hernandez-Guzman et al., 2008):

$$k_{p_i} > \sum_{j=1}^n \max_q \left| \frac{\partial g_j(q)}{\partial q_j} \right| \quad (70)$$

Note that (27) implies (70).

Therefore, in the linear region β_1 , $W_1(\tilde{q})$ is a strictly convex function with a unique minimal point $\tilde{q} = h_1(x)$ which implies that $W_1(\tilde{q} - h_1(x))$ is a locally positive definite function.

Also notice that the gradient of $W_1(\tilde{q})$ with respect to \tilde{q} is given globally by

$$\frac{\partial W_1(\tilde{q})}{\partial \tilde{q}} = \text{Sat} \left[\text{Sat} \left(K_p \tilde{q} + x + g(q_d) \right) \right] - g(q_d - \tilde{q}) \quad (71)$$

which, under Assumption 2, will have a unique critical point for all $\tilde{q}_i \in \mathbb{R}$ with $i = 1, 2, \dots, n$, and hence, the minimum point of $W_1(\tilde{q})$ results to be a global minimum point $\tilde{q} = h_1(x)$.

In order to prove radially unboundedness of $W_1(\tilde{q})$, it is possible to prove that outside of the region β_1 the function $W_1(\tilde{q})$ can be lower bounded by straight lines of the type

$$W_{\beta_1} = k_{\beta_1} \left| \tilde{q}_i - h_{1_i}(x) \right| - c_i$$

where k_{β_1} and c_i are suitable constants. So, $|\tilde{q}_i - h_{1_i}(x)| \rightarrow \infty$ implies $W_{\beta_1} \rightarrow \infty$ for $i = 1, 2, \dots, n$; therefore $W(\tilde{q}) \rightarrow \infty$ as $\|\tilde{q}\| \rightarrow \infty$, which proves that $W_1(\tilde{q})$ is radially unbounded.

Real-Time-Position Prediction Algorithm for Under-actuated Robot Manipulator Using of Artificial Neural Network

Ahmad Azlan Mat Isa, Hayder M.A.A. Al-Assadi and Ali T. Hasan
*Faculty of Mechanical Engineering, University Technology MARA (UiTM)
Shah Alam, 40450
Malaysia*

1. Introduction

Robot manipulators, in general, are required to have the same number of actuators as the number of joints to obtain full control. In the case of under-actuated robots, this condition is not satisfied which make the behavior of that class of robots very difficult to be predicted. Under-actuated robots can be a better design choice for robots in space and other industrial applications, their advantages over fully actuated robots led to many studies to predict their behavior (Yu et al., 1998; Berkemeier & Fearing, 1999; Spong, 1995; Ono et al., 2001; Nakanishi et al., 2000; Funda et al., 1996; Luca et al., 2000; Luca & Oriolo, 2002; Arai & Tachi, 1991; Mukherjee & Chen, 1993; Yu et al., 1993; Bergerman et al., 1995; Mahindrakar et al., 2006; Muscato, 2006; Begovich et al., 2002). As a first advantage, a light-weight and low power consumption manipulator can be made. This feature is required in low cost automation and space robots. Second, they can easily overcome actuator failure due to unexpected accident. The under-actuated manipulator could be the model of the direct drive manipulator that has some failed joints; such fault-tolerant behavior is highly desirable for robots in remote or hazardous environments (Yu et al., 1998). Other interesting applications include the Acrobot (Berkemeier & Fearing, 1999; Spong, 1995), the gymnast robots (Ono et al., 2001), the brachiating robots (Nakanishi et al., 2000), and surgical robots (Funda et al., 1996).

The mathematical complexity and wide variety of applications have kept under-actuated robots an area of open research. (Luca et al., 2000; Luca & Oriolo, 2002) have investigated the behavior of a 2R manipulator moving in a horizontal plane with a single actuator at the first joint, neglecting joint friction which is not easy to achieve in real world as it involves high manufacturing cost. Trying to overcome that problem, some researchers have implemented additional equipments such as brakes at the passive joint (Arai & Tachi, 1991; Mukherjee & Chen, 1993; Yu et al., 1993; Bergerman et al., 1995). In this case, the brake can generate torque that means after all that kind of systems is considered some kind of actuator. So, it will be difficult to consider that robot as an under-actuated manipulator.

Motivated by this problem, (Yu et al., 1998) have investigated the dynamic characteristics of a two-link manipulator in view of global motion including joint friction by proposing a mathematical model; they have found that the manipulator can be positioned if the friction

acts on the passive joint. In this case, any additional equipment such as brakes is not needed in positioning all the joints to desired position. Their results were verified using numerical simulation. Later on, (Mahindrakar et al., 2006) have presented a mathematical model for a two-link under-actuated manipulator wherein the motion of the system was confined to a horizontal plane; their proposed dynamic model takes into account the frictional forces acting on the joints. Results obtained were also verified through numerical simulation.

Many attempts to solve the problem have been found in the literature. Yet, solutions proposed are still lack of generality and systematization. To overcome this problem, artificial intelligence was introduced for prediction and making robot systems able to attribute more intelligence and high degree of autonomy.

Applying fuzzy logic to under-actuated robots (as an artificial intelligence method), there were few studies in recent past (Muscato, 2006; Begovich et al., 2002).

Although the results presented were promising, these results cannot be generalized to other systems, because they only came from practical considerations. Besides, despite the fact that unlike most learning control algorithms, multiple trials are not necessary for the robot to learn the desired trajectory. A major drawback was that Fuzzy Logic based approaches only remembers the most recent data points introduced (Graca & Gu, 1993). Gleaning the learning abilities of genetic algorithms GA (as another method of artificial intelligence) to solve the problem was an alternative. Blending of GA with fuzzy rules, in order to capture the hidden nonlinearities of the system will be useful in developing any learning techniques. (Lee & Zak, 2002) have presented the design criterion of a GA based neural fuzzy controller for an anti-break system. As it has been seen, each of the previously mentioned techniques has their own drawbacks. To overcome this problem researchers have recommended neural networks so that it would remember the trajectories as it traversed them (Graca & Gu, 1993). Artificial neural networks (ANNs) have been widely used for their extreme flexibility due to its learning ability and the capability of non-linear function approximation. Their ability to learn by example makes them very flexible and powerful. ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification (Kalogirou, 2001; Hasan et al., 2006). Therefore, ANNs have been intensively used for solving regression and classification problems in many fields. A number of realistic approaches have been proposed and justified for applications to robotic systems (Balakrishnan et al., 2000; Kim et al., 2002; Köker, 2005; Hasan et al., 2007; Al-Assadi et al., 2007; Siqueira & Terra, 2009).

In real world application, no physical property such as the friction coefficient can be exactly derived. Besides, there are always kinematics uncertainties presence in the real world such as ill-defined linkage parameters and backlashes in gear trains (Hasan et al., 2009; Hasan et al., 2010). In this paper, and to overcome whichever uncertainty presented in the real world, data were recorded experimentally from sensors fixed on each joint for a horizontal two-link under-actuated robot.

The developed learning algorithm is based on weight adaptation of the network, by minimizing the tracking error after each iteration process. This scheme does not require any prior knowledge of the dynamic model of the system being controlled. The basic idea of this concept is the use of the ANNs to learn the characteristics of the robot system rather than to specify an explicit robot system model, so, every uncertainty in the system will be counted for. Experimental trajectory tracking has shown the ability of the proposed approach to

overcome the disadvantages of using some schemes like the Fuzzy Learning for example that only remembers the most recent data sets introduced, as the literature has shown.

2. Equations of motion with friction effect

As Figure 1 show, the space coordinate of the manipulator is parameterized by q .

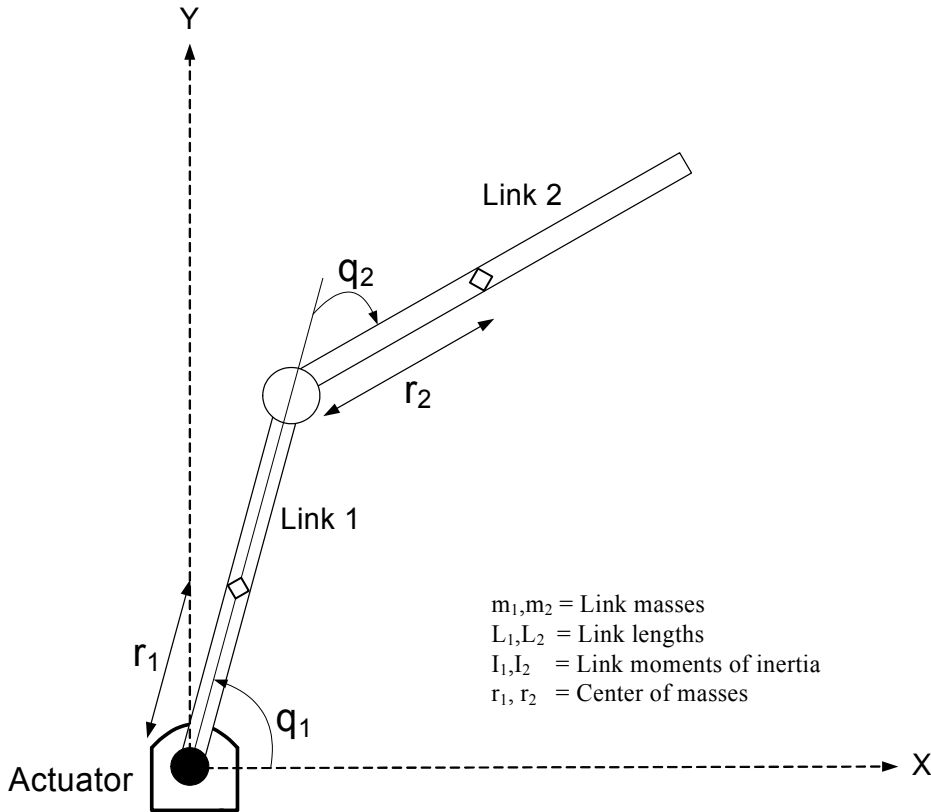


Fig. 1. Schematic diagram of the robot used

The coordinate $q_i, i = 1, 2$ are the joint angles. The Euler-Lagrange equation of motion is (Mahindrakar et al., 2006):

$$M(q)\ddot{q} + h(q, \dot{q}) = \tau, \quad (1)$$

Where \dot{q} and \ddot{q} are the generalized velocities and accelerations respectively. $M(q)$ is the inertia matrix, which is symmetric and positive definite. The centripetal and Coriolis terms are collected in the vector $h(q, \dot{q})$. The vector h contains terms purely quadratic in the velocities; gravity terms are absent since it assumed that the manipulator moves in a horizontal plane.

Define the following constants:

$$c_1 = m_1 r_1^2 + m_2 l_1^2 + I_1, \quad c_2 = m_2 r_2^2 + I_2, \quad c_3 = m_2 l_1 r_2.$$

The equations of motion accounting for the Coulomb plus viscous friction at the joints become:

$$m_{11} \ddot{q}_1 + m_{12} \ddot{q}_2 + h_1 = \tau - \text{SGN}(\dot{q}_1) F_1 - b_1 \dot{q}_1, \quad (2)$$

$$m_{21} \ddot{q}_1 + m_{22} \ddot{q}_2 + h_2 = -\text{SGN}(\dot{q}_2) F_2 - b_2 \dot{q}_2, \quad (3)$$

Where,

$$m_{11} = c_1 + c_2 + 2c_3 \cos q_2, \quad m_{12} = c_2 + c_3 \cos q_2,$$

$$m_{21} = m_{12}, \quad m_{22} = c_2,$$

$$h_1 = -c_3 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \sin q_2, \quad h_2 = c_3 \dot{q}_1^2 \sin q_2.$$

The $F_i, b_i, q_i, i = 1, 2$ represent the Coulomb and viscous friction forces respectively. The set-valued signum function is defined as:

$$\text{SGN}(x) \begin{cases} \{1\} & \text{if } x > 0, \\ \{-1\} & \text{if } x < 0, \\ [-1, 1] & \text{if } x = 0. \end{cases} \quad (4)$$

The above shown function suffers from the fact that the solution does not give a clear indication on how to select an appropriate solution from the several possible solutions for a particular arm configuration.

3. Experiment procedure

In this section, the real time implementation of the experimentally collecting data procedure is discussed. Different methods for collecting data have been found in the literature. Using a pre-specified model, using a trajectory planning method or using a simulation program for this purpose are examples for some of these methods. However, there are always kinematics uncertainties presences in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear train, in this approach, data were recorded directly from sensors fixed on each joint, so every uncertainty in the dynamics of the system will be counted for.

The manipulator used is shown in Figure 2, which is actuated only at the first joint. The actuator used is a DC motor connected to the first link through a gearbox with a reduction ratio of 100:1, while the second joint is passive.

Each of the joints have an encoder attached to it, in order to measure the rotation angle and there are torque sensors between the motor output shaft and the robot joint to measure the torque being supplied by the motor. Joints encoders are connected to a computer equipped with MATLAB software through a data acquisition card. The robot arms were made of an aluminum square section beam to ensure a resisting to bending lightweight arm. Length of arms are $l_1 = 40$ cm and $l_2 = 30$ cm respectively. The control circuit is made up of computer

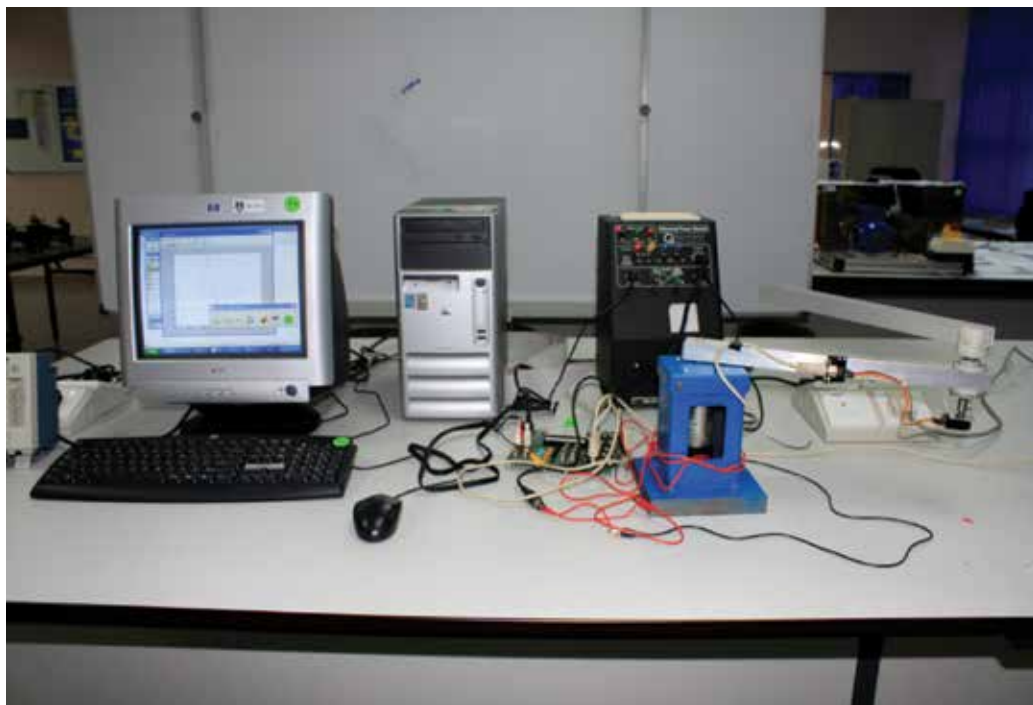


Fig. 2. The robot system used showing the computer, the data acquisition card and the robot arms

with the MATLAB software connected to the robot through a data acquisition card that acquires the motion data of the two links. Input signal is generated by the MATLAB software and transferred to the motor using the electrical board, and the robot response is recorded using the MATLAB software.

A Sinusoidal excitation signal was applied to the actuator causing different torque to the joints and the dynamic coupling effect was moving the passive joint correspondently. As a standard signal generated by the MATLAB, Sinusoidal excitation signal, was chosen in order to cause a robot motion that covers the whole working cell rather than being a specified signal to perform a pre-defined trajectory.

When the excitation signal is given, the motion of the active joint and the corresponding response of the passive joint that can be seen in Figures 3 and 4 respectively were recorded in order to be used in the training process of the ANN.

4. The adaptive learning algorithm

The fundamental idea underlying the design of the network is that the information entering the input layer is mapped as an internal representation in the units of the hidden layer and the outputs are generated by this internal representation rather than by the input vector. Given that there are enough hidden neurons, input vectors can always be encoded in a form so that the appropriate output vector can be generated from any input vector.

Figure 5 shows the network used. The output of the units in layer A are multiplied by appropriate weights W_{ij} and these are fed as inputs to the hidden layer. Hence if O_i are the output of units in layer A , then the total input to the hidden layer, i.e., layer B is:

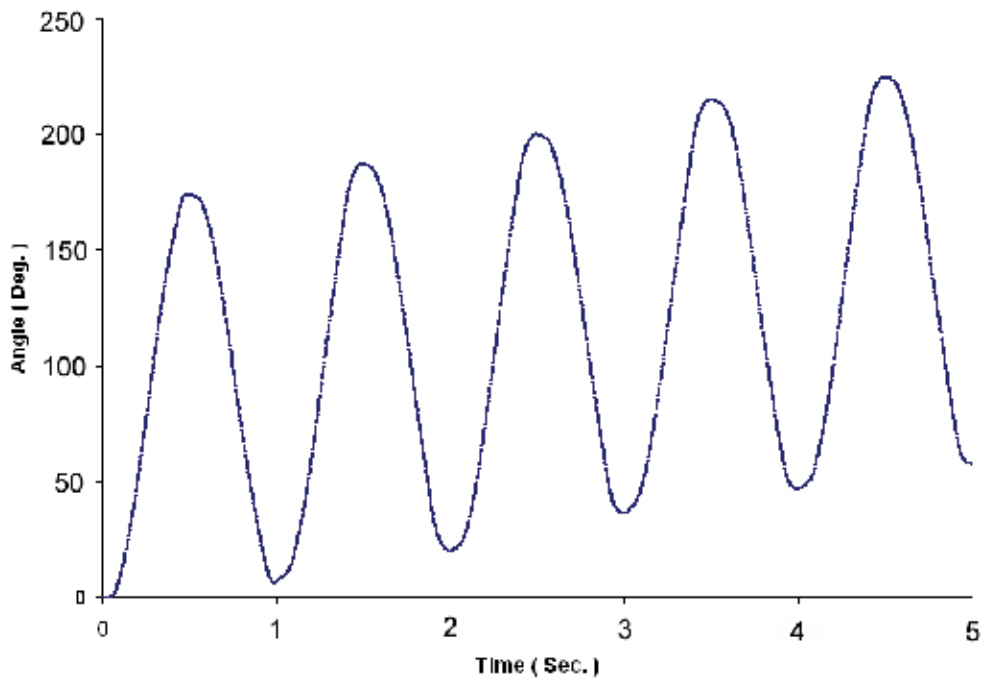


Fig. 3. Trajectory of the active joint when the excitation signal was applied

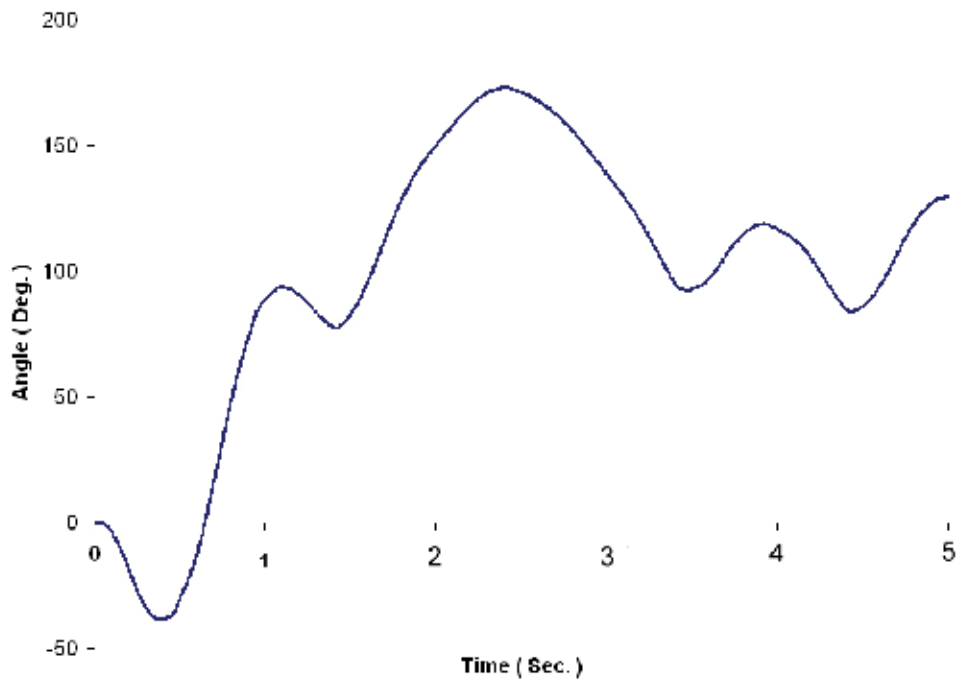


Fig. 4. Corresponding trajectory of the passive joint

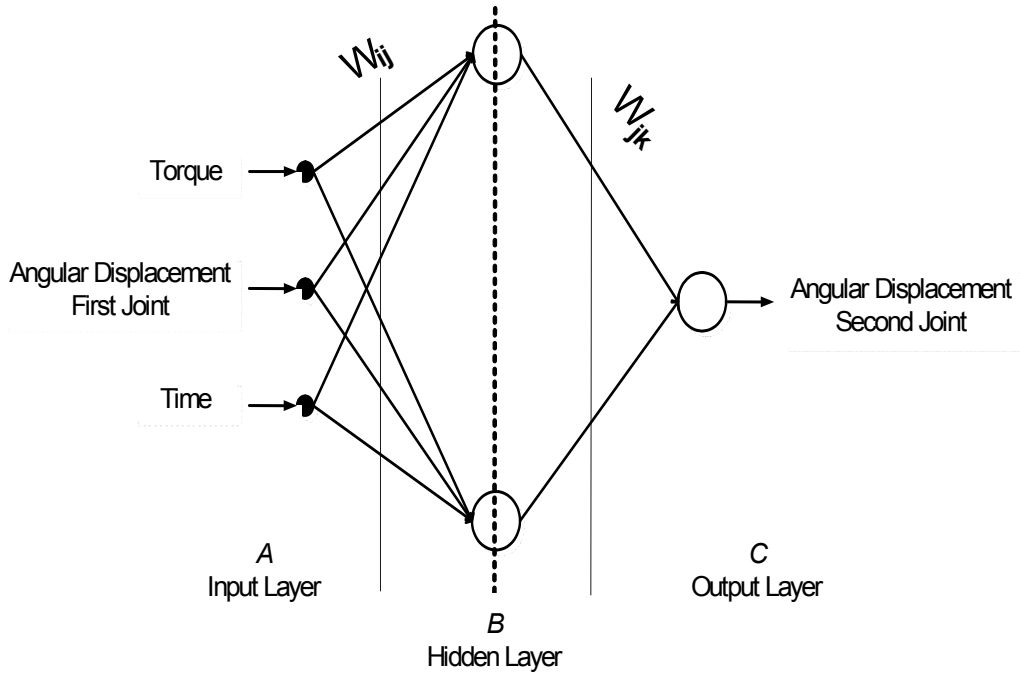


Fig. 5. The topology of the ANN used

$$Sum_B = \sum_i O_i W_{ij} \quad (5)$$

And the output O_j of a unit in layer B is:

$$O_j = f(sum_B) \quad (6)$$

Where f is a non-linear activation function, it is a common practice to choose the sigmoid function given by: -

$$f(O_j) = \frac{1}{1 + e^{-O_j}} \quad (7)$$

As a nonlinear activation function.

However, any input-output function that possesses a bounded derivative can be used in place of the sigmoid function.

If there is a fixed, finite set of input-output pairs, the total error in the performance of the network with a particular set of weights can be computed by comparing the actual and the desired output vectors for each presentation of an input vector.

Error at any output unit e_K in the layer C can be calculated by: -

$$e_K = d_K - O_K \quad (8)$$

Where d_K is the desired output for that unit in layer C and O_K is the actual output produced by the network .the total error E at the output can be calculated by: -

$$E = \frac{1}{2} \sum_K (d_K - O_K)^2 \quad (9)$$

Learning comprises changing weights so as to minimize the error function.

To minimize E by the gradient descent method. It is necessary to compute the partial derivative of E with respect to each weight in the network. Equations (5) and (6) describe the forward pass through the network where units in each layer have their states determined by the inputs they received from units of lower layer.

The backward pass through the network that involves “back propagation” of weight error derivatives from the output layer back to the input layer is more complicated. For the sigmoid activation function given in equation (7), the so-called delta-rule for iterative convergence towards a solution may be stated in general as:

$$\Delta W_{JK} = \eta \delta_K O_J \quad (10)$$

Where η is the learning rate parameter, and the error δ_K at an output layer unit K is given by: -

$$\delta_K = O_K(1 - O_K)(d_K - O_K) \quad (11)$$

And the error δ_J at a hidden layer unit is given by: -

$$\delta_J = O_J(1 - O_J) \sum_K \delta_K W_{JK} \quad (12)$$

Using the generalized delta rule to adjust weights leading to the hidden units is back propagating the error-adjustment, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons.

A back propagation network trains with two step procedure, the activity from the input pattern flows forward through the network and the error signal flows backwards to adjust the weights using the following equations: -

$$W_{IJ} = W_{IJ} + \eta \delta_J O_I \quad (13)$$

$$W_{JK} = W_{JK} + \eta \delta_K O_J \quad (14)$$

Until for each input vector the output vector produced by the network is the same as (or sufficiently close to) the desired output vector (Kalogirou, 2001; Hasan et al., 2006). Number of hidden neurons and the learning factor are determined by trial and error.

5. Results

A supervised feed forward ANN was designed using C programming language to learn the system behavior over its workspace. The network consists of input, output and one hidden layer, the input vector for the network consists of the angular displacement, the torque applied at the active joint (first joint) and the time interval, while the output vector was the angular position of the passive joint (second joint). As can be seen in Figure 5, every neuron in the network is fully connected with each other, sigmoid transfer function was used to be

the activation function, and generalized backpropagation delta learning rule (GDR) algorithm was used in the training process. All control datasets values had been scaled individually so that the overall difference in the dataset was maximized.

Training data were divided into 50 input-output sets, which covered the entire work cell of the manipulator. To build the control knowledge, a training process was carried out using the experimentally obtained data. The network was trained by presenting several target points that the network had to learn, number of neurons in the hidden layer was set to 25 with a constant learning factor of 0.9 by trial and error. Figure 6 shows the building knowledge process for the system.

To verify the success of the algorithm, the predicted values of the passive joint were compared to the experimentally collected data. The average absolute error was 4.9% after 100,000 Iterations. Figure 7 graphically shows the trajectory tracking of the passive joint, Results obtained show that the design network is capable of learning and predicting the position of the passive joint successfully.

6. Conclusions and recommendations for further research

In this paper, the Artificial Neural Network technique was applied to the problem of positioning an under-actuated robot manipulator. The position of the passive joint of under-actuated 2R manipulator is now learned through training a network based only on observation of the input-output relationship.

The proposed technique does not require any prior knowledge of the system model, the basic idea of this concept is the use of the ANN to learn the characteristics of the robot system rather than to specify explicit robot system model. Any modification in the physical set-up of the robot such as the addition of a new tool would only require training for a new trajectory without the need for any major system software modification, which is a significant advantage of using neural network approach.

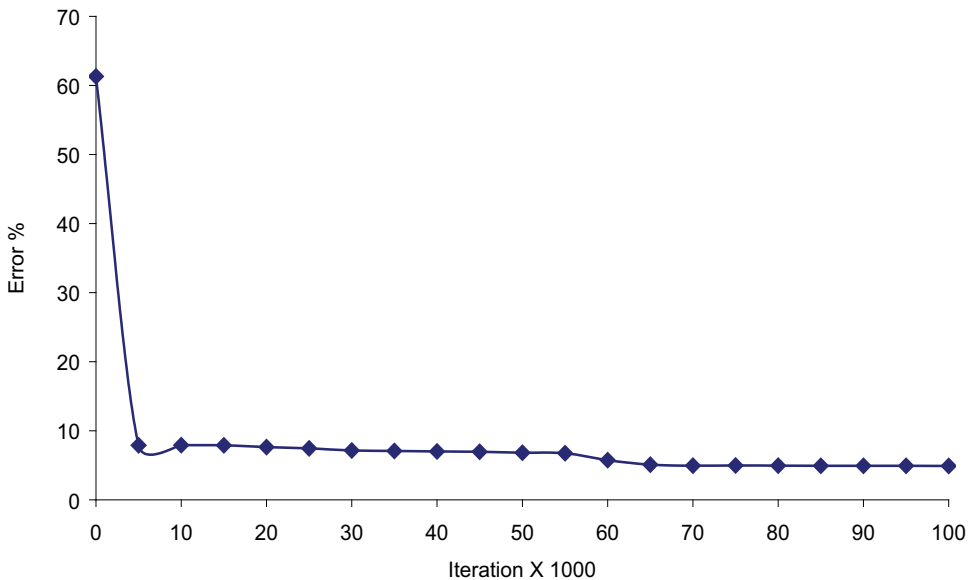


Fig. 6. Building knowledge curve of the system

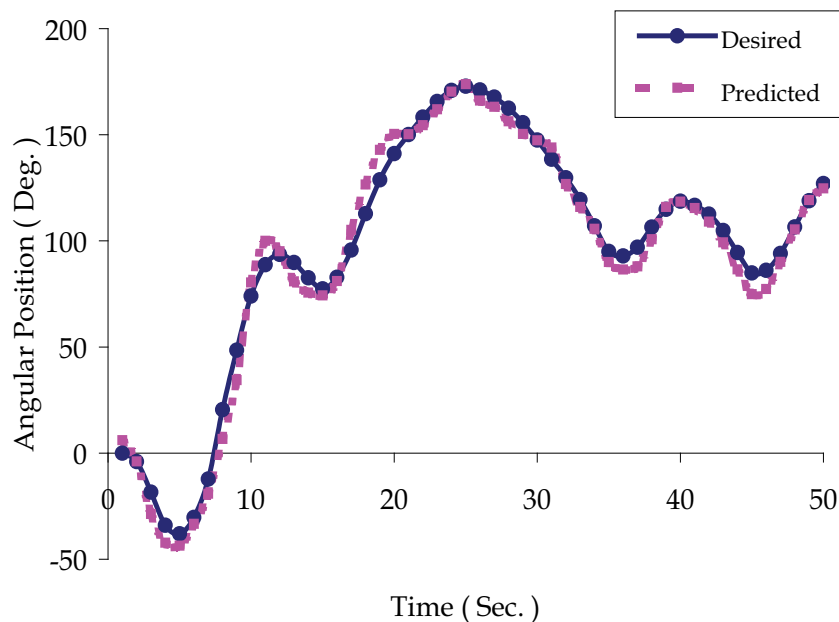


Fig. 7. Predicted trajectory tracking of the passive joint

Results obtained have shown the ability of the network to predict the trajectory of the passive joint, that is positioned by the dynamic coupling of the active joint, overcoming the disadvantages of using some schemes like the Fuzzy Learning for example that only remembers the most recent data sets introduced.

Backpropagation algorithm was used as a learning algorithm with sigmoid transfer function as an activation function in all neurons, For further research, we recommend that a different learning algorithm, different activation function and/or different number of hidden layers to be used in order to achieve, if possible, a better response in terms of precision and iteration.

7. References

- Al-Assadi, H.M.A.A., Hamouda, A.M.S., Ismail, N. and Aris, I., An Adaptive Learning Algorithm for Controlling a Two-Degree-Of-Freedom Serial Ball-and-Socket Actuator. *Proceedings of the IMechE Part I: Journal of Systems and Control Engineering*, 2007,221(7): 1001-1006.
- Arai, H. and Tachi, S., Position Control of a Manipulator With Passive Joints Using Dynamic Coupling. *IEEE Trans. On Robotics and Automation*, 1991,7(4): 528-534.
- Balakrishnan, S., Popplewell, N. and Thomlinson, M., Intelligent Robotic Assembly. *International Journal of Computers and Industrial Engineering*, 2000, 38: 467-478.
- Begovich, O., Sanchez, E.N., and Maldonado, M., Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an under-actuated robot. *IEEE Transactions on Control Systems Technology*, January 2002, 10(1): 14-20.
- Bergerman, M., Lee, C. and Xu, Y., Experimental Study of an Underactuated Manipulator. *Proc. 1995 IEEE/RSJ Int. Conf. On Intelligent Robotics and Systems*, 1995,2:317-322.
- Berkemeier, M. D. and Fearing, R. S., Tracking fast inverted trajectories of the underactuated

- acrobot, IEEE Transactions on Robotics and Automation, Aug. 1999, 15(4): 740 - 750.
- Funda, J., Taylor, R., Eldridge, B., Gomory, S. and Gruben, K., Constrained Cartesian motion control for teleoperated surgical robots. IEEE Transactions on Robotics and Automation, June 1996, 12(3): 453 - 465.
- Graca, R.A. and Gu, Y., A Fuzzy Learning Algorithm for Kinematics Control of a Robotic System. Proceeding of the 32nd Conference on Decision and Control. San Antonio, Texas. December, 1993: 1274-1279.
- Hasan, A. T., Hamouda, A.M.S., Ismail, N, Aris, I. and Marhaban, M.H., Trajectory Tracking for a Serial Robot Manipulator Passing Through Singular Configurations Based on the Adaptive Kinematics Jacobian Method. Proceedings of the IMechE Part I: Journal of Systems and Control Engineering, 2009, 223(3): 393-415.
- Hasan, A. T., Ismail, N, Hamouda, A.M.S., Aris, I., Marhaban, M.H. and Al-Assadi, H.M.A.A., Artificial Neural Network-Based Kinematics Jacobian Solution for Serial Manipulator Passing Through Singular Configurations. International Journal of Advanced in Engineering Software, 2010, 41:359-367.
- Hasan, A.T., Hamouda, A.M.S., Ismail, N., and Al-Assadi, H.M.A.A., An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. Journal of Advances in Engineering Software, 2006, 37(7): 432-438.
- Hasan, A.T., Hamouda, A.M.S., Ismail, N., and Al-Assadi, H.M.A.A., A New Adaptive Learning Algorithm for Robot Manipulator Control. Proceeding of the IMechE, Part I: Journal of System and Control Engineering, 2007, 221(4): 663-672.
- Kim, I.S., Son, J.S., Park, C.E. Lee, C.W., and Prasad, Y. K.D.V., A Study on Prediction of Bead Height in Robotic Arc Welding Using a Neural Network. International Journal of Materials Processing Technology, 2002,130-131: 229-234.
- Köker, R., Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. International Journal of Engineering Applications of Artificial Intelligence, 2005, 18: 685-693.
- Lee, Y. and Zak, S. H., Designing a genetic neural antilock-break system controller. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 198 - 211.
- Luca, A.D. and Oriolo, G., Trajectory Planning and Control for Planar Robots With Passive Last Joint. International Journal of Robotics Research, 2002, 21:575-590.
- Luca, A.D., Mattone, R. and Oriolo, G., Stabilization of an Underactuated Planar 2R Manipulator. International Journal of Robust and Nonlinear Control, 2000,24:181-198.
- Mahindrakar, A.D., Rao, S. and Banavar, R.N., Point-to Point Control of a 2R Planar Horizontal Underactuated Manipulator. International Journal of Mechanism and Machine Theory, 2006, 41:838-844.
- Mukherjee, R. and Chen, D., Control of Free-Flying Underactuated Space Manipulators To Equilibrium Manifolds. IEEE Trans. On Robotics and Automation, 1993, 9(5): 561-570.
- Muscato, G., Fuzzy Control of an Underactuated Robot With a Fuzzy Microcontroller. International Journal of Microprocessors and Microsystems, 1999,23:385-391.
- Nakanishi, J. , Fukuda, T. and Koditschek, D., A brachiating robot controller. IEEE Transactions on Robotics and Automation, April 2000,16(2): 109 - 123.

- Ono, K., Yamamoto, K. and Imadu, A. Control of giant swing motion of a two-link horizontal bar gymnastic robot. *Advanced Robotics*, 2001, 15(4): 449 – 465.
- Siqueira, A. A. G. and Terra, M. H., Neural Network-Based H_∞ Control for Fully Actuated and Underactuated Cooperative Manipulator. *International Journal of Control Engineering Practice*, 2009, 17:418-425.
- Kalogirou, S. A., Artificial Neural Networks In Renewable Energy Systems Applications: a review. *International Journal of Renewable and Sustainable Energy Reviews*. 2001, 5: 373-401.
- Spong, M. W., The swing up control problem for the acrobat. *IEEE Control Systems Magazine*, Feb. 1995, 15(1): 49 – 55.
- Yu, K-H, Shito, Y. and Inooka, H., Position Control of an Underactuated Manipulator Using Joint Friction. *International Journal of Non-Linear Mechanics*, 1998, 33(4): 607- 614.
- Yu, K-H., Takahashi, T. and Inooka, H. ,Dynamics and Motion Control of a Two-Link Robot Manipulator With a Passive Joint. *Proc. 1995 IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, 1995, 2:311-316.

On Nonlinear Control Perspectives of a Challenging Benchmark

Guangyu Liu and Yanxin Zhang
The University of Auckland
New Zealand

1. Introduction

Dynamical systems are often nonlinear in nature. It motivates people to explore various theoretical nonlinear analysis and control design tools, of which constructive nonlinear design methods are the most celebrated ones. However, applying a constructive tool faces up a big hurdle that the tool deals only with a certain dynamical structure, often not possessed by the natural dynamics. Nonlinear constructive control designs heavily relies on the identification of a particular structure via coordinate transformation and control transformation. To be realistic, these theoretical tools are not general to all of the nonlinear systems. Here, a challenging benchmark example—a four degrees of freedom inverted pendulum under the influence of a planar force—is considered that is nonlinear, multiple input and multiple output, underactuated and unstable. The benchmark is also of practical interests because it is an abstract of several applications. Three challenging control objectives are envisaged for the first time in the literature in order to how to apply various cutting-edge theoretical nonlinear control tools. In fact, the key step of all of the nonlinear designs is to identify spectral structures— certain “normal” forms. From this aspect, a sequence of preliminary designs will accompany the existing tools to construct nonlinear controllers, which is quite different from the linear control designs.

2. The benchmark problem

2.1 Modeling

The spherical inverted pendulum is subject to a holonomic constraint on the vertical direction and its self-spin about the principal axis along the pole is neglected from the context. As a result, the benchmark has only four degrees of freedom described by a set of generalized coordinates $q \in R^4$ that include two translational ones (also called external variables) and two angular ones (also called shape variables). The translational coordinates are unanimously denoted by two globally fixed Cartesian coordinates (x,y) while the angular ones have several choices as is given later. $Q \in R^4$ denotes the generalized input for the system with

$$Q = (F_x, F_y, 0, 0)^T + v_f, \quad (1)$$

where $(F_x, F_y) \triangleq F$ is the actual planar force and $v_f \in R^4$ is a collection of exogenous disturbances and unmodelled dynamics.

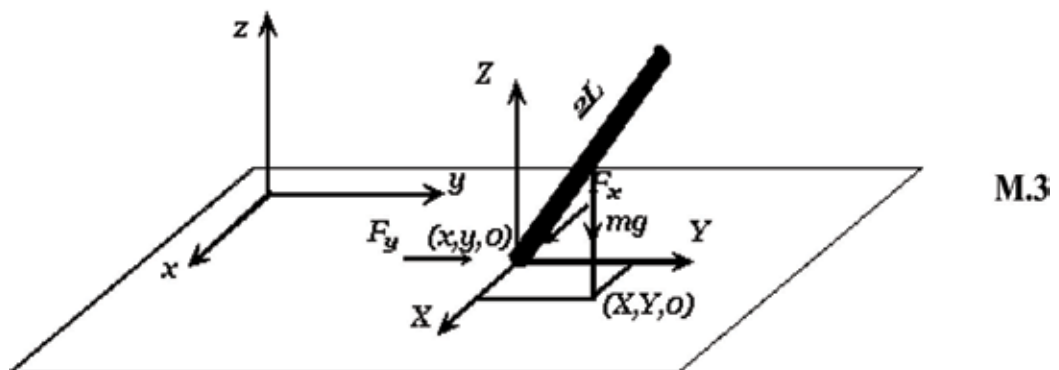
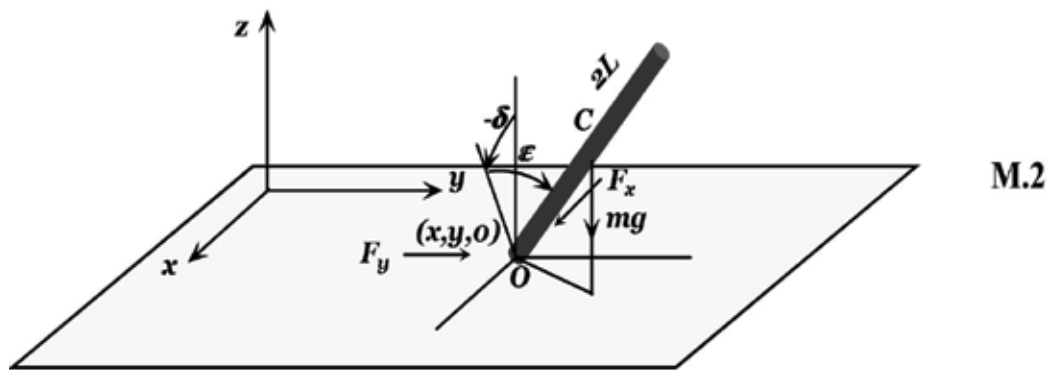
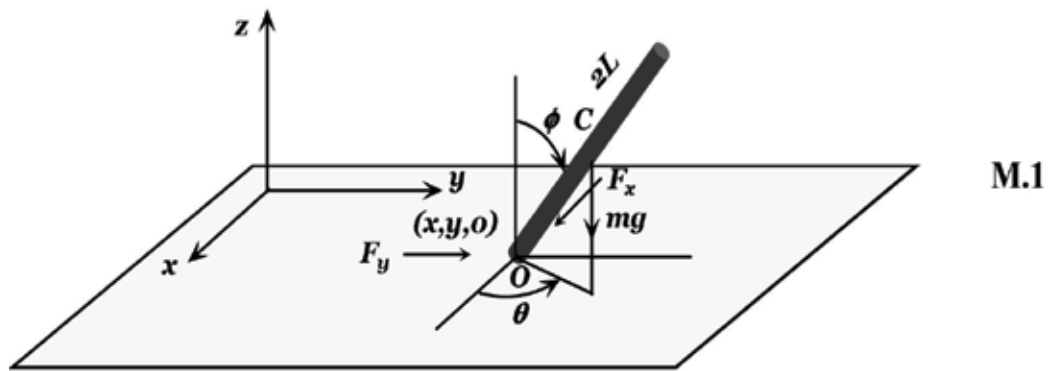


Fig. 1. The configurations of a spherical inverted pendulum

Define a Lagrangian $\mathcal{L} = K - V$ where K and V are respectively the kinetic energy and the potential energy of the benchmark. Applying the Euler-Lagrangian equations

$$\frac{d}{dt} \frac{d\mathcal{L}}{d\dot{q}} - \frac{d\mathcal{L}}{dq} = Q \quad (2)$$

for the benchmark derives the dynamics

$$\mathbf{D}(q) \cdot \{\ddot{q}\} + \mathbf{C}(q, \dot{q}) \cdot \{\dot{q}_i\} + \mathbf{G}(q) = Q, \quad (3)$$

where $\mathbf{D}(q)$ is the matrix of inertia, $\mathbf{C}(q, \dot{q})$ is the centrifugal and Coriolis matrix and $\mathbf{G}(q)$ is the gravitational matrix. Equation 3 is taken as the mathematical model of the benchmark.

Three models with respect to three sets of generalized coordinates are derived (see Fig. 1)

M.1 *The model in $q = (x, y, \theta, \phi)$ in (Liu, 2006) - θ and ϕ are the procession and nutation angles respectively; the model has singular points at $\phi = \dots, 0, \pi, 2\pi, \dots$ but the model is ideal for the objective of swing-up (e.g., (Albouy & Praly, 2000)); the upper space is defined by $U = \{(x, y, \theta, \phi, \dot{x}, \dot{y}, \dot{\theta}, \dot{\phi}) \in \mathbb{R}^8 \mid -\pi/2 < \phi < \pi/2\}$;*

M.2 *The model in $q = (x, y, \delta, \epsilon)$ in (Liu et al., 2008a) - δ and ϵ denote the heading and bank angles respectively; the model has singular points at $\delta = \pi/2, 3\pi/2, \dots$ and/or $\epsilon = \pi/2, 3\pi/2, \dots$ that does not affect the control objectives here; special structures have been derived from this model (see **S.1** and **S.2** in the sequel); the upper space is defined by $U = \{(x, y, \delta, \epsilon, \dot{x}, \dot{y}, \dot{\delta}, \dot{\epsilon}) \in \mathbb{R}^8 \mid -\pi/2 < \delta < \pi/2 \text{ and } -\pi/2 < \epsilon < \pi/2\}$;*

M.3 *The model in $q = (x, y, X, Y)$ in (Liu et al., 2008b) - X and Y are the projection of the center of mass in the horizontal plane; the model can only represent the case that the pendulum is either above the horizontal plane or below the plane but it is sufficient to the control objectives in this paper; the description of the model is technically simpler than the above two but we cannot ensure that it also implies particular structures as those derived from **M.2**; the upper space is defined by $U = \{(x, y, X, Y, \dot{x}, \dot{y}, \dot{X}, \dot{Y}) \in \mathbb{R}^8 \mid \sqrt{X^2 + Y^2} < L\}$ (L is the length of the center of mass to the pivot).*

Generally, Equation 3 can be written in a state space form

$$\dot{\eta} = f(\eta, F, v_f) \quad (4)$$

where $\eta \triangleq (q, \dot{q}) \in U$ denotes the state vector and Equation 4 is called the nominal dynamics as $v_f \equiv 0$.

2.2 Problem formulation

In the literature, a local stabilizing controller is used to switch from a swing-up strategy (Albouy & Praly, 2000) to achieve a large domain attraction. Here, three different control objectives are envisaged which are more challenging:

PF.1 *The non-local stabilization* - Find a planar force F to drive the spherical inverted pendulum in such a way that for a non-trivial set $S \subset U$ and $S \ni 0$, where the trivial solution denotes the upright position of the pendulum and a given point on the horizontal plane in (x, y) for the universal joint of the pendulum, S is contained in a domain of attraction. If $S \subseteq U$ and $U \subseteq S$, the closed loop system is said to yield a "global" stability region. If $\forall S \subseteq U$, there exist certain design parameters such that S is

contained in a domain of attraction. Then, the closed loop system is said to yield a “semi-global” stability region.

- PF.2 Exact output tracking** – Let $(x_d(t), y_d(t))$ for $t \in (-\infty, \infty)$ be a sufficiently smooth desired curvature in the globally fixed frame with respect to the time variable t . Derive a feedback control law for F such that the pivot position, denoted by triplet $(t, x(t), y(t))$, of the pendulum starting from a set of initial conditions $(t_0, x(t_0), y(t_0))$ converges to $(t, x_d(t), y_d(t))$ asymptotically, i.e., $x(t) - x_d(t) \rightarrow 0$, $y(t) - y_d(t) \rightarrow 0$ as $t \rightarrow \infty$. Meanwhile, the pendulum is kept in U .
- PF.3 Way-point tracking** – Let $p = \{p_1, p_2, \dots, p_n\}$ with $p_i = (x_{r_i}, y_{r_i})$ for $i = 1, 2, \dots, n$ be a given sequence of points on the plane $x - y$ of the globally fixed frame. Associated with each p_i , consider the closed ball $N_{\mu_i}(p_i)$ with center p_i and radius $\mu_i > 0$. Derive a feedback control law for F such that the pivot (x, y) of the pendulum converges to p_n after visiting the ordered sequence of neighborhood $N_{\mu_i}(p_i)$ for $i = 1, 2, \dots, (n - 1)$ while keeping the pendulum in the upper space U .

2.3 Derivatives of the benchmark

The system is an abstraction of many real life applications/problems (see Fig. 2)

- A.1 A juggler's balancing problem** – One of very childish games is to balance a pole using a finger. The pole may fall in any direction and its base moves together with the finger. When the finger moves to the left, to the right, forward or backward in a horizontal plane, a planar force $F = (F_x, F_y)$ is applied the pole to steer it around. The human's hand is replaced by a manipulator in an automated environment.
- A.2 The hovering of a vector thrust rocket** – This system may hover at certain altitude either staying at a point or tracking certain trajectory. The rocket may head to any direction in a horizontal plane under the influence of injection—the main thrust. In this case, the main thrust can be decoupled to a vertical thrust against the gravity force or the drag and a planar thrust $F = (F_x, F_y)$ steering the rocket in the plane.
- A.3 A personal transporter** – It is a two-wheel vehicle on which a rider stands without falling over in any direction. The rider who hold the bar bending to the left, the right, forward and backward induces the cart to move intelligently to balance the rider. Some different accelerations may yielded by two wheels that together with an acceleration yielded by the centrifugal and Coriolis effects form a planar force $F = (F_x, F_y)$ to balance the rider. There is a commercial product from Segway.
- A.4 The test bench** – A pole with a universal joint stands on a cart sliding on a beam that in turn slides in a fixed frame. The cart and the beam that are driven by two motors respectively yields a planar force $F = (F_x, F_y)$ to the pole. This is a case where the classical inverted pendulum on the cart operates in three dimensional space;
- A.5 Others** – There are other controlled systems similar to the benchmark, for example, the launching of a spacecraft (without the thrust at the beginning).
- As is given in **A.1-A.5**, a planar force $F = (F_x, F_y)$ could be derived from several different types of original actuation for different controlled systems. Without loss of generality, we take the planar force F as the “generalized” force acting on the models from **M.1-M.3**. This gives us the same benchmark when exploring various control ideas. So, one can focus on the basic dynamic behaviors and the principles.

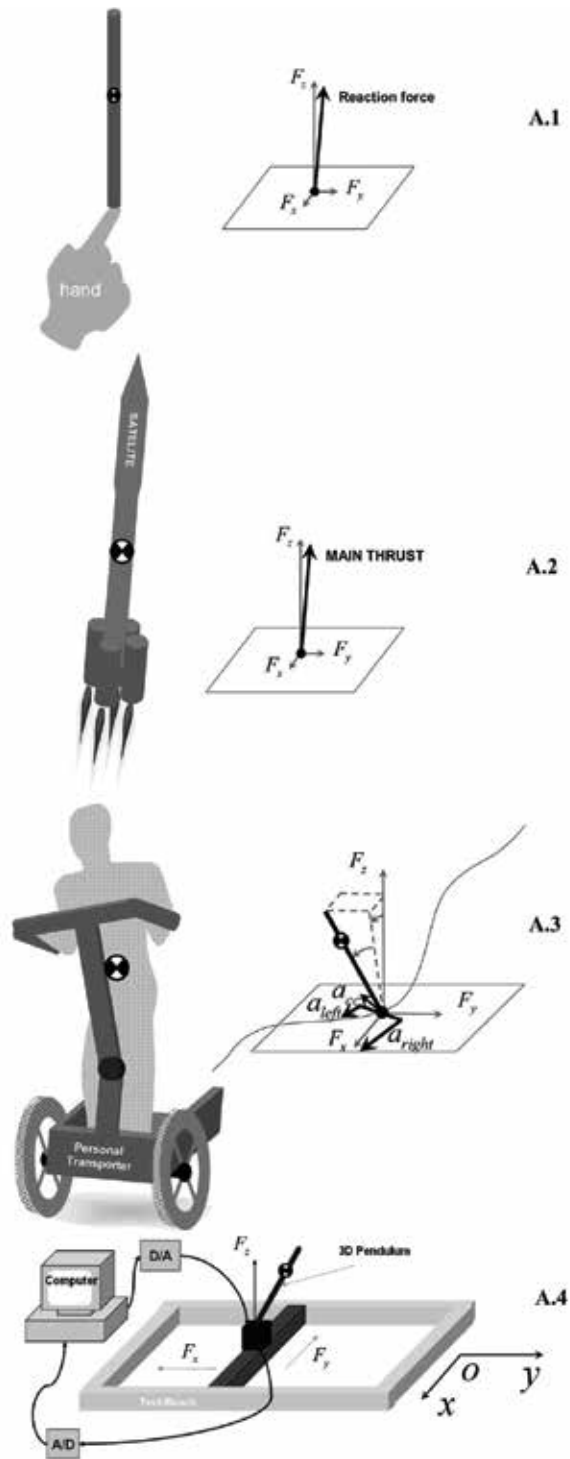


Fig. 2. Applications A.1-A.4

3. Nonlinear analysis and design tools

In the realm of various nonlinear analysis and design tools, the following concepts and tools are among the mainstream (not a complete survey), which are either used, incorporated, or related to several successful designs for the benchmark

- T.1** *The differential geometric approach* (see (Isidori, 1995)) – It is fundamental to nonlinear control systems. One of the key ideas is to transform a system to a linear one by means of feedback and coordinate transformation. The notion of “zero” dynamics plays an important role in the problem of achieving local asymptotic stability, asymptotic tracking, model matching and disturbance decoupling.
- T.2** *Input-to-state stability (ISS)* (see (Sontag, 1990; 2005)) – The concept establishes a result on feedback redesign to obtain a desirable stability condition with respect to actuator errors, and provides a necessary and sufficiency test in terms of ISS-Lyapunov function. It brings about a number of powerful analysis tools, one of which is asymptotic “ISS” gain and its small gain theorem (Teel, 1996). The latter leads to a “celebrated” design tool-forwarding.
- T.3** *Forwarding and backstepping* – Forwarding is a recursive control design procedure for nonlinear systems possessing an upper triangular structure. Nest saturating design (a low gain approach) (Teel, 1996) is the first tool in forwarding where design parameters are carefully selected to make the feedback interconnection of two systems satisfying small gain conditions. Lyapunov approaches (see (Mazenc & Praly, 1996; Sepulchre et al., 1997)) for forwarding are practically very difficult to apply because constructing an “exact” cross term in the Lyapunov function is hard. Backstepping (a high gain approach) (see (Kristić, 1995; Sepulchre et al., 1997)) is a different recursive design procedure for nonlinear systems possessing a lower triangular structure. It is a very successful tool. However, one must realize that many nature systems do not possess such a structure. A misconception is that the interlaced designs (Sepulchre et al., 1997) apply also to special structures (half upper and half lower structures). *Sliding mode control* (see (Utkin, 1992)) can be taken as a recursive design procedure similar to backstepping.
- T.4** *Singular perturbations* (see (Kokotović, 1986)) – It is a means of taking into account neglected high-frequency phenomena and considering them in a separate fast time-scale. This is achieved by treating a change in the dynamic order of a system of differential equations as a parameter perturbation, called the “singular perturbations”. It results in a structure of a dynamical system with two time scales (fast and slow) so that the control problem is simplified.
- T.5** *Controlled Lagrangians/Hamiltonians (IDA-PBC)* (see (Block et al., 2001; Ortega et al., 2002)) – The methods are constructive passivity based control tools for a physical system that can be described in Lagrangian dynamics or Hamiltonian dynamics. The key notion is the energy shaping (kinetic, potential or total energy) such that the closed loop system preserves the structure of Lagrangian or Hamiltonian dynamics with a desired behavior. For example, the unstable equilibrium of the original dynamics may become a stable equilibrium of the modified dynamics. For mechanical systems, two variations are equivalent.
- T.6** *Stable inversion/output regulation* (see (Devasia, 1996; Isidori, 1995)) – The Byrnes-Isidori (see (Isidori, 1995)) regulator generalizes internal model principle to nonlinear systems that can be applied to track any trajectory generated by a given exosystem if one can

solve the associated PDEs. The stable inversion technique (see (Devasia, 1996)) trades the requirement of solving these general PDES for a specific trajectory. Both tools can deal with the unstable “zero” dynamics that cannot be dealt with by the conventional inversion technique.

T.7 Hybrid control¹—There is no ultimate definition. It refers to a control system that mixes discrete parts (e.g., a controller, a supervisor) and continuous parts (e.g., a continuous plant).

4. Constructive control designs

4.1 Step 1 identifying “normal” forms

Unlike linear systems that can be written more or less in a unified manner, nonlinear systems are so diversified that one can only cope with a subclass of nonlinear systems even one particular example at a time. Therefore, nonlinear control designs are usually much more complex and difficult than linear ones. The situation well fits in with a famous sentence in Leo Tolstoy’s *Anna Karenina*

“All happy families (linear systems here) are happy alike, all unhappy families (nonlinear systems here) are unhappy in their own way.”

Nevertheless, the linear control theory is not a panacea to all control problems as it holds only around an operating point if and only if the first approximation principle holds at this point. In contrast, nonlinear control systems may yield a large (even “global”) region of stability, tracks asymptotically a nonlinear trajectory that exceeds the bandwidth of a linear control system, and provides more physical insights.

A significant effort in nonlinear control designs is to identify a structure that is suitable for a particular design procedure. *Ad hoc* approaches for identifying a structure of a nonlinear control system maybe

- neglecting some nonlinear effects or considering them as perturbations;
- exploring physical properties to provide insight to the dynamics;
- taking a preliminary feedback and/or a change of states to simplify the dynamics.

Neglecting some nonlinear effects in a nonlinear design should be taken carefully because the claimed properties (e.g., a “global” domain of attraction and robustness) for the reduced dynamics may not represent a real situation. In our designs, we only neglect the disturbance and the unmodelled dynamics in analysis and design. So, we guarantee that the closed loop systems represents the original *full* nonlinear control system.

The structures that are explored for our designs are listed (to compare with the different structures, we abuse notations a little bit for new states)

S.1 The original dynamics maps to an “appropriate” upper triangular structure (Liu et al., 2008a)

$$\begin{aligned}\dot{\zeta}_i &= A_i \zeta_i + g_i(\zeta_i, u) \\ \dot{\xi}_i &= f_i(\zeta_i, u),\end{aligned}\quad \text{for } i = 1, 2, 3, 4 \quad (5)$$

by a nonsingular transformation $T_1 U \rightarrow R^8$ (there is no constraint in new states) and a preliminary feedback $F = \alpha_1(\eta, u)$, where u is the new input, $\xi_{i+1} \triangleq (\xi_i, \zeta_i)$, (ξ_i, ζ_i) are the

¹ It does not mean a particular tool or method but a broad class of mixed tools and methods.

states corresponding to each augmented subsystem and $A_i = 0$. The feedback linearization technique (Isidori, 1995) in **T.1** is incorporated.

S.2 The original dynamics also maps to two interconnected subsystems (Liu et al., 2008c)

$$\begin{aligned}\dot{\omega} &= A\omega + B\xi_1 + \varphi_\eta(\xi_1, \xi_2, \zeta_1, \zeta_2) \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= u_1\end{aligned}\quad (6)$$

$$\begin{aligned}\dot{\vartheta} &= A\vartheta + B\zeta_1 + \varphi_\vartheta(\xi_1, \xi_2, \zeta_1, \zeta_2) \\ \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= u_2\end{aligned}\quad (7)$$

by a nonsingular transformation $T_2U \rightarrow R^8$ (there is no constraint in new states) and a preliminary feedback $F = \alpha_2(\eta, u)$, where $u = (u_1, u_2)$ is the new input, (ξ_1, ξ_2, ω) (with $\omega = (\omega_1, \omega_2)$) and $(\zeta_1, \zeta_2, \vartheta)$ (with $\vartheta = (\vartheta_1, \vartheta_2)$) are the states for two subsystems respectively, $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and $\varphi_\eta(\cdot)$ and $\varphi_\vartheta(\cdot)$ are interconnected terms which are high order nonlinear terms with respect to their arguments.

S.3 This structure is trivial as we can write the original unperturbed dynamics in an “appropriate” form of the Euler-Lagrangian equations (Block et al., 2001)

$$\begin{aligned}\dot{\omega} &= A_u\omega + \psi_\eta(\omega, \vartheta, u_1, u_2) \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= u_1\end{aligned}\quad (10)$$

$$\begin{aligned}\dot{\vartheta} &= A_s\vartheta + \psi_\vartheta(\omega, \vartheta, u_1, u_2) \\ \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= u_2\end{aligned}\quad (11)$$

by a nonsingular transformation $T_3U \rightarrow \chi \in R^8$ (χ is a locally bounded set about $(\omega, \vartheta) \ni 0$) and a preliminary feedback $F = \alpha_3(\eta, u)$, where $u = (u_1, u_2)$ is the new input, $(\xi_1,$

$\xi_2, \omega, \zeta_1, \zeta_2, \vartheta)$ with $\omega = (\omega_1, \omega_2)$ and $\vartheta = (\vartheta_1, \vartheta_2)$ are the new states, $A_u = \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix}$ and

$A_s = \begin{pmatrix} -c & 0 \\ 0 & -c \end{pmatrix}$ for a scalar $c > 0$. Here, a combination of a linear transformation and the

feedback linearization technique is used.

4.2 Step 2 applying nonlinear tools

The structures **S.1-S.4** enable us to complete a number of nonlinear control designs relatively easier for three control objectives **PF.1-PF.3**. Fig. 3 shows the close loop systems with the controllers **NC.1-NC.5** as follows.

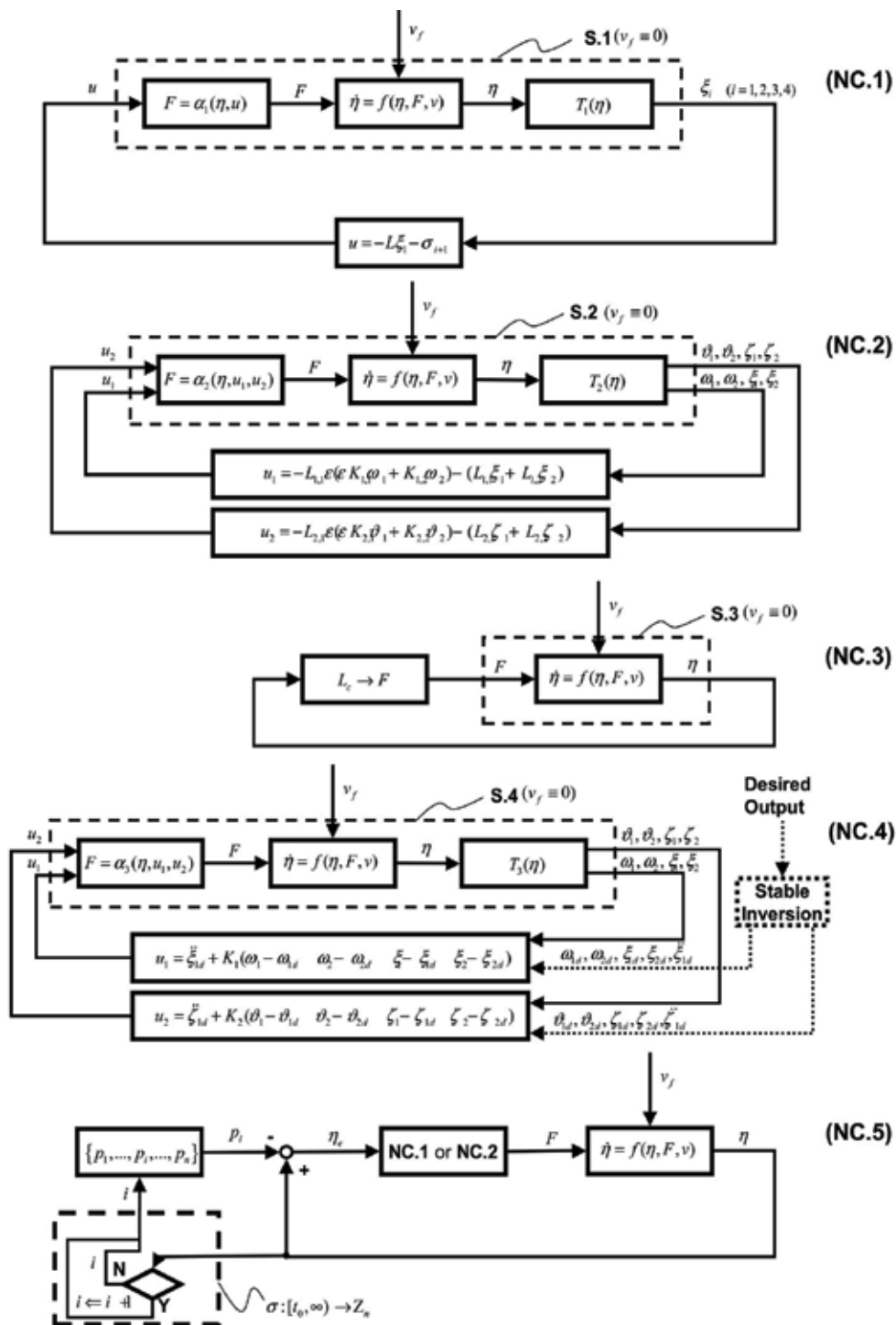


Fig. 3. Diagrams of NC.1-NC.5

NC.1 The high-low gain controller (see (Liu et al., 2008a) for **PF.1** is designed on the basis of **S.1**

$$u = -L\xi_1 - \sigma_{i+1} \text{ for } i = 1, 2, 3, 4 \quad (12)$$

where $L \in R^{4 \times 4}$ is a linear high gain matrix, $\sigma_{i+1} \triangleq \lambda_i \cdot \text{sat}(\frac{1}{\lambda_i}(K_{i+1}\xi_{i+1} + \Gamma_{i+1}v_{i+1}))$ with $v_{i+1} = \sigma_{i+2}$ (v_5 does not necessary to be given as the design is complete, K_{i+1} and λ_i are associated gain matrices and saturation levels). Nested saturating method (Teel, 1996) in **T.3** is used to design a low gain control part σ_{i+1} at the aid of a linear control design method-LQR. The controller yields a closed loop system with a “global” stability region. The design implies the existence of appropriate λ_i that is related to the domain of attraction yielded by a linear controller. Practically, λ_i is found by trails and errors. ISS (see (Sontag, 2005)) in **T.2** is a key analysis tool in both the design and the redesign.

NC.2 The decentralized controller in (Liu et al., 2008c) for **PF.1** is designed on the basis of **S.2**

$$\begin{aligned} u_1 &= -\varepsilon L_{1,1}(\varepsilon K_{1,1}\omega_1 + K_{1,2}\omega_2) \\ &\quad + (L_{1,1}\xi_1 + L_{1,2}\xi_2) \\ u_2 &= -\varepsilon L_{2,1}(\varepsilon K_{2,1}\vartheta_1 + K_{2,2}\vartheta_2) \\ &\quad + (L_{2,1}\zeta_1 + L_{2,2}\zeta_2) \end{aligned} \quad (13)$$

where $L_{\cdot, \cdot}$ and $K_{\cdot, \cdot}$ are positive scalars, $\varepsilon \in (0, 1)$ is time scaling parameters. The resultant closed loop system is a hidden singularly perturbed system that can be transformed into a standard singular perturbation form (slow) $\dot{\bar{x}} = f(\bar{x}, \bar{y})$, (fast) $\varepsilon \dot{y} = h(\bar{x}, \bar{y}, \varepsilon)$. A “strong” Lyapunov function comes with the design and the total stability of the system is ensured. A “semi-global” stability region (it increases as ε decreases) is yielded by the closed loop system. The design is heavily relying on **T.4** (see (Kokotović, 1986)).

NC.3 The controller via controlled Lagrangians in (Block et al., 2001) and (Liu et al., 2007) (a complete version) for **PF.1** is based on **S.3**

$$F \Leftarrow \mathcal{L}_c \quad (14)$$

which defines a passivity based controller F , where \mathcal{L}_c is defined as a controlled Lagrangian that satisfies the conditions in (Block et al., 2001). Although the controller is a direct result of the theory (Block et al., 2001) in **T.5**, the derivation is technically complex. A “weak” Lyapunov function comes with the design, that is, an energy function of the closed loop system. LaShall’s invariance principle is used to established the stability but the principle cannot guarantee the stability under disturbances.

NC.4 The exact output tracking controller in (Liu et al., 2008b) for **PF.2** is a designed on the basis of **S.3**

$$\begin{aligned} u_1 &= \ddot{\xi}_{1d} + K_1 (\omega_1 - \omega_{1d}\omega_2 - \omega_{2d} \\ &\quad \xi_1 - \xi_{1d}\xi_2 - \xi_{2d}) \\ u_2 &= \ddot{\zeta}_{1d} + K_2 (\vartheta_1 - \vartheta_{1d}\vartheta_2 - \vartheta_{2d} \\ &\quad \zeta_1 - \zeta_{1d}\zeta_2 - \zeta_{2d}) \end{aligned} \quad (15)$$

where $\{\xi_{1d}, \zeta_{1d} | \xi_{2d}, \zeta_{2d}, \ddot{\xi}_{1d}, \ddot{\zeta}_{1d}, \omega_{1d}, \omega_{2d}, \mathcal{G}_{1d}, \mathcal{G}_{2d}\}$ are obtained based on the stable inversion tool (Devasia, 1996) in **T.6** with respect to a desired output trajectory. K are linear feedback gain matrices obtained by a linear controller design-LQR. $(\ddot{\xi}_{1d}, \ddot{\zeta}_{1d})$ is a guidance controller (a feedforward part) and the rest is a feedback minimizing the tracking errors and rejecting exogenous disturbances. For an achievable desired trajectory that is $c_2(-\infty, \infty)$, the output (the translational variables ξ_1 and ζ_1 - the original x and y) of the closed loop system tracks exactly the desired trajectory while keeping the pendulum upward.

NC.5 The hybrid controller in (Liu & Yang, 2010) for **PF.3** is in the category of **T.7**. The result is relying on **NC.1** or **NC.2** and an event driven piecewise constant signal $\sigma[t_0, \infty) \rightarrow \mathbf{Z}_n$ that is continuous from the right at every point and is defined recursively by

$$\sigma = \bar{\alpha}(\chi, \psi, \sigma^-), \quad t \geq t_0 \quad (16)$$

where χ and ψ are metrics on the current tracking errors with respect to the neighborhood N_{μ_i} of i th way-point, $\sigma(\tau)$ is equal to the limit from the left of $\sigma(\tau)$ as $\tau \rightarrow t$ based on an event that determines the discrete value i in a set $\{1, 2, \dots, n\}$. The controller yields either “global” or “semi-global” stability region to the closed loop system inherit from **NC.1** or **NC.2**. The ordered sequence of way-points are guaranteed but the timing to a way-point is uncertain.

5. Conclusion and future work

The cutting-edge theoretical nonlinear analysis and designs tools have been used successfully to solve the challenging control goals for a four degrees of freedom spherical inverted pendulum, such as the global stabilization and the nonlinear exact tracking. However, the tools are unable to yield satisfactory controllers on their own. A designer should perform preliminary designs via identify the special structures, “normal” forms, to bridge the gap. Observed from these successful designs, a good insight to the physical dynamical system would help us to find a way, bridging the gap. The experiences obtained from the benchmark example should be extended to other nonlinear control systems. Techniques of identifying various “normal forms” should be emphasized.

6. References

- Albouy, X. & Praly, L. (2000). On the use of dynamic invariants and forwarding for swinging up a spherical inverted pendulum, in *Proceedings of 39th Conference on Decision & Control*, Sydney, Australia, pp. 1667-1672.
- Bloch, A.; Chang, D.; Leonard, N. & Marsden, J. (2001). Controlled Lagrangians and the stabilization of mechanical systems II potential shaping, *IEEE Transactions on Automatic Control*, Vol. 41, pp. 1556-1571.
- Devasia, D.; Chen, D. & Paden, B. (1996). Nonlinear inversion-based output tracking, *IEEE Transactions on Automatic Control*, Vol. 41, pp. 930-942.
- Isidori, A. (1995). *Nonlinear Control System (3rd edition)*, Springer.
- Kokotović, P. Khalil, H. & O'Reilly, J. (1986). *Singular Perturbation Methods in Control Analysis and Design*, Academic Press Inc..

- Krstić M.; Kanellakopoulos, L. & Kokotović, P. (1995). *Nonlinear and Adaptive Control Design*, John Wiley & Sons.
- Liu, G. (2006). *Modeling, Stabilizing Control and Trajectory Tracking of a Spherical Inverted Pendulum*. Ph.D Thesis, The University of Melbourne.
- Liu, G.; Challa, I. & Yu, L.(2007). Revisit controlled Lagrangians for spherical inverted pendulum , *International Journal of Mathematics and Computers in Simulation*, Vol. 1, No. 1, pp. 209-214.
- Liu, G.; Mareels, I. & Nešić, D. (2008). Decentralized control design of interconnected chains of integrators a case study, *Automatica*, Vol. 44, No. 8, pp. 2171-2178.
- Liu, G.; D. Nešić & I. Mareels (2008). Nonlinear stable-inversion based output tracking for the spherical inverted pendulum, *International Journal of Control*, Vol. 81, No.7, pp. 1035-1053.
- Liu, G.; D. Nešić & I. Mareels (2008). Nonlinear stable-inversion based output tracking for the spherical inverted pendulum, *International Journal of Control*, Vol. 81, No.1, pp. 116-133.
- Liu, G. & Yang, R. (2010). Minimizing operating points of way point tracking of an unstable nonlinear plant, *Asian Journal of Control*, Vol. 12, No. 1, pp. 84-88.
- Mazenc, F. & Praly, L. (1996). Adding integrations, saturated controls, and stabilization for feedforward systems, *IEEE Transactions on Automatic Control*, Vol.41, pp. 1559-1577.
- Ortega, R.; Spong, W.; Gomez-Estern, F. & Blankenstein, G. (2002). Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE Transactions on Automatic Control*, Vol. 47, pp. 1218-1233.
- Sepulchre, R.; Janković, M. & Kokotović, P. (1997). *Constructive Nonlinear Control*, Springer, pp. 979-984.
- Sepulchre, R.; Janković M. & Kokotović, P. (1997). Integrator forwarding a new recursive nonlinear robust design. *Automatica*, Vol. 33, pp. 979-984.
- Sontag, E. (1990). Further facts about input to state stabilization. *IEEE Transactions on Automatic Control*, Vol. 35, pp. 473-476.
- Sontag, E. (2005). *Input to state stability Basic concepts and results*, Springer Lecture Notes in Mathematics, Springer.
- Teel, A. (1996). A nonlinear small gain theorem for the analysis of control systems with saturation. *IEEE Transactions on Automatic Control*, Vol. 41, pp. 1256-1270.
- Utkin, V. (1992). *Sliding modes in control optimization*, Springer-Verlag.

A Unified Approach to Robust Control of Flexible Mechanical Systems Using \mathcal{H}_∞ Control Powered by PD Control

Masayoshi Toda

*Tokyo University of Marine Science and Technology
Japan*

1. Introduction

This chapter presents a unified approach to robust control of a variety of flexible mechanical systems, which are not only systems having flexible structure themselves such as a robotic manipulator with a flexible structure and a crane system, but also systems not having flexible structure but handling flexible objects such as a liquid container system and a fishery robot. So far, a lot of research efforts have been devoted to solve control problems of such flexible systems, one of the most typical problems among which is the problem of flexible robotic manipulators, e.g., [Sharon & Hardt (1984); Spong (1987); Wang & Vidyasagar (1990); Torres et al., (1994); Magee & Book (1995); Nenchev et al., (1996); Nenchev et al., (1997)]. As other types of applications, the problems of a crane system [Kang et al. (1999)] and of a liquid container system [Yano & Terashima (2001); Yano et al., (2001)] have been investigated. The common control problem for flexible systems can be stated as “*how to achieve required motion control with suppressing undesirable oscillation due to its flexibility*”.

From the control methodology point of view, let us review those previous works. For so-called micro-macro manipulators associated with large flexible space robots, [Torres et al (1994)] and [Nenchev et al., (1996); Nenchev et al., (1997)] have proposed path-planning based control methods using a coupling map and a reaction null-space respectively, which utilize the geometric redundancy. The control methods in [Sharon & Hardt (1984)] for a micro-macro manipulator and in [Kang et al., (1999)] for a crane system rely on the endpoint direct feedback, which require sensors to measure the endpoint. In [Wang & Vidyasagar (1990)], a passivity-based control method has been proposed for a single flexible link, and in [Spong (1987)] an exact-linearization method and an integral manifold method have been presented for a flexible-joint manipulator. The method in [Magee & Book (1995)] is based on input signal filtering where the underlying concept is pole-zero cancellation. [Ueda & Yoshikawa (2004)] has applied a mode-shape compensator based on acceleration feedback to a flexible-base manipulator. For a liquid container system, \mathcal{H}_∞ control in [Yano & Terashima (2001)] and a notch-type filter based control, that is, equivalent to pole-zero cancellation, in [Yano et al., (2001)] are utilized respectively. In general, most other works have focused on individual systems and hence their control methods are not directly available for various flexible systems. For example, the path-planning methods in [Torres et

al., (1994); Nenchev et al., (1996); Nenchev et al., (1997)] cannot be applied to non-redundant systems. The direct endpoint feedback might be difficult in such a case as of a large space robot where it is difficult to employ sensors to directly measure the endpoint.

In a stark contrast with those works, we have been tackling with a unified control design method which can be applied to various flexible mechanical systems in a uniform and systematic manner. The proposed method exploits a problem setting framework which is referred to as “generic problem setting” in the modeling phase and then, in the control design phase, \mathcal{H}_∞ control powered by PD control. In the sense of control methodology, the underlying concept is pole-zero cancellation similarly with [Magee & Book (1995); Yano et al., (2001)], however the control design approach is totally different from ones in those works. On the other hand, although [Yano & Terashima (2001)] has employed \mathcal{H}_∞ control, its usage is different from ours as explained later, and further the pole-zero cancellation is not the case in [Yano & Terashima (2001)]. In our control design method, the point to be emphasized is that PD control plays very important roles in facilitating the generic problem setting and the \mathcal{H}_∞ control design, and most importantly in enhancing the robustness of the control system. Then, the advantageous features of our control design method are:

1. The method can be applied to various flexible systems in a uniform, systematic, and simple manner where the frequency-domain perspective will be provided;
2. The robustness can easily be enhanced by appropriately choosing the PD control gains;
3. Due to the nature based on pole-zero cancellation, any oscillation sensors will not be required, which is considerably important in the practical sense.

In [Toda (2004)], we have first introduced the fundamental idea and demonstrated control simulations using linear system and weakly nonlinear system examples. Then, in [Toda (2007)], robust control has been explicitly considered and a rather strongly nonlinear system example has been tackled. Now, in this article the control design method and the previous achievements are summarized, moreover a multiple-input-multiple-output (MIMO) system and the optimality with respect to PD control are examined while those points have not been considered in [Toda (2004); Toda (2007)].

The remainder of this chapter is organized as follows. Section 2 presents the generic problem setting and an illustrative MIMO system example. Section 3 introduces the control design method and discusses its features in some detail. Then, Section 4 demonstrates control simulations using the MIMO system example. Finally, Section 5 gives some concluding remarks.

2. Generic problem setting and an illustrative example

2.1 Generic problem setting

For the purpose of accommodating a variety of flexible systems, in the modeling phase, a generic model which can represent such systems in a uniform manner is required. Hence, we consider a cascade chain of linear mass-spring-damper systems as shown in Fig. 1. m_i , k_i , d_i , f_i , and q_i denote the mass, stiffness parameter, damping parameter, exerted force, and displacement from the equilibrium of the i th component respectively. The first component is connected to the stationary base. The number of components depends on systems to be modeled. For example, a single-link flexible-joint manipulator can be modeled as a two-component model, where m_1 denotes the inertia of the actuator, f_1 the actuator torque, m_2 the inertia of the link, and f_2 must be zero, that is, the first component is directly actuated while the second one is not so, thus, is merely an oscillatory component. Applying PD control to the actuator, the corresponding dynamical model can be described as follows,

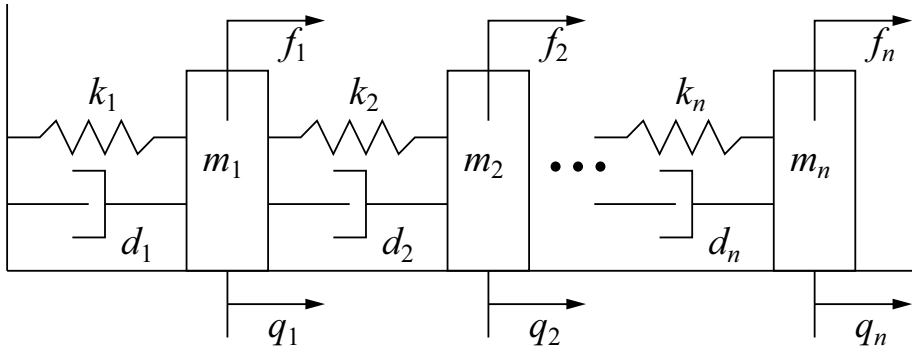


Fig. 1. Schematic diagram of the generic problem setting.

$$\begin{aligned} m_1\ddot{q}_1 + m_2(\ddot{q}_1 + \ddot{q}_2) + d_1\dot{q}_1 + k_1q_1 &= f_1 \\ m_2(\ddot{q}_1 + \ddot{q}_2) + d_2\dot{q}_2 + k_2q_2 &= 0. \end{aligned} \quad (1)$$

On the other hand, let us consider a single-link flexible-base linear manipulator. In this case, conversely, the first component is merely an oscillatory component while the second one is to be directly controlled via the actuator. The dynamical model including PD control to the actuator can be described as follows,

$$\begin{aligned} m_1\ddot{q}_1 + m_2(\ddot{q}_1 + \ddot{q}_2) + d_1\dot{q}_1 + k_1q_1 &= 0 \\ m_2(\ddot{q}_1 + \ddot{q}_2) + d_2\dot{q}_2 + k_2q_2 &= f_2. \end{aligned} \quad (2)$$

As seen from the above discussion, by assigning a component to be directly controlled via the corresponding actuator or an oscillatory component to each mass, this chain model can represent various flexible systems. This problem setting framework based on the chain model is referred to as “generic problem setting”. Then, the control problem is how to control positions of the directly controlled components with suppressing oscillations of the oscillatory components. It should be noted that with the proposed control method any sensors for the oscillatory components will not be required except such cases where, in the steady state, deformation due to the flexibility and the gravity would become a problem. In cases of nonlinear and/or uncertain systems, through some linearization procedures such as nonlinear state feedback and linear approximation around the equilibrium, the system is modeled as a linear model with parametric uncertainties and/or disturbances. Furthermore, by applying PD control to the nonlinear system, one can make the linear dynamics dominant, therefore can facilitate the generic problem setting.

2.2 Illustrative example

In [Toda (2004)], as illustrative examples, we have chosen the flexible-joint manipulator and the flexible-base linear one represented by (1) and (2) respectively, and a gantry-crane system which can be represented by the same model as the flexible-joint manipulator one by using linear approximation. Then, in [Toda (2007)], as a strongly nonlinear system example, a single-link revolutionary-joint flexible-base manipulator has been considered. Since all the examples in these previous works are of single-input-single-output (SISO) systems, in this article we choose a two-link flexible-joint manipulator as an MIMO system example as depicted in Fig. 2.

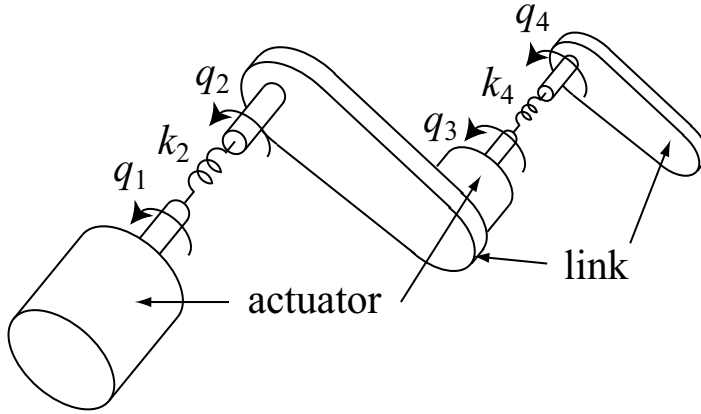


Fig. 2. Two-link flexible-joint manipulator.

$q = [q_1, q_2, q_3, q_4]^t$ denotes the position vector of the manipulator, k_2, k_4 , and d_2, d_4 denote the joint stiffness and damping parameters respectively. $[\cdot]^t$ denotes the transpose. Additionally, by introducing PD control to the actuators with the P gains k_1, k_3 and the D gains d_1, d_3 , the dynamical model is as in the following.

$$M(q)\ddot{q} + C(q, \dot{q}) + D\dot{q} + Kq = f \quad (3)$$

where $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the centripetal and Coriolis term, $D = \text{diag}[d_1, d_2, d_3, d_4]$ is the damping diagonal matrix, $K = \text{diag}[k_1, k_2, k_3, k_4]$ is the stiffness diagonal matrix, and $f = [f_1, 0, f_3, 0]^t$ is the control torque vector excluding the PD control scheme. Specifically, each element of $M(q)$, M_{ij} is as follows:

$$\begin{aligned} M_{11} &= m_1 + m_2 + m_3 + m_4 + 2R \cos(q_3 + q_4) \\ M_{12} &= M_{21} = M_{22} = m_2 + m_3 + m_4 + 2R \cos(q_3 + q_4) \\ M_{13} &= M_{23} = M_{31} = M_{32} = m_3 + m_4 + R \cos(q_3 + q_4) \\ M_{14} &= M_{24} = M_{41} = M_{42} = m_4 + R \cos(q_3 + q_4) \\ M_{33} &= m_3 + m_4 \\ M_{34} &= M_{43} = M_{44} = m_4 \end{aligned} \quad (4)$$

where m_i and R are the inertia parameters. And $C(q, \dot{q})$ is formulated as

$$C(q, \dot{q}) = \begin{bmatrix} -\{2(\dot{q}_1 + \dot{q}_2) + (\dot{q}_3 + \dot{q}_4)\}(\dot{q}_3 + \dot{q}_4)R \sin(q_3 + q_4) \\ -\{2(\dot{q}_1 + \dot{q}_2) + (\dot{q}_3 + \dot{q}_4)\}(\dot{q}_3 + \dot{q}_4)R \sin(q_3 + q_4) \\ (\dot{q}_1 + \dot{q}_2)^2 R \sin(q_3 + q_4) \\ (\dot{q}_1 + \dot{q}_2)^2 R \sin(q_3 + q_4) \end{bmatrix}. \quad (5)$$

As seen from Equations (3)–(5), it is confirmed that except the nonlinear terms the dynamical model can completely be represented in the generic problem setting with four components. Moreover, assuming that the dynamics due to the PD control scheme is more dominant than $C(q, \dot{q})$ and that $M(q)$ with $q_3 = \pi/3$ and $q_4 = 0$ is a nominal constant matrix, the proposed control design method will be applied to this problem. The physical parameters in the

dynamical model employed for the control design and simulations in the sequel are shown in Table 1, which are set by considering the experimental apparatus at hand.

parameter	value	unit
m_1	1.000e-5	kgm ²
m_2	2.027e-3	kgm ²
m_3	1.000e-6	kgm ²
m_4	1.520e-4	kgm ²
R	9.410e-5	kgm ²
d_2	0.000e 0	Nms
d_4	0.000e 0	Nms
k_2	2.180e-1	Nm
k_4	1.520e-2	Nm

Table 1. Physical parameters.

3. Control design

Here we introduce our control design method which is applied to the obtained model in the generic problem setting. In the design procedure, first one should determine the PD control gains, then proceed to the \mathcal{H}_∞ control design aiming to shape the associated sensitivity functions. However, in this section, for ease of exposition we first present the \mathcal{H}_∞ control design and after that discuss the PD control scheme in some detail.

3.1 Sensitivity function shaping by \mathcal{H}_∞ control

Once the PD control scheme has been determined, the control design procedure is almost automatically processed in the linear \mathcal{H}_∞ control framework with the aim of shaping the associated sensitivity functions. Fig. 3 depicts the augmented plant for \mathcal{H}_∞ control design where P denotes the plant incorporating the PD control scheme which consists of P_i corresponding to the components to be directly controlled and P_j to the oscillatory ones,

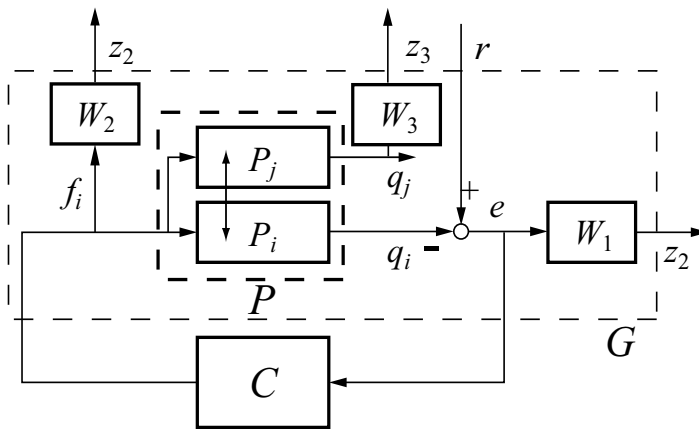


Fig. 3. Augmented plant for \mathcal{H}_∞ control design.

where P_i and P_j are coupled systems each other. The sensitivity functions taken into account are the transfer function S_1 from the reference commands r to the tracking control errors e , S_2 from r to the control inputs f_i , and S_3 from r to the oscillatory component displacements q_j . In the example given in Section 2.2, q_i are q_1, q_3 , and q_j are q_2, q_4 respectively.

Note that S_3 plays a key role in this problem and, in terms of \mathcal{H}_∞ control design, makes our method differ from the others such as [Yano & Terashima (2001)] which does not consider S_3 but only the standard mixed sensitivity problem. By explicitly employing S_3 , the resultant \mathcal{H}_∞ controller will automatically contain the corresponding zeros to the oscillatory poles of the plant and thus pole-zero cancellation will occur in the closed-loop system which leads to suppression of oscillation. Due to this nature of pole-zero cancellation, the control system will not require any sensors to measure the states of the oscillatory components q_j .

The respective weighting functions for the sensitivity functions in the example are

$$W_1(s) = \frac{20}{7} \begin{bmatrix} \frac{1}{s+0.0001} & 0 \\ 0 & \frac{1}{s+0.0001} \end{bmatrix} \quad (6)$$

$$W_2(s) = \frac{3}{7} \begin{bmatrix} \frac{s+0.1}{s+100} & 0 \\ 0 & \frac{s+0.1}{s+100} \end{bmatrix} \quad (7)$$

$$W_3(s) = \frac{20}{7} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (8)$$

W_1 is only a quasi-integrator intended for step tracking control. W_2 is a high-pass filter which will be determined by the actuator capability. W_3 for S_3 is only a constant gain. These functions are very simple, and in particular W_1 and W_3 might not depend on problems. Therefore, the designer will only need to care the constants $20/7, 3/7, 20/7$ to adjust the balance among the functions. This simplicity is one of the important advantages of the proposed method.

Then, by constructing the augmented plant G as in Fig. 3, an \mathcal{H}_∞ controller C will be synthesized such that the \mathcal{H}_∞ norm of the closed-loop system T_{rz} from r to $z = [z_1, z_2, z_3]^t$, that is, $\|T_{rz}\|_\infty$ is minimum. In this example, the resultant $\|T_{rz}\|_\infty$ was 1.

If one may wish to explicitly consider the model uncertainties in the control design, μ -synthesis [Packard & Doyle (1993); Zhou et al., (1995)] can be applied instead of merely \mathcal{H}_∞ control design. The interested readers may consult [Toda (2007)] for the specific approach in the same framework.

In addition, to improve the transient performance of the obtained control system, a low-pass filter is employed for step reference commands. In this example, the reference command filter is

$$P_r = \begin{bmatrix} \frac{100}{s^2 + 36s + 100} & 0 \\ 0 & \frac{100}{s^2 + 36s + 100} \end{bmatrix}. \quad (9)$$

3.2 PD control

3.2.1 Roles of PD control

Next, let us discuss the PD control scheme exploited for this problem. One role of the PD control scheme is, as mentioned in Section 2.1, of facilitating the generic problem setting by making the linear dynamics dominant. And as the second role, the scheme serves to facilitate the \mathcal{H}_∞ control design, that is, by eliminating the poles on the imaginary axis and turning the problem into so-called the standard \mathcal{H}_∞ control problem [Doyle et al. (1989); Zhou et al., (1995)]. However, a more important role is of enhancing the robustness with respect to the oscillation suppression capability, which is deeply connected with the pole-zero cancellation mechanism of the \mathcal{H}_∞ controller.

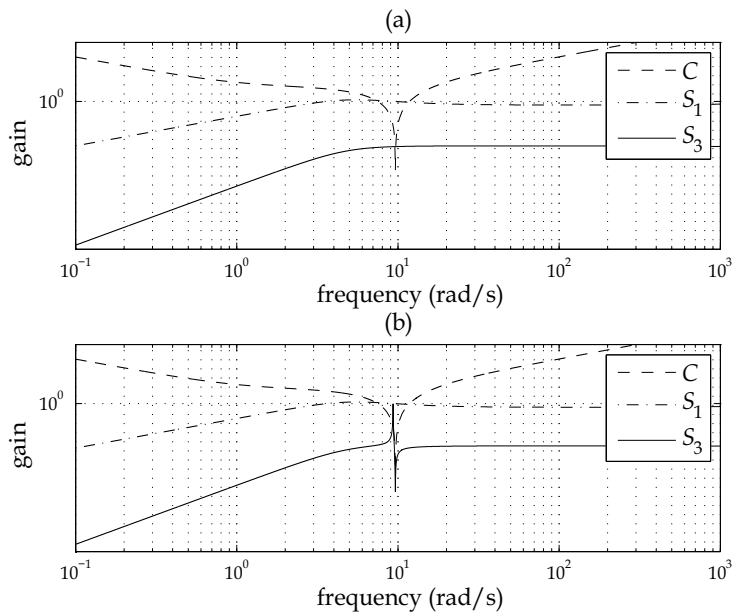
In the case of a completely linear system with neither model uncertainties nor perturbations, the pole-zero cancellation will never fail, and hence the constant oscillation suppression performance can be acquired. However, otherwise, that is, in cases of a nonlinear system and/or with model uncertainties, the pole-zero cancellation will fail since the oscillatory poles of the plant vary. In such a case, the damping property of the plant will become critical. Specifically, when the minimum among the damping factors of the plant poles is too small, the oscillation suppression performance can largely degrade in case of failure of the pole-zero cancellation. Here the damping factor ζ of a stable pole s , whose real part $\text{Re}(s) \leq 0$, is defined as

$$\zeta := -\frac{\text{Re}(s)}{|s|} \quad (10)$$

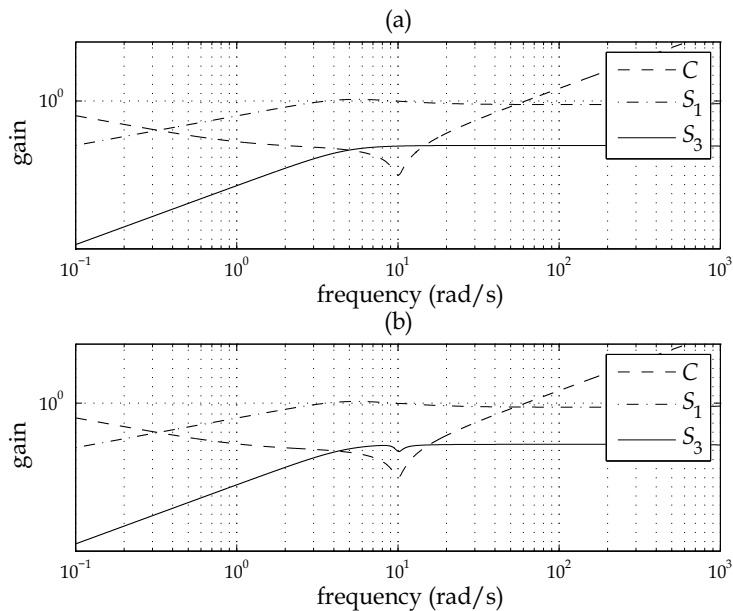
where ζ of a real s is the maximum of 1.

However, by choosing the PD control gains, this damping property can be appropriately modified. We illustrate this fact by using a nonlinear SISO system example, i.e., a single-link revolutionary-joint flexible-base manipulator, investigated in [Toda (2007)]. Fig. 4 shows the frequency responses of the \mathcal{H}_∞ controller C , sensitivity functions S_1 , S_3 of the two control systems with the different PD gains respectively. The upper figure shows the case with the minimal damping factor of 8×10^{-4} , and the lower one does the case with the factor of 6×10^{-2} . Further, in each figure, the nominal and perturbed cases are compared. As seen from the figures, in the upper case, the controller has a very stark notch compared to that in the lower case. Then, considering the sensitivity function S_1 corresponding to the tracking control performance, in both the systems and in both the nominal and perturbed cases, the properties are the same. However, when it comes to S_3 related to the oscillation suppression performance, although in the nominal case their properties are the same in both the system, in the perturbed case they are totally different. In the upper case, the stark oscillatory property has appeared due to the pole-zero cancellation failure while in the lower case it is not the case despite of such a failure. This difference stems from the difference in the minimal damping factors. Therefore, all the above discussions have been demonstrated, and it has been proved that the PD control scheme plays an important role of enhancing the robustness with respect to the oscillation suppression capability.

Additionally, note that considering the fact that the obtained \mathcal{H}_∞ controller is strictly proper, employing PD control obviously extends the class of controllers.



(a) Frequency responses of C , S_1 , and S_3 with the minimal damping factor of 8×10^{-4} .
 (a) nominal case (b) perturbed case.



(b) Frequency responses of C , S_1 , and S_3 with the minimal damping factor of 6×10^{-2} .
 (a) nominal case (b) perturbed case.

Fig. 4. Pole-zero cancellation failure examples from [Toda (2007)].

3.2.2 Optimality with respect to the PD gains

Here, one question may arise, "when is it optimal in choosing the PD control gains and/or the minimal damping factor?". To seek the answer to this question, by using the illustrative example, we have examined various PD gains, the resultant minimal damping factors and control simulation results in a trial and error manner. Then, we have found the following points:

- P1 A too small minimal damping factor leads to poor oscillation suppression performance;
- P2 The maximum of minimal damping factor however does not necessarily reveal the optimal control performance;
- P3 even if with the same minimal damping factor, the control performance varies according to the P gain.

Accordingly, in this example, we have employed the following cost function η_1 to be minimized in choosing the PD gains;

$$\eta_1(d_1, d_3, k_1, k_3) := (\zeta_{min} - 0.4)^2 + 100(k_1 + k_3) \quad (11)$$

where d_i 's and k_i 's are bounded as $2.18e-5 \leq d_1 \leq 2.18e1$, $1.52e-6 \leq d_3 \leq 1.52$, $2.18e-6 \leq k_1 \leq 2.18e2$, $1.52e-7 \leq k_3 \leq 1.52e1$, respectively. Further, to demonstrate the above point 3, the other cost function η_2 taking only ζ_{min} into account

$$\eta_2(d_1, d_3, k_1, k_3) := (\zeta_{min} - 0.4)^2 \quad (12)$$

for similarly bounded d_i 's and k_i 's has been also considered. In the next section, these optimization strategies will be discussed based on control simulations.

4. Control simulations

In this article, to prove that the proposed control method can be applied to even MIMO systems, and to demonstrate the above discussions on the optimality with respect to the PD gains, we here present control simulations. According to the last section, four cases of PD gains are considered, which includes the cases of the respective optimal gains due to η_1 and η_2 , and additional two non-optimal cases. The respective ζ_{min} and PD gains are shown in Table 2. Comparing Cases 1 and 2 in Table 2, it is noticed that the same ζ_{min} and similar D gains can be obtained, however that the P gains in Case 2 are considerably larger than those in Case 1, which indeed reflects the cost functions in (11) and (12).

Case	ζ_{min}	d_1 (Nms)	d_3 (Nms)	k_1 (Nm)	k_3 (Nm)
Case 1 (η_1)	0.40	2.25e-2	1.60e-3	2.18e-6	1.52e-7
Case 2 (η_2)	0.40	2.20e-2	1.46e-3	8.49e-2	6.22e-3
Case 3	0.06	1.02e-1	1,76e-2	4.68e-5	7.60e-7
Case 4	1.00	3.3e-3	5.67e-4	9.35e-4	1.52e-5

Table 2. ζ_{min} and PD gains.

For these cases, step tracking control simulations have been conducted. The conditions are:

1. the simulation period is 10 s;
2. all the initial states are zeros;

3. two types of references $0 \rightarrow \pi/3$ rad and $0 \rightarrow \pi/2$ rad for both r_1 and r_3 , with the step time of 1 s are applied.

The simulation results are shown in Figs. 5–7 respectively. First we shall see the two optimal cases. In Figs. 5 and 6, the upper figures show each displacement on large scale graphs while the lower ones do each tracking control error to the final goal on fine scale ones. Comparing Case 1 of η_1 and Case 2 of η_2 , that is, with the same ζ_{min} of 0.40, on large scale graphs those results are almost the same and reveal the good performances for both tracking control and oscillation suppression. On fine scale graphs, they are still very similar, however the oscillations of the oscillatory components e_2 and e_4 in Case 2 are slightly larger than those in Case 1, and slight overshoots of e_3 can be seen at around 3 s in Case 2, which might be due to the largenesses of k_1 and k_3 .

Next, let us see the non-optimal cases in Fig. 7. In the figure, the upper figure shows the results of Case 3 with the small ζ_{min} of 0.06, while the lower one does those of Case 4 with the large, in fact, maximal ζ_{min} of 1.00 on fine scale graphs respectively. As seen from the figures, as pointed out before, the results of Case 3 reveal poor oscillation suppression performances, while the results of Case 4 reveal a slightly slow response in e_3 and a slight steady error in e_1 , which thus has demonstrated P1 and P2 in the last section.

Consequently, the main goal of extending our proposed method to MIMO systems has successfully been achieved, that is, it has been confirmed that the proposed method is effective and feasible for even MIMO systems. Additionally, discussions on the optimality with respect to the PD control gains have been given in some detail. The obtained control system based on the cost function η_1 has revealed good performances in both tracking control and oscillation suppression, which therefore can be one of the promising candidates for the optimality, although it has not yet been conclusive that η_1 can be useful for other examples.

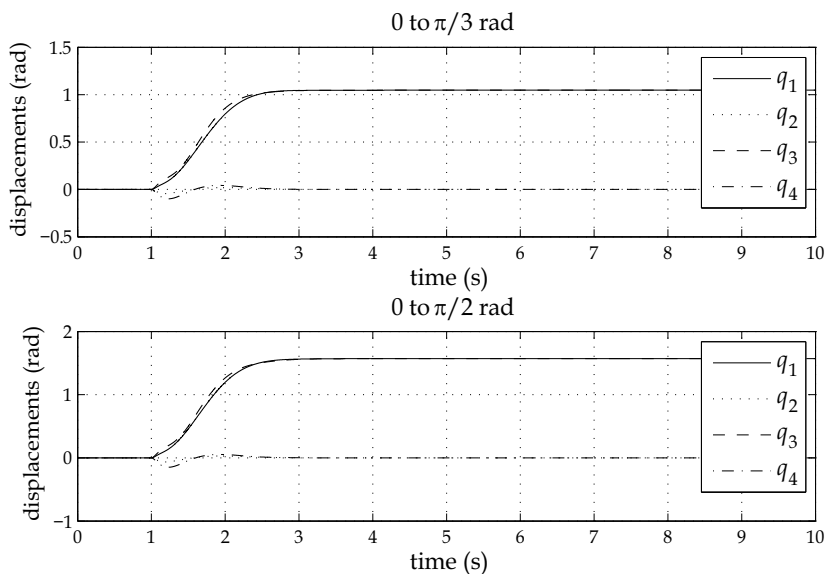
5. Conclusions

In this article, we have presented the control design method based on \mathcal{H}_∞ control and PD control aiming at a uniform approach to motion control of various flexible mechanical systems. In particular, with a special emphasis on MIMO systems and the optimal PD gains, we have introduced and demonstrated the concept of the generic problem setting in the modeling phase, the physics behind our control method, that is, how the PD control scheme elaborately powers the \mathcal{H}_∞ control system, the promising candidate of cost function for the optimal PD gains, and the control simulations which have supported all the discussions.

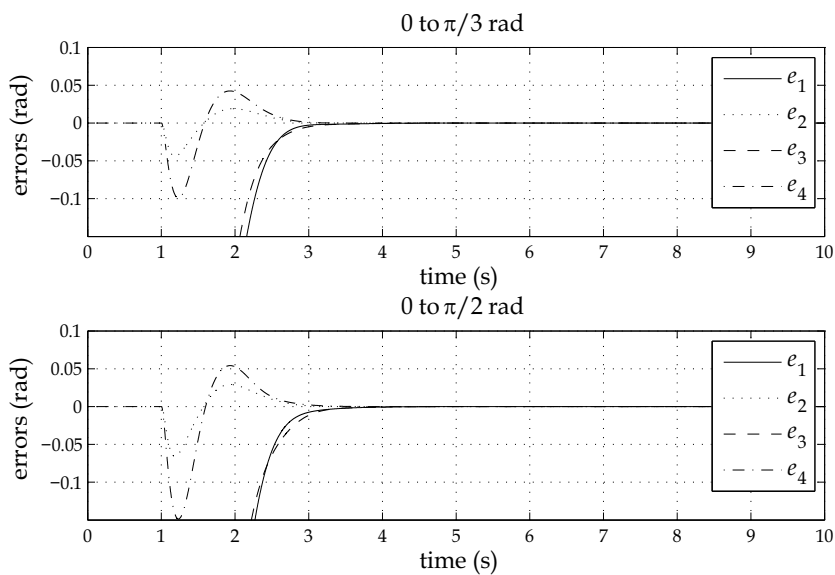
Here, again we emphasize the advantageous features of the proposed approach:

1. A variety of flexible mechanical systems can be systematically dealt with in a uniform and simple manner where the frequency-domain perspective will be provided;
2. The robustness can be easily enhanced by appropriately choosing the PD control gains;
3. Due to the nature based on pole-zero cancellation, any oscillation sensors will not be required, which is considerably important in the practical sense.

Consequently, we have shown that our methodology is easy to use and effective indeed and further will possibly evolve in the sense of optimality.

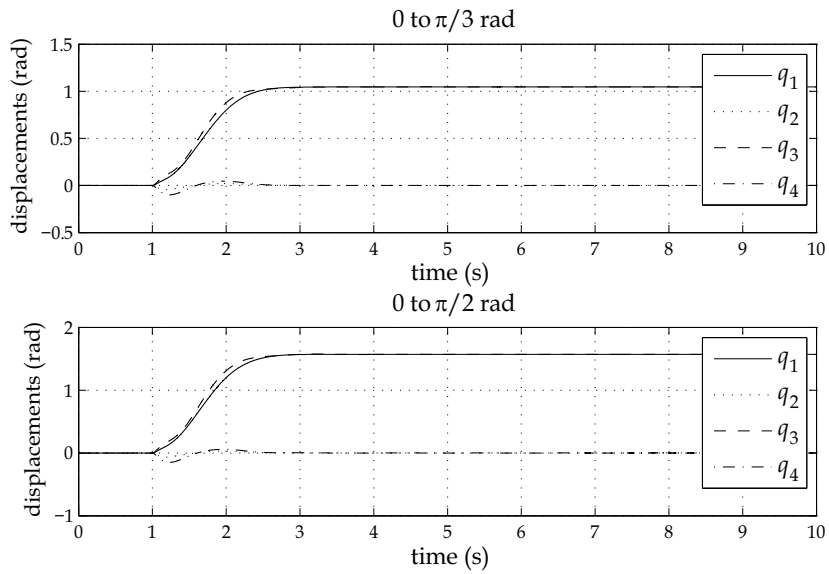


(a) Simulation results (large scale).

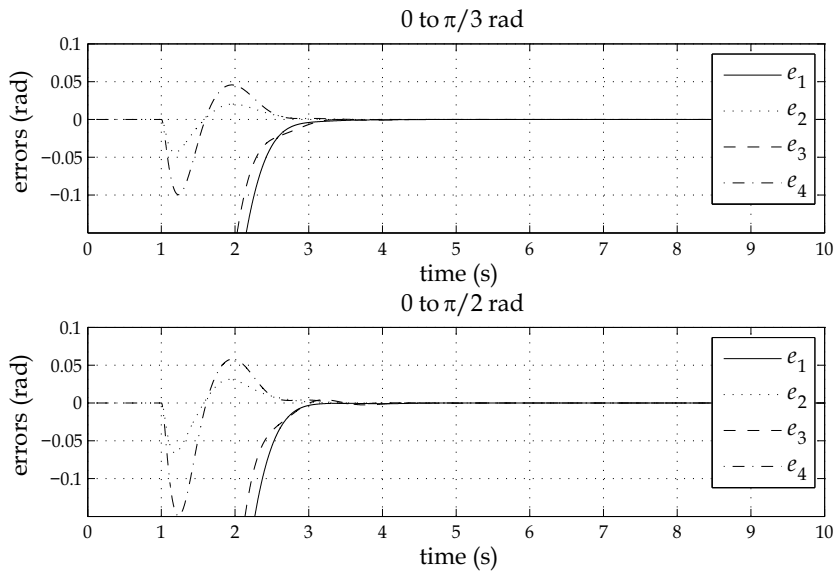


(b) Simulation results (fine scale).

Fig. 5. Simulation results using the optimal PD gains due to η_1 ($\zeta_{min}=0.40$).

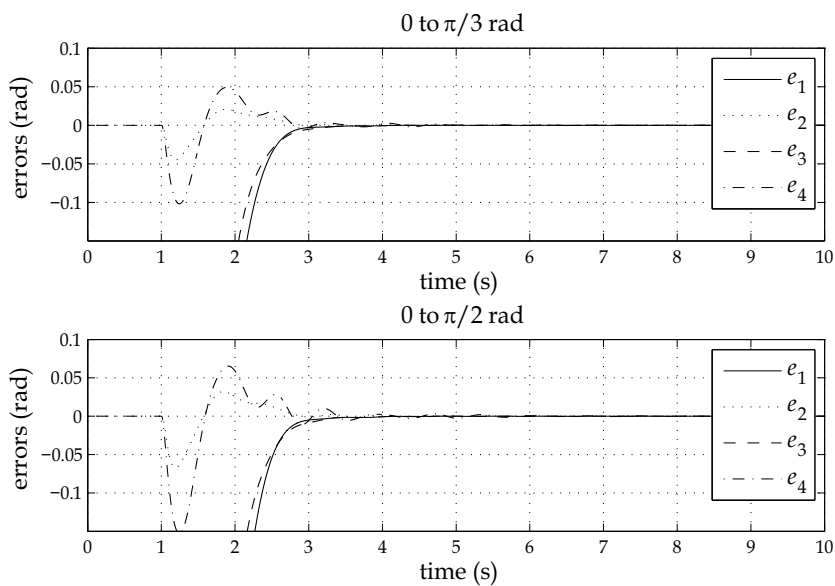


(a) Simulation results (large scale).

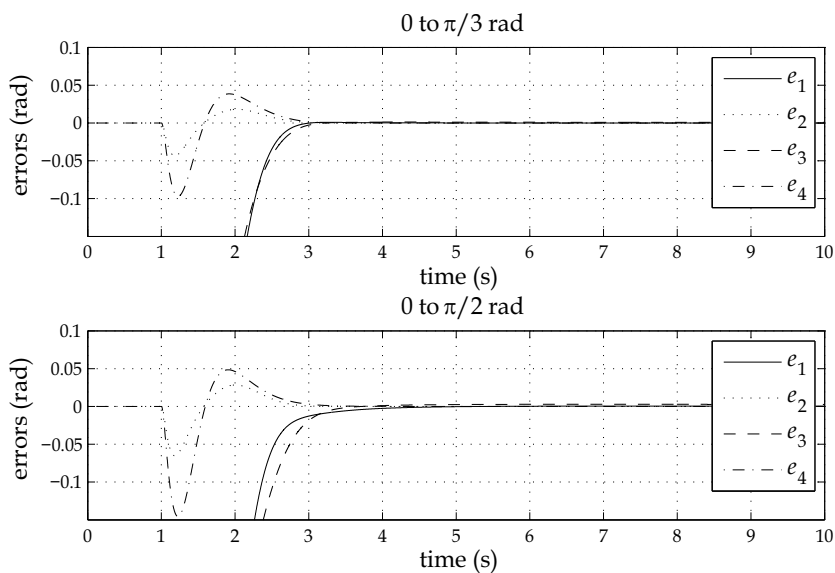


(b) Simulation results (fine scale).

Fig. 6. Simulation results using the optimal PD gains due to η_2 ($\zeta_{min}=0.40$).



(a) Simulation results with $\zeta_{min} = 0.06$.



(b) Simulation results with $\zeta_{min} = 1.00$.

Fig. 7. Simulation results using the non-optimal PD gains.

6. References

- Doyle, J. C.; Glover, K.; Khargonekar, P. P. & Francis, B. A. (1989). State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, August-1989, 831–847.
- Kang, Z.; Fuji, S.; Zhou, C. & Ogata, K. (1999). Adaptive control of a planar gantry crane by the switching of controllers. *Transactions of the Society of Instrument and Control Engineering*, Vol. 35, No. 2, Feb-1999, 253–261.
- Magee, D. P. & Book, W. J. (1995). Filtering micro-manipulator wrist commands to prevent flexible base motion. *Proceedings of American Control Conference 1995*, 924–928.
- Nenchev, D. N.; Yoshida, K. & Uchiyama, M. (1996). Reaction null-space based control of a flexible structure mounted manipulator systems. *Proceedings of IEEE International Conference on Decision and Control 1996*, 4118–4123.
- Nenchev, D. N.; Yoshida, K.; Vichitkulsawat, P.; Konno, A. & Uchiyama, M. (1997). Experiments on reaction null-space based decoupled control of a flexible structure mounted manipulator system. *Proceedings of IEEE International Conference on Robotics and Automation 1997*, 2528–2534.
- Packard, A. & Doyle, J. C. (1993). The complex structured singular value. *Automatica*, Vol. 29, No. 1, 1993, 71–110.
- Sharon, A. & Hardt, D. (1984). Enhancement of robot accuracy using end-point feedback and a macro-micro manipulator system. *Proceedings of American Control Conference 1984*, 1836–1842.
- Spong, M. W. (1987). Modeling and control of elastic joint robots. *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 109, Dec-1987, 310–319.
- Toda, M. (2004). A unified approach to control of mechanical systems with a flexible structure. *Proceedings of International Symposium on Robotics and Automation 2004*, 313–319.
- Toda, M. (2007). A unified approach to robust control of flexible mechanical systems. *Proceedings of IEEE Conference on Decision and Control 2007*, 5787–5793.
- Torres, M. A.; Dubowsky, S. & Pisoni, A. C. (1994). Path-planning for elastically-mounted space manipulators: experimental evaluation of the coupling map. *Proceedings of IEEE International Conference on Robotics and Automation 1994*, 2227–2233.
- Ueda, U. & Yoshikawa, T. (2004). Mode-shape compensator for improving robustness of manipulator mounted on flexible base. *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, April-2004, 256–268.
- Wang, D. & Vidyasagar, M. (1990). Passive control of a single flexible link. *Proceedings of IEEE International Conference on Robotics and Automation 1990*, 1432–1437.
- Yano, K. & Terashima, K. (2001). Robust liquid container transfer control for complete sloshing suppression. *IEEE Transactions on Control and Systems Technology*, Vol. 9, No. 3, May-2001, 483–493.
- Yano, K.; Higashikawa, S. & Terashima, K. (2001). Liquid container transfer control on 3D transfer path by hybrid shaped approach. *Proceedings of IEEE International Conference on Control Applications 2001*, 1168–1173.
- Zhou, K.; Doyle, J. C. & Glover, K. (1995). *Robust Control and Optimal Control*, Prentice-Hall, New Jersey.

An Improved Adaptive Kinematics Jacobian Trajectory Tracking of a Serial Robot Passing Through Singular Configurations

Ali T. Hasan¹, Hayder M.A.A. Al-Assadi² and Ahmad Azlan Mat Isa²

¹*Mechanical & Manufacturing Engineering Department, Faculty of Engineering, University Putra Malaysia 43400 UPM Serdang, Selangor*

²*Faculty of Mechanical Engineering, University Technology MARA Shah Alam, 40450 Malaysia*

1. Introduction

In real time applications, the trajectory which has to be followed and the task that has to be performed during motion planning of multi-axis non-linear mechanical systems, such as robot manipulators are of great importance. Due to the non-linear transformation between the task space and the joint space coordinates, singularities and uncertainties in the arm configuration occur, the unplanned occurrence of such problems drive the end-effector out of the desired path which may cause collision of the robot arm with objects located in its work cell (Köker, 2005; Antonelli et al., 2003).

Depending on different tasks operation requirements and circumstances, motion control algorithms can be developed either at the *kinematics* level or at the *dynamic* level (Graca & Gu, 1993; Karilk & Aydin, 2000). To develop a dynamic control algorithm, torque limits of the joint actuators are to be handled, two typical approaches were introduced which are the Computed-torque and Resolved-acceleration approach, both approaches are based on the inverse dynamic model of the robot system (Asada & Soltin, 1986; Sopng & Vinyasagar, 1998; Faiz & Agrawal, 2000). A problem with these algorithms is the remarkable computational load required to handle the dynamics of a full-sized manipulator, which is seldom affordable by current industrial control units. In addition, implementation of torque-based control laws requires replacement of the low-level joint servos typically available in industrial robots with custom control loops.

Aimed at overcoming the above drawbacks, a different approach to path tracking based on the kinematics control was proposed. In detail, kinematics control consists in an inverse kinematics transformation which sends to the joint servos the reference values corresponding to an assigned end-effector trajectory; as a first advantage, this allows simple interfacing with the standard control architecture of industrial robots. In the framework of kinematics-based methods for path tracking, the counterpart of the physically meaning joint torque limits is played by acceleration constraints and the use of full dynamic models can be avoided; this typically leads to computationally light algorithms that allow real-time implementation on standard numerical hardware even for robot arms of many Degrees of

Freedom (DOF). A further advantage of kinematics control methods is the possibility of exploiting the presence of redundant (DOF) (Antonelli et al., 2003).

A considerable research effort has been devoted to solve the Inverse Kinematics problem in past years (Yang, 1969; Duffy & Rooney, 1975; Albala & Angeles, 1979; Tsai & Morgan, 1985; Daniel & Raul, 2003). Even though, Closed-form analytical solutions can only be found for manipulators having simple geometric structures (Antonelli et al., 2003; Karilk & Aydin, 2000). A number of algorithmic techniques mainly based on inversion of the mapping established between the joint space and the task space of the manipulator's Jacobian matrix have been proposed for those structures that cannot be solved in closed form.

The Resolved Motion Rate-Control technique was the first work in this field (Whitney, 1969), in this technique the pseudoinverse of the Jacobian matrix is used to obtain the joint velocities corresponding to a given end-effector velocity, a major drawback of this method was the singularity problem. The use of a damped least-squares inverse of the Jacobian matrix has been later proposed in lieu of the pseudoinverse to overcome the problem of kinematics singularities (Nakamura & Hanafusa, 1986; Wampler, 1986).

Since in the above algorithmic methods the joint angles are obtained by numerical integration of the joint velocities, these and other related techniques suffer from errors due to both long-term numerical integration drift and incorrect initial joint angles.

To alleviate the difficulty, algorithms based on the feedback error correction are introduced (Balestrino et al., 1984; Wampler & Leifer, 1988). However, it is assumed that the exact model of manipulator Jacobian matrix of the mapping from joint coordinate to Cartesian coordinate is exactly known. It is also not sure to what extent the uncertainty could be allowed. Therefore, most research on robot control has assumed that the exact kinematics and Jacobian matrix of the manipulator from joint space to Cartesian space are known. This assumption leads to several open problems in the development of robot control laws today (Antonelli et al., 2003).

A new direction making control systems able to attribute more intelligence and high degrees of autonomy was proposed. With proper development, intelligent control systems may have great potential for solving today's and tomorrow's more complex control problems. The common objective associated with an intelligent control system can be identified to reduce accurate crisp model dependence and increase intelligent abilities of the control system.

Owing to this motivation, there have been increasing research interest of ANNs and a number of realistic control approaches have been proposed and justified for their feasible applications to robotic systems (D'Souza et al., 2001; Ogawa et al., 2005; Köker, 2005; Hasan et al., 2006; Al-Assadi et al., 2007). Artificial neural network (ANN) uses data sets to obtain the models of systems in fields such as robotics, factory automation and autonomous vehicles. Their ability to learn by example makes artificial neural networks very flexible and powerful. Therefore, neural networks have been intensively used for solving regression and classification problems in many fields. In short, neural networks are nonlinear processes that perform learning and classification. Recently neural networks have been used in many areas that require computational techniques such as pattern recognition, optical character recognition, outcome prediction and problem classification. The current focuses in learning research lies on increasingly more sophisticated algorithms for the off-line analysis of finite data sets, without severe constraints on the computational complexity of the algorithms (Bingual et al., 2005).

Kuroe and colleges (Kuroe et al., 1994) have proposed a learning method of a neural network such that the network represents the relations of both the positions and velocities from the Cartesian coordinate to the joint space coordinate. They've derived a learning algorithm for arbitrary connected recurrent networks by introducing adjoint neural networks for the original neural networks (Network inversion method). On-line training has been performed for a 2 DOF robot.

It was essentially an on-line learning process (Graca & Gu, 1993) have developed a Fuzzy Learning Control algorithm. Based on the robotic differential motion procedure, the Jacobian inverse has treated as a fuzzy matrix and has learned through the fuzzy regression process. It was significant that the fuzzy learning control algorithm neither requires an exact kinematics model of a robotic manipulator, nor a fuzzy inference engine as is typically done in conventional fuzzy control. Despite the fact that unlike most learning control algorithms, multiple trials are not necessary for the robot to "learn" the desired trajectory. A major drawback was that it only remembers the most recent data points introduced, the researchers have recommended neural networks so that it would remember the trajectories as it traversed them.

Studying the trajectory tracking of a serial manipulator by using ANNs has two problems, one of these is the selection of the appropriate type of network and the other is the generating of suitable training data set (Funahashi, 1998; Hasan et al, 2007). Researchers have applied different methods for gathering training data, while some of them have used the kinematics equations (Karilk & Aydin, 2000; Bilingual et al., 2005), others have used the network inversion method (Kuroe et al., 1994); Köker, 2005), while the cubic trajectory planning was also used (Köker et al., 2004), a simulation program has also been used for this purpose (Driscoll, 2000). However, there are always kinematics uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear train.

The proposed solution of the kinematics Jacobian in this approach, involves the determination of the end-effectors coordinates and their rate of change as a function of given positions and speed of the axes of motion, although this is very difficult in practice (Hornic, 1991), training data were recorded experimentally from sensors fixed on each joint and the Euler (RPY) representation was used to represent the orientation (as was recommended by Karilk and Aydin (Karilk & Aydin, 2000), as they have used the robot model to get the training data and used the homogeneous transformation matrix representation to represent the orientation). On the other hand, two different network's configurations were trained and compared to examine the effect of the orientation on the Inverse Kinematics solution of serial robots. Finally, the obtained results from the testing phase of the best network were verified experimentally using a six DOF serial robot manipulator.

2. Kinematics of serial robots

For serial robot manipulators, the Cartesian space coordinates x of a robot manipulator is related to the joint coordinates q by:

$$x = f(q) \quad (1)$$

where $f(\cdot)$ is a non-linear differential function.

If the Cartesian coordinates x were given, joint coordinates q can be obtained as:

$$q = f^{-1}(x) \quad (2)$$

If a Cartesian linear velocity is denoted by V , the joint velocity vector \dot{q} has the following relation:

$$V = J\dot{q} \quad (3)$$

Where J is the Jacobian matrix.

If V , is a desired Cartesian velocity which represents the linear velocity of the desired trajectory to be followed. Then, the joint velocity vector \dot{q} can be resolved by:

$$\dot{q} = J^{-1}V \quad (4)$$

In differential motion control, the desired trajectory is subdivided into sampling points separated by a time interval Δt between two terminal points of the path. Assuming that at time t_i the joint positions take on the value $q(t_i)$, the required \dot{q} at time $(t_i + \Delta t)$ is conventionally updated by using:

$$q(t_i + \Delta t) = q(t_i) + \dot{q} \Delta t \quad (5)$$

Substituting Eqs. (2) and (4) into (5) yields:

$$q(t_i + \Delta t) = f^{-1}(x)(t_i) + J^{-1}V\Delta t \quad (6)$$

Equation (6) is a kinematics control law used to update the joint position q and is evaluated on each sampling interval. The resulting $q(t_i + \Delta t)$ is then sent to the individual joint motor servo-controllers, each of which will independently drive the motor so that the robotic manipulator can be maneuvered to follow the desired trajectory (Graca & Gu, 1993).

Using ANN to solve relation (2), researchers applied two approaches. In (Ogawa et al., 2005; Hasan et al., 2006; Köker et al., 2004) only the Cartesian coordinates has been inverted, mapping from the joint space to the Cartesian space is uniquely decided when the end effector's position is calculated using direct kinematics, as shown in figure 1(a). However, the transformation from the Cartesian to the joint space is not uniquely decided in the inverse kinematics as shown in figure 1(b).

When coupling of the position and orientation e.g., (Köker, 2005; Karilk & Aydin, 2000) Denavit and Hartenberg (Denavit & Hertenberg, 1955) proposed a matrix method of systematically establishing a coordinate system to each link of an articulated chain as shown in figure 2 to describe both translational and rotational relationships between adjacent links (Fu et al., 1987; Köker, 2005).

In this method each of the manipulator links is modelled, this modelling describes the "A" homogeneous transformation matrix, which uses four link parameters. The forward kinematics solution can be obtained as:

$$A_{END-EFFECTOR} = T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 = \begin{bmatrix} \text{Rotation} & | & \text{Position} \\ \text{matrix} & | & \text{vector} \\ \text{-----} & | & \text{-----} \\ \text{Perspective} & | & \text{Scaling} \\ \text{transformation} & | & \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

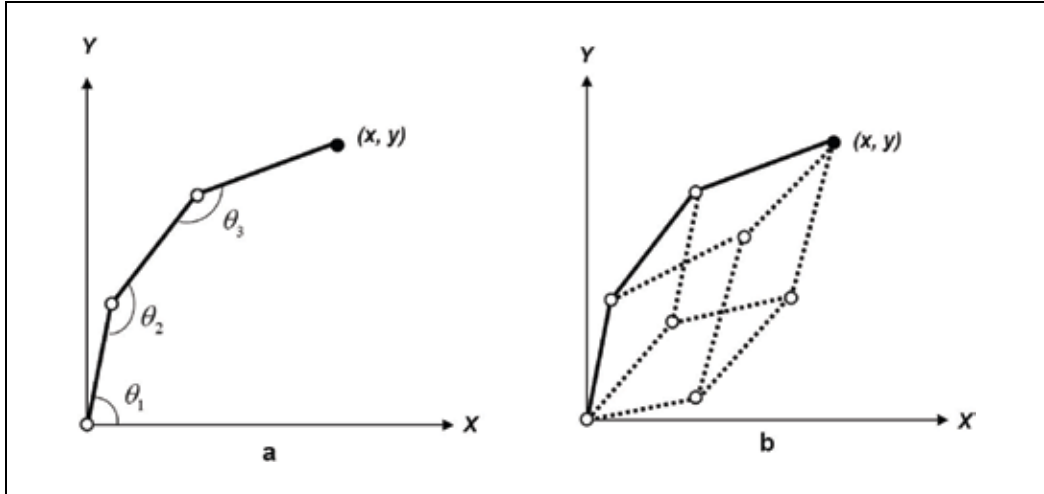


Fig. 1. a) Joint angles and end-effector's coordinates (forward kinematics).
 b) Combination of all possible joint angles (Inverse Kinematics).

Where:

- n : Normal vector of the hand. Assuming a parallel-jaw hand, it is orthogonal to the fingers of the robot arm.
- a : Sliding vector of the hand. It is pointing in the direction of the finger motion as the gripper opens and closes.
- a : Approach vector of the hand. It is pointing in the direction normal to the palm of the hand (i.e., normal to the tool mounting plate of the arm).
- p : Position vector of the hand. It points from the origin of the base coordinate system to the origin of the hand coordinate system, which is usually located at the center point of the fully closed fingers.

The orientation of the hand is described according to the RPY rotation as:

$$RPY(\varphi_x, \varphi_y, \varphi_z) = Rot(Z_w, \varphi_z) \cdot Rot(Y_w, \varphi_y) \cdot Rot(X_w, \varphi_x) \quad (8)$$

After T_6 matrix is solved:

$$\varphi_z = ATAN2(n_y, n_x) \quad (9)$$

$$\varphi_y = ATAN2(-n_z, n_x \cos \varphi_z + n_y \sin \varphi_z) \quad (10)$$

$$\varphi_x = ATAN2(a_x \sin \varphi_z - a_y \cos \varphi_z, o_y \cos \varphi_z - o_x \sin \varphi_z) \quad (11)$$

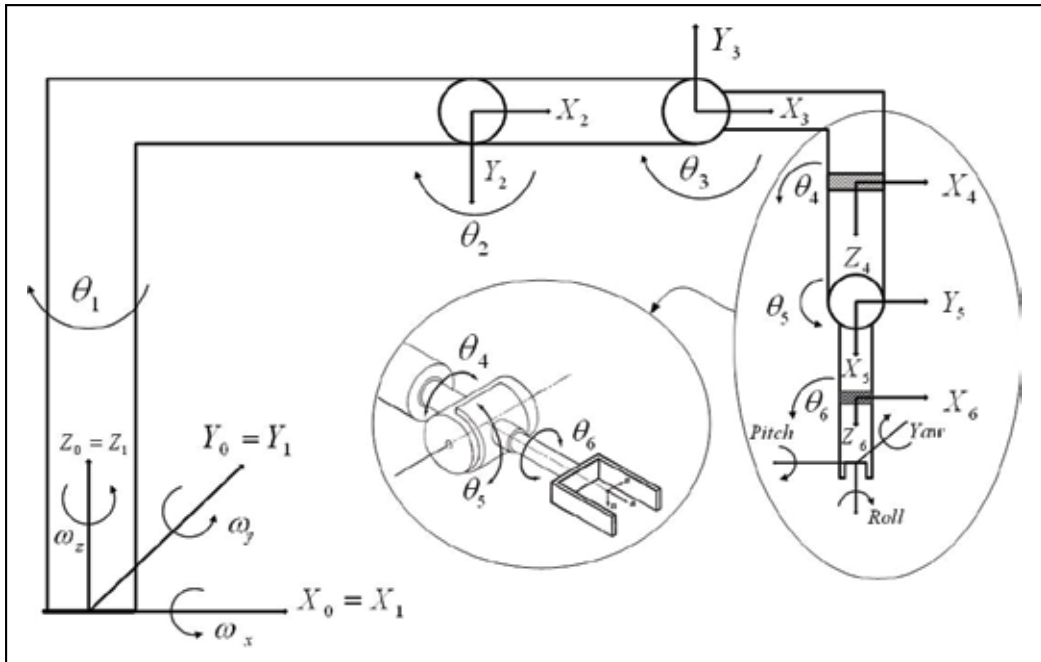


Fig. 2. Schematic diagram for a general 6 DOF serial robot showing the wrist mechanism

These equations describe the orientation according to the RPY representation (Karilk & Aydin, 2000). To find the IK solution, however, joints angles are found according to the manipulator's end position, described with respect to the world coordinate system.

IK solution can be shown as a function:

$$IK(X, Y, Z, \varphi_x, \varphi_y, \varphi_z) = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \quad (12)$$

Traditional methods for solving the IK problem are inadequate if the structure of the robot is complex, besides; these methods suffer from the fact that the solution does not give a clear indication on how to select an appropriate solution from the several possible solutions for a particular arm configuration, users often needs to rely on their intuition to choose the right answer (Fu et al., 1987; Hasan et al., 2006).

On the other hand, solving Eq. (4) for the joint velocities (Inverting the Jacobian matrix), results in the singularity problem. The manipulator singularity resolution problem has attracted many research interests, and various approaches have been proposed to tackle the problem. Techniques of coping with kinematics singularities can be divided into four groups: avoiding singular configurations, robust inverses, a normal form approach and extended Jacobian techniques.

The first approach to cope with singularities is to keep a current configuration far away from singular configurations. Unfortunately, it causes severe restrictions on the configuration space as well as the workspace because the singular configurations split the configuration space into separate components. To avoid ill conditioning of the Jacobian matrix, robust inverses are used. Instead inverting the original Jacobian matrix at singularity, a disturbed well-conditioned Jacobian matrix is inverted. The main drawback using this approach is that robust inverse methods increase errors in following a desired path.

The normal form technique, with the use of diffeomorphisms in joint and task spaces, expresses original kinematics around singularity in the simplest normal form. Then, a piece of the path to follow corresponding to the singular configuration mapped into the task space is moved from the task to the joint space and trajectory planning is performed there. Far away from singularities the basic Newton algorithm is used to generate a trajectory. Finally, trajectory pieces are joined.

For most singularities the normal form approach enables to detect their types. It provides for a smooth passing through singular configurations. The main disadvantage of the normal form approach is a significant computational load in deriving the diffeomorphisms.

Finally, The extended Jacobian technique, supplements original kinematics with auxiliary functions. Then, extended Jacobian is formulated to be well conditioned.

For nonredundant manipulators with square Jacobian matrices the extended Jacobian forms a non-square matrix and its generalized (Moore-Penrose) inversion is computationally expensive (Duleba & Sasiadek, 2000).

Therefore, to analyze the singular conditions of a manipulator and develop effective algorithms to resolve the inverse kinematics problem at or in the vicinity of singularities are of great importance.

3. Artificial neural networks

The possibility of developing a machine that would “think” has intrigued human beings since ancient times, Machinery can outperform humans physically. Similarly, computers can outperform mental functions in limited areas, notably in the speed of mathematical calculations. For example, the fastest computers developed are able to perform roughly 10 billion calculations per second. But making more powerful computers will probably not be the way to create a machine capable of thinking. Computer programs operate according to set procedures, or logic steps, called algorithms. In addition, most computers do serial processing such as operations of recognition and computations are performed one at a time. The brain works in a manner called parallel processing, performing a number of operations simultaneously. To achieve simulated parallel processing, artificial neural networks (ANNs) are collections of small individual interconnected processing units. Information is passed between these units along interconnections. An incoming connection has two values associated with it, an input value and a weight. The output of the unit is a function of the summed value. ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification (Kalogirou, 2001).

The elementary nerve cell called a neuron, which is the fundamental building block of the biological neural network. Its schematic diagram is shown in Figure 3.

A typical cell has three major regions: the cell body, which is also called the soma, the axon, and the dendrites. Dendrites form a dendritic tree, which is a very fine bush of thin fibers around the neuron's body. Dendrites receive information from neurons through axons-Long fibers that serve as transmission lines. An axon is a long cylindrical connection that carries impulses from the neuron. The end part of an axon splits into a fine arborization. Each branch of it terminates in a small end bulb almost touching the dendrites of neighbouring neurons. The axon-dendrite contact organ is called a *synapse*. The synapse is where the neuron introduces its signal to the neighbouring neuron (Zurada, 1992; Hasan et al., 2006),

to stimulate some important aspects of the real biological neuron. An ANN is a group of interconnected artificial neurons usually referred to as "node" interacting with one another in a concerted manner; Figure 4 illustrates how information is processed through a single node. The node receives weighted activation of other nodes through its incoming connections. First, these are added up (summation). The result is then passed through an activation function and the outcome is the activation of the node. The activation function can be a threshold function that passes information only if the combined activity level reaches a certain value, or it could be a continuous function of the combined input, the most common to use is a sigmoid function for this purpose. For each of the outgoing connections, this activation value is multiplied by the specific weight and transferred to the next node (Kalogirou, 2001; Hasan, 2006).

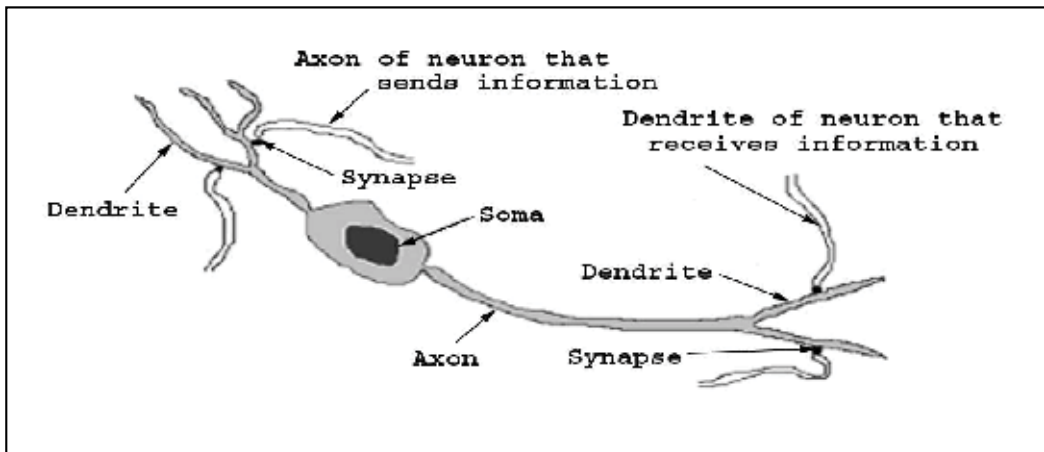


Fig. 3. Schematic diagram for the biological neuron

An artificial neural network consists of many nodes joined together usually organized in groups called 'layers', a typical network consists of a sequence of layers with full or random connections between successive layers as Figure 5 shows. There are typically two layers with connection to the outside world; an input buffer where data is presented to the network, and an output buffer which holds the response of the network to a given input pattern, layers distinct from the input and output buffers called 'hidden layer', in principle there could be more than one hidden layer, In such a system, excitation is applied to the input layer of the network.

Following some suitable operation, it results in a desired output. Knowledge is usually stored as a set of connecting weights (presumably corresponding to synapse efficiency in biological neural system) (Santosh et al., 1993). A neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two respects; the knowledge is acquired by the network through a learning process, and interneuron connection strengths known as synaptic weights are used to store the knowledge (Haykin, 1994).

Training is the process of modifying the connection weights in some orderly fashion using a suitable learning method. The network uses a learning mode, in which an input is presented to the network along with the desired output and the weights are adjusted so that the network attempts to produce the desired output. Weights after training contain meaningful information whereas before training they are random and have no meaning (Kalogirou, 2001).

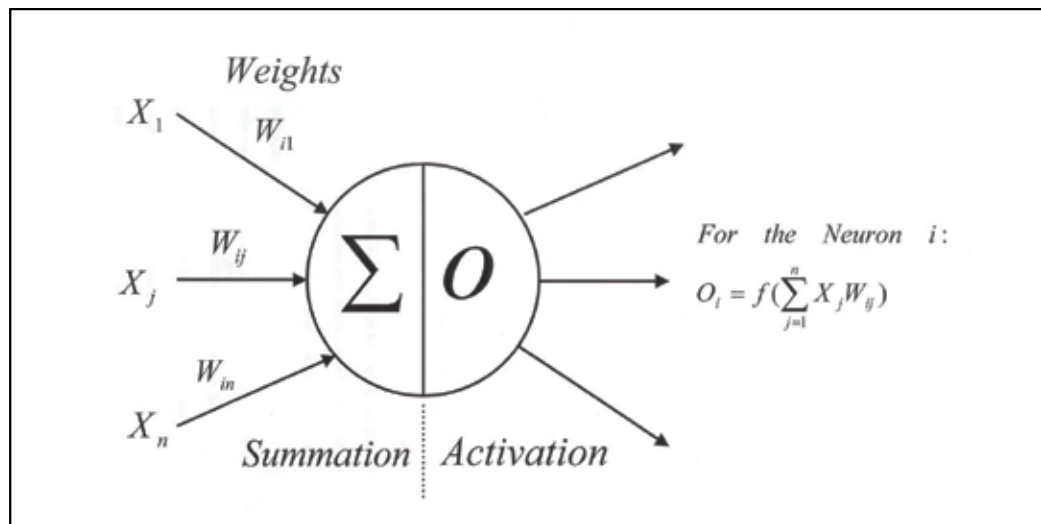


Fig. 4. Information processing in the neural unit

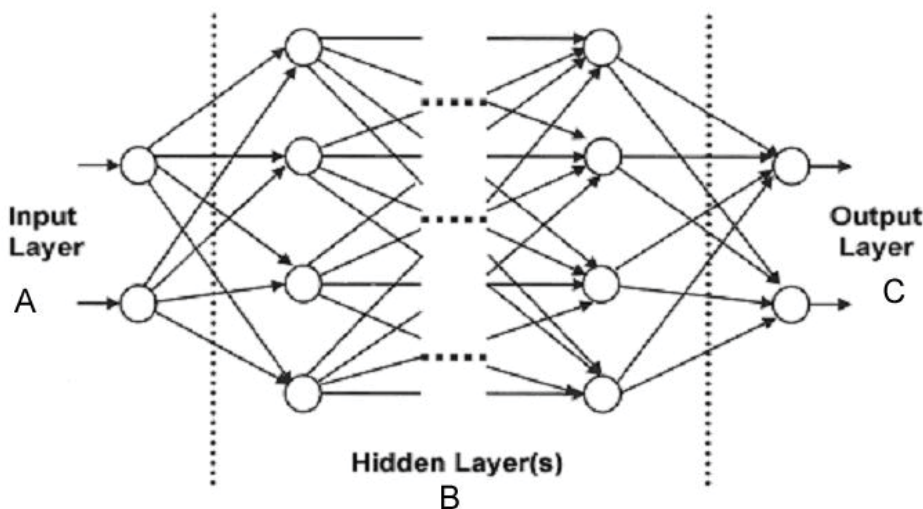


Fig. 5. Schematic diagram of a multilayer feedforward neural network

Two different types of learning can be distinguished: supervised and unsupervised learning, in supervised learning it is assumed that at each instant of time when the input is applied, the desired response d of the system is provided by the teacher. This is illustrated in Figure 6-a. The distance $\rho [d,o]$ between the actual and the desired response serves as an error measure and is used to correct network parameters externally. Since adjustable weights are assumed, the teacher may implement a reward-and-punishment scheme to adopt the network's weight. For instance, in learning classifications of input patterns or situations with known responses, the error can be used to modify weights so that the error decreases. This mode of learning is very pervasive.

Also, it is used in many situations of learning. A set of input and output patterns called a training set is required for this learning mode. Figure 6-b shows the block diagram of unsupervised learning. In unsupervised learning, the desired response is not known; thus, explicit error information cannot be used to improve network's behaviour. Since no information is available as to correctness or incorrectness of responses, learning must somehow be accomplished based on observations of responses to inputs that we have marginal or no knowledge about (Zurada, 1992).

The fundamental idea underlying the design of a network is that the information entering the input layer is mapped as an internal representation in the units of the hidden layer(s) and the outputs are generated by this internal representation rather than by the input vector. Given that there are enough hidden neurons, input vectors can always be encoded in a form so that the appropriate output vector can be generated from any input vector (Santosh et al., 1993).

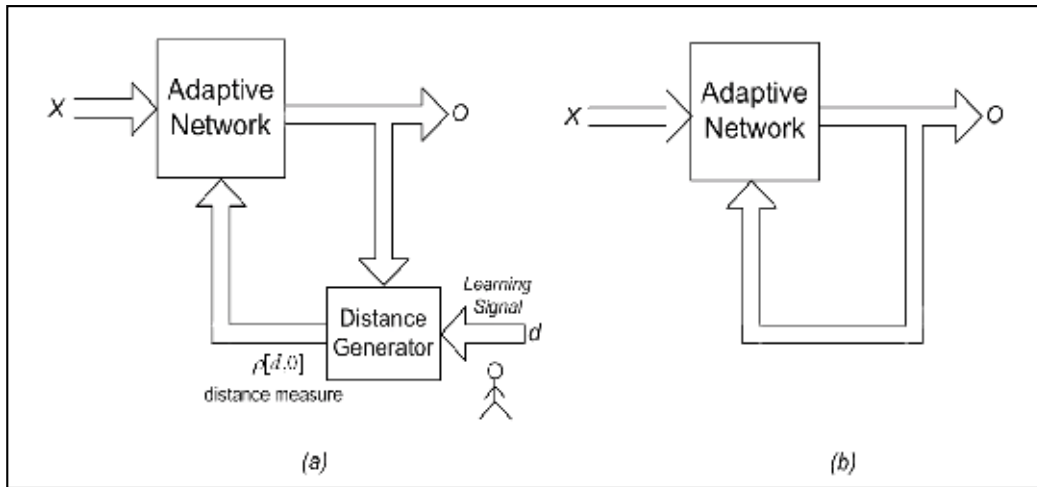


Fig. 6. Basic learning modes

As it can be seen in figure 5, the output of the units in layer A (Input Layer) are multiplied by appropriate weights W_{ij} and these are fed as inputs to the hidden layer. Hence if O_i are the output of units in layer A, then the total input to the hidden layer, i.e., layer B is:

$$Sum_B = \sum_i O_i W_{ij} \tag{13}$$

And the output O_j of a unit in layer B is:

$$O_j = f(sum_B) \tag{14}$$

Where f is the non-linear activation function, it is a common practice to choose the sigmoid function given by:

$$f(O_j) = \frac{1}{1 + e^{-O_j}} \tag{15}$$

as the nonlinear activation function. However, any input-output function that possesses a bounded derivative can be used in place of the sigmoid function. If there is a fixed, finite set

of input-output pairs, the total error in the performance of the network with a particular set of weights can be computed by comparing the actual and the desired output vectors for each presentation of an input vector. The error at any output unit e_k in the layer **C** can be calculated by:

$$e_k = d_k - O_k \quad (16)$$

Where d_k is the desired output for that unit in layer **C** and O_k is the actual output produced by the network .the total error E at the output can be calculated by:

$$E = \frac{1}{2} \sum_k (d_k - O_k)^2 \quad (17)$$

Learning comprises changing weights so as to minimize the error function and to minimize E by the gradient descent method. It is necessary to compute the partial derivative of E with respect to each weight in the network. Equations (13) and (14) describe the forward pass through the network where units in each layer have there states determined by the inputs they received from units of lower layer. The backward pass through the network that involves "back propagation " of weight error derivatives from the output layer back to the input layer is more complicated. For the sigmoid activation function given in equation (15), the so-called delta-rule for iterative convergence towards a solution maybe stated in general as:

$$\Delta W_{jk} = \eta \delta_k O_j \quad (18)$$

Where η is the learning rate parameter, and the error δ_k at an output layer unit K is given by:

$$\delta_k = O_k(1 - O_k)(d_k - O_k) \quad (19)$$

And the error δ_j at a hidden layer unit is given by:

$$\delta_j = O_j(1 - O_j) \sum_k \delta_k W_{jk} \quad (20)$$

Using the generalize delta rule to adjust weights leading to the hidden units is back propagating the error-adjustment, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons. A back propagation network trains with two step procedures as it is shown in figure 7, the activity from the input pattern flows forward through the network and the error signal flows backwards to adjust the weights using the following equations:

$$W_{ij} = W_{ij} + \eta \delta_j O_i \quad (21)$$

$$W_{jk} = W_{jk} + \eta \delta_k O_j \quad (22)$$

Until for each input vector the output vector produced by the network is the same as (or sufficiently close to) the desired output vector (Santosh et al., 1993).

ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification (Kalogirou, 2001).

4. Experiment design

Trajectory planning was performed for every 1-second interval using cubic trajectory planning method to generate the angular position and velocity for each joint, and then these generated data were fed to the robot's controller to generate the corresponding Cartesian position, orientation and linear velocity of the end-effector, which were recorded experimentally from sensors fixed on the robot joints.

In trajectory planning of a manipulator, it is interested in getting the robot from an initial position to a target position with free of obstacles path. Cubic trajectory planning method has been used in order to find a function for each joint between the initial position, θ_0 , and final position, θ_f of each joint.

It is necessary to have at least four-limit value on the $\theta(t)$ function that belongs to each joint, where $\theta(t)$ denotes the angular position at time t .

Two limit values of the function are the initial and final position of the joint, where:

$$\theta(0) = \theta_0 \quad (23)$$

$$\theta(t_f) = \theta_f \quad (24)$$

Additional two limit values, the angular velocity will be zero at the beginning and the target position of the joint, where:

$$\dot{\theta}(0) = 0 \quad (25)$$

$$\dot{\theta}(t_f) = 0 \quad (26)$$

Based on the constrains of typical joint trajectory listed above, a third order polynomial function can be used to satisfy these four conditions; since a cubic polynomial has four coefficients.

These conditions can determine the cubic path, where a cubic trajectory equation can be written as:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (27)$$

The angular velocity and acceleration can be found by differentiation, as follows:

$$\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2 \quad (28)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3t \quad (29)$$

Substituting the constrain conditions in the above equations results in four equations with four unknowns:

$$\begin{aligned}
 \theta_0 &= a_0, \\
 \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \\
 0 &= a_0, \\
 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2
 \end{aligned}
 \tag{30}$$

The coefficients are found by solving the above equations.

$$\begin{aligned}
 a_0 &= \theta_0, \\
 a_1 &= 0, \\
 a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0), \\
 a_3 &= \frac{-2}{t_f^3}(\theta_f - \theta_0)
 \end{aligned}
 \tag{31}$$

Angular position and velocity can be calculated by substituting the coefficients driven in Eq. (31) into the cubic trajectory Equations (27) and (28) respectively (Köker et al.,2004), which yield:

$$\theta_i(t) = \theta_{i0} + \frac{3}{t_f^2}(\theta_{if} - \theta_{i0})t^2 - \frac{2}{t_f^3}(\theta_{if} - \theta_{i0})t^3,
 \tag{32}$$

$$\dot{\theta}_i(t) = \frac{6}{t_f^2}(\theta_{if} - \theta_{i0})t - \frac{6}{t_f^3}(\theta_{if} - \theta_{i0})t^2
 \tag{33}$$

$i = 1, 2, \dots, n$ Where n is the joint number

Joint angles generated ranged from amongst all the possible joint angles that do not exceed the physical limits of each joint; Table 1 shows the range of angles for each joint used in this study.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Range of angles	0° – 160°	0° – 60°	0° – 150°	0° – 150°	0° – 120°	0° – 160°

Table 1. The range of angles for each joint used

Trajectory used for the training process has meant to be random trajectory rather than a common trajectory performed by the robot in order to cover as most space as possible of the robot’s working cell. The interval of 1 second was used between a trajectory segment and another where the final position for one segment is going to be the initial position for the next segment and so on for every joint of the six joints of the robot.

After generating the joint angles and their corresponding angular velocities, these data are fed to the robot controller, which is provided with a sensor system that can detect the angular position and velocity on one hand and the Cartesian position, orientation and the linear velocity of the end-effector on the other hand; which are recorded to be used for the networks’ training and testing process later.

5. ANN implementation

To avoid modeling kinematics and the determination of the inverse of the Jacobian matrix, the ANN technique has been used.

Two different configurations of supervised feed-forward ANNs were designed using C programming language, each of which consists of input, output, and one hidden layer. Every neuron in each network was fully connected with each other. Sigmoid transfer function was chosen to be the activation function, and the generalized backpropagation GDR algorithm was used in the training process.

Off-line training was implemented, every input and output values are usually scaled individually such that overall variance in the data set is maximized, this is necessary as it leads to faster learning, all the vectors were scaled to reflect continuous values ranges from -1 to 1.

FANUC M-710i robot was used in this study, which is a serial robot manipulator consisting of axes and arms driven by servomotors. The place at which arm is connected is a joint, or an axis. This type of robot has three main axes; the basic configuration of the robot depends on whether each main axis functions as a linear axis or rotation axis. The wrist axes are used to move an end effector (tool) mounted on the wrist flange. The wrist itself can be wagged about one wrist axis and the end effector rotated about the other wrist axis, this highly non-linear structure makes this robot very useful in typical industrial applications such as the material handling, assembly of parts and painting.

5.1 Training stage

In order to overcome the uncertainties in arm configuration and singularities that result from applying the robot system model, and to make sure that for a certain trajectory the angular position and velocity of each joint will be the same as desired when planning the trajectory for the robot; the ANN technique has been utilized where learning is only based on observation of the input-output relationship.

In back-propagation networks, the number of hidden neurons determines how well a problem can be learned. If too many are used, the network will tend to try to memorize the problem and thus not generalize well later; if too few are used, the network will generalize well but may not have enough power to learn the patterns well. Obtaining the correct number of hidden neurons is a matter of trial and error.

5.1.1 Networks' topologies

In this chapter, two different configurations were used in the training process to determine which configuration is better to be used corresponding to Eq. (2) previously discussed in section 2.

5.1.1.1 First Configuration (4 - 12 Network Configuration)

As can be seen in Figure 7, the input layer consists of 4 neurons the first three of them represent the Cartesian position of the X, Y and Z positions along the world coordinate system of the robot while the fourth neuron represents the linear velocity of the end-effector. The output layer consists of 12 neurons; the first 6 of them represent the angular position of the robot joints while the last 6 of them represent the angular velocity of each joint respectively. Number of neurons in the hidden layer was set to 77 with a constant learning factor of 0.9 by trial and error.

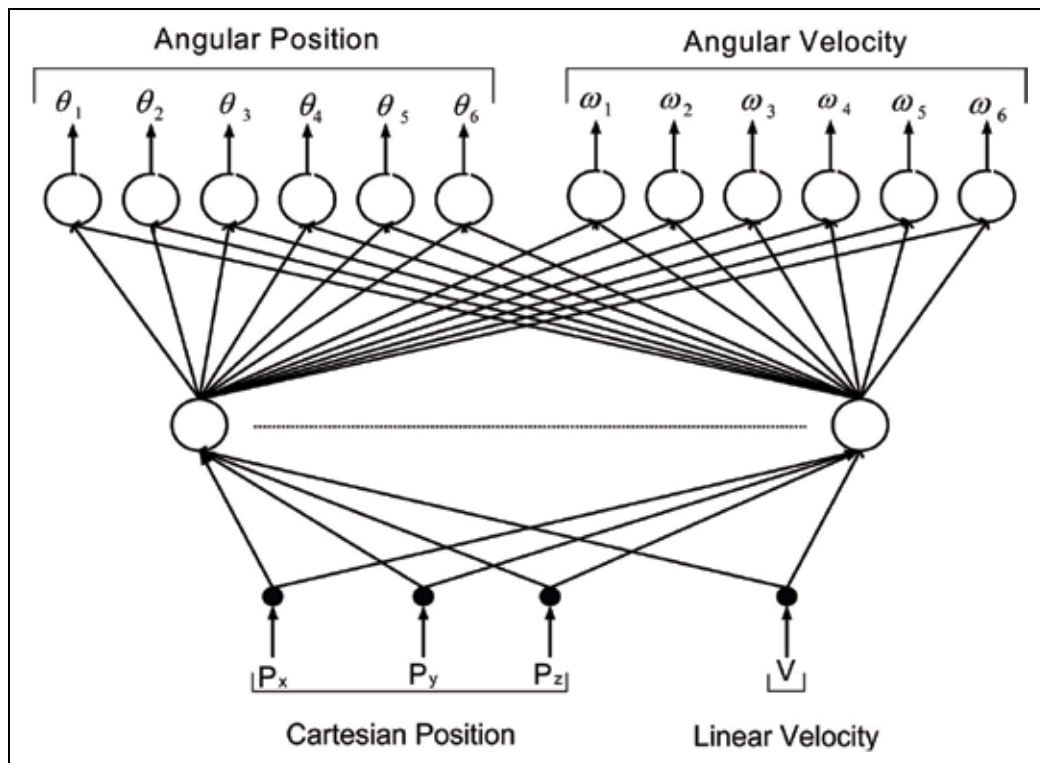


Fig. 7. The Topology of the First Network (4 -12 Network Configuration)

5.1.1.2 Second Configuration (7 - 12 Network Configuration)

In this configuration, the input layer has 7 neurons; the first three of them represent the X, Y and Z coordinates of the robot along the world coordinates system, the next three represent the orientation of the tool mounted on the last joint of the robot according to the RPY (Roll, Pitch, Yaw) representation, while the last neuron represents the linear velocity of the end-effector; as can be seen in figure 8.

Same as the first configuration, the output layer consists of 12 neurons; the first 6 of them represent the angular position of the robot joints while the last 6 of them represent the angular velocity of each joint respectively.

Number of neurons in the hidden layer was set to 55 with a constant learning factor of 0.9 by trail and error.

5.1.2 Networks' performance

The success of the ANN approach is measured according to the training error (the difference between the desired and actual system outputs). In the Generalized Delta learning Rule GDR the system is modified following each iteration, which leads to the learning curves a sample of which is shown in Figure 9 of each network configuration compared to the other (the rest of the curves have a similar behavior), as this curve shows; error is reduced in subsequent trials.

Table 2, shows the error percentages of each of the six joints compared for each other in both network configurations after the training has finished after 150 000 iteration.

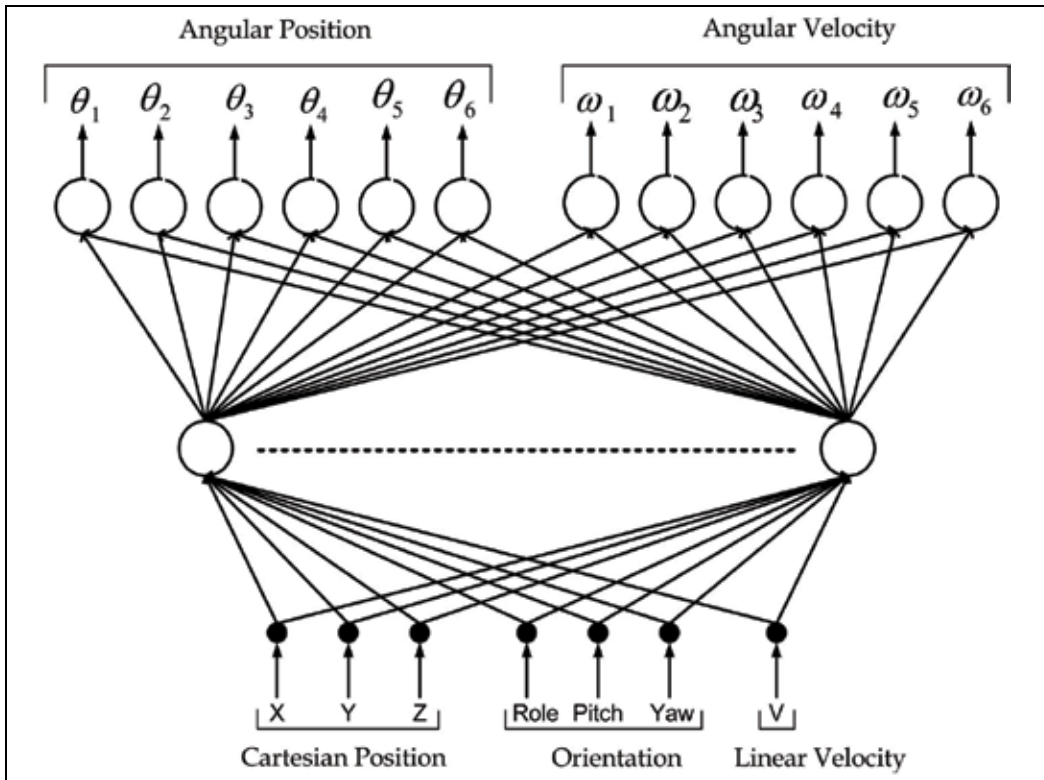


Fig. 8. The Topology of the Second Network (7 -12 Network Configuration)

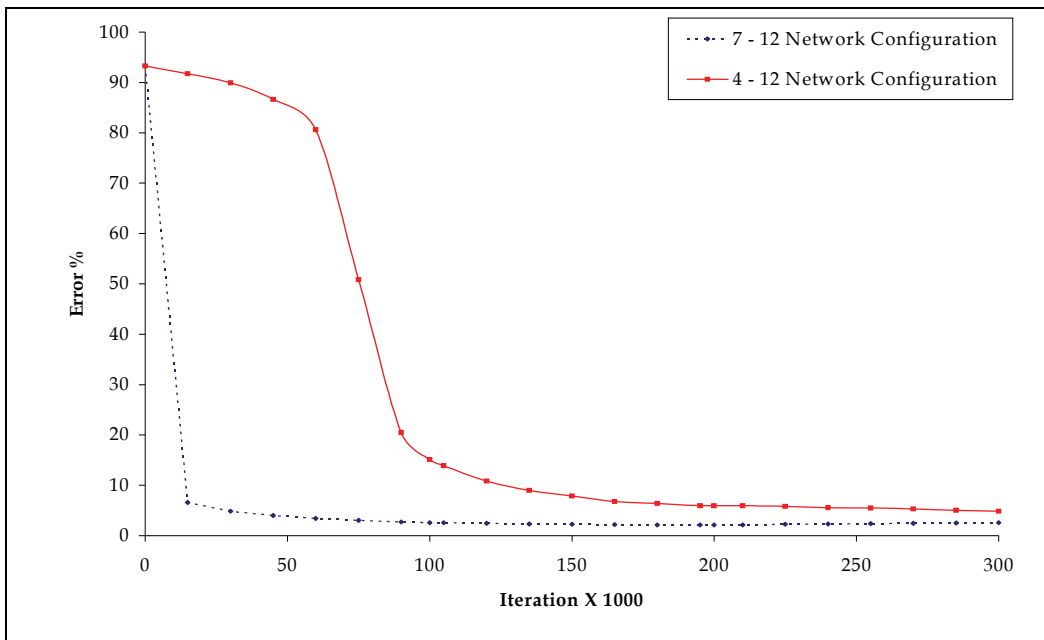


Fig. 9. The learning curve for the angular position of Joint 1 as a sample

		Network Configuration	
		4 - 12	7 - 12
Joint 1	θ	7.898%	2.27%
	ω	8.67%	1.8%
Joint 2	θ	12.432%	0.907%
	ω	39.75%	2.183%
Joint 3	θ	2.607%	1.033%
	ω	5.03%	1.775%
Joint 4	θ	9.82%	2.015%
	ω	10.4%	2.342%
Joint 5	θ	8.47%	4.435%
	ω	19.94%	1.558%
Joint 6	θ	10.86%	1.143%
	ω	5.735%	1.528%

Table 2. Training error percentages of each of the six joints compared for each other in both network configurations

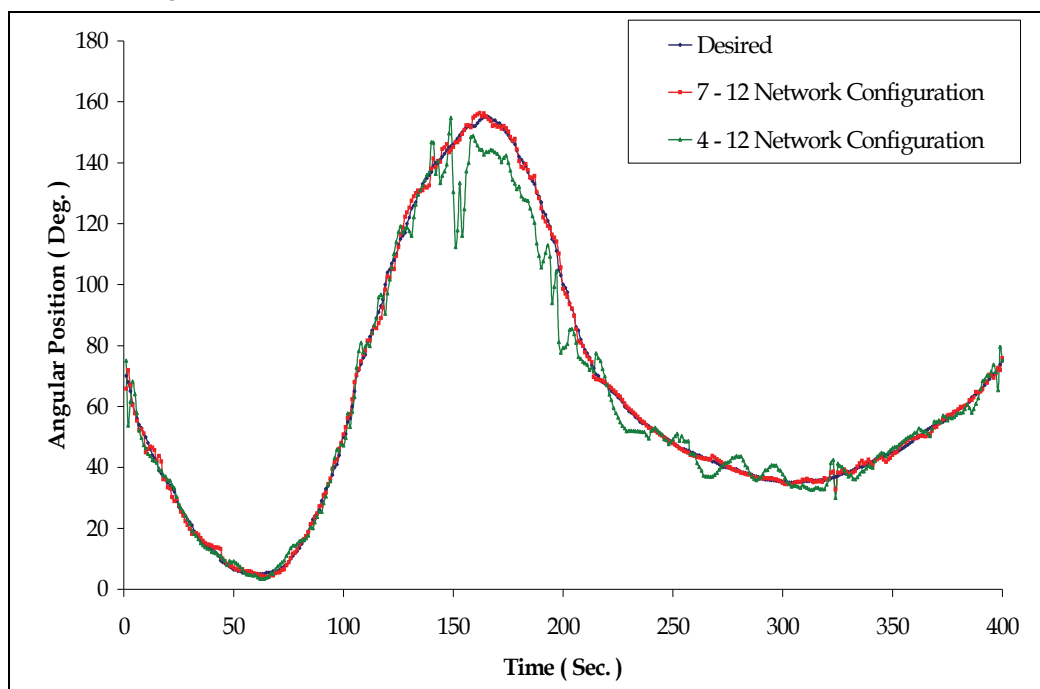


Fig. 10. The response of both network configurations compared to each other during training (The angular position of the first joint as an example)

As a result for the training stage, 7-12 network configuration has shown a better response than the 4-12 network configuration, in terms of precision and iteration (as can be seen through Table 2). Therefore, it has been chosen to apply the testing data.

To drive the robot to follow a desired path, it will be necessary to divide this path into small segments, and to move the robot through all intermediate points. To accomplish this task, at each intermediate location, the robot's trajectory equations are solved, a set of joint variables

is calculated, and the controller is directed to drive the robot to the next segment. When all segments are completed, the robot will be at the end point as desired. Figure 10 shows a sample of angular position and velocity for each joint during training (other joints have a similar behavior).

5.2 Testing phase

New data that has never been introduced to the network before have been fed to the trained network in order to test its ability to make prediction and generalization to any set of data later overcoming the singularity and uncertainty in the arm configuration resulting from applying the robot model.

Testing data were meant to pass nearby and through the singular configurations (Fourth and Fifth joints), these configurations have been determined by setting the determinant of the Jacobian matrix to zero.

Table 3 shows the percentages of error for the testing data set for each joint.

In order to verify the testing results, experiment has been performed to make sure that the output is the same or sufficiently close to the desired trajectory, and to show the combined effect of error, Figures 11 to 16 show the tracking of the Cartesian paths for the X, Y, and Z coordinates with the Roll, Pitch and Yaw orientation angles respectively.

The locus of which robot is passing through singular configurations are also shown. The error percentages in the experimental data are shown in Table 4.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Angular Position	0.06%	0.029%	0.039%	5.865%	5.065%	1.495%
Angular Velocity	3.79%	4.285%	3.745%	3.085%	4.97%	2.1%

Table 3. Error percentages for the testing data set for each joint

Cartesian Position			Orientation		
P_x	P_y	P_z	Roll	Pitch	Yaw
5.645%	1.09%	3.93%	5.95%	9.24%	5.338%

Table 4. Error percentages in the experimental data

6. Conclusions

In this approach, ANN technique has been used. The Jacobian inverse is now learned through training the network based only on observation of the input-output relationship unlike most other control schemes, which depends on the robot system model

The proposed technique does not require any prior knowledge of the kinematics model of the system being controlled, the basic idea of this concept is the use of the ANN to learn the characteristics of the robot system rather than to specify explicit robot system model.

Two different ANN configurations were used in this study. Training results have shown a better response (in terms of precision and iteration) for the configuration where the orientation of the tool is considered as an input to the network, which makes it useful in applications where a relatively accurate, minimally complex, and cheaper configuration is required.

As a conclusion, this study shows that ANNs are applicable to the Kinematics Jacobian solution of serial robots. Since one of the most important issues in using ANNs is the selection of the appropriate type of network, for future research, we suggest that different

types of networks (different topology, different activation function, different learning mode) to be used in order to get, if possible, more accurate trajectory tracking.

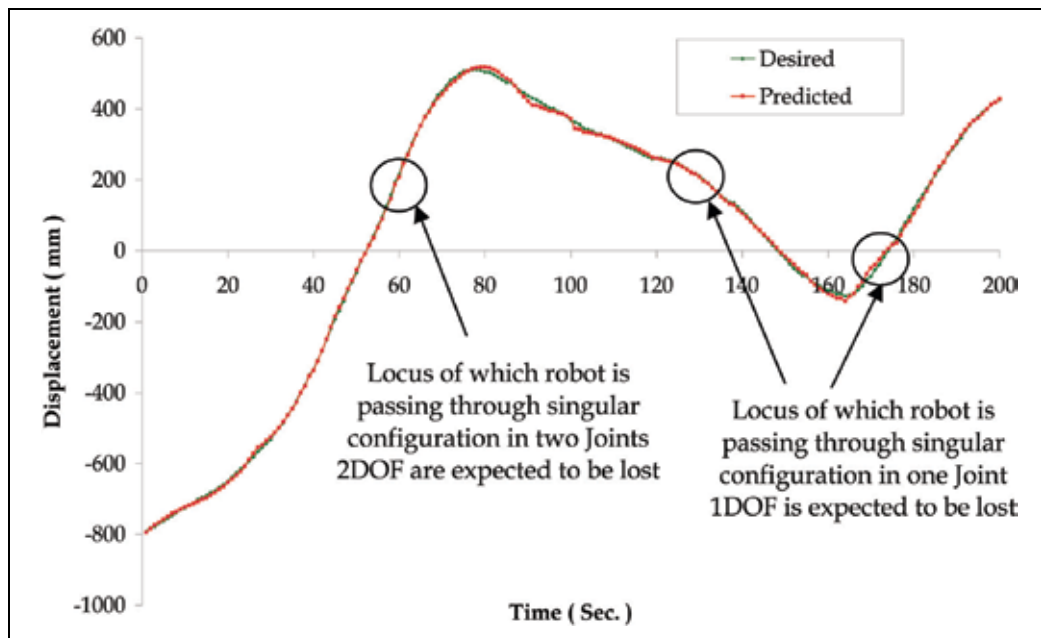


Fig. 11. Experimental trajectory tracking for the predicted X coordinate

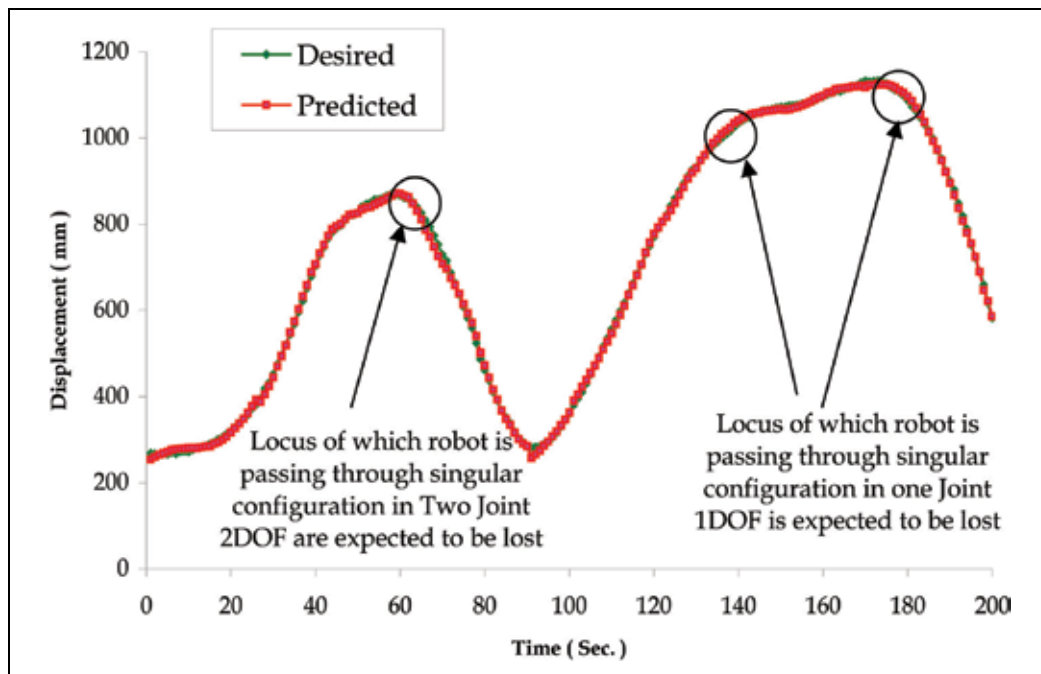


Fig. 12. Experimental trajectory tracking for the predicted Y coordinate

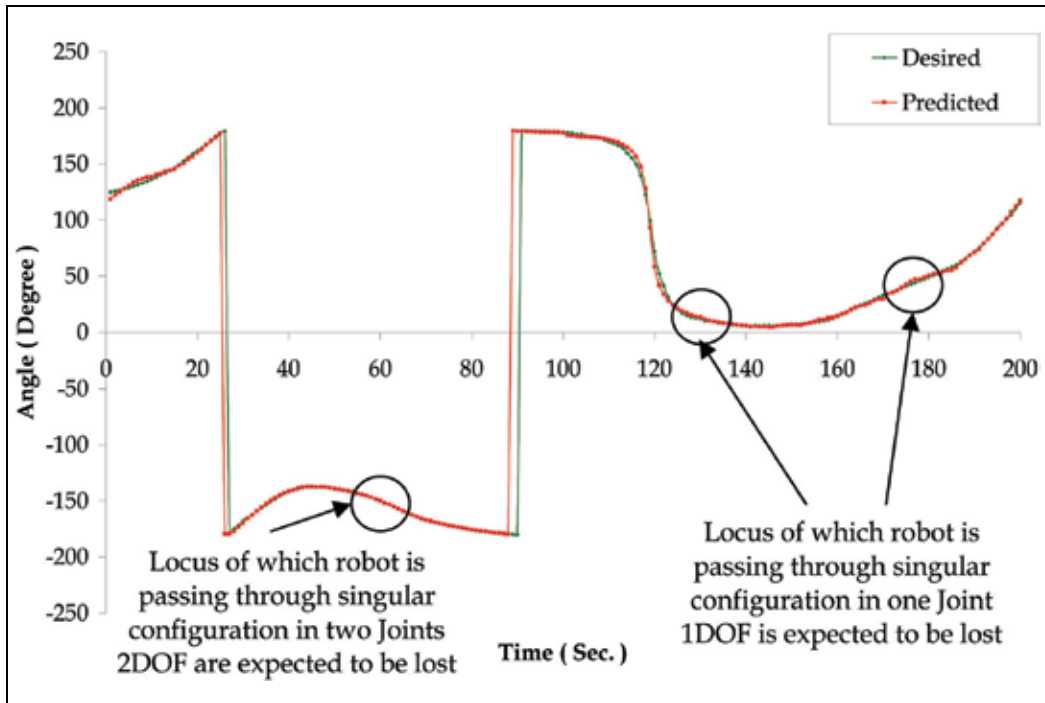


Fig. 13. Experimental trajectory tracking for the predicted Z coordinate

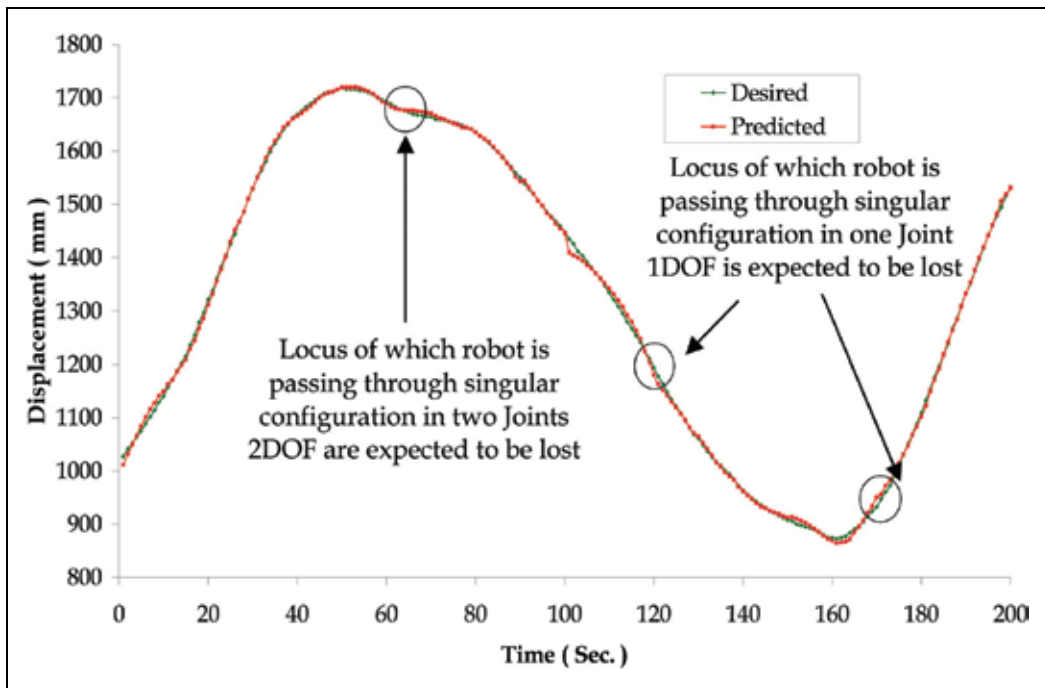


Fig. 14. Experimental trajectory tracking for the Roll orientation angle

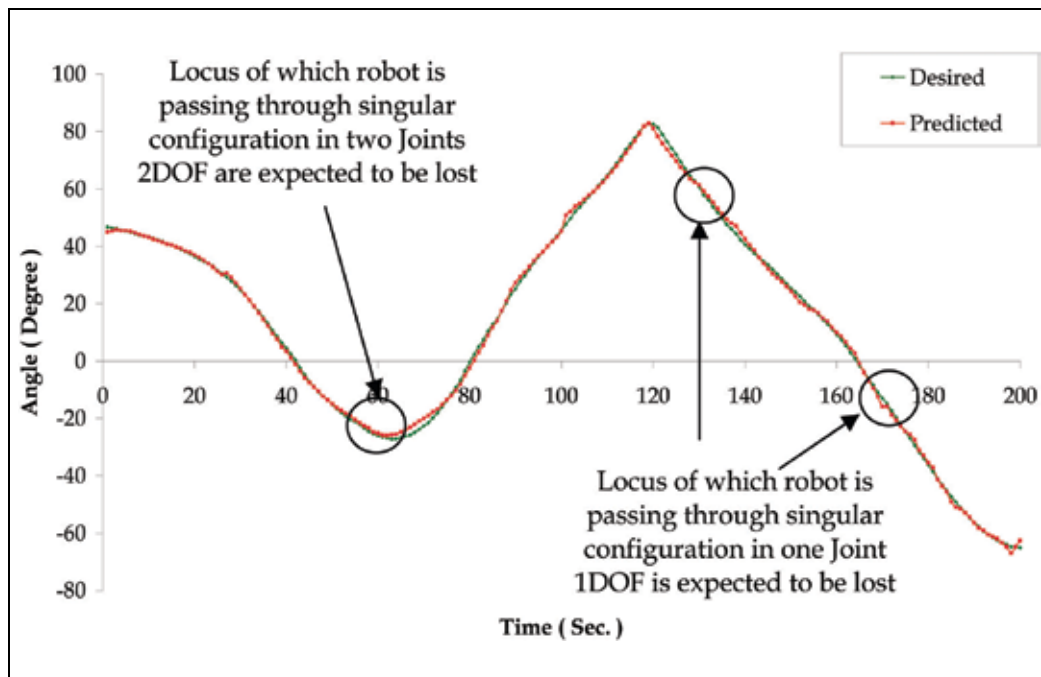


Fig. 15. Experimental trajectory tracking for the Pitch orientation angle

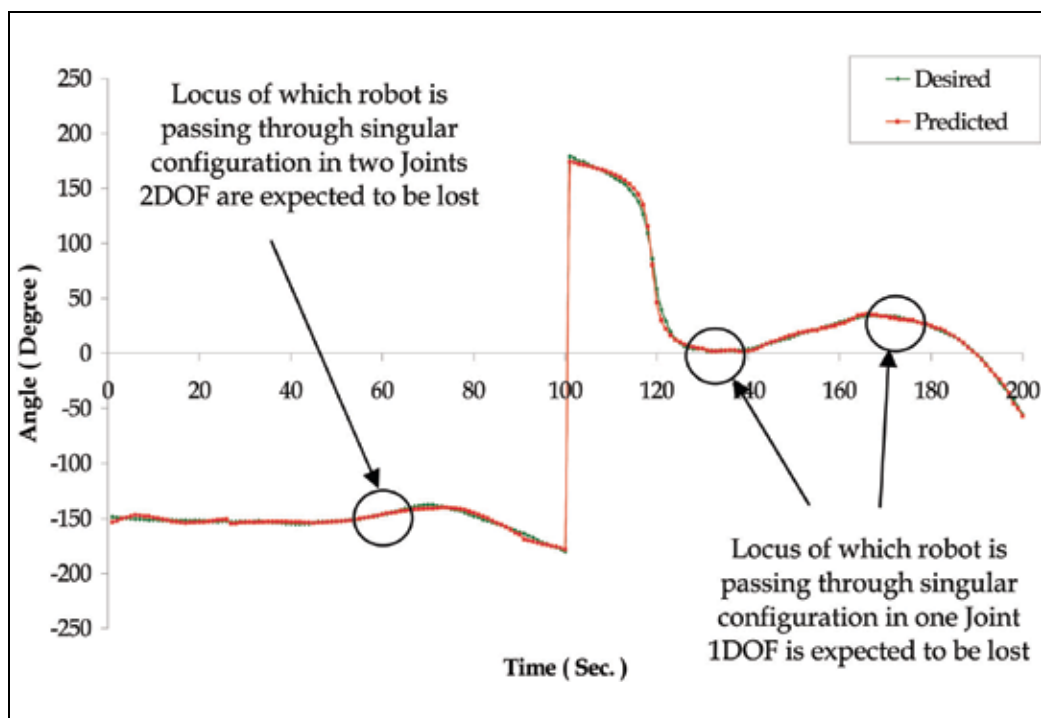


Fig. 16. Experimental trajectory tracking for the Yaw orientation angle

7. References

- Al-Assadi, H.M.A.A.; Hamouda, A.M.S.; Ismail, N. and Aris, I. (2007) .An adaptive learning algorithm for controlling a two-degree-of-freedom serial ball-and-socket actuator. *Proceedings of the I MECH E Part I Journal of Systems & Control Engineering*, Vol.221, No. 7, pp.1001-1006.
- Albala, H. & Angeles, J. (1979). Numerical solution to the input-output displacement equation of the general 7R spatial mechanism. *Proceedings of the Fifth world congress on theory of machines and mechanisms*, pp 8–11.
- Antonelli, G.; Chiaverini, S. and Fusco, G. (2003) .A new on-line algorithm for inverse kinematics of robot manipulators ensuring path-tracking capability under joint limits. *IEEE Transaction on Robotics and Automation*, Vol.19, No.1, pp. 162-167.
- Asada, H. & Soltin, J-J. E., Robot analysis and control. John Wiley and Sons Inc., New York. 1986.
- Balestrino, A., De Maria, G. and Sciavicco, L., Robust control of robotic manipulators. International Proceedings of the 9th IFAC World Congr. Budapest, Hungary .1984; 6:80-85.
- Bingual, Z., Ertunc, H.M. and Oysu, C., Comparison of Inverse Kinematics Solutions Using Neural Network for 6R Robot Manipulator with Offset. 2005 ICSC congress on Computational Intelligence.
- Daniel, M. and Raul, G., Hierarchical Kinematics analysis of robots. *Mech Mach Theory*. 2003; 33: 497–518.
- Denavit, J., and Hertenberg, R.S.A. Kinematics Notation for lower Pair Mechanism Based on Matrices. *Applied mechanics* 1955; 77: 215-221.
- Driscoll, J.A., Comparison of neural network architectures for the modeling of robot inverse kinematics. *Proceedings of the IEEE*, south astcon. 2000:44-51.
- D'Souza, A., Vijayakumar, S., and Schaal, S. (2001) 'Learning Inverse Kinematics'. *Proceedings of the 2001 IEEE/ RSJ International Conference on Intelligent Robots and Systems Maui, Haw- USA*, pp.298-303.
- Duffy, J. and Rooney, J., A foundation for a unified theory of analysis of spatial mechanism. *Eng Ind*. 1975; 97(4): 1159–64.
- Duleba, I., and Sasiadek, J.Z. (2000) 'Modified Jacobian method of transversal passing through singularities of nonredundant manipulators'. *Proceedings of the American Control Conference Chicago, Illinois June*, pp.2839-2843.
- Faiz, N. and Agrawal, S. K., Trajectory planning of robots with dynamics and inequalities. In *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA.2000: 3977–3983.
- Fu, K.S., Gonzalez, R.C. and Lee, C.S.G. *Robotics Control, Vision, and Intelligence*. McGraw-Hill book Co. Singapore, international edition. 1987.
- Funahashi, K.I., On the approximate realization of continuous mapping by neural networks. *Neural Networks*.1998; 2(3): 183-192.
- Graca, R.A. and Gu, Y., A fuzzy learning algorithm for kinematics control of a robotic system. *Proceeding of the 32nd conference on decision and control*. San Antonio, Texas. December 1993:1274-1279.

- Hasan, A.T., Hamouda, A.M.S., Ismail, N., and Al-Assadi, H.M.A.A. (2006) 'An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator'. *Int. J. Advances in Engineering Software*, Vol.37, pp. 432-438.
- Hasan, A.T., Hamouda, A.M.S., Ismail, N., and Al-Assadi, H.M.A.A., A new adaptive learning algorithm for robot manipulator control. *Proceeding of the I Mech E, Part I: Journal of System and Control Engineering*, .2007; 221(4): 663-672.
- Haykin S. *Neural Networks. A Comprehensive Foundation*. New York: Macmillan, 1994.
- Hornik, K., Approximation capabilities of multi-layer feed forward networks. *IEEE Trans. Neural Networks*. 1991; 4(2): 251-257.
- Karilk, B., Aydin, S., An improved approach to the solution of inverse kinematics problems for robot manipulators. *Journal of Engineering applications of artificial intelligence*. 2000; 13: 159-164.
- Köker, R. (2005) 'Reliability-based approach to the inverse kinematics solution of robots using Elman's networks'. *Int. J. Engineering Applications of Artificial Intelligence*, Vol.18, pp. 685-693.
- Köker, R., Öz, C., Çakar.T. and Ekiz, H., A study of neural network based inverse kinematics solution for a three-joint robot. *Robotics and Autonomous Systems*. 2004; 49: 227-234.
- Kuroe, Y., Nakai, Y. and Mori, T., A new Neural Network Learning on Inverse Kinematics of Robot Manipulators. *International Conference on Neural Networks, IEEE world congress on computational Intelligence*. 1994; 5: 2819-2824.
- Nakamura, Y. and Hanafusa, H., Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems Measurements Control*.1986; 108: 163-171.
- Ogawa, T., Matsuura, H., and Kanada, H. (2005) 'A Solution of Inverse Kinematics of Robot Arm Using Network Inversion'. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*.
- Santosh, A. Devendra P. Garg. Training back propagation and CMAC neural networks for control of a SCARA robot. *Engineering Applications of Artificial Intelligence*. Vol.6.No.2. pp.105-115. 1993.
- Spong, M.W., and Vinyasagar, M., *Robot Dynamics and control*. John Wiley and Sons Inc., New York. 1998.
- Soteris, A.Kalogirou. *Artificial Neural Networks In Renewable Energy Systems Applications: a review*. *Renewable and Sustainable Energy Reviews*. Vol. 5:pp.373-401. 2001.
- Tsai, L.W. and Morgan, A., Solving the Kinematics of the most general six and five degree of freedom manipulators by continuation methods. *Mech Transm Autom Des* 1985; 107:189-200.
- Wampler, C. W. and Leifer, L. J., Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators, *Journal of Dynamic Systems Measurements Control*.1988; 110: 31-38.
- Wampler, C. W., Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transaction Syst., Man, Cybernetics*. 1986; 16: 93-101.

- Whitney. E., Resolved motion rate control of manipulators and human prostheses, IEEE Transaction Man-Mach. Systems.1969; 10:47-53.
- Yang, A.T., Displacement analysis of spatial five-link mechanism using (3×3) matrices with dual-number element. Eng Ind. 1969; 9(1): 152-7.
- Zurada, M. J. (1992). *Introduction to Artificial Neural System Network*. West Publishing Companies, ISBN 0-314-93397-3, St. Paul, MN, USA.

Development of Fuzzy-logic-based Self Tuning PI Controller for Servomotor

Oyas Wahyunggoro and Nordin Saad
*Department of Electrical and Electronics Engineering,
 Universiti Teknologi
 PETRONAS
 Malaysia*

1. Introduction

Servomotors are used in a variety of applications in industrial electronics and robotics that includes precision positioning as well as speed control. Basically, any motor can be used in a servo system (Kissell, 2002). There are two types of motors: (a) DC motors, and (b) AC motors. DC motors have better starting torque than AC motors although they are more expensive than AC motors. Servomotors use feedback controller to control the speed or the position, or both. The basic continuous feedback control is PID controller. The PID controller possesses good performance but is not adaptive enough. This is appealing when the load is changed, where the original controller generally cannot maintain the design performance and thus should be re-designed for the new system conditions (Shieh & Li, 1998).

The pioneering work dealing with expert knowledge that can be well applied to the control of systems with uncertain, nonlinear dynamics is credited to Zadeh (Zadeh, 1968) who proposed fuzzy control theory to overcome the weakness of conventional controllers, and investigated by which owns good robustness (Yu *et. al.*, 2004). Fuzzy systems are capable of handling complex, non-linear and sometimes mathematically intangible dynamic systems using simple solutions. Fuzzy logic uses human-like but systematic properties of converting linguistic control rules based on expert knowledge into automatic control strategies. But, the response of a fuzzy logic controller is slower than a PID controller. It has been reported in a number of papers that hybrid of PID or PI, with fuzzy logic in control system can overcome the set-back of fuzzy logic controller, see (Yeh & Tsao, 1994), (He *et. al.*, 1993), (Ga & Feng, 2005) and (Jee & Koren, 2004). In fuzzy systems, the numerical input values should be first converted into the corresponding fuzzy representations by using 'fuzzifiers'. The fuzzy output are then provided by a fuzzy model, which could be a set of fuzzy logic rules, fuzzy relations or even a simple fuzzy table, with or without deep fuzzy reasoning. Finally, the fuzzy output can be converted back into their relevant numerical (crisp) output through 'defuzzifiers' (Lu, 1996). Basic configuration of fuzzy systems with fuzzifier and defuzzifier is shown in Figure 1.

Singleton fuzzifier maps a real-valued point $x^* \in U$ into a fuzzy singleton A' in U , which has membership value 1 at x^* and 0 at all other points in U (Wang, 1997). If \bar{y}_l is the center of the l 'th fuzzy set and w_l is its height, then the center of average defuzzifier determines y^* as (Wang, 1997).

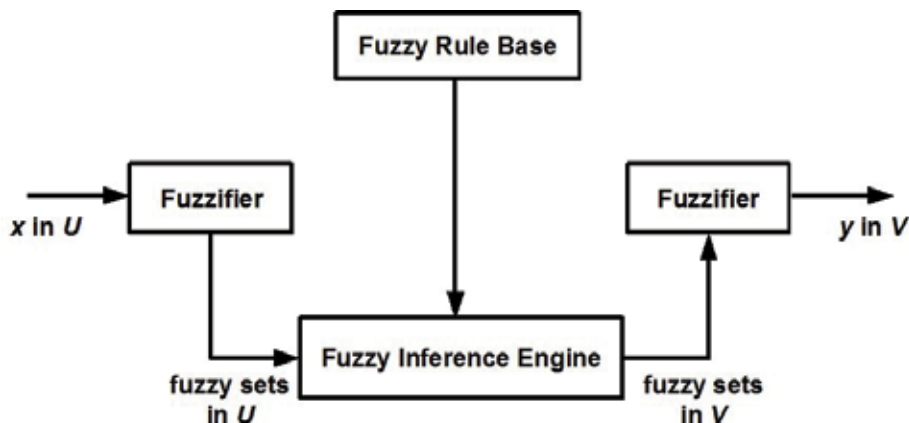


Fig. 1. Basic configuration of fuzzy systems with fuzzifier and defuzzifier (Wang, 1997)

$$y^* = \frac{\sum_{l=1}^M y_l w_l}{\sum_{l=1}^M w_l} \quad (1)$$

A study on induction machine speed control proved that PID controller's parameters can be tuned on-line by an adaptive mechanism based on a fuzzy logic (Bousserhanel *et. al.*, 2006). It is expected that hybrid of PID and fuzzy logic in control system can overcome the problem of fuzzy logic controller.

Servomotor controllers need optimization to give a good performance as desired. Sometimes, it requires time and has big risk in the optimization process. To investigate this issue, a detail study on servomotor control is conducted. A real plant, constituting of a DC motor and its controller has been built and this system is modeled and simulated to allow detail analysis about its control system. The system identification problem is to estimate a model of a system based on the observed input-output data (Takami & Mahmoudi, 2007). The expected output of identification is an s-domain model of a real system. System identification technique is used to get a transfer function of the plant and is used to build a virtual controller, which is basically the software equivalent of the real controller at the abstraction level. The virtual controller is optimized, and then the optimized parameters are applied to the real controller in the real control system.

Therefore, this work discusses the modeling, simulation and hardware implementation of a DC servomotor controller built using MATLAB/Simulink, and the analysis of controller's performance, namely a PID controller, PI controller, fuzzy logic controller, fuzzy-logic-based self tuning PI controller, and a fuzzy-logic-based self tuning PID controller on the system. In this work the aim is to improve the controller's performance using hybrid fuzzy and PID. Figure 2 outlines the background and purpose of this work.

2. The experimental DC servomotor system

The experimental rig constituting the servo speed and position controller system consists of the servomotor and load, measuring and controlling devices. The servo system contains a DC motor driven by an IGBT chopper inverter. The measuring device is the speed sensor (tachogenerator), ADC and a digital filter i.e., finite input response (FIR), while the controlling

devices are DAC, differential amplifier, and the IGBT inverter circuit. The measuring devices provide status of the output responses of the speed and position where the information about the speed and position is fed through signal conditioning circuit and anti-aliasing filter for analysis and calculation of the control signal. The speed and position requirements proportional to the manipulated variable of the controller's output are fed to a computer.

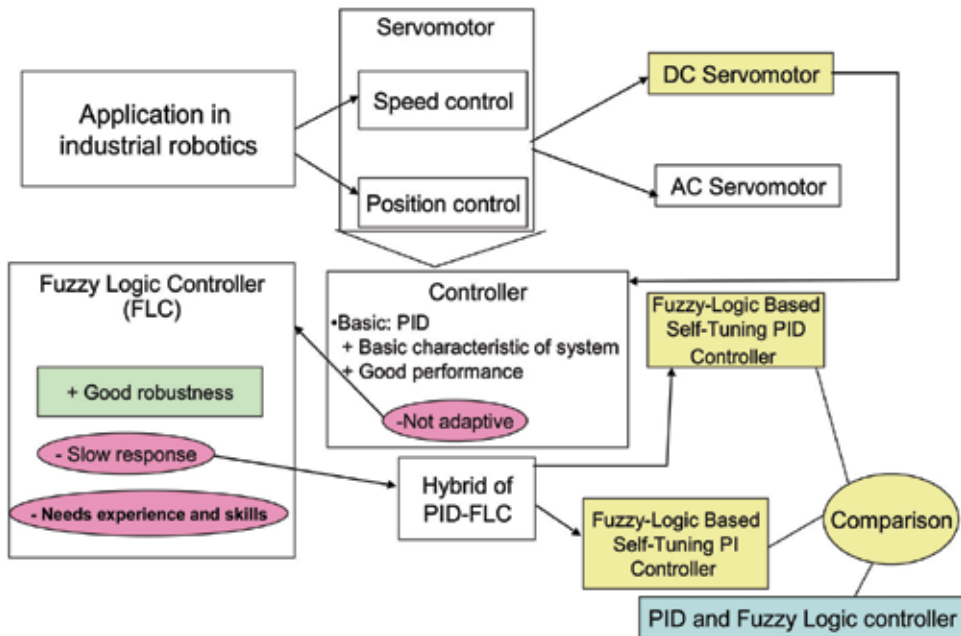


Fig. 2. Overview of the study background

The modeling and analysis of the DC servomotor for the speed and position was conducted using Simulink, with the block diagram is as shown in Figure 3. In this arrangement, the speed control loop is in the position control loop (Lacevic *et. al.*, 2007). Basically, the position control loop work until reaching the position set point while the speed is limited to the speed set point. Position control systems with conventional position sensor have well-known drawbacks. Notably, friction, stiction, backlash, and hysteresis limit the positioning accuracy. This would lead to mechanical wear that introduces impurities in the form of dust particles into the manufacturing environment. The other consequence would be mechanical coupling transmits to the microstepper vibrations from the surrounding environment, (Fulford *et. al.*, 2009). Notably, in this experimental set-up the position control is sensorless. Practically, position control is preferred to be sensorless to reduce cost and size and increase the reliability of overall system (Montanari *et. al.*, 2007).

The input, feedback, and output elements for position and speed are implemented in the Simulink diagram with specifications as follows:

- $A_p=1$
- $A_v=0.002$
- $K_v=9.5455$
- $H_v=0.002$
- $H_p=0.005$
- $K_p=0.005$

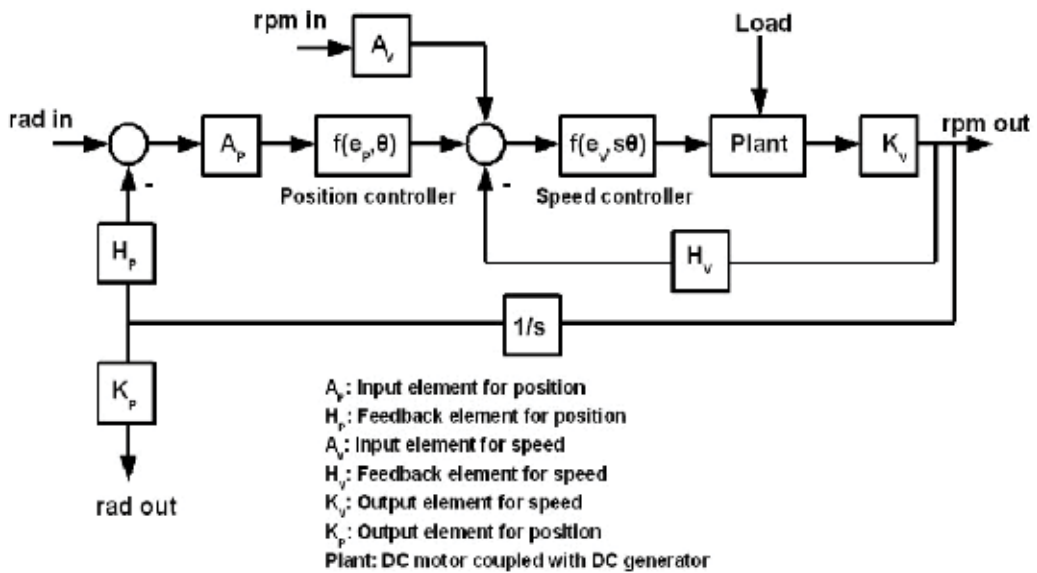


Fig. 3. Structure of feedback controller (Lacevic *et. al.*, 2007)

These specifications have been based on realistic assumptions.

The hardware implementation for the block diagram is as follows:

DC motor: 175 W, 1500 rpm, 240 V, 1.1 A

Load: Dynamometer load controller

Input elements:

Tacho-generator with 500 rpm/volt

ADC: 1 channel 0 to 10 volts

Filtering: FIR with 30 points

Output elements:

DAC: 2 channels 0 to 4 volts

Differential amp: HA-17741

Power amp: Chopper/Inverter and IGBT

Controller elements:

Computer with Intel Pentium Dual Core T2080 processor, Windows XP SP3, MATLAB/SIMULINK software.

The block diagram of hardware design is shown in Figure 4, and the power amplifier and differential amplifier are presented more detail in Figures 5 and 6.

As an illustration to substantiate the applicability of this approach, the following test parameters have been chosen. The set point for position is 5 rad, and the set point for speed equals 250 rpm. The load is initially at 0 Nm and then abruptly increases to 1 Nm after about 15 sec. The total duration of experiment is 90 sec. The speed of DC motor is detected by a tacho-generator, sampled every 0.01 sec, and filtered by 30-point FIR.

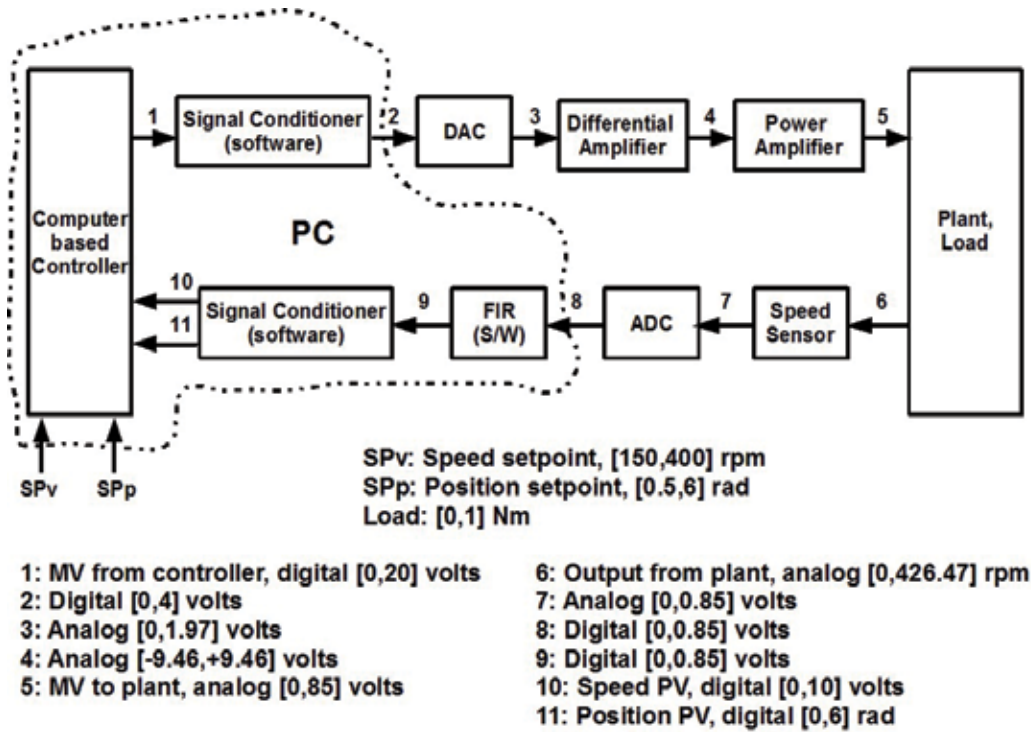


Fig. 4. Block diagram of hardware design

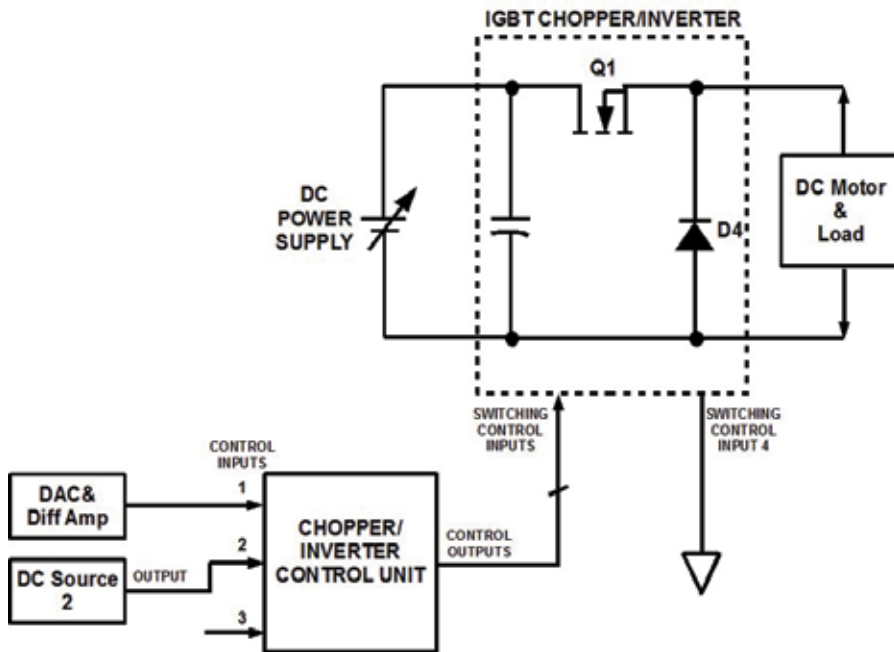


Fig. 5. Power amplifier circuit diagram

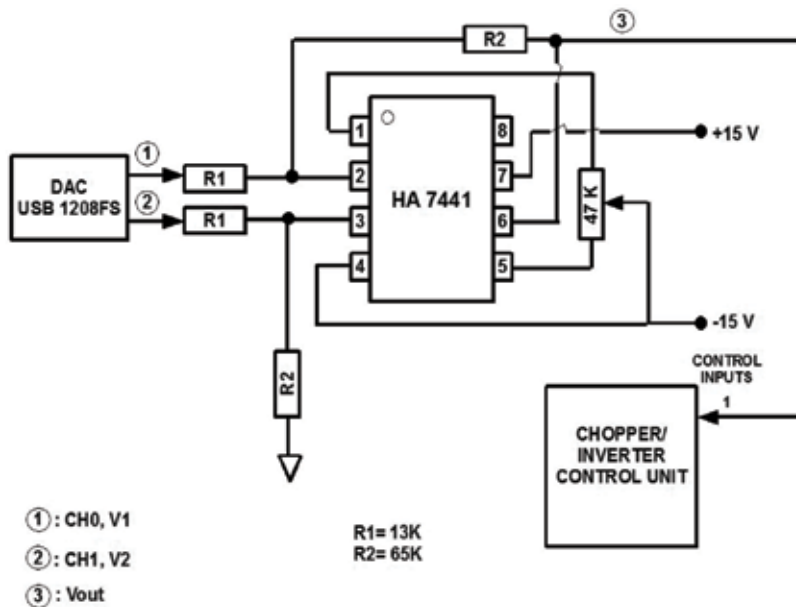


Fig. 6. Differential amplifier circuit diagram

3. Experimental procedures

3.1 The identification process

The typical identification process consists of stages where the model structure is iteratively selected, computed and updated for the best model in the structure, and finally the evaluation of the model's properties (Takami & Mahmoudi, 2007). The steps can be itemized as follows:

- Design an experiment and collect input-output data from the process to be identified.
- Examine the data. Polish the data by removing trends and outliers, and select useful portions of the original data. Apply filters to the data to enhance important frequency ranges.
- Select and define a model structure (a set of candidate system descriptions), within which a model is to be found.
- Compute the best model in the model structure according to the input-output data and a given criterion for goodness of fit.
- Examine the properties of the model obtained. If the model is good enough, then stop; otherwise go back to step c to try another model structure. Attempt other estimation methods (step 4), or work further on the input-output data (steps a and b).

3.1.1 Process I: Open loop analysis

Open loop characteristic of the gray box is tested using Simulink, which is applied to the plant through a data acquisition (DAQ), 30-point FIR, chopper/inverter control unit, and IGBT. The input voltage is varied randomly in the range of 0 to 20 volts for 100 seconds. The output of the gray box model is in rad/sec. There are five variations of input, and the characteristic of one of the input-output is shown in Figure 7.

3.1.2 Process II: Best model selection

There are six process models that can represent the system. To select the most suitable model, these models will be analysed and compared:

- Process model using 1st order transfer function without zero,
- Process model using 1st order transfer function with zero,
- Process model using 2nd order transfer function without zero,
- Process model using 2nd order transfer function with zero,
- Process model using 3rd order transfer function without zero, and
- Process model using 3rd order transfer function with zero.

The DC motor is operated for 150 sec with variations of input voltages as shown in Figure 7. The sampling time is 0.01 sec. The result (output voltage) and the input are then saved in workspace. The data for identification is obtained from the MATLAB using the command as below:

```
data=iddata(workspace_output,workspace_input,0.01);
```

Using MATLAB command, the general command to get the process model is as follows:

```
model=pem(data, 'PnDZU');
```

where:

- P : (required) for process model
- n : 0,1,2, or 3 (required) for the number of poles
- D : (optional) to include a time-delay term
- Z : (optional) to include a process zero (numerator term)
- U : (optional) to indicate possible complex-valued (underdamped) poles

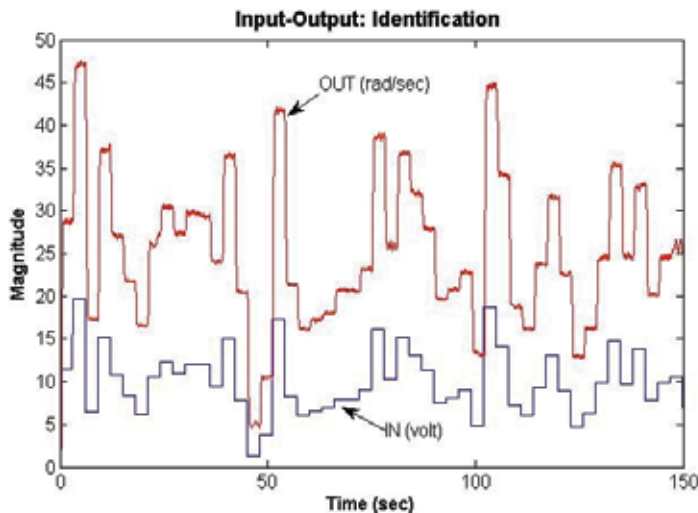


Fig. 7. Graphical process of identification

To select the best process model, the fitness value has to be obtained. If i_FIT is the fitness value for identification process, Y is the real (measured) output, \hat{Y} is the estimated output,

and \bar{Y} is the mean value of real output, then the fitness value can be obtained from the formula below, see (Montanari *et. al.*, 2007):

$$i_FIT = \left(\frac{1 - NORM(Y - \hat{Y})}{NORM(Y - \bar{Y})} \right) * 100 \quad (2)$$

3.2 The controllers

3.2.1 Conventional controllers

In this work the following two conventional controllers, namely the PI and PID are used. Each controller is optimized using Ziegler-Nichols (ultimate cycle) method. If K_{pu} is the minimum value of K_P resulting undamped oscillation and T_u is the oscillation period, then

$$\begin{aligned} K_p &= 0.45 * K_{pu} \\ K_I &= \frac{1}{0.83 * T_u} \end{aligned} \quad (2a)$$

for the PI controller, and

$$\begin{aligned} K_p &= 0.6 * K_{pu} \\ K_D &= \frac{T_u}{8} \\ K_I &= \frac{2}{T_u} \end{aligned} \quad (2b)$$

for the PID controller.

3.2.2 Fuzzy logic controllers

In this work, a fuzzifier is a singleton mode with two inputs consisting of error and change of error, each with seven uniform triangular membership functions, and one output with seven triangular membership functions. As an inference engine, the Mamdani product is used and as a defuzzifier, the center of average is used. The controllers use the rules as shown in Table I.

E, error	D, change of error						
	NB	NM	NS	Z	PS	PM	PB
NB	NB	NM	NM	NM	NM	NM	NM
NM	NM	NM	NM	NM	NM	NM	NM
NS	NS	NS	NS	NS	NS	NS	NS
Z	Z	Z	Z	Z	PS	PM	PB
PS	Z	PS	PM	PB	PB	PB	PB
PM	Z	PS	PM	PB	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

Table I. The rules for FLC

NB: Negative Big; NM: Negative Medium; NS: Negative Small Z: Zero; PS: Positive Small; PM: Positive Medium; PB: Positive Big

The fuzzy sets and their corresponding membership functions for input (error, and change of error) are shown in Figure 8, and for output are shown in Figure 9.

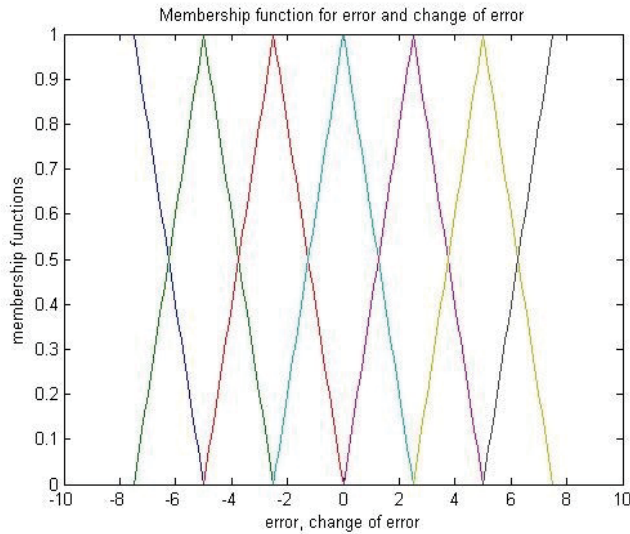


Fig. 8. Fuzzy sets for input

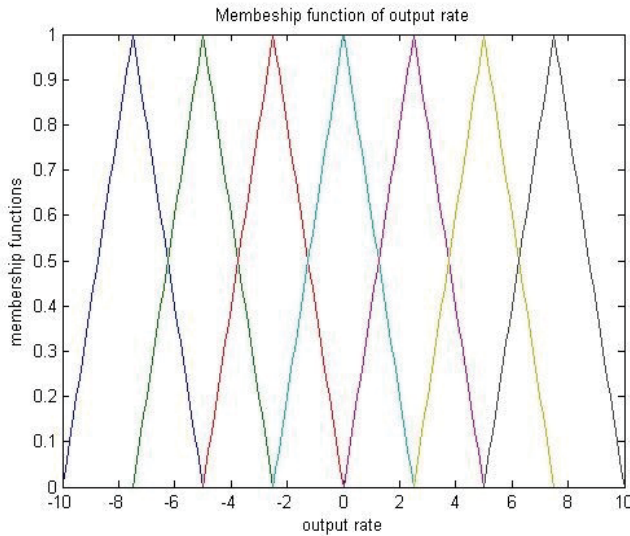


Fig. 9. Fuzzy sets for output rate

3.2.3 Hybrid controllers

There are two kinds of hybrid controllers that be the focus of this work: FLBPI and FLBPID. FLBPID uses the basis of FLBPI with additional of fixed value of K_D . The block diagram of fuzzy-logic-based self tuning PI for speed controller is shown in Figure 10. In this figure, $\omega_m(k)$ is the process value of speed and $\omega_m^*(k)$ is the setpoint of speed.

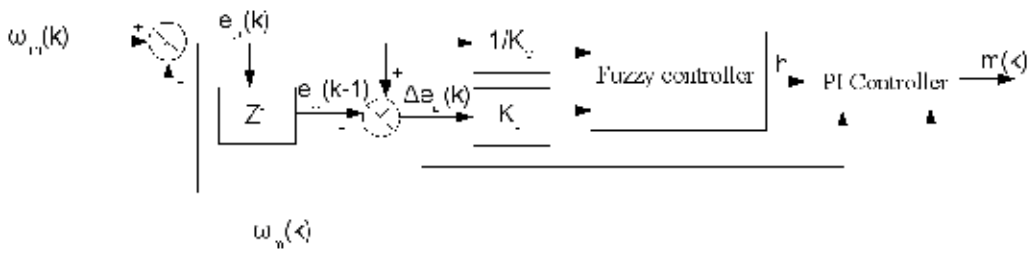


Fig. 10. Block diagram of fuzzy-logic-based self-tuning PI for the speed controller [11]

The fuzzy sets and their corresponding membership functions for input (error, $e_ω(k)$ and change of error, $Δe_ω(k)$) and output (h) are shown in Figure 11. The rules for FLBPI and FLBPID (K_P and K_I) are shown in Table II. The value of K_D is constant based on Ziegler-Nichols method in PID tuning.

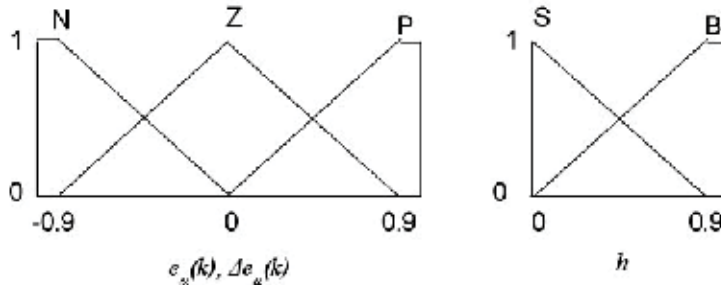


Fig. 11. Fuzzy sets and their corresponding membership functions, (Mannan *et. al.*, 2004).

Change of Error	Error		
	N	Z	P
N	S	B	S
Z	S	B	S
P	S	B	S

N: Negative, Z: Zero, P: Positive, S: Small, B: Big

Table II. Fuzzy Rules Base for K_P and K_I in Hybrid (Mannan *et. al.*, 2004).

The values of K_P , K_I , and K_D for hybrid controllers are obtained from the following formula:

$$K_p = h.K_{pm}; K_I = h^2.K_{Im}; K_D = K_{Dm} \tag{3}$$

Proportional based controller is used for the position control since the requirement is such that the motor would rotate in one direction. The proportional controller for position (K_{PP}) is tuned using experimental method with no overshoot criteria.

Integral of Absolute value of Error (IAE) is used as a performance index for overall speed control (IAE_v) in which the formula is as follows, see (Marlin, 2000):

$$IAE_v = \int_0^{90} |SP(t) - PV(t)| dt \tag{4}$$

Integral of Time Absolute value of Error (ITAE) is used as a performance index for position control ($ITAE_p$), the first 8-second of starting speed control ($ITAE_{vp}$), and the 9-second of loading speed control ($ITAE_{vpl}$) which the formulas are as follows, see (Marlin, 2000):

$$ITAE_{vp} = t \int_0^8 |SP(t) - PV(t)| dt \quad (5a)$$

$$ITAE_{vpl} = t \int_{14}^{23} |SP(t) - PV(t)| dt \quad (5b)$$

$$ITAE_p = t \int_0^{90} |SP(t) - PV(t)| dt \quad (5c)$$

and the fitness value is obtained from the basic formula as follows, see (Xiu & Ren, 2004):

$$f(i) = \frac{ITAE - ITAE(i)}{\sum_i ITAE_{\max} - ITAE(i)} \quad (6)$$

Based on Eq (6), the fitness values for overall speed (f_v), first 8-second starting speed (f_{vp}), 9-second loading speed (f_{vpl}) and position (f_p) are as follows:

$$f_v(i) = \frac{10(IAE_{v,\max} - IAE_v(i))}{\sum_{i=1}^5 (IAE_{v,\max} - IAE_v(i))} \quad (7a)$$

$$f_{vp}(i) = \frac{10(ITAE_{vp,\max} - ITAE_{vp}(i))}{\sum_{i=1}^5 (ITAE_{vp,\max} - ITAE_{vp}(i))} \quad (7b)$$

$$f_{vpl}(i) = \frac{10(ITAE_{vpl,\max} - ITAE_{vpl}(i))}{\sum_{i=1}^5 (ITAE_{vpl,\max} - ITAE_{vpl}(i))} \quad (7c)$$

$$f_p(i) = \frac{10(ITAE_{p,\max} - ITAE_p(i))}{\sum_{i=1}^5 (ITAE_{p,\max} - ITAE_p(i))} \quad (7d)$$

4. Experimental results

The comparison on the effectiveness of modeling the DC motor based on the fitness value is presented in Table III.

Selecting the best process model in Table III, the transfer function (s-domain model) of the plant based on the best model is as follows:

$$\frac{Speed_{OUT}(s)}{Voltage_{IN}(s)} = G(s) = \left[\frac{456.3713}{s^3 + 9.5040s^2 + 80.7000s + 204.5000} \right] e^{-0.1682s} \quad (8)$$

Process Model	Fitness	Remark
1	84.1416	
2	84.4121	
3	84.8531	
4	-34.0071	
5	86.5877	Best
6	75.5157	

Table III. Fitness value modelling the DC motor

Graphical comparison of actual (real time) response and s-model (estimated) response in open loop analysis is shown in Figure 12.

The PI, PID, FLBPI and FLBPID have parameters as follows:

$K_P=10.58$ and $K_I=1.47$ for PI controller

$K_P=14.10$, $K_D=0.1$, and $K_I=2.44$ for PID controller

$K_{Pm}=6$ and $K_{Im}=7.9$ for FLBPI controller (experiment)

$K_{Pm}=10.4$ (experiment), $K_{Dm}=0.1$ (Ziegler-Nichols), and $K_{Im}=7.2$ (experiment) for FLBPID controller

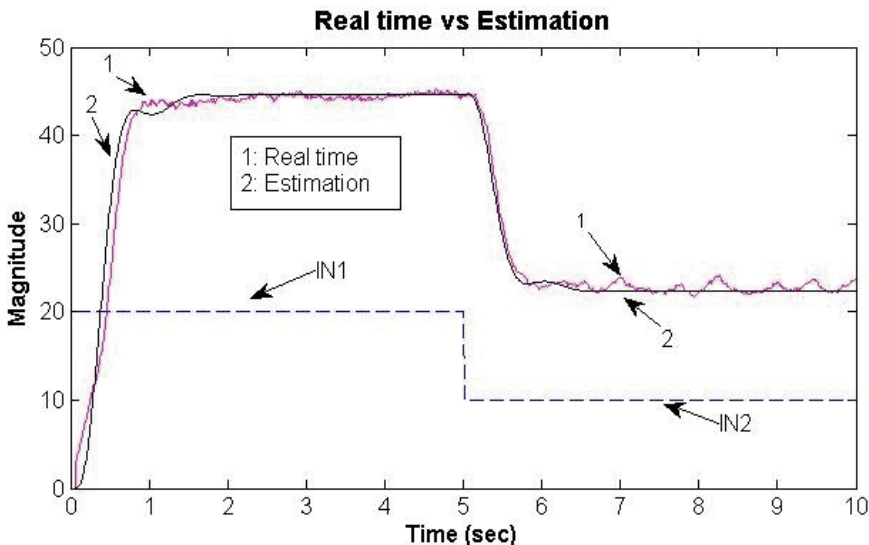


Fig. 12. Graphical comparison of actual (real time) and estimation (s-model)

The relevant parameters were obtained during the simulation (offline optimization) using the transfer function as shown in Eq (8). The comparison on the effectiveness of implementing FLBPI as compared to FLBPID, FLC, PID, and PI controller based on the performance metrics for real time experiment is presented in Table IV.

The ability to understand the influence of the different controllers on the servomotor speed and achieving its target position is one of the important aspects of this study. It is noted that the speed response with the FLBPID-type controller has a relatively good settling time, but

the fitness values for the motor during starting and loading is not good enough due to the fact that the noise is a bit larger as compared to FLBPI, albeit the fitness value of the overall speed for the FLBPI and the FLBPID are similar. A closer observation at the starting and loading performances during the real time experimentations, FLBPI performances is relatively better as compared to the PI, PID, FLBPID or FLC for speed control. A similar performance is also observed in the fitness value of position for real time experiment, where the FLBPI is relatively better as compared to the PI, PID, FLBPI or FLC for position control.

NO.	PERFORMANCE INDEX	PI	PID	FLC	FLBPI	FLBPID
1	Overshoot (%)	0.0000	0.0000	11.4299	9.5200	0.0000
2	Settling Time (sec)	3.4900	3.6200	8.9400	2.1700	1.1100
3	#SSEP (%)	0.3914	0.0334	0.5290	0.0896	0.0083
4	Undershoot (%)	38.4279	36.5108	41.5249	41.2515	40.7199
5	IAE v all	1.15E+03	1.22E+03	1.34E+03	7.12E+02	7.13E+02
6	ITAE v start	2.54E+02	1.84E+02	5.17E+02	1.45E+02	1.49E+02
7	ITAE v load	2.28E+03	1.59E+03	2.42E+03	1.54E+03	1.62E+03
8	ITAE p	1.40E+03	1.30E+03	1.49E+03	1.29E+03	1.30E+03
9	fv all	1.2137	0.7666	0.0000	4.0140	4.0057
10	fv start	1.9715	2.4915	0.0000	2.7820	2.7550
11	fv load	0.0528	0.3132	0.0000	0.3321	0.3019
12	fp	1.3433	2.8358	0.0000	2.9851	2.8358

#SSEP: Steady State Error of Position

Table IV. Performance Index Comparison for Real time experiment

Comparing the conventional and hybrid controllers based on fitness values in real time experiments, the PID performs better than the PI for speed and position control in the case for conventional controllers. Interestingly, for hybrid controllers, the FLBPI is better than the FLBPID for speed and position control, whilst the FLBPI is better than the PID for speed and position control. This shows that the performance has improved when having the hybrid fuzzy and PI as a controller. In the standalone condition and using default parameters, the FLC is not as good as its competitors for both speed and position controls.

Plots for the speed control in response to a set-point specified in both simulation and real time experiment for FLBPI controller are presented in Figure 13 and for the position control are presented in Figure 14. Comparing the simulation and real time experiment results as shown in Figures 13 and 14, the performances are not exactly the same but the output patterns are similar and the results in real time experiments are validly representing the real system's performance. This shows that identification (s-modeling) is an estimation of real hardware plant.

Plots for the speed control in response to a set-point specified in the real time experiment for FLBPI compared to PID controller are presented in Figure 15 and for the position control are as presented in Figure 16. A sudden change in load requirements that happens at $t=15$ seconds causes the speed to fall and then rises and stabilized at the original level within a reasonable range of time. This demonstrates the action of the controllers to regulate the speed.

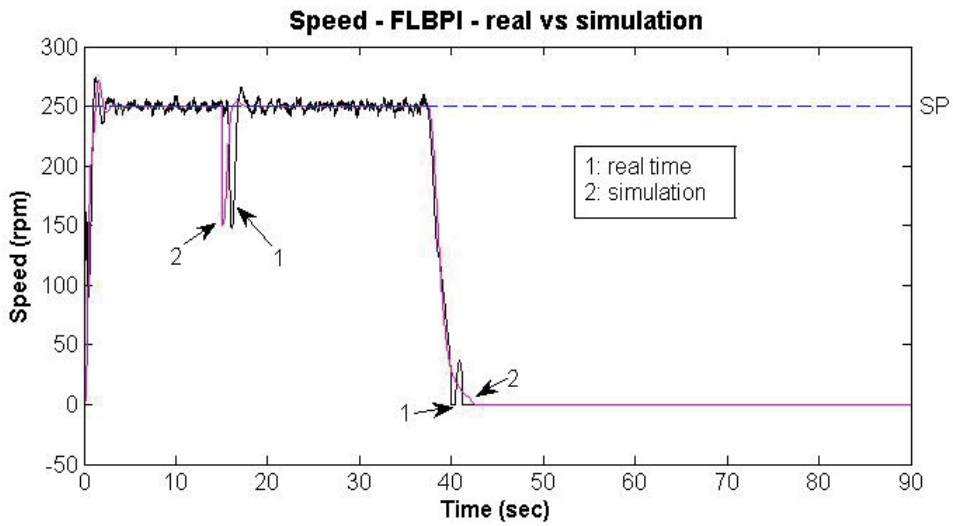


Fig. 13. Speed control of DC servomotor using FLBPI in real time experiment vs simulation

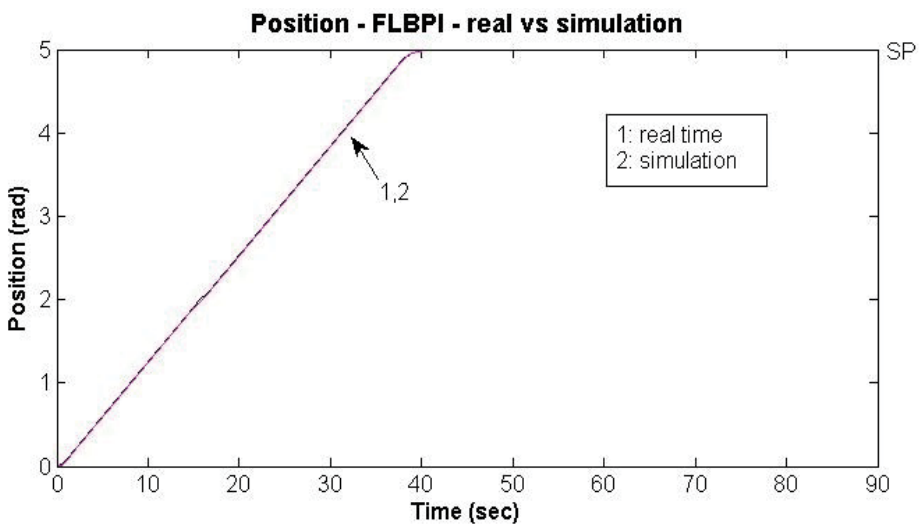


Fig. 14. Position control of DC servomotor using FLBPI in real time vs simulation

Plots for the speed control in response to a set-point specified in the real time experiment for FLBPI compared to FLC are presented in Figure 17 and for the position control are presented in Figure 18. A sudden change in load requirements that happens at $t=15$ seconds causes the speed to fall and then rises and stabilized at the original level within a reasonable range of time. Interestingly, it can be seen the delay in speed and position when using FLC alone as the controller. This demonstrates the action of the controllers to regulate the speed, and the effect of the PI in improving the response time of the hybrid controller.

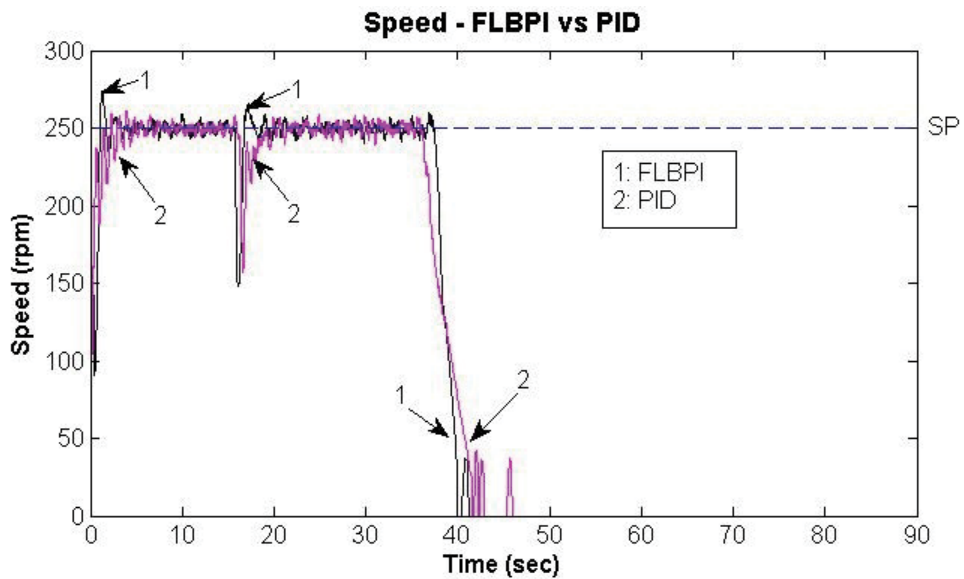


Fig. 15. Speed control of DC servomotor using FLBPI vs PID in real time experiment

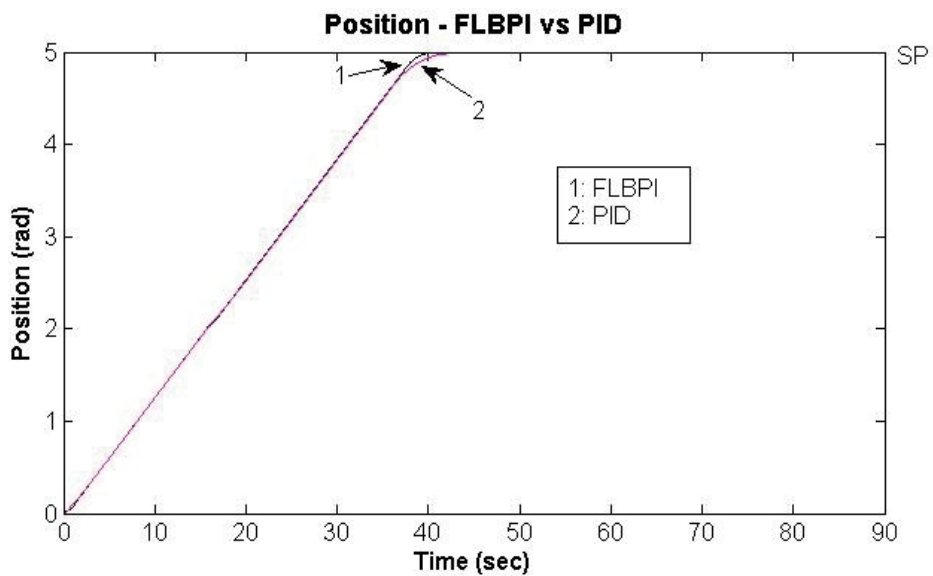


Fig. 16. Position control of DC servomotor using FLBPI vs PID in real time experiment

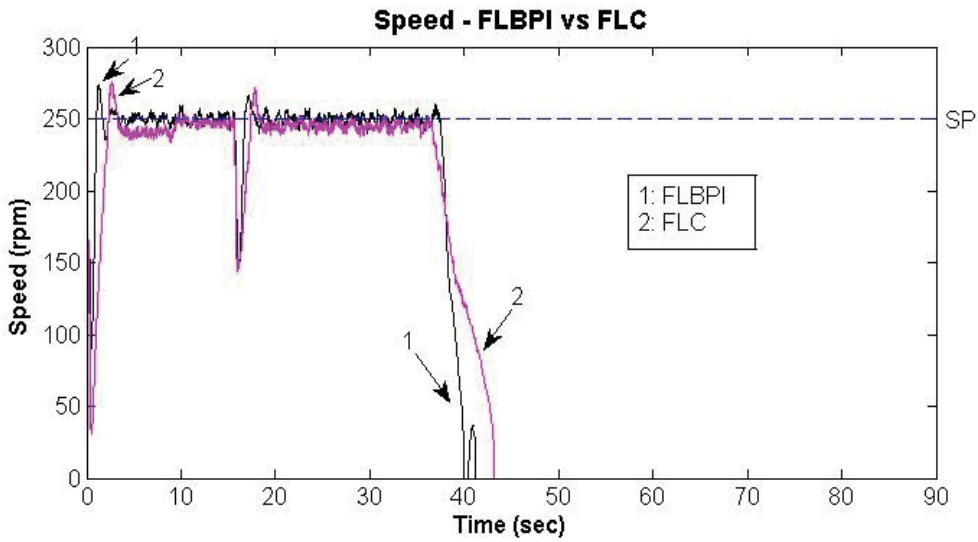


Fig. 17. Speed control of DC servomotor using FLBPI vs FLC in real time experiment

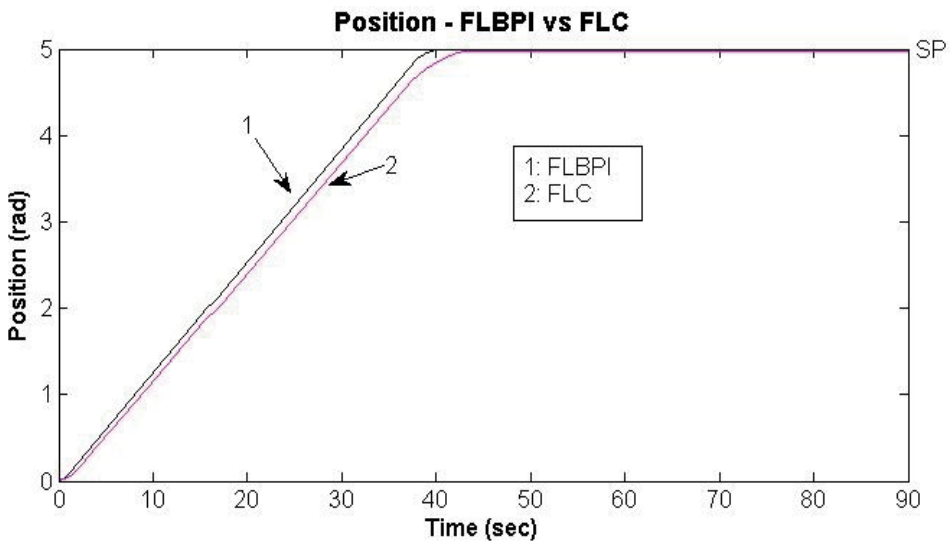


Fig. 18. Position control of DC servomotor using FLBPI vs FLC in real time experiment

5. Conclusions

This work discusses the modeling of a DC servomotor from gray box identification and performance evaluations of real time experiment using a fuzzy-logic-based self tuning PI controller as compared to fuzzy-logic-based self tuning PID controller, fuzzy logic controller, PID controller and PI controller on the DC servomotor system. Here, the s-model transfer function of a DC servomotor is identified as a third order transfer function without

zero. This identification is useful in offline optimization of a DC servomotor control, and the performance of s-modeled and real DC servomotor are similar.

Two control modes are applied in sequential to the plant: speed control in the position control loop. The open loop characteristic of a DC servomotor is sampled at 0.01 sec interval by a DAQ based on Simulink platform. Some controllers are applied to both s-modeled and real DC servomotor. It has been demonstrated that defining the fuzzy rules for the fuzzy logic-based self-tuning PI/PID controller is a much simpler task than for fuzzy logic controller. Based on the real time experiment, hybrid controllers are better than conventional controllers and fuzzy logic-based self-tuning PI controller tends to be the better choice for implementation in the hybrid controller.

6. References

- Bousserhanel, I.K., Hazzabl, A., Rahli, M., Kamli, M. and Mazari, B. (2006). Adaptive PI Controller using Fuzzy System Optimized by Genetic Algorithm for Induction Motor Control, presented at CIEP-IEEE, Puebla, Mexico.
- Fulford, C., Maggiore, M. and Apkarian, J. (2009). Control of a 5DOF Magnetically Levitated Positioning Stage, *IEEE Transaction on Control System Technology*, vol. 17 no. 4, pp. 844-852.
- Ga, X. and Feng, Z.J. (2005). Design study of an adaptive fuzzy-PD controller for pneumatic servo system, *Control Engineering Practice*, vol. 13, issue 1, pp 55-65.
- He, S.Z., Tan S., and Xu, F.L (1993). Fuzzy self-tuning of PID controllers, *Fuzzy Sets and Systems*, vol. 56, pp. 37-46.
- Jee, S. and Koren, Y. (2004). Adaptive fuzzy-logic controller for feed drive of a CNC machine tool, *Mechatronics*, vol: 14, pp. 299-326.
- Kissell, T. (2002). *Motor Control Technology for Industrial Maintenance*. New Jersey 07458: Prentice-Hall, Inc., Upper Saddle River.
- Lacevic, B., Velagic, J. and Osmic, N. (2007). Design of Fuzzy Logic Based Mobile Robot Position Controller Using Genetic Algorithm," presented at International Conference on Advanced Intelligent Mechatronics, IEEE/ASME 2007, vol., no., pp.1-6.
- Lu, Y.Z. (1996). *Industrial Intelligent Control, Fundamentals and Applications*. West Sussex PO19 1UD, England: John Wiley&Sons Ltd.
- Mannan, M.A., et al. (2004). Fuzzy-Logic-Based Self-Tuning PI Controller for Speed Control of Indirect Field-Oriented Induction Motor Drive, in *SICE Annual Conference*. Sapporo: Hokkaido Institute of Technology.
- Marlin, T.E. (2000). *Process Control: Designing Processes and Control Systems for Dynamic Performance*, 2nd International ed. Singapore: McGraw-Hill Book Companies, Inc.
- Montanari, M., Peresada, S.M., Rossi, C. and Tilli, A. (2007). Speed Sensorless Control of Induction Motors Based on a Reduced-Order Adaptive Observer," *IEEE Transaction on Control System Technology*, vol. 15 no. 6, pp. 1049-1064.
- Shieh, M. Y. and Li, T.H.S. (1998). Design and implementation of integrated fuzzy logic controller for servomotor system, *Mechatronics*, vol. 8, pp. 217-240.
- Takami, K.M. and Mahmoudi, J. (2007). Identification of a Best Thermal Formula and Model for Oil and Winding of Power Transformers Using Prediction Methods, presented at The 48th Scandinavian Conference on Simulation and Modeling (SIMS 2007), pp. 182-188.

- Wang, L.X. (1997). *A Course in Fuzzy Systems and Control*. New Jersey 07458: Prentice-Hall, Inc, A Division of Simon and Schuster Upper Saddle River, New Jersey 07458.
- Xiu, Z. and Ren, G. (2004). "Optimization Design of TS-PID Fuzzy Controllers Based on Genetic Algorithms," presented at 5th World Congress on Intelligent Control and Automation, Hangzhou, P.R. China.
- Yeh, E.C., and Tsao, Y.J. (1994). "A fuzzy preview control scheme of active suspension for rough road, Inc. *Journal of Vehicle Design*, vol. 15, pp. 166-180.
- Yu, G.R., Hwang, R.C. and Lin, C.P. (2004). "Optimal Fuzzy Control of the Spindle Motor in a CD-ROM Drive Using Genetic Algorithms," presented at Asian Control Conference, vol. 5, pp. 51-57.
- Zadeh, L.A. (1988). *Fuzzy Logic, Computer*, vol. 21 no. 24, pp. 83-93.

Distributed Particle Filtering over Sensor Networks for Autonomous Navigation of UAVs

Gerasimos G. Rigatos
*Industrial Systems Institute
Greece*

1. Introduction

State estimation and control over sensor networks is a problem met in several applications such as surveillance and condition monitoring of large-scale systems, multi-robot systems and cooperating UAVs. In sensor networks the simplest kind of architecture is centralized. Distributed sensors send measurement data to a central processing unit which provides the state estimate for the monitored system. Such an approach has several weaknesses: (i) it lacks fault tolerance: if the central processing unit is subject to a fault then state estimation becomes impossible, (ii) communication overhead often prohibits proper functioning in case of a large number of distributed measurement units. On the other hand decentralized architectures are based on the communication between neighboring measurement units. This assures scalability for the network since the number of messages received or sent by each measurement unit is independent of the total number of measurement units in the system. It has been shown that scalable decentralized state estimation can be achieved for linear Gaussian models, when the measurements are linear functions of the state and the associated process and measurement noise models follow a Gaussian distribution (Nettleton et al. 2003). A solution to decentralized sensor fusion over sensor networks with the use of distributed Kalman Filtering has been proposed in (Olfati-Saber 2006), (Watanabe & Tzafestas 1992), (Olfati-Saber 2005), (Gan & Harris 2001), (Gao et al. 2009). Distributed state estimation in the case of non-Gaussian models has been studied in (Rosencrantz et al. 2003) where decentralized sensor fusion with the use of distributed particle filters has been proposed in several other research works (Mahler 2007), (Makarenko & Durrant-Whyte 2006), (Deming & Perlovsky 2007).

In this paper autonomous navigation of UAVs will be examined and a solution to this problem will be first attempted with the use of the Extended Information Filter and the Unscented Kalman filter (Shima et al. 2007), (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005). Comparatively, autonomous UAV navigation with the use of the Distributed Particle Filter will be studied. This problem belongs to the wider area of multi-source multi-target tracking (Coué et al. 2006), (Hue et al. 2002), (Ing & Coates 2005), (Coué et al. 2003), (Morelande & D. Mušicki 2005). Subproblems to be solved for succeeding autonomous navigation of the UAVs are: (i) implementation of sensor fusion with the use of distributed filtering. In this approach the goal is to consistently combine the local particle distribution with the communicated particle distribution coming from particle filters running on nearby

measurement stations (Caballero et al. 2008). It is assumed that each local measurement station runs its own local filter and communicates information to other measurement stations close to it. The motivation for using particle filters is that they can represent almost arbitrary probability distributions, thus becoming well-suited to accommodate the types of uncertainty and nonlinearities that arise in the distributed estimation (Rigatos 2009a), (Rigatos 2009b) (ii) nonlinear control of the UAVs based on the state estimates provided by the particle filtering algorithm. Various approaches have been proposed for the UAV navigation using nonlinear feedback control (Ren & Beard 2004), (Beard et al. 2002), (Singh & Fuller 2001). The paper proposes flatness-based control for the UAV models. Flatness-based control theory is based on the concept of differential flatness and has been successfully applied to several nonlinear dynamical systems. Flatness-based control for a UAV helicopter-like model has been developed in (Léchevin & Rabbath 2006), assuming that the UAV performs manoeuvres at a constant altitude.

The paper proposes first the Extended Information Filter (EIF) and the Unscented Information Filter (UIF) as possible approaches for fusing the state estimates provided by the local monitoring stations, under the assumption of Gaussian noises. The EIF and UIF estimated state vector is in turn used by a flatness-based controller that makes the UAV follow the desirable trajectory. The Extended Information Filter is a generalization of the Information Filter in which the local filters do not exchange raw measurements but send to an aggregation filter their local information matrices (local inverse covariance matrices) and their associated local information state vectors (products of the local information matrices with the local state vectors) (Shima et al. 2007), (Lee et al. 2008). In the case of the Unscented Information Filter there is no linearization of the UAVs observation equation. However the application of the Information Filter algorithm is possible through an implicit linearization which is performed by approximating the Jacobian matrix of the system's output equation by the product of the inverse of the state vector's covariance matrix (which can be also associated to the Fisher Information matrix) with the cross-correlation covariance matrix between the system's state vector and the system's output (Lee et al. 2008)], (Vercauteren & Wang 2005). Again, the local information matrices and the local information state vectors are transferred to an aggregation filter which produces the global estimation of the system's state vector.

Next, the Distributed Particle Filter (DPF) is proposed for fusing the state estimates provided by the local monitoring stations (local filters). The motivation for using DPF is that it is well-suited to accommodate non-Gaussian measurements. A difficulty in implementing distributed particle filtering is that particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter) (Ong et al. 2008), (Ong et al. 2006). This can be resolved by transforming the particles sets into Gaussian mixtures, and defining the global probability distribution on the common support set of the probability density functions associated with the local filters. The state vector which is estimated with the use of the DPF is used again by a flatness-based controller to make each UAV follow a desirable flight path.

The structure of the chapter is as follows: in Section 2 the Distributed Extended Kalman Filter (Extended Information Filter) is studied. In Section 3, the Distributed Unscented Kalman Filter (Unscented Information Filter) is analyzed and its use for distributed sensor fusion and state estimation is explained. In Section 4 Distributed Particle Filtering for sensor fusion-based state estimation will be analyzed. In Section 5 nonlinear control will be

proposed for succeeding trajectory tracking by the UAVs. In Section 6 simulation experiments will be provided about UAVs autonomous navigation using the proposed distributed particle filtering algorithm. The test case will be concerned with m helicopter models monitored by n different ground stations. By fusing the measurements from the distributed observation units with the use of the Extended Information Filter and the proposed Particle Filter algorithm, state estimates of the UAVs are obtained. These in turn are used by local nonlinear controllers for succeeding trajectory tracking. Finally in Section 7 concluding remarks will be provided.

2. Distributed Extended Kalman Filtering

2.1 Extended Kalman Filtering at local processing units

The distributed Extended Kalman Filter, also know as Extended Information Filter, performs fusion of the state estimates which are provided by local Extended Kalman Filters. Thus, the functioning of the local Extended Kalman Filters should be analyzed first. The following nonlinear state model is considered (Rigatos & Tzafestas 2007):

$$\begin{aligned} x(k+1) &= \phi(x(k)) + L(k)u(k) + w(k) \\ z(k) &= \gamma(x(k)) + v(k) \end{aligned} \quad (1)$$

where $x \in R^{m \times 1}$ is the system's state vector and $z \in R^{p \times 1}$ is the system's output, while $w(k)$ and $v(k)$ are uncorrelated, zero-mean, Gaussian zero-mean noise processes with co-variance matrices $Q(k)$ and $R(k)$ respectively. The operators $\phi(x)$ and $\gamma(x)$ are $\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_m(x)]^T$, and $\gamma(x) = [\gamma_1(x), \gamma_2(x), \dots, \gamma_p(x)]^T$, respectively. It is assumed that ϕ and γ are sufficiently smooth in x so that each one has a valid series Taylor expansion.

Following a linearization procedure, ϕ is expanded into Taylor series about \hat{x} :

$$\phi(x(k)) = \phi(\hat{x}(k)) + J_\phi(\hat{x}(k))[x(k) - \hat{x}(k)] + \dots \quad (2)$$

where $J_\phi(x)$ is the Jacobian of ϕ calculated at $\hat{x}(k)$:

$$J_\phi(x) = \frac{\partial \phi}{\partial x} \Big|_{x=\hat{x}(k)} = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \dots & \frac{\partial \phi_1}{\partial x_m} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \dots & \frac{\partial \phi_2}{\partial x_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \frac{\partial \phi_m}{\partial x_2} & \dots & \frac{\partial \phi_m}{\partial x_m} \end{pmatrix} \quad (3)$$

Likewise, γ is expanded about $\hat{x}^-(k)$

$$\gamma(x(k)) = \gamma(\hat{x}^-(k)) + J_\gamma[x(k) - \hat{x}^-(k)] + \dots \quad (4)$$

where $\hat{x}^-(k)$ is the estimation of the state vector $x(k)$ before measurement at the k -th instant to be received and $\hat{x}(k)$ is the updated estimation of the state vector after measurement at the k -th instant has been received. The Jacobian $J_\gamma(x)$ is

$$J_\gamma(x) = \frac{\partial \gamma}{\partial x} \Big|_{x=\hat{x}^-(k)} = \begin{pmatrix} \frac{\partial \gamma_1}{\partial x_1} & \frac{\partial \gamma_1}{\partial x_2} & \dots & \frac{\partial \gamma_1}{\partial x_m} \\ \frac{\partial \gamma_2}{\partial x_1} & \frac{\partial \gamma_2}{\partial x_2} & \dots & \frac{\partial \gamma_2}{\partial x_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \gamma_p}{\partial x_1} & \frac{\partial \gamma_p}{\partial x_2} & \dots & \frac{\partial \gamma_p}{\partial x_m} \end{pmatrix} \quad (5)$$

The resulting expressions create first order approximations of ϕ and γ . Thus the linearized version of the system is obtained:

$$\begin{aligned} x(k+1) &= \phi(\hat{x}(k)) + J_\phi(\hat{x}(k))[x(k) - \hat{x}(k)] + w(k) \\ z(k) &= \gamma(\hat{x}^-(k)) + J_\gamma(\hat{x}^-(k))[x(k) - \hat{x}^-(k)] + v(k) \end{aligned} \quad (6)$$

Now, the EKF recursion is as follows: First the time update is considered: by $\hat{x}(k)$ the estimation of the state vector at instant k is denoted. Given initial conditions $\hat{x}(0)$ and $P^-(0)$ the recursion proceeds as:

- *Measurement update.* Acquire $z(k)$ and compute:

$$\begin{aligned} K(k) &= P^-(k)J_\gamma^T(\hat{x}^-(k)) \cdot [J_\gamma(\hat{x}^-(k))P^-(k)J_\gamma^T(\hat{x}^-(k)) + R(k)]^{-1} \\ \hat{x}(k) &= \hat{x}^-(k) + K(k)[z(k) - \gamma(\hat{x}^-(k))] \\ P(k) &= P^-(k) - K(k)J_\gamma(\hat{x}^-(k))P^-(k) \end{aligned} \quad (7)$$

- *Time update.* Compute:

$$\begin{aligned} P^-(k+1) &= J_\phi(\hat{x}(k))P(k)J_\phi^T(\hat{x}(k)) + Q(k) \\ \hat{x}^-(k+1) &= \phi(\hat{x}(k)) + L(k)u(k) \end{aligned} \quad (8)$$

The schematic diagram of the EKF loop is given in Fig. 1.

2.2 Calculation of local estimations in terms of EIF information contributions

Again the discrete-time nonlinear system of Eq. (1) is considered. The Extended Information Filter (EIF) performs fusion of the local state vector estimates which are provided by the local Extended Kalman Filters, using the *Information matrix* and the *Information state vector* (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005), (Manyika & H. Durrant-Whyte 1994). The Information Matrix is the inverse of the state vector covariance matrix, and can be also associated to the Fisher Information matrix." (Rigatos & Zhang 2009). The Information state vector is the product between the Information matrix and the local state vector estimate

$$\begin{aligned} Y(k) &= P^{-1}(k) = I(K) \\ \hat{y}(k) &= P^{-1}\hat{x}(k) = Y(k)\hat{x}(k) \end{aligned} \quad (9)$$

The update equation for the Information Matrix and the Information state vector are given by

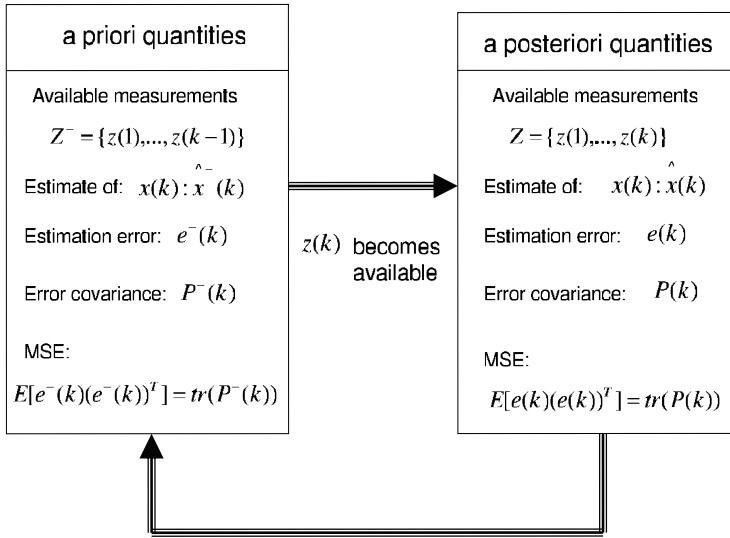


Fig. 1. Schematic diagram of the EKF loop

$$\begin{aligned} Y(k) &= P^-(k)^{-1} + J_\gamma^T(k)R^{-1}(k)J_\gamma(k) \\ &= Y^-(k) + I(k) \end{aligned} \quad (10)$$

$$\begin{aligned} \hat{y}(k) &= \hat{y}^-(k) + J_\gamma^T R(k)^{-1} [z(k) - \gamma(x(k)) + J_\gamma \hat{x}^-(k)] \\ &= \hat{y}^-(k) + i(k) \end{aligned} \quad (11)$$

where

$$\begin{aligned} I(k) &= J_\gamma^T(k)R^{-1}(k) - 1J_\gamma(k) \text{ is the associated information matrix and} \\ i(k) &= J_\gamma^T R(k)^{-1} [(z(k) - \gamma(x(k))) + J_\gamma \hat{x}^-(k)] \text{ is the information state contribution} \end{aligned} \quad (12)$$

The predicted information state vector and Information matrix are obtained from

$$\begin{aligned} \hat{y}^-(k) &= P^-(k)^{-1} \hat{x}^-(k) \\ Y^-(k) &= P^-(k)^{-1} = [J_\phi(k)P^-(k)J_\phi(k)^T + Q(k)]^{-1} \end{aligned} \quad (13)$$

The Extended Information Filter is next formulated for the case that multiple local sensor measurements and local estimates are used to increase the accuracy and reliability of the estimation. It is assumed that an observation vector $z^i(k)$ is available for N different sensor sites $i = 1, 2, \dots, N$ and each sensor observes a common state according to the local observation model, expressed by

$$z^i(k) = \gamma(x(k)) + v^i(k), i = 1, 2, \dots, N \quad (14)$$

where the local noise vector $v^i(k) \sim N(0, R^i)$ is assumed to be white Gaussian and uncorrelated between sensors. The variance of a composite observation noise vector v_k is expressed in terms of the block diagonal matrix

$$R(k) = \text{diag}[R(k)^1, \dots, R^N(k)]^T \quad (15)$$

The information contribution can be expressed by a linear combination of each local information state contribution i^i and the associated information matrix I^i at the i -th sensor site

$$i(k) = \sum_{i=1}^N J_\gamma^{iT}(k) R^i(k)^{-1} [z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k) \hat{x}^-(k)] \quad (16)$$

$$I(k) = \sum_{i=1}^N J_\gamma^{iT}(k) R^i(k)^{-1} J_\gamma^i(k)$$

Using Eq. (16) the update equations for fusing the local state estimates become

$$\hat{y}(k) = \hat{y}^-(k) + \sum_{i=1}^N J_\gamma^{iT}(k) R^i(k)^{-1} [z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k) \hat{x}^-(k)] \quad (17)$$

$$Y(k) = Y^-(k) + \sum_{i=1}^N J_\gamma^{iT}(k) R^i(k)^{-1} J_\gamma^i(k)$$

It is noted that in the Extended Information Filter an aggregation (master) fusion filter produces a global estimate by using the local sensor information provided by each local filter.

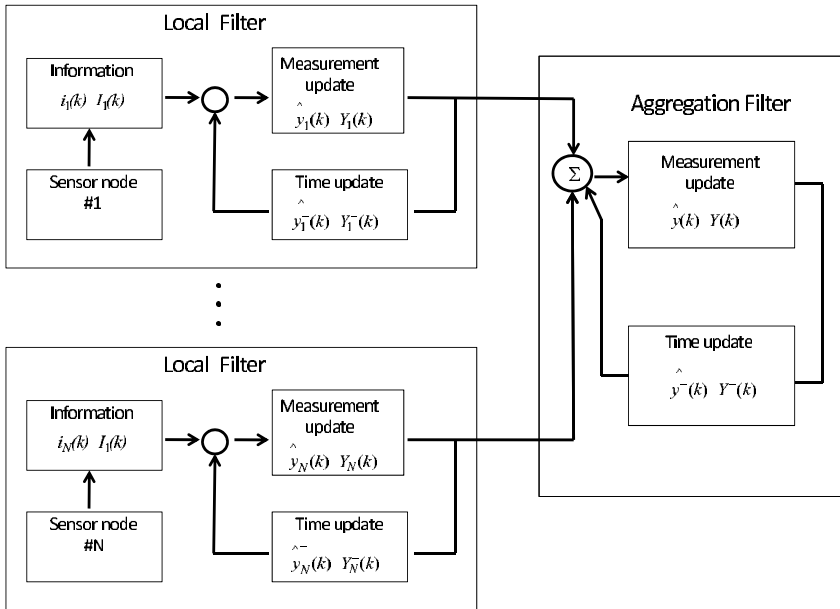


Fig. 2. Fusion of the distributed state estimates with the use of the Extended Information Filter
As in the case of the Extended Kalman Filter the local filters which constitute the Extended Information Filter can be written in terms of *time update* and a *measurement update* equation.
Measurement update: Acquire $z(k)$ and compute

$$Y(k) = P^-(k)^{-1} + J_\gamma^T(k) R(k)^{-1} J_\gamma(k) \quad (18)$$

or $Y(k) = Y^-(k) + I(k)$ where $I(k) = J_\gamma^T(k) R^{-1}(k) J_\gamma(k)$

$$\begin{aligned}\hat{y}(k) &= \hat{y}^-(k) + J_\gamma^T(k)R(k)^{-1}[z(k) - \gamma(\hat{x}(k)) + J_\gamma \hat{x}^-(k)] \\ \text{or } \hat{y}(k) &= \hat{y}^-(k) + i(k)\end{aligned}\quad (19)$$

Time update: Compute

$$Y^-(k+1) = P^-(k+1)^{-1} = [J_\phi(k)P(k)J_\phi(k)^T + Q(k)]^{-1} \quad (20)$$

$$y^-(k+1) = P^-(k+1)^{-1} \hat{x}^-(k+1) \quad (21)$$

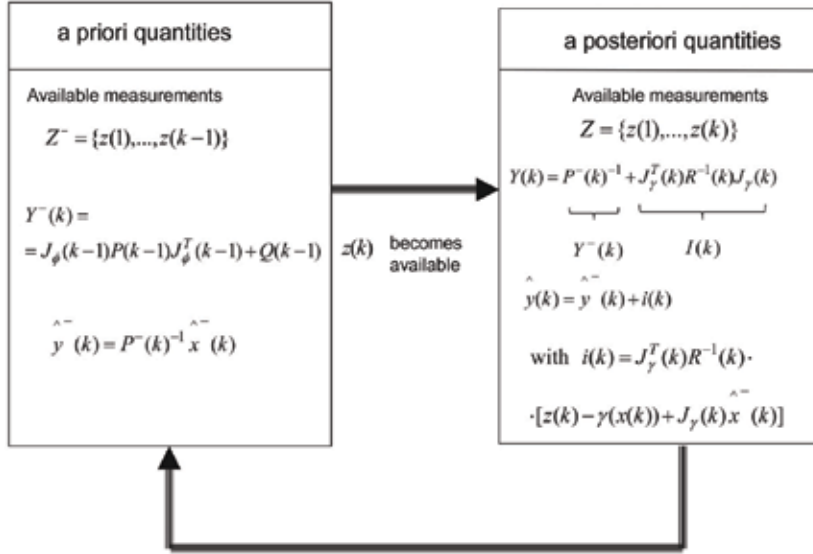


Fig. 3. Schematic diagram of the Extended Information Filter loop

2.3 Extended Information Filtering for state estimates fusion

In the Extended Information Filter each one of the local filters operates independently, processing its own local measurements. It is assumed that there is no sharing of measurements between the local filters and that the aggregation filter (Fig. 2) does not have direct access to the raw measurements feeding each local filter. The outputs of the local filters are treated as measurements which are fed into the aggregation fusion filter (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005). Then each local filter is expressed by its respective error covariance and estimate in terms of information contributions given in Eq.(13)

$$\begin{aligned}P_i^{-1}(k) &= P_i^-(k)^{-1} + J_\gamma^T(k)R(k)^{-1}J_\gamma(k) \\ \hat{x}_i(k) &= P_i(k)(P_i^-(k)^{-1} \hat{x}_i^-(k) + J_\gamma^T(k)R(k)^{-1} \cdot [z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k) \hat{x}_i^-(k)])\end{aligned}\quad (22)$$

It is noted that the local estimates are suboptimal and also conditionally independent given their own measurements. The global estimate and the associated error covariance for the aggregate fusion filter can be rewritten in terms of the computed estimates and covariances from the local filters using the relations

$$\begin{aligned}
J_\gamma^T(k)R(k)^{-1}J_\gamma(k) &= P_i^{-1}(k) - P_i^-(k)^{-1} \\
J_\gamma^T(k)R(k)^{-1}[z^i(k) - \gamma^k(x(k)) + J_\gamma^i(k)\hat{x}_i^-(k)] &= P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k)
\end{aligned} \tag{23}$$

For the general case of N local filters $i = 1, \dots, N$, the distributed filtering architecture is described by the following equations

$$\begin{aligned}
P(k)^{-1} &= P^-(k)^{-1} + \sum_{i=1}^N [P_i(k)^{-1} - P_i^-(k)^{-1}] \\
\hat{x}(k) &= P(k)[P^-(k)^{-1}\hat{x}^-(k) + \sum_{i=1}^N (P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k))]
\end{aligned} \tag{24}$$

It is noted that the global state update equation in the above distributed filter can be written in terms of the information state vector and of the information matrix

$$\begin{aligned}
\hat{y}(k) &= \hat{y}^-(k) + \sum_{i=1}^N (\hat{y}_i(k) - \hat{y}_i^-(k)) \\
\hat{Y}(k) &= \hat{Y}^-(k) + \sum_{i=1}^N (\hat{Y}_i(k) - \hat{Y}_i^-(k))
\end{aligned} \tag{25}$$

The local filters provide their own local estimates and repeat the cycle at step $k + 1$. In turn the global filter can predict its global estimate and repeat the cycle at the next time step $k + 1$ when the new state $\hat{x}(k + 1)$ and the new global covariance matrix $P(k + 1)$ are calculated. From Eq. (24) it can be seen that if a local filter (processing station) fails, then the local covariance matrices and the local state estimates provided by the rest of the filters will enable an accurate computation of the system's state vector.

3. Distributed Sigma-Point Kalman Filtering

3.1 Unscented Kalman Filtering at local processing units

It is also possible to estimate the state vectors of the distributed UAVs which constitute the multi-UAV system through the fusion of the estimates provided by local Sigma-Point Kalman Filters. This can be succeeded using the Distributed Sigma-Point Kalman Filter, also known as *Unscented Information Filter* (UIF) (Lee et al. 2008), (Lee et al. 2008). First, the functioning of the local Sigma-Point Kalman Filters will be explained. Each local Sigma-Point Kalman Filter generates an estimation of the UAV's state vector by fusing measurement from distributed sensors (e.g. IMU and GPS). Sigma-Point Kalman Filtering is proposed (Julier et al. 2000), (Julier et al. 2004), (Särkkä 2007). The Sigma-Point Kalman Filter overcomes the flaws of Extended Kalman Filtering. Unlike EKF no analytical Jacobians of the system equations need to be calculated as in the case for the EKF. This makes the sigma-point approach suitable for application in "black-box" models where analytical expressions of the system dynamics are either not available or not in a form which allows for easy linearization. This is achieved through a different approach for calculating the posterior 1st and 2nd order statistics of a random variable that undergoes a nonlinear transformation. The state distribution is represented again by a Gaussian Random Variable but is now specified using a minimal set of deterministically chosen weighted sample points. The basic sigma-point approach can be described as follows:

1. A set of weighted samples (sigma-points) are deterministically calculated using the mean and square-root decomposition of the covariance matrix of the system's state vector. As a minimal requirement the sigma-point set must completely capture the first and second order moments of the prior random variable. Higher order moments can be captured at the cost of using more sigma-points.
2. The sigma-points are propagated through the true nonlinear function using functional evaluations alone, i.e. no analytical derivatives are used, in order to generate a posterior sigma-point set.
3. The posterior statistics are calculated (approximated) using tractable functions of the propagated sigma-points and weights. Typically, these take on the form of a simple weighted sample mean and covariance calculations of the posterior sigma points.

It is noted that the sigma-point approach differs substantially from general stochastic sampling techniques, such as Monte-Carlo integration (e.g Particle Filtering methods) which require significantly more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state. The deceptively simple sigma-point approach results in posterior approximations that are accurate to the third order for Gaussian inputs for all nonlinearities. For non-Gaussian inputs, approximations are accurate to at least the second-order, with the accuracy of third and higher-order moments determined by the specific choice of weights and scaling factors.

The Unscented Kalman Filter (UKF) is a special case of Sigma-Point Kalman Filters. The UKF is a discrete time filtering algorithm which uses the unscented transform for computing approximate solutions to the filtering problem of the form

$$\begin{aligned} x(k+1) &= \phi(x(k)) + L(k)U(k) + w(k) \\ y(k) &= \gamma(x(k)) + v(k) \end{aligned} \quad (26)$$

where $x(k) \in R^n$ is the system's state vector, $y(k) \in R^m$ is the measurement, $w(k) \in R^n$ is a Gaussian process noise $w(k) \sim N(0, Q(k))$, and $v(k) \in R^m$ is a Gaussian measurement noise $v(k) \sim N(0, R(k))$. The mean and covariance of the initial state $x(0)$ are $m(0)$ and $P(0)$, respectively.

Some basic operations performed in the UKF algorithm (*Unscented Transform*) are summarized as follows:

1. Denoting the current state mean as \hat{x} , a set of $2n+1$ sigma points is taken from the columns of the $n \times n$ matrix $\sqrt{(n+\lambda)P_{xx}}$ as follows:

$$\begin{aligned} x^0 &= \hat{x} \\ x^i &= \hat{x} + [\sqrt{(n+\lambda)P_{xx}}]_i, \quad i = 1, \dots, n \\ x^i &= \hat{x} - [\sqrt{(n+\lambda)P_{xx}}]_i, \quad i = n+1, \dots, 2n \end{aligned} \quad (27)$$

and the associate weights are computed:

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{(n+\lambda)} & W_0^{(c)} &= \frac{\lambda}{(n+\lambda) + (1-\alpha^2 + b)} \\ W_i^{(m)} &= \frac{1}{2(n+\lambda)}, \quad i = 1, \dots, 2n & W_i^{(c)} &= \frac{1}{2(n+\lambda)} \end{aligned} \quad (28)$$

where $i = 1, 2, \dots, 2n$ and $\lambda = \alpha^2(n + \kappa) - n$ is a scaling parameter, while α, β and κ are constant parameters. Matrix P_{xx} is the covariance matrix of the state x .

2. Transform each of the sigma points as

$$z^i = h(x^i) \quad i = 0, \dots, 2n \quad (29)$$

3. Mean and covariance estimates for z can be computed as

$$\begin{aligned} \hat{z} &\simeq \sum_{i=0}^{2n} W_i^{(m)} z^i \\ P_{zz} &= \sum_{i=0}^{2n} W_i^{(c)} (z^i - \hat{z})(z^i - \hat{z})^T \end{aligned} \quad (30)$$

4. The cross-covariance of x and z is estimated as

$$P_{xz} \simeq \sum_{i=0}^{2n} W_i^{(c)} (x^i - \hat{x})(z^i - \hat{z})^T \quad (31)$$

The matrix square root of positive definite matrix P_{xx} means a matrix $A = \sqrt{P_{xx}}$ such that $P_{xx} = AA^T$ and a possible way for calculation is SVD.

Next the basic stages of the *Unscented Kalman Filter* are given:

As in the case of the Extended Kalman Filter and the Particle Filter, the Unscented Kalman Filter also consists of prediction stage (time update) and correction stage (measurement update) (Julier et al. 2004), (Särrkä 2007).

Time update: Compute the predicted state mean $\hat{x}^-(k)$ and the predicted covariance $P_{xx}^-(k)$ as

$$\begin{aligned} [\hat{x}^-(k), P_{xx}^-(k)] &= UT(f_d, \hat{x}(k-1), P_{xx}(k-1)) \\ P_{xx}^-(k) &= P_{xx}(k-1) + Q(k-1) \end{aligned} \quad (32)$$

Measurement update: Obtain the new output measurement z_k and compute the predicted mean $\hat{z}(k)$ and covariance of the measurement $P_{zz}(k)$, and the cross covariance of the state and measurement $P_{xz}(k)$

$$\begin{aligned} [\hat{z}(k), P_{zz}(k), P_{xz}(k)] &= UT(h_d, \hat{x}^-(k), P_{xx}^-(k)) \\ P_{zz}(k) &= P_{zz}(k) + R(k) \end{aligned} \quad (33)$$

Then compute the filter gain $K(k)$, the state mean $\hat{x}(k)$ and the covariance $P_{xx}(k)$, conditional to the measurement $y(k)$

$$\begin{aligned} K(k) &= P_{xz}(k)P_{zz}^{-1}(k) \\ \hat{x}(k) &= \hat{x}^-(k) + K(k)[z(k) - \hat{z}(k)] \\ P_{xx}(k) &= P_{xx}^-(k) - K(k)P_{zz}(k)K(k)^T \end{aligned} \quad (34)$$

The filter starts from the initial mean $m(0)$ and covariance $P_{xx}(0)$. The stages of state vector estimation with the use of the Unscented Kalman Filter algorithm are depicted in Fig. 6.

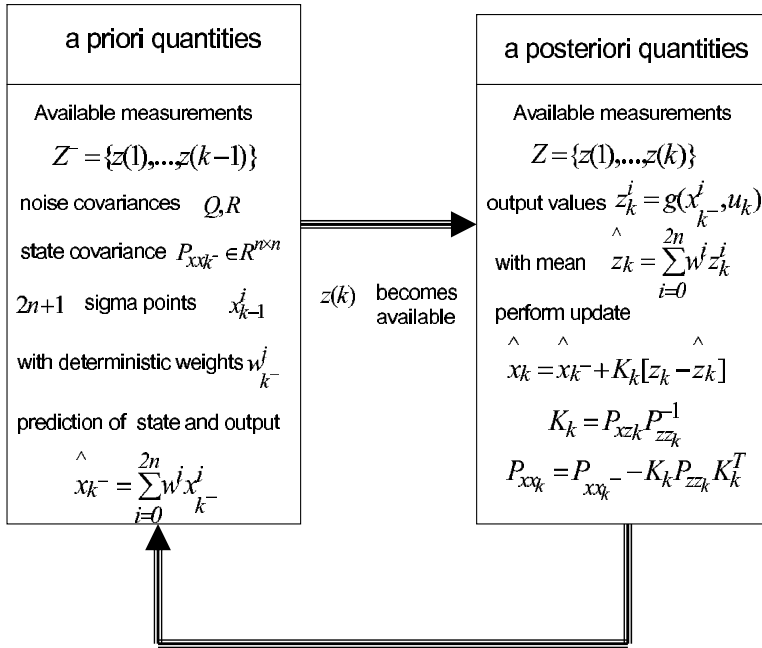


Fig. 4. Schematic diagram of the Unscented Kalman Filter loop

3.2 Unscented Information Filtering

The Unscented Information Filter (UIF) performs fusion of the state vector estimates which are provided by local Unscented Kalman Filters, by weighting these estimates with local Information matrices (inverse of the local state vector covariance matrices which are again recursively computed) (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005). The Unscented Information Filter is derived by introducing a linear error propagation based on the unscented transformation into the Extended Information Filtering structure. First, an augmented state vector $x_{\alpha}^{-}(k)$ is considered, along with the process noise vector, and the associated covariance matrix is introduced

$$\hat{x}_{\alpha}^{-}(k) = \begin{pmatrix} \hat{x}^{-}(k) \\ \hat{w}^{-}(k) \end{pmatrix}, P^{\alpha-}(k) = \begin{pmatrix} P^{-}(k) & 0 \\ 0 & Q^{-}(k) \end{pmatrix} \quad (35)$$

As in the case of local (lumped) Unscented Kalman Filters, a set of weighted sigma points $X_{\alpha}^{i-}(k)$ is generated as

$$\begin{aligned} X_{\alpha,0}^{-}(k) &= \hat{x}_{\alpha}^{-}(k) \\ X_{\alpha,i}^{-}(k) &= \hat{x}_{\alpha}^{-}(k) + [\sqrt{(n_{\alpha} + \lambda)} P_{\alpha}^{-}(k-1)]_i, i = 1, \dots, n \\ X_{\alpha,i}^{-}(k) &= \hat{x}_{\alpha}^{-}(k) + [\sqrt{(n_{\alpha} + \lambda)} P_{\alpha}^{-}(k-1)]_i, i = n+1, \dots, 2n \end{aligned} \quad (36)$$

where $\lambda = \alpha^2(n_{\alpha} + \kappa) - n_{\alpha}$ is a scaling, while $0 \leq \alpha \leq 1$ and κ are constant parameters. The corresponding weights for the mean and covariance are defined as in the case of the lumped Unscented Kalman Filter

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{n_\alpha + \lambda} & W_0^{(c)} &= \frac{\lambda}{(n_\alpha + \lambda) + (1 - \alpha^2 + \beta)} \\
W_i^{(m)} &= \frac{1}{2(n_\alpha + \lambda)}, i = 1, \dots, 2n_\alpha & W_i^{(c)} &= \frac{1}{2(n_\alpha + \lambda)}, i = 1, \dots, 2n_\alpha
\end{aligned} \tag{37}$$

where β is again a constant parameter. The equations of the prediction stage (measurement update) of the information filter, i.e. the calculation of the information matrix and the information state vector of Eq. (13) now become

$$\begin{aligned}
\hat{y}^-(k) &= Y^-(k) \sum_{i=0}^{2n_\alpha} W_i^m X_i^x(k) \\
Y^-(k) &= P^-(k)^{-1}
\end{aligned} \tag{38}$$

where X_i^x are the predicted state vectors when using the sigma point vectors X_i^w in the state equation $X_i^x(k+1) = \phi(X_i^w(k)) + L(k)U(k)$. The predicted state covariance matrix is computed as

$$P^-(k) = \sum_{i=0}^{2n_\alpha} W_i^{(c)} [X_i^x(k) - \hat{x}^-(k)][X_i^x(k) - \hat{x}^-(k)]^T \tag{39}$$

As noted, the equations of the Extended Information Filter (EIF) are based on the linearized dynamic model of the system and on the inverse of the covariance matrix of the state vector. However, in the equations of the Unscented Kalman Filter (UKF) there is no linearization of the system dynamics, thus the UKF cannot be included directly into the EIF equations. Instead, it is assumed that the nonlinear measurement equation of the system given in Eq. (1) can be mapped into a linear function of its statistical mean and covariance, which makes possible to use the information update equations of the EIF. Denoting $Y_i(k) = \gamma(X_i^x(k))$ (i.e. the output of the system calculated through the propagation of the i -th sigma point X_i^i through the system's nonlinear equation) the observation covariance and its cross-covariance are approximated by

$$\begin{aligned}
P_{YY}^-(k) &= E[(z(k) - \hat{z}^-(k))(z(k) - \hat{z}^-(k))^T] \\
&\cong J_\gamma(k)P^-(k)J_\gamma(k)^T
\end{aligned} \tag{40}$$

$$\begin{aligned}
P_{XY}^-(k) &= E[(x(k) - \hat{x}^-(k))(z(k) - \hat{z}^-(k))^T] \\
&\cong P^-(k)J_\gamma(k)^T
\end{aligned} \tag{41}$$

where $z(k) = \gamma(x(k))$ and $J_\gamma(k)$ is the Jacobian of the output equation $\gamma(x(k))$. Next, multiplying the predicted covariance and its inverse term on the right side of the information matrix Eq. (12) and replacing $P(k)J_\gamma(k)^T$ with $P_{XY}^-(k)$ (k) gives the following representation of the information matrix (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005)

$$\begin{aligned}
I(k) &= J_\gamma(k)^T R(k)^{-1} J_\gamma(k) \\
&= P^-(k)^{-1} P^-(k) J_\gamma(k)^T R(k)^{-1} J_\gamma(k) P^-(k)^T (P^-(k)^{-1})^T \\
&= P^-(k)^{-1} P_{XY}(k) R(k)^{-1} P_{XY}(k)^T (P^-(k)^{-1})^T
\end{aligned} \tag{42}$$

where $P^-(k)^{-1}$ is calculated according to Eq. (39) and the cross-correlation matrix $P_{XY}(k)$ is calculated from

$$P_{XY}^-(k) = \sum_{i=0}^{2n_g} W_i^{(c)} [X_i^x(k) - \hat{x}^-(k)][Y_i(k) - \hat{z}^-(k)]^T \quad (43)$$

where $Y_i(k) = \gamma(X_i^x(k))$ and the predicted measurement vector $\hat{z}^-(k)$ is obtained by $\hat{z}^-(k) = \sum_{i=0}^{2n} W_i^{(m)} Y_i(k)$. Similarly, the information state vector i_k can be rewritten as

$$\begin{aligned} i(k) &= J_\gamma^T(k)R(k)^{-1}[z(k) - \gamma(x(k)) + J_\gamma^T(k)\hat{x}^-(k)] \\ &= P^-(k)^{-1}P^-(k)J_\gamma^T(k)R(k)^{-1} \cdot \\ &\quad \cdot [z(k) - \gamma(x(k)) + J_\gamma^T(k)\hat{x}^-(k)(P^-(k))^T(P^-(k)^{-1})^T\hat{x}^-(k)] \\ &= P^-(k)^{-1}P_{XY}^-(k)R(k)^{-1}[z(k) - \gamma(x(k)) + P_{XY}^-(k)(P^-(k)^{-1})^T\hat{x}^-(k)] \end{aligned} \quad (44)$$

To complete the analogy to the information contribution equations of the EIF a "measurement" matrix $H^T(k)$ is defined as

$$H(k)^T = P^-(k)^{-1}P_{XY}^-(k) \quad (45)$$

In terms of the measurement matrix $H(k)$ the information contributions equations are written as

$$\begin{aligned} i(k) &= H^T(k)R(k)^{-1}[z(k) - \gamma(x(k)) + H(k)\hat{x}^-(k)] \\ I(k) &= H^T(k)R(k)^{-1}H(k) \end{aligned} \quad (46)$$

The above procedure leads to an implicit linearization in which the nonlinear measurement equation of the system given in Eq. (1) is approximated by the statistical error variance and its mean

$$z(k) = h(x(k)) \simeq H(k)x(k) + \bar{u}(k) \quad (47)$$

where $\bar{u}(k) = \gamma(\hat{x}^-(k)) - H(k)\hat{x}^-(k)$ is a measurement residual term. (47).

3.3 Calculation of local estimations in terms of UIF information contributions

Next, the local estimations provided by distributed (local) Unscented Kalman filters will be expressed in terms of the information contributions (information matrix I and information state vector i) of the Unscented Information Filter, which were defined in Eq. (46) (Lee et al. 2008), (Lee et al. 2008), (Vercauteren & Wang 2005). It is assumed that the observation vector $\bar{z}_i(k+1)$ is available from N different sensors, and that each sensor observes a common state according to the local observation model, expressed by

$$\bar{z}_i(k) = H_i(k)x(k) + \bar{u}_i(k) + v_i(k) \quad (48)$$

where the noise vector $v_i(k)$ is taken to be white Gaussian and uncorrelated between sensors. The variance of the composite observation noise vector v_k of all sensors is written in terms of the block diagonal matrix $R(k) = \text{diag}[R_1(k)^T, \dots, R_N(k)^T]^T$. Then one can define the local information matrix $I_i(k)$ and the local information state vector $i^i(k)$ at the i -th sensor, as follows

$$\begin{aligned} i_i(k) &= H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)] \\ I_i(k) &= H_i^T(k)R_i(k)^{-1}H_i(k) \end{aligned} \quad (49)$$

Since the information contribution terms have group diagonal structure in terms of the innovation and measurement matrix, the update equations for the multiple state estimation and data fusion are written as a linear combination of the local information contribution terms

$$\begin{aligned} \hat{y}(k) &= \hat{y}^-(k) + \sum_{i=1}^N i_i(k) \\ Y(k) &= Y^-(k) + \sum_{i=1}^N I_i(k) \end{aligned} \quad (50)$$

Then using Eq. (38) one can find the mean state vector for the multiple sensor estimation problem.

As in the case of the Unscented Kalman Filter, the Unscented Information Filter running at the i -th measurement processing unit can be written in terms of *measurement update* and *time update* equations:

Measurement update: Acquire measurement $z(k)$ and compute

$$\begin{aligned} Y(k) &= P^-(k)^{-1} + H^T(k)R^{-1}(k)H(k) \\ \text{or } Y(k) &= Y^-(k) + I(k) \text{ where } I(k) = H^T(k)R^{-1}(k)H(k) \end{aligned} \quad (51)$$

$$\begin{aligned} \hat{y}(k) &= \hat{y}^-(k) + H^T(k)R^{-1}(k)[z(k) - \gamma(\hat{x}(k)) + H(k)\hat{x}^-(k)] \\ \text{or } \hat{y}(k) &= \hat{y}^-(k) + i(k) \end{aligned} \quad (52)$$

Time update: Compute

$$\begin{aligned} Y^-(k+1) &= (P^-(k+1))^{-1} \\ \text{where } P^-(k+1) &= \sum_{i=0}^{2n_\alpha} W_i^{(e)} [X_i^x(k+1) - \hat{x}^-(k+1)][X_i^x(k+1) - \hat{x}^-(k+1)]^T \end{aligned} \quad (53)$$

$$\begin{aligned} \hat{y}(k+1) &= Y(k+1) \sum_{i=0}^{2n_\alpha} W_i^{(m)} X_i^x(k+1) \\ \text{where } X_i^x(k+1) &= \phi(X_i^w(k)) + L(k)U(k) \end{aligned} \quad (54)$$

3.4 Distributed Unscented Information Filtering for state estimates fusion

It has been shown that the update of the aggregate state vector of the Unscented Information Filter architecture can be expressed in terms of the local information matrices I_i and of the local information state vectors i_i , which in turn depend on the local covariance matrices P and cross-covariance matrices P_{XY} . Next, it will be shown that the update of the aggregate state vector can be also expressed in terms of the local state vectors $x_i(k)$ and in terms of the local covariance matrices $P_i(k)$ and cross-covariance matrices $P_{XY}^i(k)$. It is assumed that the local filters do not have access to each other row measurements and that they are allowed to communicate only their information matrices and their local information state vectors. Thus each local filter is expressed by its respective error covariance and estimate in terms of the local information state contribution i_i and its associated information matrix I_i at the i -th filter site. Then using Eq. (38) one obtains

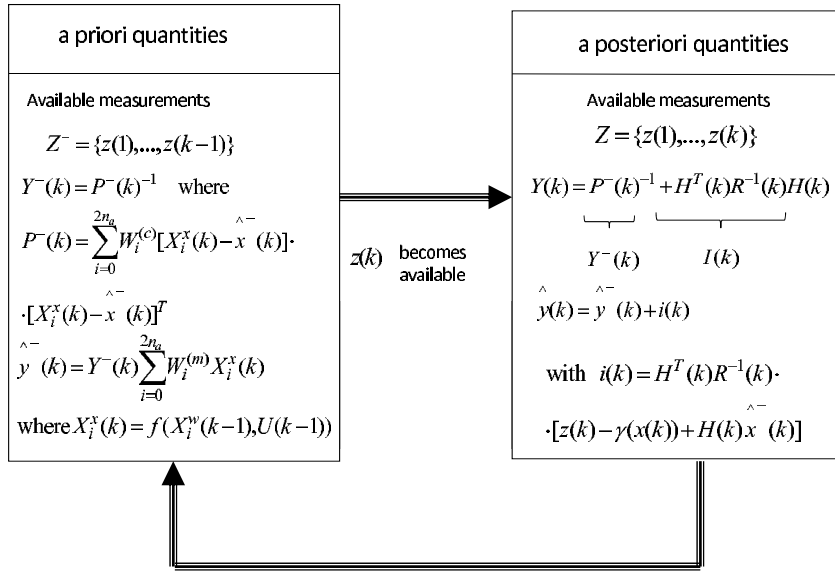


Fig. 5. Schematic diagram of the Unscented Information Filter loop

$$\begin{aligned}
 P_i(k)^{-1} &= P_i^-(k)^{-1} + H_i^T(k)R_i(k)^{-1}H_i(k) \\
 \hat{x}_i &= P_i(k)(P_i^-(k)\hat{x}_i^-(k) + H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)])
 \end{aligned} \quad (55)$$

Using Eq. (55), each local information state contribution i_i and its associated information matrix I_i at the i -th filter are rewritten in terms of the computed estimates and covariances of the local filters

$$\begin{aligned}
 H_i^T(k)R_i(k)^{-1}H_i(k) &= P_i(k)^{-1} - P_i^-(k)^{-1} \\
 H_i^T(k)R_i(k)^{-1}[z_i(k) - \gamma_i(x(k)) + H_i(k)\hat{x}^-(k)] &= P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k)
 \end{aligned} \quad (56)$$

where according to Eq.(45) it holds $H_i(k) = P_i^-(k)^{-1}P_{xY,i}^-(k)$. Next, the aggregate estimates of the distributed unscented information filtering are derived for a number of N local filters $i = 1, \dots, N$ and sensor measurements, first in terms of covariances (Vercauteren & Wang 2005), (Lee et al. 2008), (Lee et al. 2008)

$$\begin{aligned}
 P(k)^{-1} &= P^-(k)^{-1} + \sum_{i=1}^N [P_i(k)^{-1} - P_i^-(k)^{-1}] \\
 \hat{x}(k) &= P(k)[P^-(k)^{-1}\hat{x}^-(k) + \sum_{i=1}^N (P_i(k)^{-1}\hat{x}_i(k) - P_i^-(k)^{-1}\hat{x}_i^-(k))]
 \end{aligned} \quad (57)$$

and also in terms of the information state vector and of the information state covariance matrix

$$\begin{aligned}
 \hat{y}(k) &= \hat{y}^-(k) + \sum_{i=1}^N (\hat{y}_i(k) - \hat{y}_i^-(k)) \\
 Y(k) &= Y^-(k) + \sum_{i=1}^N [Y_i(k) - Y_i^-(k)]
 \end{aligned} \quad (58)$$

State estimation fusion based on the Unscented Information Filter (UIF) is fault tolerant. From Eq. (57) it can be seen that if a local filter (processing station) fails, then the local covariance matrices and local estimates provided by the rest of the filters will enable a reliable calculation of the system's state vector. Moreover, it is and computationally efficient comparing to centralized filters and results in enhanced estimation accuracy.

4. Distributed Particle Filter

4.1 Particle Filtering at local processing units

4.1.1 The particle approximation of probability density functions

One can also estimate the state vector of the UAVs that constitute the multi-UAV system through the fusion of estimates provided by local Particle Filters. This can be succeeded using the Distributed Particle Filter (DPF). First, the functioning of the local Particle Filters will be explained. Each local Particle Filter generates an estimation of the UAV's state vector by fusing measurements from distributed sensors. Particle Filtering is a method for state estimation that is not dependent on the probability density function of the measurements. In the general case the equations of the optimal filter used for the calculation of the state-vector of a dynamical system do not have an explicit solution. This happens for instance when the process noise and the noise of the output measurement do not follow a Gaussian distribution. In that case approximation through Monte-Carlo methods can be used (Thrun et al. 2005). A sampling of size N is assumed, i.e. N i.i.d. (independent identically distributed) variables $\xi^1, \xi^2, \dots, \xi^N$. This sampling follows the p.d.f. $p(x)$ i.e. $\xi^{1:N} \sim p(x)$. Instead of $p(x)$ the function $p(x) \simeq p^N(x) = \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}(x)$ can be used. It is assumed that all points ξ^i have an equal weighted contribution to the approximation of $p(x)$. A more general approach would be if weight factors were assigned to the points ξ^i , which will also satisfy the normality condition $\sum_{i=1}^N w^i = 1$. In the latter case

$$p(x) \simeq p^N(x) = \sum_{i=1}^N w^i \delta_{\xi^i}(x) \quad (59)$$

If $p(\xi^i)$ is known then the probability $P(x)$ can be approximated using the discrete values of the p.d.f. $p(\xi^i) = w^i$. If sampling over the p.d.f. $p(x)$ is unavailable, then one can use a p.d.f. $\bar{p}(x)$ with similar support set, i.e. $p(x) = 0 \Rightarrow \bar{p}(x) = 0$. Then it holds

$$E(\phi(x)) = \int \phi(x) p(x) dx = \int \phi(x) \bar{p}(x) \frac{p(x)}{\bar{p}(x)} dx. \text{ If the } N \text{ samples of } \bar{p}(x) \text{ are available at the points}$$

$\tilde{\xi}^1 \dots \tilde{\xi}^N$, i.e. $\bar{p}(\tilde{\xi}^i) = \delta_{\tilde{\xi}^i}(x)$ and the weight coefficients w^i are defined as $w^i = \frac{p(\tilde{\xi}^i)}{\bar{p}(\tilde{\xi}^i)}$, then it is

easily shown that

$$E(\phi(x)) \simeq \sum_{i=1}^N w^i \phi(\tilde{\xi}^i), \text{ where } \begin{cases} \tilde{\xi}^{1:N} \sim \bar{p}(x) \\ w^i = p(\tilde{x}^i) / \bar{p}(\tilde{x}^i) \end{cases} \quad (60)$$

The meaning of Eq. (60) is as follows: assume that the p.d.f. $p(x)$ is unknown (target distribution), however the p.d.f. $\bar{p}(x)$ (importance law) is available. Then, it is sufficient to sample on $\bar{p}(x)$ and find the associated weight coefficients w^i so as to calculate $E(\phi(x))$.

4.1.2 The prediction stage

As in the case of the Kalman Filter or the Extended Kalman Filter the particles filter consists of the measurement update (correction stage) and the time update (prediction stage) (Rigatos 2009b),(Thrun wt al. 2005). The prediction stage calculates $p(x(k) | Z^-)$ where $Z^- = \{z(1), z(2), \dots, z(n-1)\}$ according to Eq. (59). It holds that:

$$p(x(k-1) | Z^-) = \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-1}^i} (x(k-1)) \quad (61)$$

while from Bayes formula it holds $p(x(k) | Z^-) = \int p(x(k) | x(k-1))p(x(k-1) | Z^-)dx$. Using also Eq. (61) one finally obtains

$$p(x(k) | Z^-) = \sum_{i=1}^N w_{k-1}^i \delta_{\xi_{k-1}^i} (x(k)) \quad (62)$$

with $\xi_{k-1}^i \sim p(x(k) | x(k-1)) = \xi_{k-1}^i$

The meaning of Eq. (62) is as follows: the state equation of the system is executed N times, starting from the N previous values of the state vectors $x(k-1) = \xi_{k-1}^i$

$$\begin{aligned} \hat{x}(k+1) &= \phi(\hat{x}(k)) + L(k)u(k) + w(k) \\ z(k) &= \gamma(\hat{x}(k)) + v(k) \end{aligned} \quad (63)$$

Thus estimations of the current value of the state vector $\hat{x}(k)$ are obtained, and consequently the mean value of the state vector will be given from Eq. (62). This means that the value of the state vector which is calculated in the prediction stage is the result of the weighted averaging of the state vectors which were calculated after running the state equation, starting from the N previous values of the state vectors ξ_{k-1}^i .

4.1.3 The correction stage

The a-posteriori probability density is found using Eq. (62). Now a new position measurement $z(k)$ is obtained and the objective is to calculate the corrected probability density $p(x(k) | Z)$, where $Z = \{z(1), z(2), \dots, z(k)\}$. From Bayes law it holds that

$$p(x(k) | Z) = \frac{p(Z | x(k))p(x(k))}{p(Z)}$$
 which can be also written as

$$p(x(k) | Z) = \frac{p(z(k) | x(k))p(x(k) | Z^-)}{\int p(z(k) | x(k), Z^-)p(x(k) | Z^-)dx} \quad (64)$$

Substituting Eq. (62) into Eq. (64) and after intermediate calculations one finally obtains

$$p(x(k) | Z) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i} (x(k))$$

where $w_k^i = \frac{w_{k-1}^i p(z(k) | x(k) = \xi_{k-1}^i)}{\sum_{j=1}^N w_{k-1}^j p(z(k) | x(k) = \xi_{k-1}^j)}$ (65)

Eq. (65) denotes the corrected value for the state vector. The recursion of the Particle Filter proceeds in a way similar to the update of the Kalman Filter or the Extended Kalman Filter, i.e.:

- *Measurement update:* Acquire $z(k)$ and compute

new value of the state vector

$$p(x(k) | Z) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i}^i(x(k))$$

with corrected weights

(66)

$$w_k^i = \frac{w_{k-1}^i p(z(k) | x(k) = \xi_{k-1}^i)}{\sum_{j=1}^N w_{k-1}^j p(z(k) | x(k) = \xi_{k-1}^j)} \text{ and } \xi_k^i = \xi_{k-1}^i$$

Resampling for substitution of the degenerated particles

- *Time update:* compute state vector $x(k + 1)$ according to the pdf

$$p(x(k + 1) | Z) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i}^i(x(k))$$

(67)

where $\xi_k^i \sim p(x(k + 1) | x(k) = \xi_k^i)$

The stages of state vector estimation with the use of the Particle Filtering algorithm are depicted in Fig. 6.

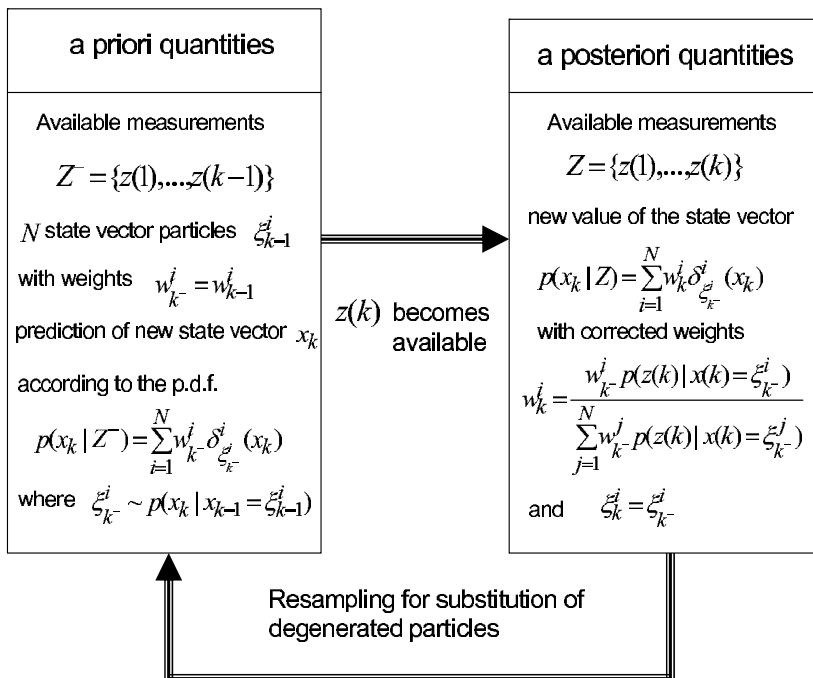


Fig. 6. Schematic diagram of the Particle Filter loop

4.1.4 Resampling issues in particle filtering

The algorithm of particle filtering which is described through Eq. (62) and Eq. (65) has a significant drawback: after a certain number of iterations k , almost all the weights w_k^i become 0. In the ideal case all the weights should converge to the value $\frac{1}{N}$, i.e. the particles should have the same significance. The criterion used to define a sufficient number of

particles is $N_k^{\text{eff}} = \frac{1}{\sum_{i=1}^N w_k^i} \in [1, N]$. When N_k^{eff} is close to value N then all particles have

almost the same significance. However using the algorithm of Eq. (62) and Eq. (65) results in $N_k^{\text{eff}} \rightarrow 1$, which means that the particles are degenerated, i.e. they lose their effectiveness.

Therefore, it is necessary to modify the algorithm so as to assure that degeneration of the particles will not take place (Rigatos 2009a), (Thrun et al. 2005), (Zhang et al. 2005).

When N_k^{eff} is small then most of the particles have weights close to 0 and consequently they have a negligible contribution to the estimation of the state vector. To overcome this drawback of the PF algorithm weakens such particles in favor of particles that have a non-negligible contribution. Therefore, the particles of low weight factors are removed and their place is occupied by duplicates of the particles with high weight factors. The total number of particles remains unchanged (equal to N) and therefore this procedure can be viewed as a "resampling" or "redistribution" of the particles set.

The particles resampling presented above maybe slow if not appropriately tuned. There are improved versions of it which substitute the particles of low importance with those of higher importance. A first choice would be to perform a multinomial resampling. N particles are chosen between $\{\xi_k^1, \dots, \xi_k^N\}$ and the corresponding weights are w_k^1, \dots, w_k^N . The number of times each particle is selected is given by $[j_1, \dots, j_n]$. Thus a set of N particles is again created, the elements of which are chosen after sampling with the discrete distribution

$\sum_{i=1}^N w_k^i \delta_{\xi_k^i}(x)$. The particles $\{\xi_k^1, \dots, \xi_k^N\}$ are chosen according to the probabilities $\{w_k^1, \dots, w_k^N\}$. The selected particles are assigned with equal weights $\frac{1}{N}$.

Although sorting of the particles' weights is not necessary for the convergence of the particle filter algorithm, there are variants of the resampling procedure of $(\xi_k^i, w_k^i, i = 1, \dots, N)$ which are based on previous sorting in decreasing order of the particles' weights (efficient sorting approaches make the complexity of the particle filtering to be $O(N \log(N))$, while the avoidance of resampling results in a faster algorithm of complexity $O(N)$). Sorting of particles' weights gives $w^{s[1]} > w^{s[2]} > \dots > w^{s[N]}$. A random numbers generator is evoked and the resulting numbers $u^{i:N} \sim U[0,1]$ fall in the partitions of the interval $[0,1]$. The width of these partitions is w^i and thus a redistribution of the particles is generated. For instance, in a wide partition of width w^j will be assigned more particles than to a narrow partition of width w^m . A detailed analysis on the tuning of the resampling procedure in Particle Filtering has been given in (Rigatos 2009a).

4.2 Distributed Particle Filtering for state estimation fusion

The Distributed Particle Filter performs fusion of the state vector estimates which are provided by the local Particle Filters. This is succeeded by fusing the discrete probability

density functions of the local Particle Filters into a common probability distribution of the system's state vector. Without loss of generality fusion between two estimates which are provided by two different probabilistic estimators (particle filters) is assumed. This amounts to a multiplication and a division operation to remove the common information, and is given by (Ong et al. 2008), (Ong et al. 2006)

$$p(x(k)|Z_A \cup Z_B) \propto \frac{p(x(k)|Z_A)p(x(k)|Z_B)}{p(x(k)|Z_A \cap Z_B)} \quad (68)$$

where Z_A is the sequence of measurements associated with the i -th processing unit and Z_B is the sequence of measurements associated with the j -th measurement unit. In the implementation of distributed particle filtering, the following issues arise:

1. Particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter). Therefore a point-to-point application of Eq. (68) is not possible.
2. The communication of particles representation (i.e. local particle sets and associated weight sets) requires significantly more bandwidth compared to other representations, such as Gaussian mixtures.

Fusion of the estimates provided by the local particle filters (located at different processing units) can be performed through the following stages. First, the discrete particle set of Particle Filter A (Particle Filter B) is transformed into a continuous distribution by placing a Gaussian kernel over each sample (Fig. 7) (Musso et al. 2001)

$$K_h(x) = h^2 K(x) \quad (69)$$

where $K()$ is the rescaled Kernel density and $h > 0$ is the scaling parameter. Then the continuous distribution A (B) is sampled with the other particles set B (A) to obtain the new importance weights, so that the weighted sample corresponds to the numerator of Eq. (68) (Fig. 8). Such a conversion from a discrete particle probability distribution functions

$\sum_{i=1}^N w_A^{(i)} \delta(x_A^{(i)})$ ($\sum_{i=1}^N w_B^{(i)} \delta(x_B^{(i)})$) into continuous distributions is denoted as

$$\sum_{i=1}^N w_A^{(i)} \delta(x_A^{(i)}) \rightarrow p_A(x) \quad \left(\sum_{i=1}^N w_B^{(i)} \delta(x_B^{(i)}) \rightarrow p_B(x) \right) \quad (70)$$

The common information appearing in the processing units A and B should not be taken into account in the joint probability distribution which is created after fusing the local probability densities of A and B. This means that in the joint p.d.f. one should sample with importance weights calculated according to Eq. (68). The objective is then to create an importance sampling approximation for the joint distribution that will be in accordance to Eq. (68). A solution to this can be obtained through Monte Carlo sampling and suitable selection of the so-called "proposal distribution" (Ong et al. 2008), (Ong et al. 2006)]

According to the above, for the joint distribution the idea behind Monte Carlo sampling is to draw N i.i.d samples from the associated probability density function $p(x)$, such that the target density is approximated by a point-mass function of the form

$$p(x) \simeq \sum_{i=1}^N w_k^{(i)} \delta(x_k^{(i)}) \quad (71)$$

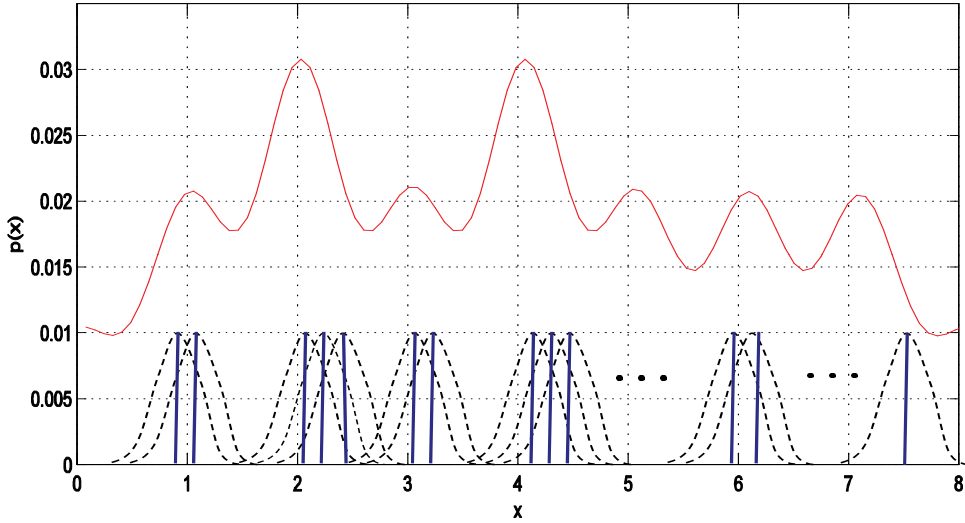


Fig. 7. Conversion of the particles discrete probability density function to a continuous distribution, after allocating a Gaussian kernel over each particle

where $\delta(x_k^{(i)})$ is a Dirac delta mass located at $x_k^{(i)}$. Then the expectation of some function $f(x)$ with respect to the pdf $p(x)$ is given by

$$I(f) = E_{p(x)}[f(x)] = \int f(x)p(x)dx \tag{72}$$

the Monte-Carlo approximation of the integral with samples is then

$$I_N(f) = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \tag{73}$$

where $x^{(i)} \approx p(X)$ and $I_N(f) \rightarrow I(f)$ for $N \rightarrow \infty$. since, the true probability distribution $p(x)$ is hard to sample from, the concept of importance sampling is to select a proposal distribution $\bar{p}(x)$ in place of $p(x)$, with the assumption that $\bar{p}(x)$ includes the support space of $p(x)$. Then the expectation of function $f(x)$, previously given in Eq. (72), is now calculated as

$$I(f) = \int f(x) \frac{p(x)}{\bar{p}(x)} \bar{p}(x) dx = \int f(x) w(x) \bar{p}(x) dx \tag{74}$$

where $w(x)$ are the importance weights

$$w(x) = \frac{p(x)}{\bar{p}(x)} \tag{75}$$

Then the Monte-Carlo estimation of the mean value of function $f(x)$ becomes

$$I_N(f) = \sum_{i=1}^N f(x^{(i)}) w(x^{(i)}) \tag{76}$$

For the division operation, the desired probability distribution is

$$p(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})} \quad (77)$$

In that case the important weights of the fused probability density functions become

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})\bar{p}(x^{(i)})} \quad (78)$$

which is then normalized so that $\sum_{i=1}^N w(x^{(i)}) = 1/N$, where N is the number of particles. The next step is to decide what will be the form of the proposal distribution $\bar{p}(x)$. A first option is to take $\bar{p}(x)$ to be a uniform distribution, with a support that covers both of the support sets of the distributions A and B .

$$\bar{p}(x) = U(x) \quad (79)$$

Then the sample weights $\bar{p}(x^{(i)})$ are all equal at a constant of value C . Hence the importance weights are

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})C} \quad (80)$$

Another suitable proposal distribution that takes more into account the new information received (described as the probability distribution of the second processing unit) is given by

$$\bar{p}(x) = p_B(x) \quad (81)$$

and the important weights are then adjusted to be

$$w(x^{(i)}) = \frac{p_A(x^{(i)})}{p_B(x^{(i)})^2} \quad (82)$$

5. Nonlinear control for autonomous UAV navigation

5.1 Kinematic model of the UAV

For the design of the autonomous navigation system of the UAVs a suitable control scheme has to be chosen. In this control loop there will be processing of the estimated UAV state vector, as obtained through the distributed filtering algorithms which were presented in Sections 2 to 4. To this end, the kinematic model the kinematic model of the UAVs has to be analyzed first. Based on this kinematic model a flatness-based controller will be derived. The UAV dynamics suggest the following structure for constant altitude manoeuvres (Léchevin & Rabbath 2006):

$$\begin{aligned} \dot{x} &= v \cos(\theta), \dot{y} = v \sin(\theta), \dot{\theta} = u_1 \\ \dot{v} &= u_2, \dot{h} = 0 \end{aligned} \quad (83)$$

where (x,y) is the desired inertial position of the UAV, θ is the UAV's heading, v is the UAV's velocity, h is the UAV's attitude, and u_1, u_2 are constrained by the dynamic capability of the UAVs namely the heading rate constraint and the acceleration constraint respectively.

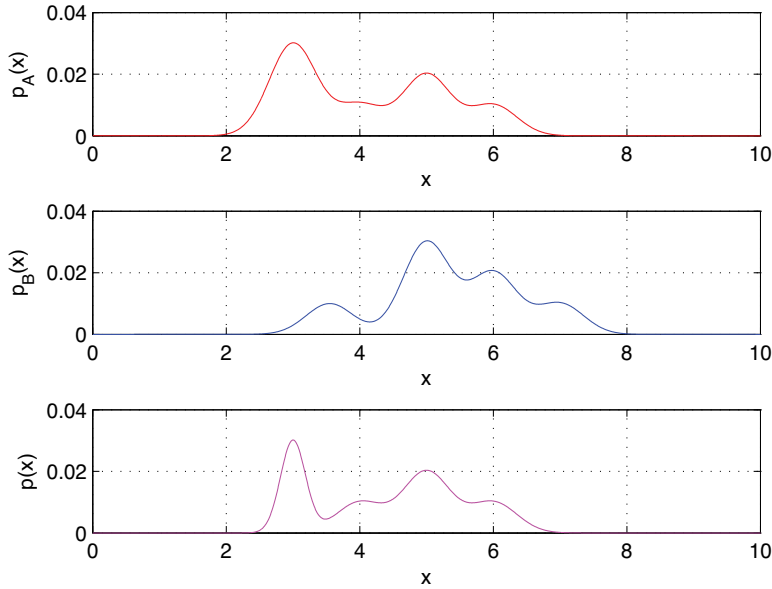


Fig. 8. Fusion of the probability density functions produced by the local particle filters

An inertial measurement unit (IMU) of a UAV usually consists of a three axis gyroscope and a three axis accelerometer. A vision sensor can be also mounted underneath the body of the UAV and is used to extract points of interest in the environment. The UAV also carries a barometric pressure sensor for aiding of the platform attitude estimation. A GPS sensor, can be also mounted on the board. The sensor data is filtered and fused to obtain estimates of the desired entities such as platform and feature position (Vissière et al. 2008).

5.2 Differential flatness for finite dimensional systems

Flatness-based control is proposed for steering the UAV along a desirable trajectory (Oriolo et al. 2002), (Villagra et al. 2007), (Fliess et al. 1999). The main principles of flatness-based control are as follows: A finite dimensional system is considered. This can be written in the general form of an ODE, i.e.

$$S_i(w, \dot{w}, \ddot{w}, \dots, w^{(i)}), \quad i = 1, 2, \dots, q \quad (84)$$

The quantity w denotes the system variable while $w^{(i)}, i = 1, 2, \dots, q$ are its derivatives (these can be for instance the elements of the system's state vector). The system of Eq. (1) is said to be differentially flat if there exists a collection of m functions $y = (y_1, \dots, y_m)$ of the system variables $w_i, i = 1, \dots, s$ and of their time-derivatives, i.e.

$$y_i = \phi(w, \dot{w}, \ddot{w}, \dots, w^{\alpha_i}), \quad i = 1, \dots, m \quad (85)$$

such that the following two conditions are satisfied (Fliess et al. 1999), (Rigatos 2008):

1. There does not exist any differential relation of the form

$$R(y, \dot{y}, \dots, y^{(\beta)}) = 0 \quad (86)$$

which implies that the derivatives of the flat output are not coupled in the sense of an ODE, or equivalently it can be said that the flat output is differentially independent.

2. All system variables, i.e. the components of w (elements of the system's state vectors) can be expressed using only the flat output y and its time derivatives

$$w_i = \psi_i(y, \dot{y}, \dots, y^{(i)}), i = 1, \dots, s \quad (87)$$

An equivalent definition of differentially flat systems is as follows:

Definition: The system $\dot{x} = f(x, u)$, $x \in R^n$, $u \in R^m$ is differentially flat if there exist relations $h : R^n \times R^m \rightarrow R^m$, $\phi : (R^m)^r \rightarrow R^n$ and $\psi : (R^m)^{r+1} \rightarrow R^m$, such that $y = h(x, u, \dot{u}, \dots, u^{(r)})$, $x = \phi(y, \dot{y}, \dots, y^{(r-1)})$ and $u = \psi(y, \dot{y}, \dots, y^{(r-1)}, y^{(r)})$. This means that all system dynamics can be expressed as a function of the flat output and its derivatives, therefore the state vector and the control input can be written as $x(y) = \phi(y(t), \dot{y}(t), \dots, y^{(r)}(t))$ and $u = \psi(y(t), \dot{y}(t), \dots, y^{(r)}(t))$.

It is noted that for linear systems the property of differential flatness is equivalent to that of controllability.

5.3 Differential flatness of the UAV kinematic model

It is assumed that the helicopter-like UAV, performs manoeuvres at a constant altitude. Then, from Eq. (83) one can obtain the following description for the UAV kinematics

$$\dot{x} = v \cos(\theta), \dot{y} = v \sin(\theta), \dot{\theta} = \frac{v}{l} \tan(\phi) \quad (88)$$

where using the analogous of the unicycle robot v is the velocity of the UAV, l is the UAV's length, θ is the UAV's orientation (angle between the transversal axis of the UAV and axis OX), and ϕ is a steering angle. The flat output is the cartesian position of the UAV's center of gravity, denoted as $\eta = (x, y)$, while the other model parameters can be written as:

$$v = \pm \|\dot{\eta}\| \quad \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} = \frac{\dot{\eta}}{v} \quad \tan(\phi) = l \det(\dot{\eta} \dot{\eta}) / v^3 \quad (89)$$

These formulas show simply that θ is the tangent angle of the curve traced by P and $\tan(\phi)$ is the associated curvature. With reference to a generic driftless nonlinear system

$$\dot{q}, q \in R^n, w \in R^m \quad (90)$$

dynamic feedback linearization consists in finding a feedback compensator of the form

$$\begin{aligned} \dot{\xi} &= \alpha(q, \xi) + b(q, \xi)u \\ w &= c(q, \xi) + d(q, \xi)u \end{aligned} \quad (91)$$

with state $\xi \in R^v$ and input $u \in R^m$, such that the closed-loop system of Eq. (90) and Eq. (91) is equivalent under a state transformation $z = T(q, \xi)$ to a linear system. The starting point is the definition of a m -dimensional output $\eta = h(q)$ to which a desired behavior can be assigned. One then proceeds by successively differentiating the output until the input appears in a non-singular way. If the sum of the output differentiation orders equals the dimension $n + v$ of the extended state space, full input-state-output linearization is obtained

(In this case η is also called a flat output). The closed-loop system is then equivalent to a set of decoupled input-output chains of integrators from u_i to η_i . The exact linearization procedure is illustrated for the unicycle model of Eq. (21). As flat output the coordinates of the center of gravity of the vehicle is considered $\eta = (x, y)$. Differentiation with respect to time then yields (Oriolo et al. 2002), (Rigatos 2008)

$$\dot{\eta} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \end{pmatrix} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (92)$$

showing that only v affects $\dot{\eta}$, while the angular velocity ω cannot be recovered from this first-order differential information. To proceed, one needs to add an integrator (whose state is denoted by ξ) on the linear velocity input

$$v = \xi, \quad \dot{\xi} = \alpha \Rightarrow \dot{\eta} = \xi \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad (93)$$

where α denotes the linear acceleration of the UAV. Differentiating further one obtains

$$\ddot{\eta} = \dot{\xi} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} + \xi \dot{\theta} \begin{pmatrix} \sin(\theta) \\ \cos(\theta) \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\xi \sin(\theta) \\ \sin(\theta) & \xi \cos(\theta) \end{pmatrix} \begin{pmatrix} \alpha \\ \omega \end{pmatrix} \quad (94)$$

and the matrix multiplying the modified input (α, ω) is nonsingular if $\xi \neq 0$. Under this assumption one defines

$$\begin{pmatrix} \alpha \\ \omega \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\xi \sin(\theta) \\ \sin(\theta) & \xi \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (95)$$

$\ddot{\eta}$ is denoted as

$$\ddot{\eta} = \begin{pmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = u \quad (96)$$

which means that the desirable linear acceleration and the desirable angular velocity can be expressed using the transformed control inputs u_1 and u_2 . Then, the resulting dynamic compensator is (return to the initial control inputs v and ω)

$$\begin{aligned} \dot{\xi} &= u_1 \cos(\theta) + u_2 \sin(\theta) \\ v &= \xi \\ \omega &= \frac{u_2 \cos(\theta) - u_1 \sin(\theta)}{\xi} \end{aligned} \quad (97)$$

Being $\xi \in \mathbb{R}$, it is $n + v = 3 + 1 = 4$, equal to the output differentiation order in Eq. (29). In the new coordinates

$$\begin{aligned} z_1 &= x \\ z_2 &= y \\ z_3 &= \dot{x} = \xi \cos(\theta) \\ z_4 &= \dot{y} = \xi \sin(\theta) \end{aligned} \quad (98)$$

The extended system is thus fully linearized and described by the chains of integrators, in Eq. (29), and can be rewritten as

$$\begin{aligned}\ddot{z}_1 &= u_1 \\ \ddot{z}_2 &= u_2\end{aligned}\quad (99)$$

The dynamic compensator of Eq. (97) has a potential singularity at $\xi = v = 0$, i.e. when the UAV is not moving, which is a case never met when the UAV is in flight. It is noted however, that the occurrence of such a singularity is structural for non-holonomic systems. In general, this difficulty must be obviously taken into account when designing control laws on the equivalent linear model.

A nonlinear controller for output trajectory tracking, based on dynamic feedback linearization, is easily derived. Assume that the UAV must follow a smooth trajectory $(x_d(t), y_d(t))$ which is persistent, i.e. for which the nominal velocity $v_d = (\dot{x}_d^2 + \dot{y}_d^2)^{\frac{1}{2}}$ along the trajectory never goes to zeros (and thus singularities are avoided). On the equivalent and decoupled system of Eq. (32), one can easily design an exponentially stabilizing feedback for the desired trajectory, which has the form

$$\begin{aligned}u_1 &= \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + k_{p1}(y_d - y) + k_{d1}(\dot{y}_d - \dot{y})\end{aligned}\quad (100)$$

and which results in the following error dynamics for the closed-loop system

$$\begin{aligned}\ddot{e}_x + k_{d1}\dot{e}_x + k_{p1}e_x &= 0 \\ \ddot{e}_y + k_{d2}\dot{e}_y + k_{p2}e_y &= 0\end{aligned}\quad (101)$$

where $e_x = x - x_d$ and $e_y = y - y_d$. The proportional-derivative (PD) gains are chosen as $k_{p1} > 0$ and $k_{d1} > 0$ for $i = 1, 2$. Knowing the control inputs u_1, u_2 , for the linearized system one can calculate the control inputs v and ω applied to the UAV, using Eq. (91). The above result is valid, provided that the dynamic feedback compensator does not meet the singularity. In the general case of design of flatness-based controllers, the following theorem assures the avoidance of singularities in the proposed control law (Oriolo et al. 2002):

Theorem: Let $\lambda_{11}, \lambda_{12}$ and $\lambda_{21}, \lambda_{22}$ be respectively the eigenvalues of two equations of the error dynamics, given in Eq. (91). Assume that, for $i = 1, 2$ it is $\lambda_{i1} < \lambda_{i2} < 0$ (negative real eigenvalues), and that λ_{i2} is sufficiently small. If

$$\min_{t \geq 0} \left\| \begin{pmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \end{pmatrix} \right\| \geq \begin{pmatrix} \dot{e}_x^0 \\ \dot{e}_y^0 \end{pmatrix}\quad (102)$$

with $\dot{e}_x^0 = \dot{e}_x(0) \neq 0$ and $\dot{e}_y^0 = \dot{e}_y(0) \neq 0$, then the singularity $\xi = 0$ is never met.

6. Simulation tests

6.1 Autonomous UAV navigation with Extended Information Filtering

It was assumed that $m = 2$ helicopter models were monitored by $n = 2$ different ground stations. At each ground station an Extended Kalman Filter was used to track each UAV. By

fusing the measurements provided by the sensors mounted on each UAV, each local EKF was able to produce an estimation of a UAV's motion. Next, the state estimates obtained by the pair local EKFs associated with each UAV were fused with the use of the Extended Information Filter. This fusion-based state estimation scheme is depicted in Fig. 2. As explained in Section 2 the weighting of the state estimates of the local EKFs was performed using the local information matrices. The distributed filtering architecture is shown in Fig. 9.

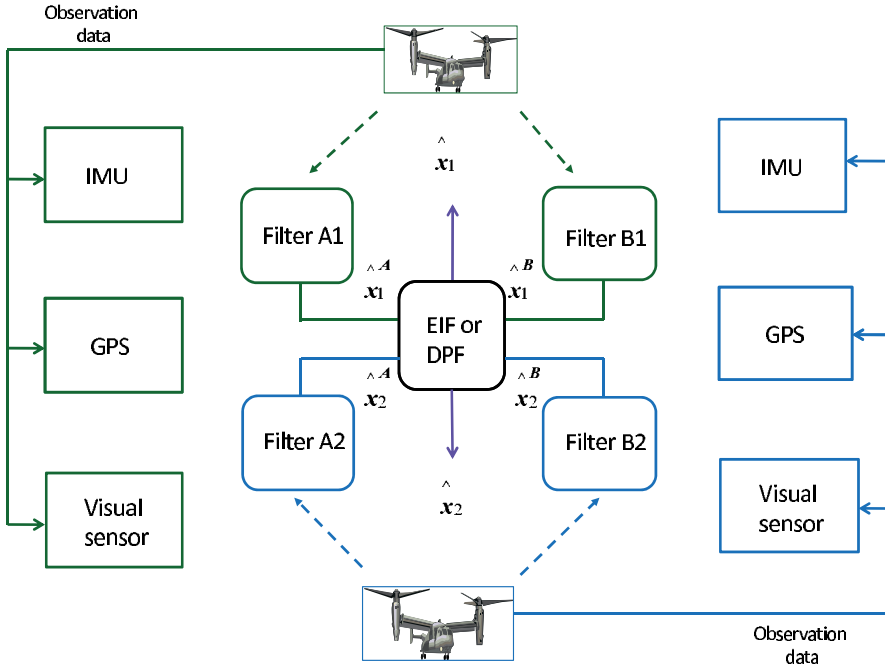


Fig. 9. Distributed Filtering over WSN

Next, some details will be given about the local EKF design for the UAV model of Eq. (88). The UAV's continuous-time kinematic equation is:

$$\dot{x}(t) = v(t)\cos(\theta(t)), \dot{y}(t) = v(t)\sin(\theta(t)), \dot{\theta}(t) = \omega(t) \tag{103}$$

The IMU system provides measurements of the UAV's position $[x, y]$ and the UAV's orientation angle θ over a sampling period T . These sensors are used to obtain an estimation of the displacement and the angular velocity of the UAV $v(t)$ and $\omega(t)$, respectively. The IMU sensors can introduce incremental errors, which result in an erroneous estimation of the orientation θ . To improve the accuracy of the UAV's localization, measurements from the GPS (or visual sensors) can be used. On the other hand, the GPS on its own is not always reliable since its signal can be intermittent. Therefore, to succeed accurate localization of the UAV it is necessary to fuse the GPS measurements with the IMU measurements of the UAV or with measurements from visual sensors (visual odometry).

The inertial coordinates system OXY is defined. Furthermore the coordinates system $O'X'Y'$ is considered (Fig. 10). $O'X'Y'$ results from OXY if it is rotated by an angle θ . The coordinates of the center of symmetry of the UAV with respect to OXY are (x, y) , while the coordinates of

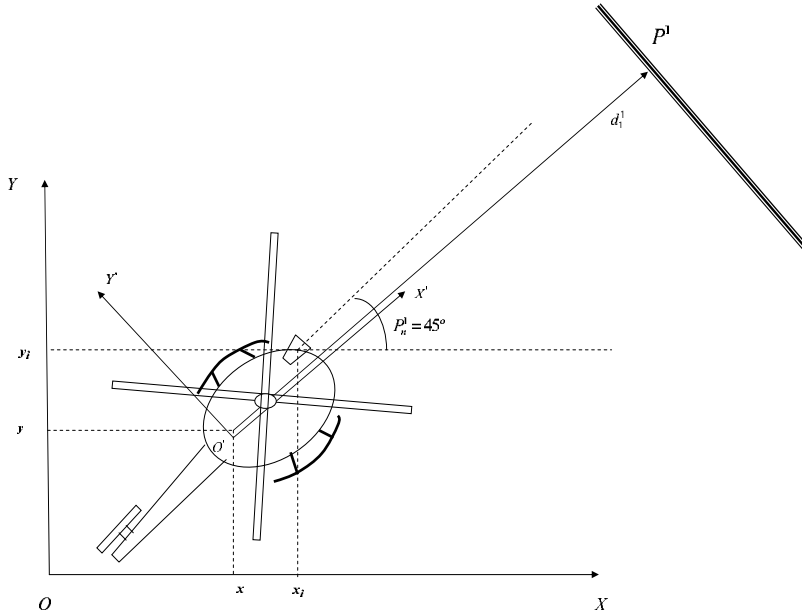


Fig. 10. Reference frames for the UAV

the GPS or visual sensor that is mounted on the UAV, with respect to $O'X'Y'$ are x'_i, y'_i . The orientation of the GPS (or visual sensor) with respect to OXY is θ'_i . Thus the coordinates of the GPS or visual sensor with respect to OXY are (x_i, y_i) and its orientation is θ_i , and are given by:

$$\begin{aligned} x_i(k) &= x(k) + x'_i \sin(\theta(k)) + y'_i \cos(\theta(k)) \\ y_i(k) &= y(k) - x'_i \cos(\theta(k)) + y'_i \sin(\theta(k)) \\ \theta_i(k) &= \theta(k) + \theta_i \end{aligned} \quad (104)$$

For manoeuvres at constant altitude the GPS measurement (or the visual sensor measurement) can be considered as the measurement of the distance from a reference surface P^j . A reference surface P^j in the UAVs 2D flight area can be represented by P_r^j and P_n^j , where (i) P_r^j is the normal distance of the plane from the origin O, (ii) P_n^j is the angle between the normal line to the plane and the x-direction.

The GPS sensor (or visual sensor i) is at position $x_i(k), y_i(k)$ with respect to the inertial coordinates system OXY and its orientation is $\theta_i(k)$. Using the above notation, the distance of the GPS (or visual sensor i), from the plane P^j is represented by P_r^j, P_n^j (see Fig. 10):

$$d_i^j(k) = P_r^j - x_i(k) \cos(P_n^j) - y_i(k) \sin(P_n^j) \quad (105)$$

Assuming a constant sampling period $\Delta t_k = T$ the measurement equation is $z(k+1) = \gamma(x(k)) + v(k)$, where $z(k)$ is the vector containing GPS (or visual sensor) and IMU measures and $v(k)$ is a white noise sequence $\sim N(0, R(kT))$.

By definition of the measurement vector one has that the output function is $\gamma(x(k)) = [x(k), y(k), \theta(k), d_1(k)]^T$. The UAV state is $[x(k), y(k), \theta(k)]^T$ and the control input is denoted by

$U(k) = [v(k), \omega(k)]^T$. To obtain the Extended Kalman Filter (EKF), the kinematic model of the UAV is linearized about the estimates $\hat{x}(k)$ and $\hat{x}^-(k)$ the control input $U(k-1)$ is applied.

The *measurement update* of the EKF is

$$\begin{aligned} K(k) &= P^-(k)J_\gamma^T(\hat{x}^-(k))[J_\gamma(\hat{x}^-(k))P^-(k)J_\gamma^T(\hat{x}^-(k)) + R(k)]^{-1} \\ \hat{x}(k) &= \hat{x}^-(k) + K(k)[z(k) - \gamma(\hat{x}^-(k))] \\ P(k) &= P^-(k) - K(k)J_\gamma^T P^-(k) \end{aligned}$$

The *time update* of the EKF is

$$\begin{aligned} P^-(k+1) &= J_\phi(\hat{x}(k))P(k)J_\phi^T(\hat{x}(k)) + Q(k) \\ \hat{x}^-(k+1) &= \phi(\hat{x}(k)) + L(k)U(k) \end{aligned}$$

$$\text{where } L(k) = \begin{pmatrix} T\cos(\theta(k)) & 0 \\ T\sin(\theta(k)) & 0 \\ 0 & T \end{pmatrix} \text{ and } J_\phi(\hat{x}(k)) = \begin{pmatrix} 1 & 0 & -v(k)\sin(\theta)T \\ 0 & 1 & -v(k)\cos(\theta)T \\ 0 & 0 & 1 \end{pmatrix} \quad (106)$$

while $Q(k) = \text{diag}[\sigma^2(k), \sigma^2(k), \sigma^2(k)]$, with $\sigma^2(k)$ chosen to be 10^{-3} and $\phi(\hat{x}(k)) = [\hat{x}(k), \hat{y}(k), \hat{\theta}(k)]^T$, $\gamma(\hat{x}(k)) = [\hat{x}(k), \hat{y}(k), \hat{\theta}(k), d(k)]^T$, i.e.

$$\gamma(\hat{x}(k)) = \begin{pmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \\ P_r^j - x_i(k)\cos(P_n^j) - y_i(k)\sin(P_n^j) \end{pmatrix} \quad (107)$$

In the calculation of the observation equation Jacobian one gets

$$J_\gamma^T(\hat{x}^-(k)) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\cos(P_n^j) & -\sin(P_n^j) & \{x'_i\cos(\theta - P_n^j) - y'_i\sin(\theta - P_n^j)\} \end{pmatrix} \quad (108)$$

The UAV is steered by a dynamic feedback linearization control algorithm which is based the flatness-based control analyzed in Section 5:

$$\begin{aligned} u_1 &= \ddot{x}_d + K_{p_1}(x_d - x) + K_{d_1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + K_{p_2}(y_d - y) + K_{d_2}(\dot{y}_d - \dot{y}) \\ \dot{\xi} &= u_1\cos(\theta) + u_2\sin(\theta) \\ v = \xi, \quad \omega &= \frac{u_2\cos(\theta) - u_1\sin(\theta)}{\xi} \end{aligned} \quad (109)$$

Under the control law of Eq. (109) the dynamics of the tracking error finally becomes

$$\begin{aligned}\ddot{e}_x + K_{d_1}\dot{e}_x + K_{p_1}e_x &= 0 \\ \ddot{e}_y + K_{d_2}\dot{e}_y + K_{p_2}e_y &= 0\end{aligned}\quad (110)$$

where $e_x = x - x_d$ and $e_y = y - y_d$. The proportional-derivative (PD) gains are chosen as K_{p_1} and K_{d_1} , for $i = 1, 2$.

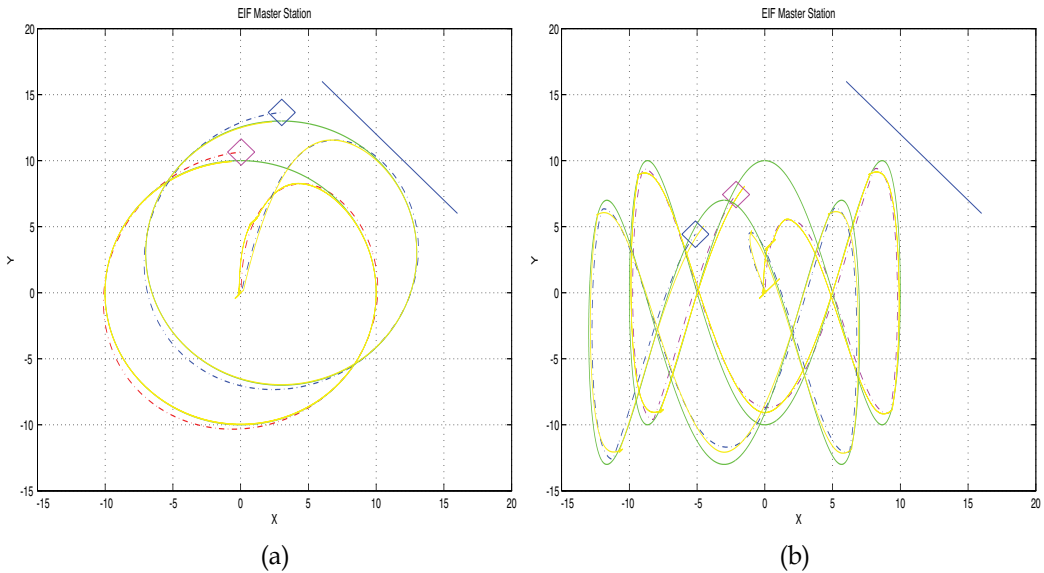


Fig. 11. Autonomous navigation of the multi-UAV system when the UAVs state vector is estimated with the use of the Extended Information Filter (a) tracking of circular reference trajectory (b) tracking of a curve-shaped reference trajectory

Results on the performance of the Extended Information Filter in estimating the state vectors of multiple UAVs when observed by distributed processing units is given in Fig. 11. Using distributed EKFs and fusion through the Extended Information Filter is more robust comparing to the centralized EKF since (i) if a local processing unit is subject to a fault then state estimation becomes is still possible and can be used for accurate localization of the UAV, as well as for tracking of desirable flight paths, (ii) communication overhead remains low even in the case of a large number of distributed measurement units, because the greatest part of state estimation is performed locally and only information matrices and state vectors are communicated between the local processing units, (iii) the aggregation performed on the local EKF also compensates for deviations in state estimates of local filters (which can be due to linearization errors).

6.2 Autonomous UAV navigation with Distributed Particle Filtering

Details on the implementation of the local particle filters are given first. Each local particle filter provides an estimation of the UAV's state vector using sensor fusion. The UAV model described in Eq. (103), and the control law given in Eq. (109) are used again.

The *measurement update* of the PF is $p(x(k)|Z) = \sum_{i=1}^N w_k^i \delta_{\xi_k^-}^{x(k)}$ with

$$w_k^i = \frac{w_{k-1}^i p(z(k)|x(k) = \xi_k^-)}{\sum_{j=1}^N w_{k-1}^j p(z(k)|x(k) = \xi_k^-)}$$

where the measurement equation is given by $\hat{z}(k) = z(k) + v(k)$

with $z(k) = [x(k), y(k), \theta(k), d(k)]^T$, and $v(k)$ = measurement noise.

The *time update* of the PF is $p(x(k+1)|Z) = \sum_{i=1}^N w_k^i \delta_{\xi_k^i}^{x(k)}$ where $\xi_k^i \sim p(x(k+1)|x(k) = \xi_k^-)$

and the state equation is $\hat{x}^- = \phi(x(k)) + L(k)U(k)$, where $\phi(x(k))$, $L(k)$, and $U(k)$ are defined in subsection 6.1. At each run of the time update of the PF, the state vector estimation $\hat{x}^-(k+1)$ is calculated N times, starting each time from a different value of the state vector ξ_k^i . Although the Distributed Particle Filter can function under any noise distribution in the simulation experiments the measurement noise was assumed to be Gaussian. The obtained results are given in Fig. 12.

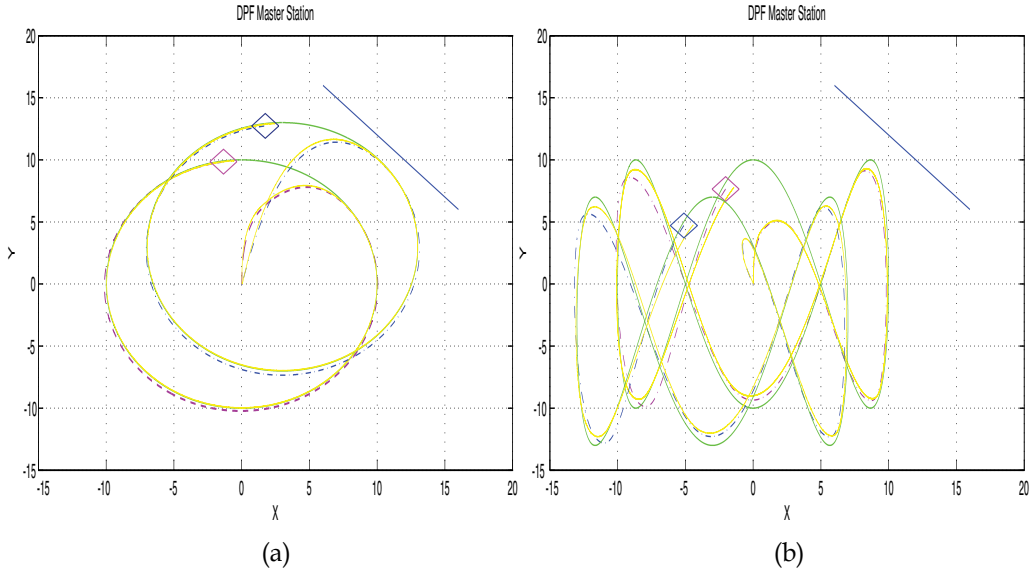


Fig. 12. Autonomous navigation of the multi-UAV system when the UAVs state vector is estimated with the use of the Distributed Particle Filter (a) tracking of circular reference trajectory (b) tracking of a curve-shaped reference trajectory

In the simulation experiments it was observed that the Distributed Particle Filter, for $N = 1000$ particles, succeeded more accurate state estimation (smaller variance) than the EIF and consequently enables better tracking of the desirable trajectories by the UAVs. This improved performance of the DPF over the EIF is due to the fact that the local EKF that constitute the EIF introduce cumulative errors due to the EKF linearization assumption (truncation of higher order terms in the Taylor expansion of Eq. (2) and Eq. (4)). Comparing to the Extended Information Filter, the Distributed Particle Filter demands more computation resources and its computation cycle is longer. However, the computation cycle of PF can be drastically reduced on a computing machine with a fast processor or with

parallel processors (Míguez 2007). Other significant issues that should be taken into account in the design of the Distributed Particle Filter are the consistency of the fusion performed between the probability density functions of the local filters and the communication overhead between the local filters.

The simulation results presented in Fig. 12 show the efficiency of the Distributed Particle Filtering in providing accurate localization for the multi-UAV system, as well as for implementing state estimation-based control schemes. The advantages of using Distributed Particle Filtering are summarized as follows: (i) there is robust state estimation which is not constrained by the assumption of Gaussian noises. The fusion performed between the local probability density functions enables to remove outlier particles thus resulting in an aggregate state distribution that confines with accuracy the real state vector of each UAV. If a local processing unit (local filter) fails the reliability of the aggregate state estimation will be preserved (ii) computation load can be better managed comparing to a centralized particle filtering architecture. The greatest part of the necessary computations is performed at the local filters. Moreover the advantage of communicating state posteriors over raw observations is bandwidth efficiency, which is particularly useful for control over a wireless sensor network.

7. Conclusions

The paper has examined the problem of localization and autonomous navigation of a multi-UAV system based on distributed filtering over sensor networks. Particular emphasis was paid to distributed particle filtering since this decentralized state estimation approach is not constrained by the assumption of noise Gaussian distribution. It was considered that m UAV (helicopter) models are monitored by n different ground stations. The overall concept was that at each monitoring station a filter should be used to track each UAV by fusing measurements which are provided by various UAV sensors, while by fusing the state estimates from the distributed local filters an aggregate state estimate for each UAV should be obtained.

The paper proposed first the Extended Information Filter (EIF) and the Unscented Information Filter (UIF) as possible approaches for fusing the state estimates obtained by the local monitoring stations, under the assumption of Gaussian noises. It was shown that the EIF and UIF estimated state vector can be used by a flatness-based controller that makes the UAV follow the desirable trajectory. The Extended Information Filter is a generalization of the Information Filter in which the local filters do not exchange raw measurements but send to an aggregation filter their local information matrices (inverse covariance matrices which can be also associated to the Fisher Information matrices) and their associated local information state vectors (products of the local Information matrices with the local state vectors). In case of nonlinear system dynamics, such as the considered UAV models, the calculation of the information matrices and information state vectors requires the linearization of the local observation equations in the system's state space description and consequently the computation of Jacobian matrices is needed.

In the case of the Unscented Information Filter there is no linearization of the UAVs observation equation. However the application of the Information Filter algorithm is possible through an implicit linearization which is performed by approximating the Jacobian matrix of the system's output equation by the product of the inverse of the state vector's covariance matrix (Fisher information matrix) with the cross-covariance matrix

between the system's state vector and the system's output. Again, the local information matrices and the local information state vectors are transferred to an aggregation filter which produces the global estimation of the system's state vector.

Next, the Distributed Particle Filter (DPF) was proposed for fusing the state estimates provided by the local monitoring stations (local filters). The motivation for using DPF was that it is well-suited to accommodate non-Gaussian measurements. A difficulty in implementing distributed particle filtering is that particles from one particle set (which correspond to a local particle filter) do not have the same support (do not cover the same area and points on the samples space) as particles from another particle set (which are associated with another particle filter). This can be resolved by transforming the particles set into Gaussian mixtures, and defining the global probability distribution on the common support set of the probability density functions associated with the local filters. Suitable importance resampling is proposed so as to derive the weights of the joint distribution after removing the common information contained in the probability density functions of the local filters. The state vector which is estimated with the use of the DPF was again used by the flatness-based controller to make each UAV follow a desirable flight path.

Comparing to centralized state estimation and control the proposed distributed state estimation and control schemes have significant advantages: (i) they are fault tolerant: if a local processing unit is subject to a fault then state estimation is still possible and accurate, (ii) the computation load is distributed between local processing units and since there is no need to exchange a large amount of information, the associated communication bandwidth is low. In the case of the Extended Information Filter and of the Unscented Information Filter the information transmitted between the local processing units takes the form of the information covariance matrices and the information state vectors. In the case of Distributed Particle Filtering the information transmitted between the local processing units takes the form of Gaussian mixtures. The performance of the Extended Information Filter and of the Distributed Particle Filter was evaluated through simulation experiments in the case of a 2-UAV model monitored and remotely navigated by two local stations.

Comparing the DPF to the EIF through simulation experiments it was observed that the Distributed Particle Filter, succeeded more accurate state estimation (smaller variance) than the EIF and consequently enabled better tracking of the desirable trajectories by the UAVs. This improved performance of the DPF over the EIF is explained according to the fact that the local EKFs that constitute the EIF introduce cumulative errors due to the EKF linearization assumption. It was also observed that the Distributed Particle Filter demands more computation resources than the Extended Information Filter and that its computation cycle is longer. However, the computation cycle of the DPF can be drastically reduced on a computing machine with a fast processor or with parallel processors. Other issues that should be taken into account in the design of the Distributed Particle Filter are the consistency of the fusion performed between the probability density functions of the local filters and the communication overhead between the local filters.

8. References

- [Beard et al. 2002] Beard, R.W.; McLain, T.W., Goodrich, M., & Anderson, E.P. (2002). Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, pp. 911-922, 2002.

- [Caballero et al. 2008] Caballero, F.; Merino, L., Ferruz, J., & Ollero, A. (2008). A particle filtering method for Wireless Sensor Network localization with an aerial robot beacon. *Proc. IEEE International Conference on Robotics and Automation 2006*, pp. 2860-2865, 2008.
- [Coué et al. 2003] Coué, C.; Pradalier, C. & Laugier, C. (2003). Bayesian Programming MultiTarget Tracking: an Automotive Application. *Int. Conf. on Field and Service Robotics*, Lake Yamanaka (Japan), July 2003.
- [Coué et al. 2006] Coué, C.; Pradalier, C., Laugier, C., Fraichard, T. & Bessiére, P. (2006). Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. *The International Journal of Robotics Research*, Vol. 25, No. 1, pp. 19-30, 2006.
- [Deming & Perlovsky 2007] Deming, R.W. & Perlovsky, L.I. (2007). Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors. *Information Fusion*, Elsevier, vol.8, no.3, pp. 316-330, 2007.
- [Fliess et al. 1999] Fliess, M. & Mounier, H. (1999). *Tracking control and π -freeness of infinite dimensional linear systems*, *Dynamical Systems, Control, Coding and Computer Vision*, (G. Picci and D.S. Gilliam, Eds.), Vol. 258, pp. 41-68, Birkhäuser, 1999.
- [Gan & Harris 2001] Gan Q. & Harris, C.J. (2001). Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No.1, pp. 273-280, 2001.
- [Gao et al. 2009] Gao, S; Zhong, Y. Zhang, X., & Shirinzadeh, B. (2009). Multi-sensor optimal data fusion for INS/GPS/SAR integrated navigation system. *Aerospace Science and Technology*, Elsevier, Vol. 13, pp. 232-237, 2009.
- [Hue et al. 2002] Hue, C.; Le Cadre, J.P. & Perez, P. (2002). Tracking Multiple Objects with Particle Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38. No.3, pp. 791-812, 2002.
- [Ing & Coates 2005] Ing, J. & Coates, M.G. (2005). Parallel particle filters for tracking in wireless sensor networks. *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC 2005*, art. no. 1506277, pp. 935-939, 2005.
- [Julier et al. 2000] Julier, S.; Uhlmann, J. & Durrant-Whyte, H.F. (2000). A new method for the nonlinear transformations of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, Vol.45, No.3, pp. 477-482, 2000.
- [Julier et al. 2004] Julier, S.J. & Uhlmann, J.K. (2004). Unscented Filtering and Nonlinear Estimation, *Proceedings of the IEEE*, Vol.92, pp. 401-422, 2004.
- [Léchevin & Rabbath 2006] Léchevin, N. & Rabbath, C.A. (2006). Sampled-data control of a class of nonlinear flat systems with application to unicycle trajectory tracking. *ASME Journal of Dynamical Systems, Measurement and Control*, vol. 128, No.3, pp. 722-728, 2006.
- [Lee et al. 2008] Lee D.J. (2008). Unscented Information Filtering for Distributed Estimation and multiple sensor fusion. *AIAA Guidance, Navigation and Control Conference and Exhibit*, Aug. 2008, Hawaii, USA.
- [Lee et al. 2008] Lee, D.J. (2008). Nonlinear estimation and multiple sensor fusion using unscented information filtering. *IEEE Signal Processing Letters*, Vol. 15, pp. 861-864, 2008.
- [Mahler 2007] Mahler, R.P.S. (2003). *Statistical Multisource-Multitarget Information Fusion*. Artech House Inc. 2007.

- [Makarenko & Durrant-Whyte 2006] Makarenko, A. & Durrant-Whyte, H. (2006). Decentralized Bayesian algorithms for active sensor networks. *Information Fusion*, Elsevier, Vol.7, pp. 418-433, 2006.
- [Manyika & H. Durrant-Whyte 1994] Manyika, J. & Durrant-Whyte, H. (1994). *Data fusion and sensor management: a decentralized information theoretic approach*. Englewood Cliffs, NJ, Prentice Hall, 1994.
- [Míguez 2007] Míguez, J. (2007). Analysis of parallelizable resampling algorithms for particle filtering. *Signal Processing*, Elsevier, Vol. 87, 3155-3174, 2007.
- [Morelande & D. Mušicki 2005] Morelande, M.R. & Mušicki, D. (2005). Fast multiple target tracking using particle filters. *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, December 12-15, 2005.
- [Musso et al. 2001] Musso, C.; Oudjane, N. & Le Gland, F. (2001). *Improving regularized particle filters*, In: *Sequential Monte Carlo Methods in Practice*. A. Doucet, N. de Freitas and N. Gordon Eds., Springer-Verlag 2001, pp. 247-272, 2001.
- [Nettleton et al. 2003] Nettleton, E.; Durrant-Whyte & H. Sukkarieh, S. (2003). A robust architecture for decentralized data fusion. *ICAR03, 11th International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.
- [Olfati-Saber 2005] Olfati-Saber, R. (2005). Distributed Kalman Filter with Embedded Consensus Filters. *Proc. 44th IEEE Conference on Decision and Control*, pp. 8179-8184, Seville, Spain, 2005.
- [Olfati-Saber 2006] Olfati-Saber, R. (2006). Distributed Kalman Filtering and Sensor Fusion in Sensor Networks. *Lecture notes in control and information sciences*, Vol. 331, pp. 157-167.
- [Ong et al. 2006] Ong, L.L.; Upcroft, B., Bailey, T., Ridley, M. Sukkarieh, S. & Durrant-Whyte, H. (2006). A decentralized particle filtering algorithm for multi-target tracking across multiple flight vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.
- [Ong et al. 2008] Ong, L.L.; Bailey, T., Durrant-Whyte, H. & Upcroft, B. (2008). Decentralized Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks. *Fusion 2008, The 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.
- [Oriolo et al. 2002] Oriolo, G.; De Luca, A. & Vendittelli, M. (2002). WMR Control Via Dynamic Feedback Linearization: Design, Implementation and Experimental Validation. *IEEE Transactions on Control Systems Technology*, Vol. 10, No.6, pp. 835-852, 2002.
- [Ren & Beard 2004] Ren, W. & Beard, R.W. (2004). Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 5, pp. 706-716, 2004.
- [Rigatos & Tzafestas 2007] Rigatos, G.G. & Tzafestas, S.G. (2007). Extended Kalman Filtering for Fuzzy Modeling and Multi-Sensor Fusion. *Mathematical and Computer Modeling of Dynamical Systems*, Vol. 13, No 3, Taylor and Francis, 2007.
- [Rigatos 2008] Rigatos, G.G. (2008). Autonomous robots navigation using flatness-based control and multi-sensor fusion. *Robotics, Automation and Control*, (P. Pecherkova, M. Fliidr and J. Dunik, Eds), I-Tech Education and Publishing KG, Vienna Austria, pp. 394-416, 2008.

- [Rigatos 2009a] Rigatos, G.G. (2009). Particle Filtering for State Estimation in Nonlinear Industrial Systems. *IEEE Transactions on Instrumentation and Measurement*, Vol. 58, No. 11, pp. 3885-3900, 2009.
- [Rigatos 2009b] Rigatos, G.G. (2009). Sigma-point Kalman Filters and Particle Filters for integrated navigation of unmanned aerial vehicles. *Intl. Workshop on Robotics for Risky Interventions and Environmental Surveillance, RISE 2009*, Brussels, Belgium, Jan. 2009.
- [Rigatos & Zhang 2009] Rigatos, G. & Zhang, Q. (2009). Fuzzy model validation using the local statistical approach. *Fuzzy Sets and Systems*, Elsevier, Vol. 60, No.7, pp. 882-904, 2009.
- [Rosencrantz et al. 2003] Rosencrantz, M.; Gordon, G., Thrun, S. (2003). Decentralized data fusion with distributed particle filtering. *Proceedings of the Conference of Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.
- [Särrkä 2007] Särrkä, S. (2007). On Unscented Kalman Filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, Vol.52, No.9, pp.1631-1641, 2007.
- [Shima et al. 2007] Shima, T.; Rasmussen, S.J. & Chandler, P. (2007). UAV team decision and control using efficient collaborative estimation. *Journal of Dynamic Systems, Measurement and Control*, Transactions of the ASME, Vol. 129, No 5, pp. 609-619, 2007.
- [Singh & Fuller 2001] Singh, L. & Fuller, J. (2001). Trajectory generation for a UAV in urban terrain using nonlinear MPC. *Proceedings of American Control Conference*, pp. 2301-2308, 2001.
- [Thrun et al. 2005] Thrun, S.; Burgard, M. & Fox, D. (2005). *Probabilistic Robotics*, MIT Press, 2005.
- [Vercauteren & Wang 2005] Vercauteren, T. & Wang, X. (2005). Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks. *IEEE Transactions on Signal Processing*, Vol.53, No.8, pp. 2997-3009, 2005.
- [Villagra et al. 2007] Villagra, J.; d'Andrea-Novel, B., Mounier, H. & Pengov, M. (2007). Flatness-based vehicle steering control strategy with SDRE feedback gains tuned via a sensitivity approach. *IEEE Transactions on Control Systems Technology*, Vol. 15, pp. 554-565, 2007.
- [Vissière et al. 2008] Vissière, D.; Bristeau, P.-J., Martin, A.P. & N. Petit (2008). Experimental autonomous flight of a small-scaled helicopter using accurate dynamics model and low-cost sensors. *Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 2008*.
- [Watanabe & Tzafestas 1992] Watanabe, K. & Tzafestas, S.G. (1992). Filtering, Smoothing and Control in Discrete-Time Stochastic Distributed-Sensor Networks, *In: Stochastic Large-Scale Engineering Systems* (S.G. Tzafestas and K. Watanabe Eds), pp. 229-252, Marcel Dekker, 1992.
- [Zhang et al. 2005] Zhang, Campillo, Q.F., Cerou, F. & F. Legland (2005). Nonlinear fault detection and isolation based on bootstrap particle filters, *Proc. of the 44th IEEE Conference on Decision and Control, and European Control Conference, Seville Spain, Dec. 2005*.

Design and Control of a Compact Laparoscope Manipulator: A Biologically Inspired Approach

Atsushi Nishikawa¹, Kazuhiro Taniguchi², Mitsugu Sekimoto³,
Yasuo Yamada³, Norikatsu Miyoshi³, Shuji Takiguchi³,
Yuichiro Doki³, Masaki Mori³ and Fumio Miyazaki³

¹*Shinshu University*

²*Kogakuin University*

³*Osaka University*

Japan

1. Introduction

Laparoscopic surgery is a technique where surgical tools and a laparoscope are inserted into the patient's body through small holes in the abdomen, and the surgeon carries out the surgery while viewing the images from the laparoscope on a TV monitor (see Fig. 1(left)). Laparoscopic surgery has grown rapidly in popularity in recent years, not only because it is less invasive and produces less visible scarring, but also because of its benefits in terms of healthcare economy, such as shorter patient stays. The most important characteristic of this technique is that the surgeon performs the operation while watching the video image from the laparoscope on a monitor instead of looking directly at the site of the operation. Thus, an important factor affecting the safety and smoothness of the operation is the way in which the video images are presented in a field of view suitable for the surgical operation. Manipulation of the laparoscope is not only needed for orienting the laparoscope towards the parts requiring surgery, but also for making fine adjustments to ensure that the field of view, viewing distance and so on are suitable for the surgical operation being performed. A camera assistant operates the laparoscope according to the surgeon's instructions, but must also make independent decisions on how to operate the laparoscope in line with the surgeon's intentions as the surgery progresses. Consequently even the camera assistant that operates the laparoscope must have the same level of experience in laparoscopic surgery as the surgeon. However, not many surgeons are skilled in the special techniques of laparoscopic surgery. It is therefore not uncommon for camera assistants to be inexperienced and unable to maintain a suitable field of view, thus hindering the progress of the operation. To address this problem, laparoscope manipulating robots are expected as a substitute for the human camera assistant and have already been made commercially available (see Fig. 1(right)). However, there are several problems to be solved:

1. **Hardware problems:** A large apparatus sometimes interferes with the surgeon. The setting and repositioning is awkward. Furthermore, the initial and maintenance costs are expensive.

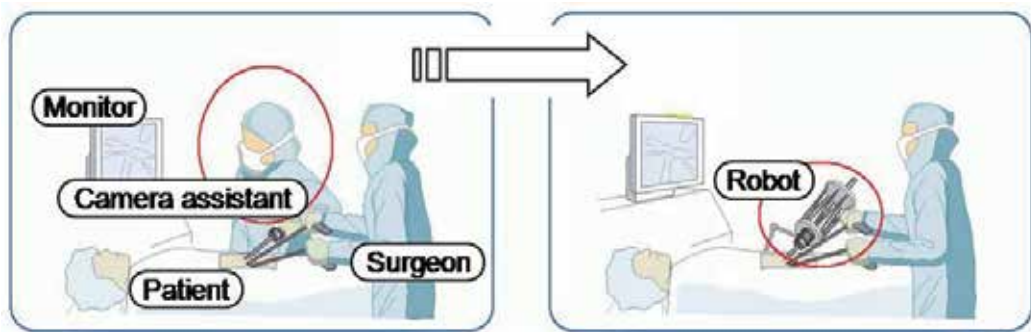


Fig. 1. Laparoscopic surgery. (left) Conventional laparoscopic surgery where the laparoscope is operated by a human camera assistant. (right) Robot-assisted surgery where the laparoscope is operated by a laparoscope manipulator.

2. **Software problems:** It is difficult to build and implement the accurate laparoscope manipulating model and consequently the conventional systems may not always offer the optimal view that the surgeon wants.

In this chapter, we will introduce a biologically inspired approach to the development of a new laparoscope manipulating robot to overcome those problems.

2. Related works

Laparoscope manipulators have been developed in the last fifteen years and there are at least 27 kinds of laparoscope controlling robots which are commercialized or published in refereed articles as of September 2009 (Taniguchi et al. (2010)). Some of them have already been made commercially available and are in widespread use. These include AESOP made in the US by Computer Motion Inc. (now known as Intuitive Surgical Inc.) (Sackier & Wang (1994)), EndoAssist made in the UK by Armstrong Healthcare Ltd. (now known as Prosurge Inc.) (Aiono et al. (2002)), LapMan made in Belgium by Medsys s.a. (Polet & Donnez (2004)), and Naviot made in Japan by Hitachi Co.,Ltd. (Tanoue et al. (2006)). Although these commercialized manipulators have various merits such as stable view and reduction of need for medical staff, several problems have been noted. First, the bulky manipulator and the supporting arm often interfere with the surgical procedures. Second, the setting and detaching of the robot is frequently awkward, causing an extension of the time required for the operation. Furthermore, the initial and maintenance costs are expensive. In addition to such hardware problems, they usually must be controlled by the operating surgeon himself/herself using a human-machine interface such as an instrument-mounted joystick, foot pedal, voice controller, or head/face motion-activated system. This is an additional task that distracts the surgeon's attention from the main region of interest and may result in frustration and longer surgery time.

To free the surgeon from the task of controlling the view and to automatically offer an optimal and stable view during laparoscopic surgery, several automatic camera positioning systems have been devised (Casals et al. (1996), Wei et al. (1997), Wang et al. (1998), Nishikawa et al. (2003), Ko & Kwon (2004), Nishikawa et al. (2006)). These systems visually extract the shape and/or position of the surgical instrument from the laparoscopic images in

real time, and automatically manipulate the laparoscope to always center the tip of the instrument in the displayed image. Such systems are based on the simple idea that the projected position of the distal end of the surgical tool corresponds to the surgeon's region of interest in a laparoscopic image. Besides centering on the most interesting area, there is an additional and important factor that defines a good image of the surgical scene—*zooming ratio* (Nishikawa et al. (2008)) — that corresponds to the depth of insertion of the laparoscope along its longitudinal axis. The pioneering studies of fully automatic camera positioning systems defined the zooming ratio as a “uniform” function of the estimated distance between the tip of the tool and the laparoscope (Wei et al. (1997)) or the area ratio between the visible tool and the whole image (Casals et al. (1996)). Although these approaches may completely remove the surgeon's camera control burden, they may not provide the specific view that the surgeon wants, because the most appropriate zooming ratio varies widely during surgery. The best zooming ratio depends on both the surgical procedure/phase and the habits/preferences of the operating surgeon. For this reason, most of the instrument tracking systems recently developed (Wang et al. (1998), Nishikawa et al. (2003), Ko & Kwon (2004), Nishikawa et al. (2006)) have abandoned the idea of systematic control of zooming parameters; instead, the surgeon is required to define the parameters preoperatively or adjust them intraoperatively through conventional human-machine interfaces, which again means an extra control burden for the surgeon.

3. Hardware design: analogy to human muscular structure

We developed a compact and lightweight robot manipulator, named P-arm (Sekimoto et al. (2009)), in collaboration with Daiken Medical Co., Ltd., Japan.

3.1 Parallel mechanism

There are several parallel robots (Kobayashi et al. (1999), Tanoue et al. (2006), Pisla et al. (2008)), which operates a laparoscope through the incision point on the abdominal wall of the patient. These systems have “less than 4” DOF and set up the laparoscope “outside” the parallel mechanism. Unlike the previous systems, the proposed manipulator is composed of a Stewart-Gough platform equipped with “six” linear actuators arranged in parallel “around” the laparoscope (see Fig. 2). This novel mechanism has an analogy to human muscular structure in which many extensors and flexors interact with each other; the rigid laparoscope corresponds to a bone of the human body and the linear actuators correspond to the muscles attached to the bone. This bio-inspired structure enables both the manipulator itself and the space necessary for operating the manipulator to be simple and small. The size of the P-arm is 120 mm in maximum diameter and 297.5 mm in length. Consequently, the manipulator can avoid interference with the surgeon's work during surgery. The Stewart-Gough platform has 6 DOF, whereas laparoscope movements are kinematically restricted to 4 DOF, due to the constraints imposed by operating through the incision point. In our case, even when two of the six actuators stop and are dislocated, the manipulator works safely because the system uses the remaining four actuators to produce constrained 4 DOF motion. Thus, our laparoscope manipulating robot based on the use of Stewart-Gough platform architecture provides both flexibility and accuracy while maintaining safety.

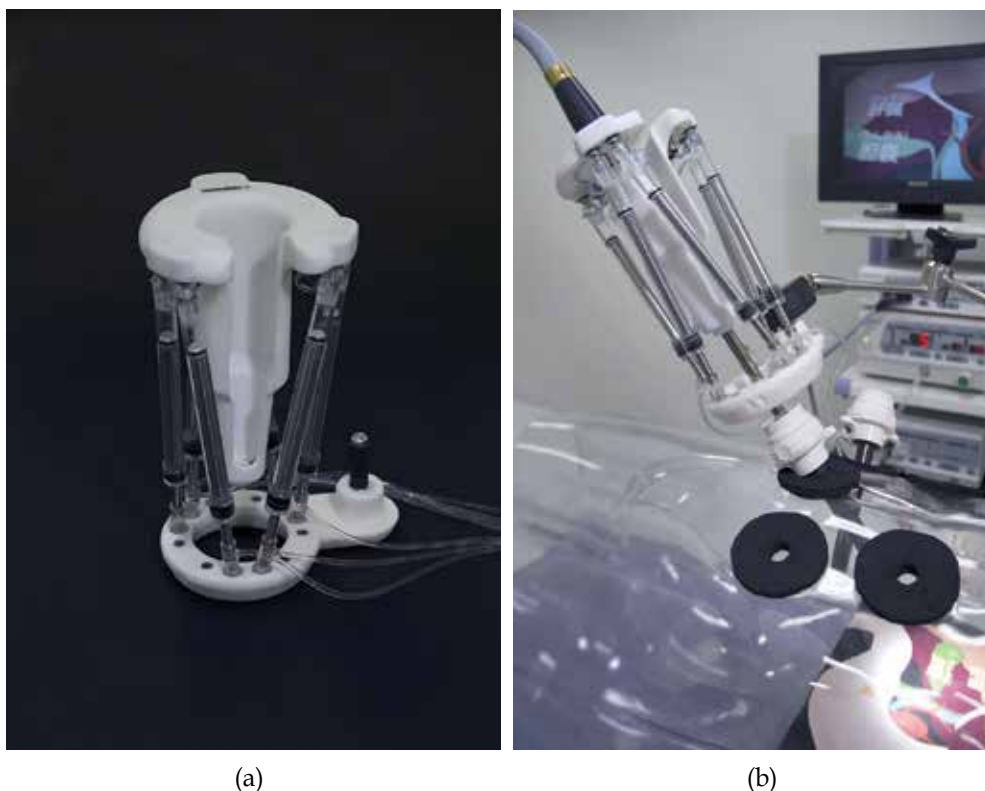


Fig. 2. Compact and lightweight laparoscope manipulator, named P-arm. (a) The P-arm is composed of a Stewart-Gough platform equipped with six linear hydraulic actuators. (b) The P-arm can hold a general laparoscope and can be supported by the conventional instrument holder.

3.2 Hydraulic actuators

The “artificial” muscles of the manipulator, that is, the linear actuators, are driven by hydraulic pressure transmitted via tubes connecting to the water cylinders in the controller unit. The actuator, tube, and the cylinder containing water were assembled en bloc and packaged in sterilized condition for clinical use. Also, the materials that were as inexpensive as possible were selected for all the parts of the manipulator including the actuators among those suitable for medical use and sterilization. All of the previously developed robots had to be wrapped in a sterilized plastic bag preoperatively, because the robot itself was not suitable for sterilization. The proposed manipulator was designed to be disposable and to be provided in a sterilized condition to make the preparation for the operation easy and quick and lessen the maintenance cost of the robot. Furthermore, materials in the manipulator were also selected in consideration of their weight. The actuator, which was mainly made of polycarbonate, weighed only 30 g. In total, the manipulator weighed only 580 g. The light weight allows that the manipulator to be fixed to the operating table with a conventional slim instrument holder. This makes the setting and repositioning of the manipulator easier and quicker. Also, the operating table can be tilted without repositioning the manipulator.

The actuators are attached to the manipulator using a permanent magnet. Therefore, when excess force is applied to the manipulator, the actuator is readily dislocated so it does not injure the patient. In addition, even when two of the six actuators are dislocated, the manipulator works safely as discussed above. The dislocated actuators can be easily reattached to the manipulator. Furthermore, in the case of an emergency, the robot can be stopped promptly by the emergency stop system, which is controlled by a circuit independent of the operating system.

3.3 Results

The robot was evaluated by performing the following three types of operations using a living swine: a laparoscopic cholecystectomy, a laparoscopic anterior resection of the rectum, a laparoscopic distal gastrectomy (Sekimoto et al. (2009)). As a result, it worked steadily for all the operations, without interfering with the surgeon's work (see Fig. 3). Also,



Fig. 3. View of an *in vivo* experiment (laparoscopic cholecystectomy) using a living swine. The P-arm and its supporting arm were so small that they did not interfere with the surgeon's work.

it contributed to shortening the setting and detaching time. The setting times were 66, 93, 104 seconds and the detaching times were 24 and 17 seconds, respectively. Wagner reported the setting time of 2 minutes for AESOP and 5.3 minutes for EndoAssist (Wagner et al. (2006)). Compared with these results, the P-arm was considered to be superior. The facility of the system is essential for the robot to be accepted by surgeons.

4. Software design: Use of biological fluctuation

Recent studies revealed that biological systems did not require the precise environmental model but rather made use of “fluctuation” in order to adapt to the environment. This adaptation mechanism can be represented by the following equation (Kashiwagi et al. (2006)):

$$\frac{d\bar{x}}{dt} = \bar{f}(\bar{x}) \times activity + \bar{\eta} \quad (1)$$

where \bar{x} and $\bar{f}(\bar{x})$ are the state and the dynamics of the system, and $\bar{\eta}$ indicates noise (fluctuation). A scalar variable *activity* indicates the fitness of the state \bar{x} to the environment and controls the behavior of the system. The term $\bar{f}(\bar{x}) \times activity$ becomes dominant in the above equation when the variable *activity* is large, and the state transition becomes deterministic. On the other hand, the noise $\bar{\eta}$ becomes dominant when *activity* is small, and the state transition becomes probabilistic. If the function $\bar{f}(\bar{x})$ has several attractors, the state of the system \bar{x} is entrained into one attractor when *activity* is large, while the behavior of the system becomes like a random walk when *activity* is small. The variable *activity* is designed to be large (small) when the state \bar{x} is suited (not suited) to the environment. The function $\bar{f}(\bar{x})$ is designed to have several attractors and updated in real-time based on the present *activity* information such that the state \bar{x} may efficiently become suited to the environment. As a result, the state of the system is entrained into an attractor that is suited to the environment and *activity* becomes large. Otherwise *activity* remains small and the system searches for a suitable attractor by a random walk.

By letting the state \bar{x} be the desired position of the tip of the right-hand surgical instrument in terms of laparoscopic camera coordinates, we developed a novel laparoscope positioning system that did not require any precise camera manipulating models (Nishikawa et al. (2009a)).

4.1 Design of activity

In order to find the *activity*—the most important factor for offering the specific view that the surgeon wants during laparoscopic surgery, a number of *in-vitro* laparoscopic cholecystectomy tests were performed. For each test, a swine liver with a gallbladder was placed in a training box and the gallbladder was removed by an operating surgeon with the use of the laparoscope robot P-arm controlled through a joystick interface by a camera assistant (another surgeon). In order to gather the positional relationship between the right and left surgical instruments and the laparoscope during surgery, a 3D tracking system (Polaris Accedo, NDI Corporation) was used. As a result, it was found that the velocity of the tip of the left-hand instrument and the velocity of the tip of the laparoscope had a high correlation (the cross correlation coefficient between the two was +0.765, (Nishikawa et al. (2009b))). We hypothesized that, at least in case of laparoscopic cholecystectomy, the camera

assistant changed the field of view when the magnitude of the acceleration of the tip of the left-hand instrument was large, and employed the following equation as the activity:

$$activity = \frac{1}{N} \sum_{n=0}^{N-1} activity_{i-n} \tag{2}$$

$$activity_i = \frac{1}{N} \sum_{n=0}^{N-1} p_{i-n} \tag{3}$$

$$p_i = \begin{cases} 0 & \text{if } |v_i - v_{i-1}| > K \\ 1 & \text{if } |v_i - v_{i-1}| \leq K \end{cases} \tag{4}$$

where i means time, N indicates the positive number for calculating the moving average. v_i indicates the magnitude of the velocity of the tip of the left-hand instrument at time i , and $K(> 0)$ is a threshold value.

4.2 Design of attractors

In Eq. 1, $\vec{f}(\vec{x})$ must have several attractors. Fukuyori et al. (2008) pointed out that the attractor should be adaptively allocated where the activity becomes large. Based on this concept of “adaptive attractors”, we regard the position of the tip of the right-hand instrument at time j as the center of the j -th attractor with the magnitude of $C^{i-j} \times activity_j$ at time $i(\geq j)$, and employ the following equation to design attractors:

$$\vec{f}(\vec{x}) = \sum_{j \leq i, C^{i-j} \times activity_j > M} \left[C^{i-j} \times activity_j \times \frac{B}{\|\vec{r}_j - \vec{x}\|^2 + A} \right] \cdot \frac{\vec{r}_j - \vec{x}}{\|\vec{r}_j - \vec{x}\|} \tag{5}$$

$$\vec{x}(t)|_{t=0} = \frac{1}{N} \sum_{n=0}^{N-1} \vec{r}_{i-n} \tag{6}$$

where i means the present time, N indicates the positive number for calculating the moving average. The vector \vec{r}_j represents the position of the tip of the right-hand instrument at time j . A , B , and C are all the positive constants: the parameters A and B respectively set the range and power of attractors, and the parameter $C(<1)$ indicates a forgetting factor. $M(>0)$ is a threshold for ignoring weak attractors. The term $\frac{B}{\|\vec{r}_j - \vec{x}\|^2 + A}$ acts like the Gaussian function

whose center is \vec{r}_j .

4.3 Results

We implemented this bio-inspired method on our robotic laparoscope positioner described in section 3. Fig. 4 shows the overview of our automatic laparoscope positioning system. The position/pose of the three tools: the right and left instruments and the laparoscope can be obtained simultaneously by the commercial 3D tracking system, Polaris Accedo (NDI Corporation) (See Blasinski et al. (2007) and Nishikawa et al. (2008) for the details). Then

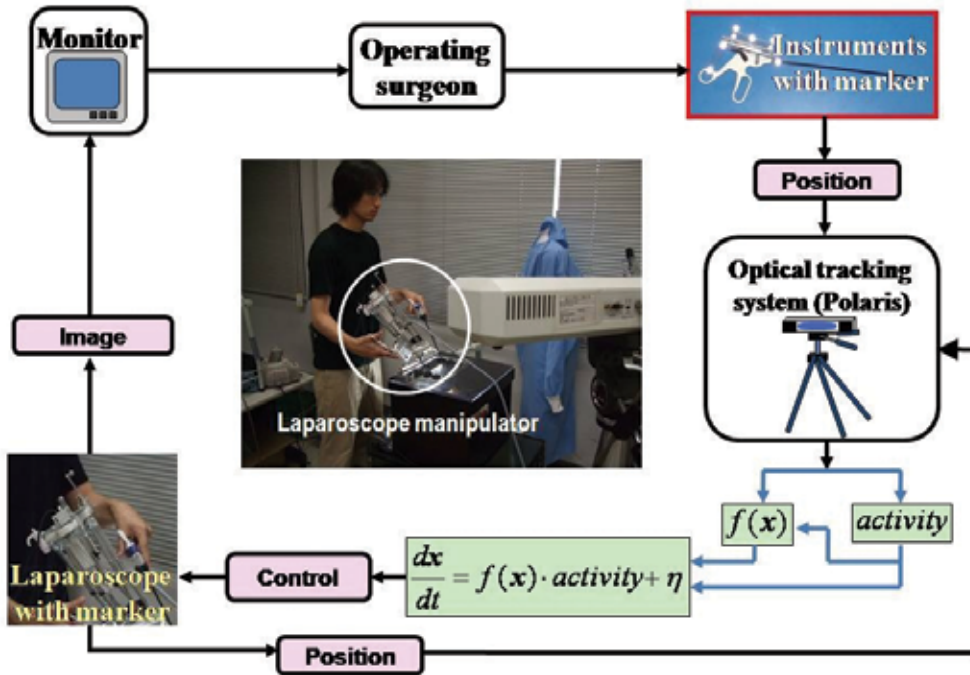


Fig. 4. Overview of automatic laparoscope positioning system. The proposed system uses “fluctuation” to determine and update in real-time the desired position of the tip of the righthand instrument, \bar{x} , during surgery.

both \bar{r}_i (the position of the tip of the right-hand instrument) and \bar{v}_i (the velocity of the tip of the left-hand instrument) are estimated in terms of laparoscopic camera coordinates, and *activity* and $\bar{f}(\bar{x})$ are calculated from Eqs. 2 and 5 respectively. As a result, we can determine and update also in real-time the desired position of the tip of the right-hand instrument, \bar{x} , during surgery, by substituting the resulting values: *activity* and $\bar{f}(\bar{x})$ into Eq. 1 and solving the Eq. 1 numerically (e.g., by the Runge-Kutta method) under the initial condition given by Eq. 6.

To validate the proposed system, a number of *in-vitro* laparoscopic cholecystectomy tests were performed. For each test, a swine liver with a gallbladder was placed in the training box and the gallbladder was removed by an operating surgeon with the support of the laparoscope robot P-arm controlled by Eq. 1. As a result, our system successfully and automatically controlled the position of a laparoscope during all the operations (Figs. 5–10).

5. Concluding remarks

A compact and lightweight laparoscope manipulator was developed. Also, a novel method for controlling the position of a laparoscope was inspired by biological systems dynamics. Our approach opens potential applications to skill transfer and adaptive behavior in medicine.



Fig. 5. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (1/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.



Fig. 6. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (2/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.



Fig. 7. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (3/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.



Fig. 8. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (4/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.



Fig. 9. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (5/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.



Fig. 10. View of an *in vitro* experiment (laparoscopic cholecystectomy) using a swine liver with a gallbladder (6/6). (left) a surgeon and the laparoscope robot P-arm, (mid) image from the laparoscope, (right) visualization of attractors as the contour map on the image plane.

6. Acknowledgements

This research was supported by “Special Coordination Funds for Promoting Science and Technology: Yuragi Project” of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and Grant-in-Aid for Scientific Research (A) of the Japan Society for the Promotion of Science (Representative Researcher: Atsushi Nishikawa, Project Number 19206047). We cordially appreciate the cooperation of Mr. Takeharu Kobayashi, Mr. Kouhei Kazuhara, Mr. Takaharu Ichihara, and Mr. Naoto Kurashita of Daiken Medical Co. Ltd. Also, we would like to express our sincere gratitude that the surgeons of Osaka University Hospital agreed to help our research.

7. References

- Aiono, S.; Gilbert, J. M.; Soin, B.; Finlay, P. A. & Gordan, A. (2002). Controlled trial of the introduction of a robotic camera assistant (Endo Assist) for laparoscopic cholecystectomy. *Surgical Endoscopy*, Vol. 16, No. 9, September 2002, 1267–1270, ISSN 0930-2794 (Print) 1432-2218 (Online)
- Blasinski, H.; Nishikawa, A. & Miyazaki, F. (2007). The application of adaptive filters for motion prediction in visually tracked laparoscopic surgery. *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO2007)*, pp. 360–365, Sanya, China, December 2007, IEEE Press
- Casals, A.; Amat, J. & Laporte, E. (1996). Automatic guidance of an assistant robot in laparoscopic surgery. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 895–900, Minneapolis, USA, April 1996, IEEE Press
- Fukuyori, I.; Nakamura, Y.; Matsumoto, Y. & Ishiguro, H. (2008). Flexible control mechanism for multi-DOF robotic arm based on biological fluctuation. In: *From Animals to Animats 10*, 22–31, Springer, ISBN 978-3-540-69133-4
- Kashiwagi, A.; Urabe, I.; Kaneko, K. & Yomo, T. (2006). Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS ONE*, Vol. 1, No. 1, December 2006, e49
- Ko, S. Y. & Kwon, D. S. (2004). A surgical knowledge based interaction method for a laparoscopic assistant robot. *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2004)*, pp. 313–318, Kurashiki, Japan, September 2004, IEEE Press
- Kobayashi, E.; Masamune, K.; Sakuma, I.; Dohi, T. & Hashimoto, D. (1999). A new safe laparoscopic manipulator system with a five-bar linkage mechanism and an optical zoom. *Computer Aided Surgery*, Vol. 4, No. 4, 1999, 182–192
- Nishikawa, A.; Asano, S.; Fujita, R.; Yamaguchi, S.; Yohda, T.; Miyazaki, F.; Sekimoto, M.; Yasui, M.; Takiguchi, S.; Miyake, Y. & Monden, M. (2003). Selective use of face gesture interface and instrument tracking system for control of a robotic laparoscope positioner. In: *Medical Image Computing and Computer Assisted Intervention –MICCAI2003*, 973–974, Springer, ISBN 978-3-540-20464-0
- Nishikawa, A.; Ito, K.; Nakagoe, H.; Taniguchi, K.; Sekimoto, M.; Takiguchi, S.; Seki, Y.; Yasui, M.; Okada, K.; Monden, M. & Miyazaki, F. (2006). Automatic positioning of a laparoscope by preoperative workspace planning and intraoperative 3D instrument tracking. *MICCAI2006 Workshop Proceedings, Workshop on Medical Robotics: Systems and Technology towards Open Architecture*, pp. 82–91, Copenhagen, Denmark, October 2006

- Nishikawa, A.; Nakagoe, H.; Taniguchi, K.; Yamada, Y.; Sekimoto, M.; Takiguchi, S.; Monden, M. & Miyazaki, F. (2008). How does the camera assistant decide the zooming ratio of laparoscopic images? Analysis and implementation. In: *Medical Image Computing and Computer Assisted Intervention -MICCAI2008*, 611-618, Springer, ISBN 978-3-540- 85989-5
- Nishikawa, A.; Yamada, Y.; Taniguchi, K. & Miyazaki, F. (2009). Automatic endoscope positioning algorithm based on biological fluctuation. *Proceedings of the 5th Asian Conference on Computer Aided Surgery (ACCAS2009)*, p. 125, Changhua, Taiwan, July 2009
- Nishikawa, A.; Yamada, Y.; Toda, S.; Sekimoto, M.; Miyoshi, N.; Takiguchi, S.; Doki, Y.; Mori, M. & Miyazaki, F. (2009). Analysis of Decisions by Camera Assistants on the Field of View for Laparoscopic Surgery and Its Application to Automatic Positioning of a Laparoscope. *Proceedings of the 21st International Conference of the Society for Medical Innovation and Technology (SMIT2009)*, pp. 99-100, Sinaia, Romania, October 2009
- Pisla, D.; Plitea, N. & Vaida, C. (2008). Kinematic modeling and workspace generation for a new parallel robot used in minimally invasive surgery. In: *Advances in Robot Kinematics: Analysis and Design*, Lenarcic, J & Wenger, P, (Ed.), 459-468, Springer, ISBN 978-1-4020-8599-4 (Print) 978-1-4020-8600-7 (Online)
- Polet, R. & Donnez, J. (2004). Gynecologic laparoscopic surgery with a palm-controlled laparoscope holder. *The Journal of the American Association of Gynecologic Laparoscopists*, Vol. 11, No. 1, February 2004, 73-78
- Sackier, J. M. & Wang, Y. (1994). Robotically assisted laparoscopic surgery. From concept to development. *Surgical Endoscopy*, Vol. 8, No. 1, January 1994, 63-66, ISSN 0930-2794 (Print) 1432-2218 (Online)
- Sekimoto, M.; Nishikawa, A.; Taniguchi, K.; Takiguchi, S.; Miyazaki, F.; Doki, Y. & Mori, M. (2009). Development of a compact laparoscope manipulator (P-arm). *Surgical Endoscopy*, Vol. 23, No. 11, November 2009, 2596-2604, ISSN 0930-2794 (Print) 1432-2218 (Online)
- Taniguchi, K.; Nishikawa, A.; Sekimoto, M.; Kobayashi, T.; Kazuhara, K.; Ichihara, T.; Kurashita, N.; Takiguchi, S.; Doki, Y.; Mori, M. & Miyazaki, F. (2010). Classification, design and evaluation of endoscope robots. In: *Robot Surgery*, Seung Hyuk Baik, (Ed.), 1-24, INTECH, ISBN 978-953-7619-77-0
- Tanoue, K.; Yasunaga, T.; Kobayashi, E.; Miyamoto, S.; Sakuma, I.; Dohi, T.; Konishi, K.; Yamaguchi, S.; Kinjo, N.; Takenaka, K.; Maehara, Y. & Hashizume, M. (2006). Laparoscopic cholecystectomy using a newly developed laparoscope manipulator for 10 patients with cholelithiasis. *Surgical Endoscopy*, Vol. 20, No. 5, May 2006, 753-756, ISSN 0930-2794 (Print) 1432-2218 (Online)
- Wagner, A.; Varkarakis, I. M.; Link, R. E.; Sullivan, W. & Su, L. M. (2006). Comparison of surgical performance during laparoscopic radical prostatectomy of two robotic camera holders, EndoAssist and AESOP: a pilot study. *Urology*, Vol. 68, 2006, 70-74
- Wang, Y. F.; Uecker, D. R. & Wang, Y. (1998). A new framework for vision-enabled and robotically assisted minimally invasive surgery. *Computerized Medical Imaging and Graphics*, Vol. 22, No. 6, December 1998, 429-437.
- Wei, G. Q.; Arbter, K. & Hirzinger, G. (1997). Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation. *IEEE Engineering in Medicine and Biology Magazine*, Vol. 16, No. 1, January-February 1997, 40-45, ISSN 0739-5175

Open Software Architecture for Advanced Control of Robotic Manipulators

J. Gomez Ortega, J. Gamez García,
L. M. Nieto Nieto and A. Sánchez García
*System Engineering and Automation Department at Jaén University,
Spain*

1. Introduction

So far, the robotic applications has been dominated by proprietary based hardware and software devices developed for industrial applications with a large volume manufacturing, like the automotive and electronics industries. Then, the main goal of the automation technologies has been an optimized robot design for precise assembly tasks, resulting in complex systems with a reduced flexibility.

Traditional robotic applications have a fixed configuration, with the advantages of high accuracy and a well studied kinematics. However, since recent years, the number of service robots in our daily life environments is increasing. There are many new applications, i.e. teleoperation, human-robot-collaborative works, etc. that require reconfigurable hardware and expansibility to accomplish new working modes in not-structured scenarios and not-intensive manufacturing tasks. However, these systems must meet diverse user requirements and integrate different hardware and software systems developed for a particular proprietary platform. As a result, many different researchers have solved similar issues with non-interchangeable products, working from scratch each time, adapting the traditional industrial robots platform for the new applications.

Today, a new robotic system is an integration of different processors and hardware platforms manufactured by different vendors, controlled by software modules developed using different programming languages and different communication protocols. In addition, as robotic manipulator is expected to accomplish more complex tasks, it needs the integration of multiple sensors working with different time bases and bandwidths (Gamez et al., 2009; Luo et al., 2002), and new capabilities are needed that traditional control technology of current industrial robots is not offering. In order to solve these problems different open robotics platforms have been presented.

1.1 Open robotic platforms: an overview

From the definition of an Open Systems (IEEE 1003.0), an Open (Robotic) Platform should "provide capabilities that enable properly implemented applications to run on a variety of platforms from multiple vendors interoperate with other system applications and present a consistent style of interaction with the user". This leads to the following properties:

- Portability of the software, to reuse it in other platforms with minor changes.

- Reusability is an issue that should be addressed from the beginning of the development process, identifying common problems that could be solved with reusable solutions and shared within the robotics community.
- Extensibility to change or add several component (of hardware or software) to the system from different vendors.
- Adaptability/dynamic reconfigurability, providing mechanism of easy adaptation of its parameters according to the application requirements.
- Interoperability refers to the ability to support interchange of information between robotic modules designed by different vendors, providing effective communication and working in a coordinated manner. In particular it relies on the network and communication protocols that must provide effective real-time communication among distributed components, independently of the system specific particularities.

Diverse approaches have been proposed to achieve these capabilities. Some solutions use Matlab/Simulink and Real Time Workshop to generate control applications for robotic systems with a proprietary operating systems (Gamez et al., 2007), but with the disadvantage of a limited interoperability. Today, most of the research and robotic applications developed based on proprietary hardware used a layered software architecture. This approach typically includes a standard based middleware to provide integration, efficient communication, interoperability, abstraction of software components, also providing portability. At the top level different reusable software components are used. In the low level layer, the hardware is controlled by drivers developed to run on a proprietary RTOS. However, since last decade, developers have a growing interest on developing open source applications based on Linux RTOS. Thus, vendors are offering commercial-grade Linux operating systems (Saravanan et al., 2009; Gamez et al. 2009).

Another approach based on hardware modularity can support integration of new components from various vendors. The corresponding software must provide a well defined interface to provide easily integration between interconnected devices, and the capabilities of extensibility and modification (Xuemei & Liangzhong, 2007).

As the hardware is always vendor-dependent, the integration of different devices may be difficult due to incompatibility reasons. To overcome this problem, some hardware standards have been proposed, however this method is considered too restrictive to achieve reusability of existing hardware (Hong et al. 2001).

1.2 Related research

In recent years, an increasing number of initiatives have been presented:

- OROCOS (Open Robot Control Software) project (OROCOS, 2010), is a European initiative for providing free software project to develop advanced robotics applications. The project supports different C++ libraries for creating control applications over different proprietary operating systems (e.g. Win32, Mac OS). Also includes the Real-Time Toolkit (RTT) library for writing hard real-time control applications in C++ for Linux based systems, and tools from contributors to generate components using Real-Time Workshop from Matlab/Simulink. To achieve reusability, the framework supports standard component interfaces and CORBA for interoperability between distributed components over a network. Some others not real-time projects have derived from OROCOS, like ORCA (ORCA, 2010) and SmartSOFT (Schlegel, 1999).
- RT-Middleware (from Robot-Technology) (Ando et al., 2006; Chishiro et al. 2009) is a CORBA based software platform for robot system integration developed in Japan, with

the participation of the Japan Robot Association (JARA). One of the objectives of the project is to simplify the construction of customized robot combining selected RT-components. In recent years, the Object Management Group (OMG) (OMG, 2008) started a standardization process for these RT-components to achieve interoperability, interconnectivity and integration of components from different manufacturers.

- In Korea, the Open Platform for Robotic Services (OPRoS) (Park & Han, 2009) is another open software project promoted to unify different robots platforms. The framework includes standardized components, an integrated development environment (IDE) and a simulation and testing environment. OPRoS supports CORBA and the Universal Plug and Play (UPnP) (Ahn et al., 2006) standards for modular integration. The operational scheme employs a server-client model to interact with the robot system as a target robot, and external servers for heavy computation.
- The Coupled Layered Architecture for Robotic Autonomy (CLARAty) (Nesnas et al., 2003) was initiated in the NASA to provide a software framework to develop advanced robotic technologies for robotic platforms employed in other NASA programs. Unlike others architectures, CLARAty is a two-level architecture where the system decomposition allows for intelligent behavior at low levels while the structure of the different levels is maintained. In this scheme, the high level Decision Layer sends command to the Functional Layer and in a client-server model, and the Functional Layer provides different levels of abstractions to achieve adaptation of the reusable components to the hardware of different robots. Also, the Decision Layer provides a unified representation of activity plans based on a declarative model.
- For the Mobile and Autonomous Robotics Integration Environment (MARIE), the main goal was to provide a common component-based middleware to reuse and interconnect different programming environments (Cote et al., 2006). The framework followed a one-to-many interaction model between different components to coordinate the interaction within a virtual shared space, and allowing each component to use its own communication protocol.
- MIRO (Utz et al., 2002) is a CORBA based middleware organized in three layers: a device layer provides object-oriented interface abstraction for the hardware, and a service layer provides CORBA interface services between the device layer and the top layer. This layer provides reusability and easy integration in an object oriented framework.
- In recent years, several RT-Linux based open projects are developed: RTOC (Xu & Jia, 2006) is a RT-Linux based architecture based upon the OSACA model (OSACA, 1996) that can be ported to not PC-based platforms. In its layered model, a database stores universal application modules for control, path planning, etc. Other Linux based platforms use ST-RTL to generate control applications from Simulink models (Ostrovrsnik et al., 2003). Xenomai (Xenomai, 2010) is another Linux-based Real Time operating system used to develop robot control systems using open source and standardized communications protocols (Sarker et al. 2006).

The remainder of this paper is organized as follows. Firstly, a brief explanation of the necessity of these platforms is introduced in Section 2. Later, Section 3 describes the hardware structure. In this section the main characteristics of both hardware configurations are presented. In Sec. 4, the software structure is presented, while Sec. 5 presents experimental results which validate the performance of the proposed architecture. Finally, Discussions and Conclusions are presented in Section 6 and 7, respectively.

2. Why the necessity of these robotic platforms

It has been long recognized that multisensor-based control is an important problem in robotics (Gamez et al. 2008), the need to take advantage of multiple sensors in controlling a system becomes increasingly important. On the other hand, to the purpose of getting an adequate interaction between the manipulator and its environment, force/position feedback control is necessary, above all, if the environment where the robot wants to interact is unknown or changing (Gamez et al., 2005). In general, given the classical hierarchical control structure of a robot microcomputer controller (Groover, 2008) (Fig. 1), the possibilities of control or the integration of new sensors into the setup, are not offered nowadays by the robot manufacturers.

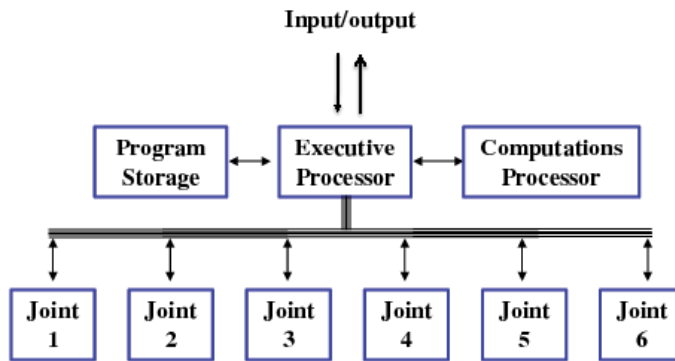


Fig. 1. Classical hierarchical control structure of a robot microcomputer controller.

A representative example of implementation of a force/position controller could be the impedance controller (Hogan, 1985). The purpose is to ensure that the manipulator is able to operate in a non-ideally structured constrained environment while maintaining contact forces within suitable limits. A description of this system is summarized in fig. 2:

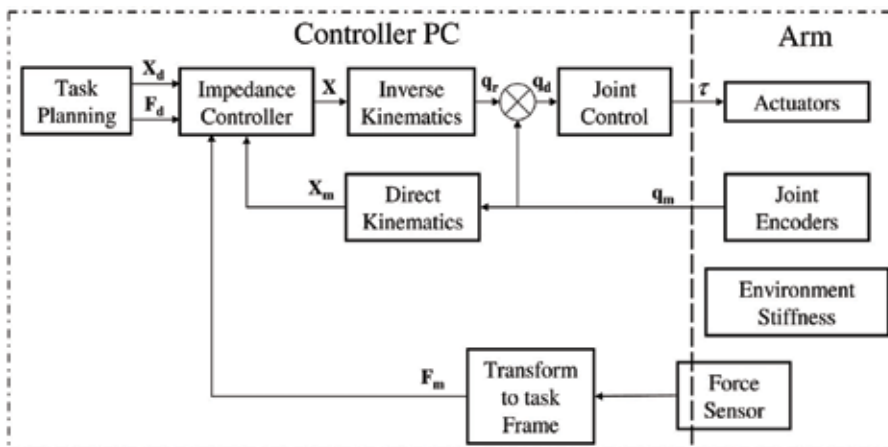


Fig. 2. Impedance controller structure.

However, an intrinsic problem occurs when trying the application of this control algorithm, if only a wrist force sensor has been used, in a dynamic situation, where the manipulator is

moving in either free or constraint space, the interaction forces and moments at the contact point, and also the noncontact ones, are measured by this sensor (Gamez et al., 2004). Furthermore, the magnitude of these dynamics disturbances cannot be ignored when large accelerations and fast motions are considered (Khatib & Burdick, 1986), when the manipulator carries out tasks with heavy tools (Johansson & Robertsson, 2003), or when the environment is not perfectly known (not allowing the use of switching strategies that compensate for the free space phase).

To solve this problem, the integration of different sensors such as a force/torque and a acceleration sensor could be use to solve this problem (Gamez et al., 2008; Kroger et al., 2007); however, fusion of data from multiple sensors into a robust and consistent model meets some difficulties such as measurements with different time bases (Luo et al., 2002) or noise and incompleteness of sensor data (Larsson et al., 1996). Another problem could be to easily connect these sensors, which are from diverse manufacturers, to the hardware setup (Gamez et al. 2009).

Thus, observing these problems, it can be guessed why a complex dynamic system, such as robotic manipulator, is demanding new and highly sophisticated capabilities that traditional control technology of current industrial robots is not offering (Wills et al., 2001).

3. Hardware elements of the platform

This section describes the hardware components that convert this platform in a non-conventional one from an industrial point of view. Also, we will describe the necessity of these elements that were used to test and validate new control concepts for manipulators that interacts with an unknown environments. In this point, it is necessary to point out that two different hardware configurations, and thus two software structure, were carried out. In both cases, the experimental setup contained the following elements: an anthropomorphic 6-DOF Stäubli RX60 industrial manipulator and a CS8 controller, a Phantom 6D Haptic Device, a vision system composed of two cameras, a 6-DOF ATI wrist force/torque sensor, a 3-DOF capacitive accelerometer, a 3-DOF gyroscope, a special purpose end effector and the teach pendant, an acquisition board integrated in the robot controller, a workcell and a number PCs to mainly develop software and to collect data.

3.1 Old hardware configuration

Initially we designed a hardware scheme that had the structure shown in Figure 3 (Gamez et al., 2009).

The kernel of this architecture is the CS8 controller PC. It is in charged of the high-level operations (execution of the path planner, trajectory generation, sercos communication, etc.), and also of reading external sensors such us the wrist force and torque sensor or the acceleration sensor. These elements were connected to the open PCI slot in the controller PC. With this structure, software modules for collecting data where mainly resident in this PC. The main advantage of using a PC-based standard interface is that it ensures that the extensibility and scalability are available. Therefore, the hardware and software components can be integrated or replaced easily.

Due to proprietary reasons, the operating system running on this PC is VxWorks (Wind-River, 2005), which allows easy integration of many commercially available add-on peripherals such as acquisition boards, ethernet boards, etc.. It also provides deterministic

context switching, timeliness, support for open standard. The external sensors used to model the environment, and thus to make the robot capable of interacting with it, are: a ATI wrist force/torque (f/t) sensor (MINI SI80-4) where the f/t strain gauge signals are conditioned using an intermediate module, called supply board, and later transmitted through a DAQ acquisition board which processes the strain gauge information and offers it through the PCI slot. A 3D accelerometer, which was attached to the end-effector of the manipulator and a 3D gyroscope of CFX Technology (an UCG-TX model). These two last sensors were also read by the same acquisition board.

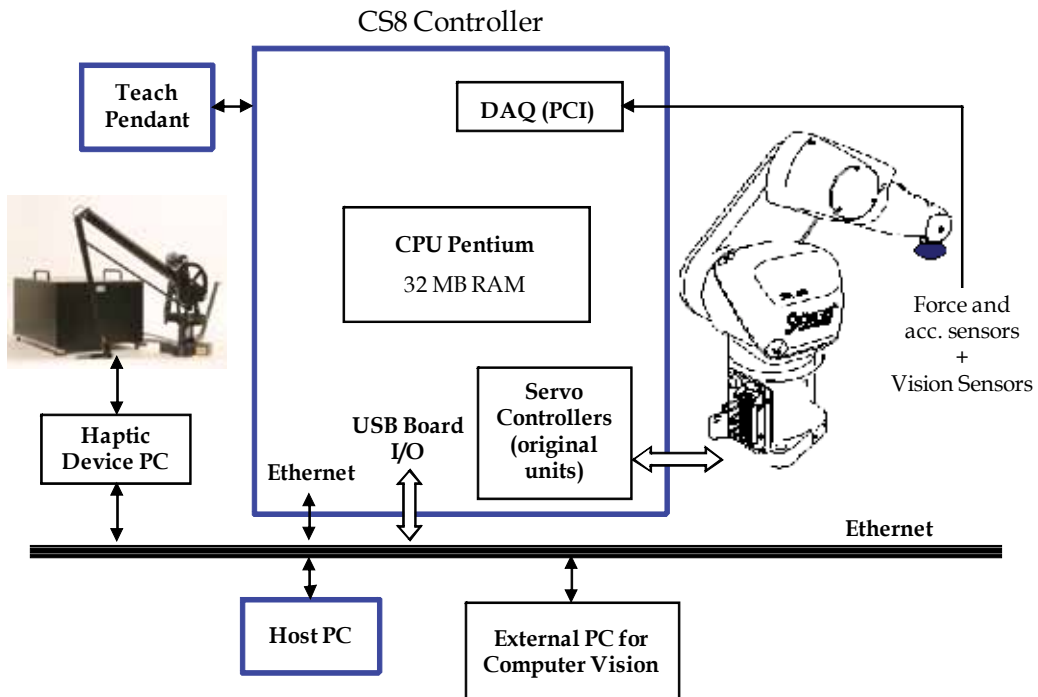


Fig. 3. Hardware configuration for the old system.

Regarding the vision system, the cameras are connected directly to a dedicated computer vision PC. Later, the image is processed and the required information -normally a vector with coordinates of positions and orientations- sent to the controller PC through ethernet. The haptic device is a PHANTOM 3/6DOF with six degrees of freedom in position and force feedback in three translational degrees of freedom. The 3.0/6 DOF has a range of motion approximating full arm movement pivoting at the shoulder. This device is connected to an extra PC via the parallel port (EPP) interface. The sample time of the haptic device is higher than the controller loop (1 KHz against 250 Hz.), and since there is no physical distance between them, the delay is one controller sample time at maximum. The bandwidth of all the mentioned sensors apart from the vision sensor is 250 Hz. This sample time has been chosen in order to synchronize the sensor readings with the robot control loop. The bandwidth of the vision sensor is smaller (around 30 Hz), because of the high computational effort required and the camera speed. Shortly, we are going to change these cameras to new ones with a bandwidth of 120 frames/sec.

3.2 New hardware configuration

The main drawbacks that can be found in the former robotic system are:

- The master PC, where a huge number of applications are running -sensor readings, control algorithm execution, robot movements-, is placed in the controller PC, so this structure is subject to a PC that in few years is antiquated and cannot be changed (without the manufacturer collaboration).
- A great part of the code running in the controller is unknown -belongs to the manufacturer- and sometimes it occurs problems because of the inconsistency between the original code and the experimental one (as uncontrolled modifications of some common data). These problems are difficult to solve, again, without a close collaboration with the manufacturer. Also, while the experimental code is more complex and bigger, the inconsistencies are more probable.
- The time synchronization is easier and more robust if we have an external master PC that configures and controls all the sensors, the control algorithms and the actuation system.

To solve the problems that the first configuration presented, a new hardware and software structure is being developed. Similar to first one, the main difference can be found in where the main executions are carried out. Figure 4 shows a scheme of this new configuration.

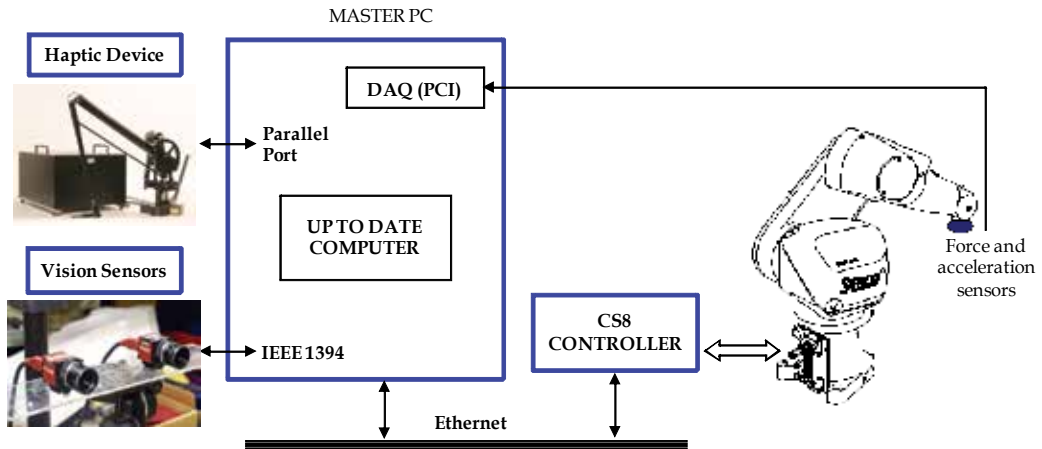


Fig. 4. Hardware configuration for the new system.

With respect to the old system, this new system introduces the following changes:

- The operating system is based on LINUX with a real time framework called XENOMAI (Xenomai, 2010).
- The vision sensors are read through a IEEE 1394 port placed in the master PC. From the experimental tests, it was checked that the computational effort required by a normal vision system processing does not bother to the rest of the tasks.
- The haptic device is connected directly to the Master PC through a parallel port.
- The external sensors, i.e. the wrist force sensor, the accelerometer, or other sensors, are connected to a Data Acquisition Board plugged into a PCI slot of the Master PC.

4. Software structure

In this section, we describe a component-based control software architecture developed in order to get a robust and easy-to-maintain experimental robotic platform. Two fundamental

goals were established for the architecture: first, it should standardize functions that are common across sensors and open robotic platforms; second, the architecture should enable design by composition. Since the most interesting configuration is the new one, we limit this section to its description. Further information about the old software configuration can be found in (Gamez et al. 2010).

4.1 Layer architecture and component definitions

Although the software structure of the experimental setup contains basically two PCs: the master PC and the controller PC, it consists of a hierarchy of components that are divided into four main layers proposed originally by (Nilsson & Johansson, 1999): lowest layer, middle layer, high layer and end-user layer. Each layer contains different types of components, which are classified depending on their functionality (Fig. 5). This components are related, in the major cases, to a block or system of the hardware structure. The four layers are:

1. Lowest layer: whose components correspond to those ones closer to the physical environment. Examples are the different sensor components or the joint control components.
2. Middle layer: Components can use the information of the lowest layer and the high layer. Examples could be a virtual sensor component or a manipulator control component.
3. High layer: Trajectory generator components.
4. End-user layer: Task planner components.

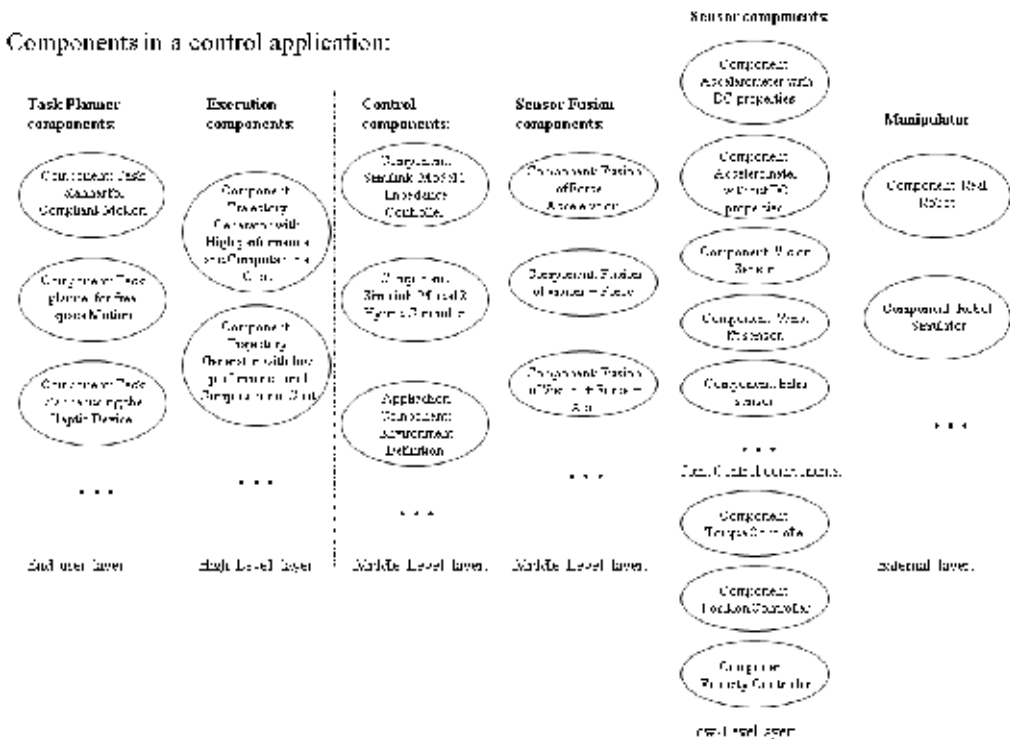


Fig. 5. Structure of the components developed for the platform.

The end-user layer describes the task to be carried out in terms of final positions, orientations and velocities of the robot end-effector. Different components have been developed and they are used depending if the task to be carried out is in open space, with constraint motion or with both. In addition, another component has been designed which is in charge of controlling the haptic device. The inputs to the components of this layer can be the reference position-orientation of the robot TCP, the desired contact forces exerted by the manipulator to the environment or even vision features. Currently, these inputs can only be defined off-line, not taking the most significant advantage of on-line programming, that is, the robot can be programmed in accordance with the actual position of equipment and pieces of these modules.

The high-level layer is compound basically of two components with the functions of a path planner. This planner generates trajectory set points for the robot, according to motion command which it receives from task specification. The commands these components offer to their lower layer can be either the joints trajectory or the Cartesian trajectory of the robot end-effector. It is necessary to point out that both the original task planning and the original trajectory generator developed by Stäubli were not used in this platform due to proprietary reasons. For our applications, the components designed for the special-purpose planner calculate the joint coordinates from the Cartesian references solving the inverse kinematics on line (Gamez, 2006). In this sense, a second component has been developed to reduce the computational cost of the previous block if necessary. Specifically, it consists of the decomposition of the robot geometric structure into two subsystems: one for position and one for orientation. This option offers an analytic solution that simplifies the singularities problem. Furthermore, a number of restrictions have been imposed to prevent special singularities such as shoulder and wrist ones. Although the developed trajectory is not robust, the resultant workspace is acceptable for most of the practical cases. Currently, these components are used from the former configuration and, in the future, we expect to improve them using more sophisticated trajectory generators than can be found, for instant, in (Bruyninckx, 2001).

For the middle-level layer, and from an engineering point of view, we note that tailoring the motion control requires control engineering competence while application support does not (Nilsson & Johansson, 1999). Although is therefore reasonable and appropriate to define two different sub-layers for these types of programming: application control layer, (movement constraints, tool mass, etc.) and control layer (to configure the control loop, tunes the gains, etc.), this level is built, on the one hand, using manipulator control components. On the other hand, other kind of components that are used in this layer are the virtual sensor components. These elements allowed the application of sensor fusion strategies in a structured way. Both components are designed with Simulink and the Real Time Toolbox of Matlab.

Using the property that any Simulink control model is an interconnection of signals (reference commands, position feedback, velocity feedback, torque feedback, sensors feedback) and mathematical operations, a generic block has been designed with a predefined number of inputs and outputs. Inside each block, one can implement different control algorithms, or sensor fusion strategies, combining a high-performance language for technical computing with a fast prototyping of the robotic platform since all the inputs and outputs are readable and writable.

In the lowest layer dedicated to the sensor, each one is modeled by a component that contains basically two parts: one is for data structure building and the other is for sensory data sharing. One of the function of the sensor components is to process the information of a specific sensor and to provide a unified sensory data bank manager. The main advantage of this manager is that it can directly offer the calibrated sensor data. Furthermore, sensor data must be shared with every necessary function in the software architecture. Another important function of the sensor components, and on the rest of components, is to stamp a time when a set of measurements, or interaction, is obtained. It helps to obtain a history of the events.

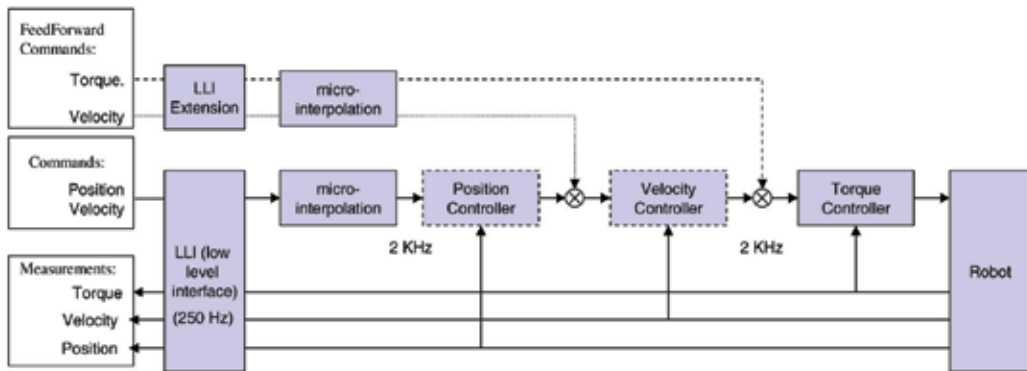


Fig. 6. Low level structure of the joint controllers.

Regarding the joints control sub-layer, it uses a low level interface designed by Stäubli Robots (Pertin & Bonnet, 2004); in fact, this is the only software module that remains from the original Stäubli system. This level obeys the structure presented in Fig. 6 and its mission is to allow the low level control of each joint. Although three different components were defined in the previous software structure (given their possibility of control: torque, position and velocity controller), only a generic one is considered in this structure. This sub-layer is placed currently in the robot controller PC and we are working on how to define the component automatically -in terms of torque, position or velocity-, given the programming of manipulator control component.

4.2 Middleware

In our case, we have to different software contexts: this one placed at the controller PC and the second one running on the master PC. In the controller PC, where the component of the joints control sub-layer is running, to guarantee that the shared memory constraints are fulfilled, the system has to protect itself from invalid memory accesses that otherwise could compromise the system.

In this case, to avoid this problem, between the component and the monitoring task, the tasks are synchronized following a structure "top to bottom" where the maximum priority is given to the joints control task. The operating system running on this PC was VxWorks (Wind-River, 2005).

Another problem was to synchronize different components that are placed in a master PC with interconnections with external systems and a Real Time Linux operating system. The solution selected was to choose XENOMAI (Xenomai, 2010) with the RTNet (Real Time

Network) package. For our case, the synchronization scheme is not based on a master clock (as it was in the former configuration, where it followed a "top to bottom" structure). Each component has its own clock, updating data with their respective bandwidth. Currently, we are implementing a middleware using concepts similar to those ones defined in OROCOS project (Bruyninckx, 2001). In a middle-long term, our intention is to obtain a user-friendly API that allows fast and easy prototyping. Figure 7 shows the block diagram of the hardware and software communications differentiating between the master PC and the controller PC. It can be guessed from this figure that each component communicates with other ones mainly through shared memory, or through Ethernet depending on where it is placed.

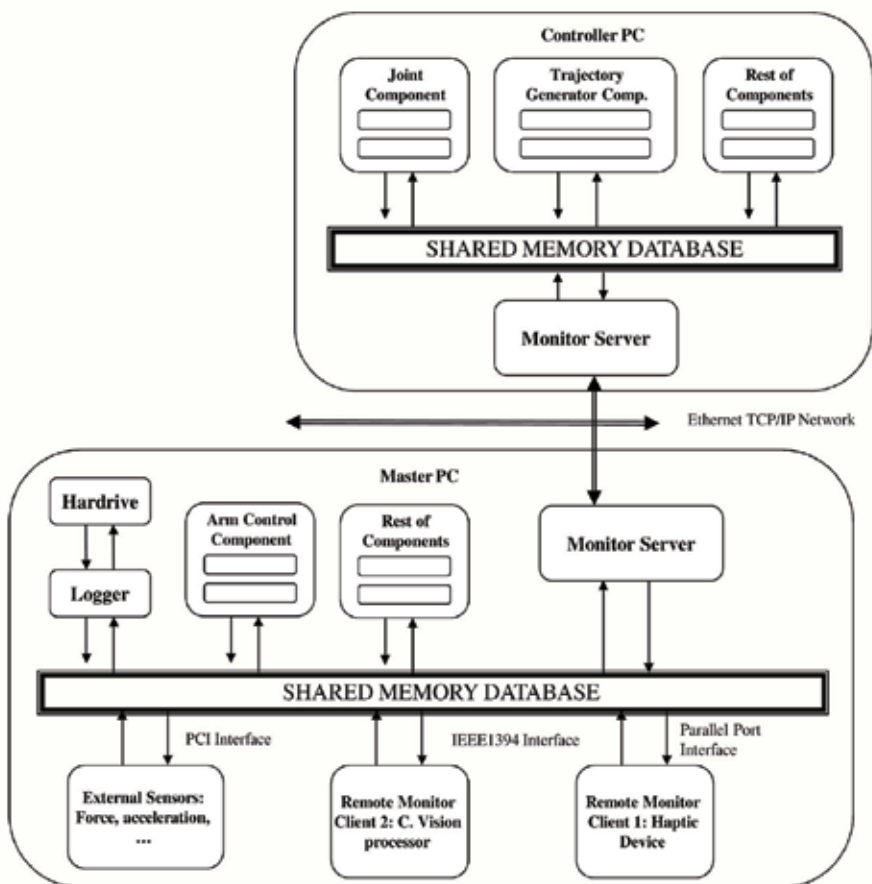


Fig. 7. Block diagram of the hardware-software communications.

5. Experimental validation

Different experiments have been carried out to validate the performance of the proposed architecture, noting that these results are obtained from the old hardware configuration. They consisted in the application of a compliant motion controller where the environment

information was obtained fusing different sensors. In particular, for the case shown in this paper, the sensors used were a force/torque sensor, an accelerometer and the joint sensors. The objective of this integration was to develop a force observer capable of estimating, accurately and from the f/t sensor measurements -which reflect the contact forces, the inertial ones and the gravity forces-, the contact force exerted by a manipulator to its environment (Gamez et al., 2008).

To carry out this test, some of the components that were described before in the previous section were used.

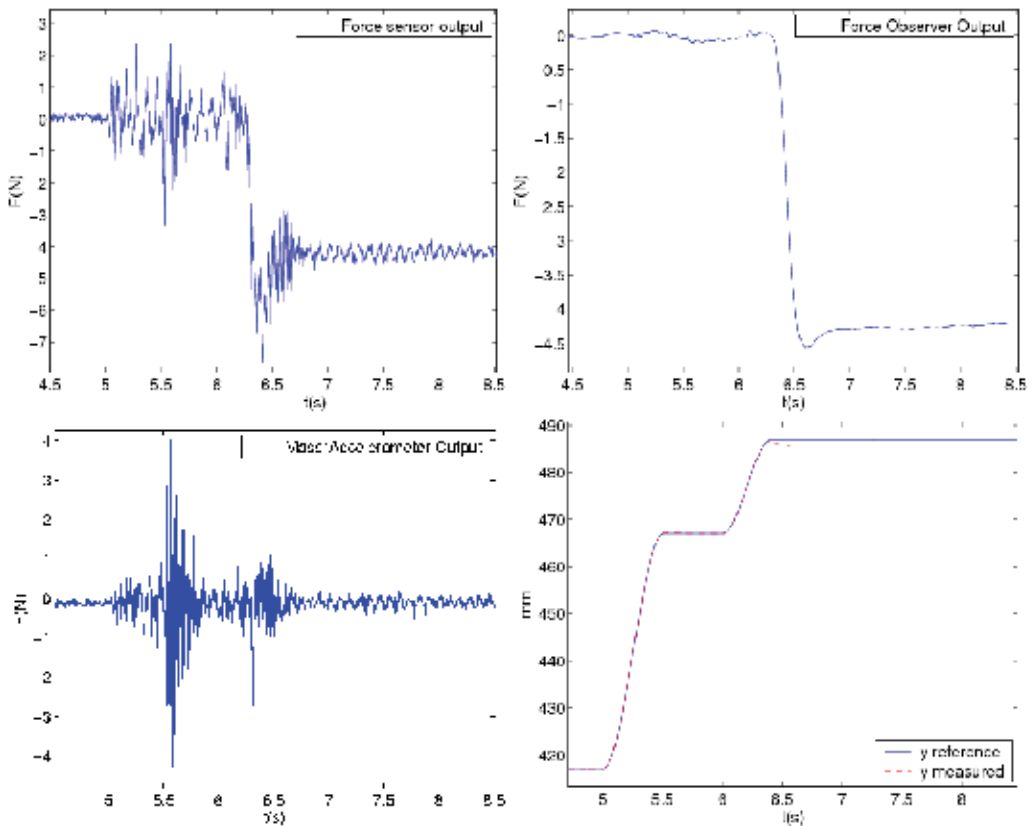


Fig. 8. Force measurement from the wrist sensor ATI (upper-left), force observer output (upper-right), accelerometer output (lower-left) and real and measured position of the robot tip for y-axis (lower-right).

The results obtained applying the force/position controller, where the information of the force observer is used, are presented in Fig. 8. The experiment consisted of a movement in the axis z of three phases: an initial movement in free space (from $t=5s$ to $t=6.2s$), a contact transition (from $t=6.2s$ to $t=6.4s$) and a movement in constrained space (from $t=6.4s$ to $t=9s$). Apart from the force compensation shown in Fig. 8, it can be also compared how the observer eliminates the inertial effects and the noise introduced by the sensors. Note the time lag between the filtered signal and the original one. It was because the selection of the gains made the poles of observer to be quite near the unit-circle (Gamez, 2006). The force control loop applied was an impedance controller (Hogan, 1985).

6. Discussion

The construction of the open robotic system developed in the framework of this work was necessary because, as it is well known, industrial manipulators do not offer, with an appropriate bandwidth, the possibility of integrating either advanced control algorithms or new sensors into the software-hardware architecture. This fact forced the research community to extend an industrial manipulator architecture in order to get a completely open one in both senses: hardware and software.

The proposed platform was designed considering a multi-layer structure that simplified the integration of external functionality in several ways. The first one consisted of offering different interfaces where the user was capable of reading all the parameters and variables, besides modifying the commanded signals, with a considerable bandwidth. The second one pretended to avoid the limitation of the industrial robots where the current methodology is to control exclusively the position without considering high level strategies for task decision making, or without taking into account new sensors that could improve the environment modelling.

Certainly, a pending aspect of this platform is the the man-machine interaction. New solutions in the area of software technology have to be included in order to create a more friendly-interface that permits to modify easily the requirements of the system, especially for the experiment generation. Perhaps, creating pseudo intelligent task interpreters, as a experimental interface, will play an important role.

Furthermore, it has to be pointed out that the design solutions have been driven following a trade-off between mass products (paying attention to the cost) and standardization requirements (leading edge technology).

On the other hand, the proposed architecture was based on consolidated open robotic platforms, specially on those ones developed in Lund University (Sweden) and Leuven University (Belgium). These platforms have been developed during several decades and have accumulated a great deal of experience, representing an excellent paradigm for initial developments. In addition, a narrow collaboration with the company of the manipulator robot has existed, what allowed access to internal functions and hardware what would be impossible in other conditions.

To conclude this section, the development of this kind of platforms does not only prove to be useful for testing advance control algorithms, but also it is necessary to emphasize the necessity of building such systems since, from a robotic research point of view, and mainly, from a robotic manufacturers overview, it helps to increase the development speed opening up the systems for third party.

7. Conclusions

This work describes an experimental platform that allows the implementation of model-based and sensor-based control algorithms in robotic manipulator. Particulary, this new system allows to easily integrate new sensors and advance control algorithms in an Stäubli industrial 6-dof robot using a component-based software methodology.

Based on a component-based development approach, two possible configurations were described, explaining why the original structure was modified migrating to a new one where the Master PC was different to the controller PC. It is also explained how the fact of using this paradigm allowed an easy reconfiguration of the robotic platform, demonstrating

that the use of components -i. e. Sensor components-, that are independent of the context, allowed as well an important restructuring of the new robotic architecture.

The resulting architecture has been designed, among other objectives, to allow different sensors to be easily switched and rewired depending on the new sensor fusion or control strategy that must be tested. Together with the component-based development approach, a software structure of layers has been proposed to facilitate the design, configuration and testing of new control algorithms and sensor fusion techniques. This structure allows systems of components, with standardized interfaces, to be connected while abstracting away implementation details of components.

Eventually, a number of experiments were performed to validate the performance of the proposed architecture and its capacity of allowing a fast and easy implementation of advance control algorithms in non-structured environments.

8. References

- Ahn, S. C., Lee, J.-W., Lim, K.-W., Ko, H., Kwon, Y.-M. & Kim, H.-G. (2006). Requirements to UPnP for Robot Middleware, *Proc. of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, pp. 4716 -4721.
- Ando, N., Suehiro, T., Kitagaki, K. & Kotoku, T. (2006). RT (Robot Technology)-Component and its Standarization, *SICE-ICASE International Joint Conference, 2006*, pp. 2633-2638.
- Bruyninckx, H. (2001). Open robot control software: the OROCOS project. *Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2523-2528.
- Chishiro, H., Fujita, Y., Takeda, A., Kojima, Y., Funaoka, K., Kato, S. & Yamasaki, N. (2009). Extended RT-Component Framework for RT-Middleware, *IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, 2009 (ISORC)*, pp. 161-168.
- Cote, C., Brosseau, Y., Letourneau, D., Raievsky, C. & Michaud, F. (2006). Robotic Software Integration Using MARIE, *International Journal of Advanced Robotic Systems, 2006*, pp. 055-060.
- Gamez, J., Robertsson, A., Gomez, J. & Johansson, R. (2004). Sensor fusion of force and acceleration for robot force control. *Int. Conf. Intelligent Robots and Systems (IROS 2004)*, 2004, pp. 3009-3014.
- Gamez, J., Robertsson, A., Gomez, J. & Johansson, R. (2005). Force and acceleration sensor fusion for compliant robot motion control. *IEEE Int. Conf. on Robotics and Automation (ICRA2005)*, 2005, pp. 2709 - 2714.
- Gamez, J. Sensor Fusion of Force. (2006). Acceleration and Position for Compliant Robot Motion Control. *Phd thesis, Jaen University, Spain, 2006*.
- Gamez, J., Gomez, J. Nieto, L. & Sanchez Garcia, A. (2007). Design and validation of an open architecture for an industrial robot control, *IEEE International Symposium on Industrial Electronics (IEEE ISIE 2007)*, 2007, pp. 2004-2009.
- Gamez, J., Robertsson, A., Gomez, J. & Johansson, R. Sensor fusion for compliant robot motion control. *IEEE Trans. on Robotics*, 2008, pp. 430-441.
- Gamez, J., Gomez, J., Sanchez, A. & Satorres, S. (2009). Robotic software architecture for multisensor fusion system. *IEEE Trans. on Industrial Electronics*, 2009, pp. 766-777.
- Groover, M. P. (2008). Automation, Production Systems and Computer-Integrated Manufacturing. Pearson Education, Upper Saddle River, New Jersey, USA, 2008.

- Hogan, N. (1985). Impedance control: An approach to manipulation, parts 1-3. *J. of Dynamic Systems, Measurement and Control*. ASME, 1985, pp. 1-24.
- Hong, K. Kim, J. Huh, C., Choi, K. & Lee, S. (2001). A pc-based open robot control system: PC-ORC. *IEEE International Symposium on Industrial Electronics, ISIE 2001*. 2001, pp. 1901 -1906.
- Johansson, R. & Robertsson, A. (2003). Robotic force control using observer-based strict positive real impedance control. *IEEE Proc. Int. Conf. Robotics and Automation*, 2003, pp. 3686-3691.
- Khatib, O. & Burdick, J. (1986). Motion and force control of robot manipulators. *IEEE Int. Conf. Robotics and Automation*, 1986, pp 1381- 1386.
- Kröger, T., Kubus, D. & Wahl, F. (2007). Force and acceleration sensor fusion for compliant manipulation control in 6 degrees of freedom. *Advanced Robotics*, 2007, pp. 1603-1616.
- Larsson, U., Forsberg, J. & Wenersson, A. (1996). Mobile robot localization: integrating measurements from a time-of-flight laser. *IEEE Trans. Industrial Electronics*, 1996, pp. 422-431.
- Luo, R., Yih, C. & Su, K. (2002). Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors J.*, 2002, pp. 107-119.
- Nernas, I., Wrigh, A., Bajracharya, M., Simmons, R. & Estlin, T. (2003). CLARATy and Challenges of Developing Interoperable Robotic Software, *Proceedings of the 2003 IEEE/RSJ. Intl. Conference on Intelligent Robots and Systems*, 2003, pp. 2428 - 2435.
- Nilsson, K. & Johansson, R. (1999). Integrated architecture for industrial robot programming and control. *J. Robotics and Autonomous Systems*, 1999, pp. 205-226.
- OMG Robotic Technology Component Specification, formal/08-04-04 edition. Object Management Group, 2008.
- ORCA, <http://orca-robotic.sourceforge.net/>, 2010.
- OROCOS-Simulink, <http://www.orocos.org/simulink/>, 2010.
- OSACA, Open System Architecture for Controls within Automation Systems, ESPRIT III Project 6379/9115, 1996.
- Ostrovrsnik, R., Hace, A. & Terbuc, M. (2003). Use of open source software for hard real-time experiments, *IEEE International Conference on Industrial Technology*, 2003, pp. 1243 - 1246.
- Park, H. & Han, S. (2009). Development of an Open Software Platform for Robotics Services, *ICCAS-SICE Int. Joint Conference*, 2009, pp. 4773 - 4775.
- Pertin, F. & Bonnet des Tuves, J. (2004). Real time robot controller abstraction layer. *Proc. Int. Symposium on Robots (ISR)*, Paris, France, March 2004.
- Saravanan, K., Thangavelu, A. & Rameshbabu, K. (2009). A middleware architectural framework for vehicular safety over vanet (InVANET). *International Conference on Networks & Communications*, 2009, pp. 277 - 282.
- Sarker, M., Kim, C., Cho, J. & You, B. (2006). Development of a Network-based Real-Time Robot Control System over IEEE 1394: Using Open Source Software Platform, *IEEE International Conference on Mechatronics*, 2006, pp. 563 - 568.
- Schlegel, C. & Worz, R. (1999). The Software Framework {SMARTSOFT} for implementing Sensorimotor Systems, *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 1999, pp 1610-1616.

- Utz, H., Sablatnög, S., Enderle, S. & Kraetzschmar, G. (2002). MIRO - Middleware for mobile robot applications, *IEEE Transactions on Robotics and Automation*, 2002, pp. 493 - 497.
- Wills, L., Kannan, S., Sander, S., Guler, M., Heck, B., Prasad, J., Schrage, D. & Vachtsevanos, G. (2001). An Open Platform for Reconfigurable Control, *IEEE Control Systems Magazine*, 2001, pp. 49 - 64.
- Wind-River. VxWorks: Reference Manual. Wind River Systems, 2005.
- Wind-River Linux. <http://www.windriver.com/products/linux/>, 2010.
- Xenomai: Real-Time Framework for Linux. <http://www.xenomai.org/>, 2010.
- Xu, H. & Jia, P. (2006). RTOC: A RT-Linux Based Open Robot Controller, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1644 - 1649.
- Xuemei, L. & Liangzhong, J. (2007). Study on control system architecture of modular robot. *Proc. of the 2007 IEEE Int. Conf. on Robotics and Biometrics*, 2007, pp. 508 - 512.

Structure and Property of the Singularity Loci of Gough-Stewart Manipulator

Y. Cao¹, Y. W. Li² and Z. Huang²

¹*School of Mechanical Engineering, Jiangnan University, 1800 Lihu Avenue, Wuxi, Jiangsu, 214122,*

²*Robotics Research Center, Yanshan University, Qinhuangdao, Hebei, 066004, China*

1. Introduction

During the past two decades, parallel manipulator system has become one of the research attentions in robotics. This popularity has been motivated by the fact that parallel manipulators possess some specific advantages over serial manipulators, i.e., higher rigidity and load-carrying capacity, better dynamic performance and a simpler inverse position kinematics, etc. Among various manipulators, the best-known is the Gough-Stewart Platform (GSP) that was introduced as a tire performance (Gough 1956-57) and an aircraft simulator (Stewart 1965).

One of the important problems in robot kinematics is special configuration or singularity. As to parallel manipulators, in such configurations, the end-effector keeps at least one remnant freedom while all the actuators are locked. This transitorily puts the end-effector out of control. Meanwhile, the articular forces may go to infinity and cause mechanical damages.

Determination of the special configurations of the six-DOF Gough-Stewart parallel manipulators is a very important problem. It is one of the main concerns in the analysis and design of manipulators. The singularity analysis of parallel manipulators has attracted a great deal of attention in the past two decades. Hunt (1983) first discovered a special configuration for this manipulator that occurs when the moving triangle-platform is coplanar with two legs meeting at a vertex of the triangle, and all the six segments associated with six prismatic actuators intersect a common line. Fichter (1986) discovered a singularity of the parallel manipulator. That occurs when the moving platform rotates $\psi = \pm\pi/2$ around Z-axis, whatever the position of the moving platform is. That mechanism has a triangular mobile platform and a hexagonal base platform. It may be named a 3/6-GSP. Huang and Qu (1987) and Huang, Kong and Fang (1997) also studied the singularity of the parallel manipulator, whose moving and basic platforms are both semi-regular hexagons (6/6-GSP). It also occurs when $\psi = \pm\pi/2$. Merlet (1988, 1989) studied the singularity of the six-DOF 3/6-GSP more systematically based on Grassman line geometry. He discovered many new singularities including 3c, 4b, 4d, 5a and 5b. 3c occurs when four lines of the six legs intersect at a common point; 4b occurs when five lines are concurrent with two skew lines; 4d occurs when all the five lines are in one plane or pass through one common point

in that plane; 5a is in general complex; 5b occurs when the six segments cross the same line. Based on line geometry, wrench singularity analyses for platform devices have been presented by Collins & Long (1995), and Hao & McCarthy (1998). Gosselin and Angeles (1990) pointed that singularities of closed-loop mechanisms can be classified into three different groups based on the Jacobian matrices. This classification was further discussed by Zlatanov, Fenton and Benhabib (1994, 1995). Zlatanov, Bonev and Gosselin (2002) discussed constraint singularities. Ma and Angeles (1991) studied architecture singularities of parallel manipulators. Kong (1998) also discussed architecture singularities of the general GSP. McAree and Daniel (1999) discussed the singularity and motion property of a 3/3-parallel manipulator. Karger and Husty (1998), Karger (2001) described the singular positions and self-motions of a special class of planar parallel manipulators where the platform is similar to the base one. It is shown that it has no self-motions unless it is architecturally singular. Kong (1998), Kong and Gosselin (2002) also studied self-motion. Chan and Ebert-Uphoff (2000) studied the nature of the kinematic deficiency in a singular configuration by calculating the nullspace of the Jacobian matrix. Di Gregorio (2004) studied the SX-YS-ZS Structures and Singularity.

Many researches dealt only with isolated singular points in space. However, in the practical configuration space of parallel manipulators the singularity configuration should be a continuous singularity curve or even be high-dimension surface. One of the main concerns is further to find out its singularity loci and their graphical representations, as well as the structure and property of the singularity loci. That is of great significance in a context of analysis and design since it allows one to obtain a complete picture of the location of the singular configurations in the workspace. For a given practical application, it is therefore easy to decide whether the singularities can be avoided. Sefrioui and Gosselin (1994, 1995) studied singularity loci of planar and spherical parallel mechanisms. Wang and Gosselin (1996, 1997) used the numerical method to study the singularity loci of spatial four- and five-DOF parallel manipulators. Collins and McCarthy (1997, 1998) studied singularity loci of the planar 3-RPR parallel manipulator, and 2-2-2 and 3-2-1 platforms and obtained cubic singularity surfaces. For the six-DOF Gough-Stewart Platform, however, the singularity expression generally is quite complicated, and difficult to analyze. Recently, Wang (1998) presented a method to analyze the singularity of a special form of the GSP and derived corresponding analytical singularity conditions. Di Gregorio (2001, 2002) also discussed the singularity loci of 3/6 and 6/6 fully-parallel manipulators. In particular, Mayer St-Onge and Gosselin (2000) analyzed the Jacobian matrix of general Stewart manipulators by two different new approaches. They derived a simpler explicit expression from the Jacobian matrix, and pointed out that the singularity locus of the general Gough-Stewart manipulator should be a polynomial expression of degree three. They also gave the graphical representations of the singularity loci.

For practical application, we want to obtain a simpler algebra expression of the singularity loci, their accurate graphical representations and know whether it consists of some typical geometrical figures. But this is very difficult for the Gough-Stewart manipulator. Huang et al. (1999, 2003) studied the singularity kinematics principle of parallel manipulators, and proved a new kinematics sufficient and necessary condition to determine the singularity. Using this method he discovered the characters of singularity locus of the 3/6-Gough-Stewart platform firstly. It shows that the singularity locus of the 3/6-Gough-Stewart platform is resolvable and consists of two typical geometrical graphs, a plane and a

hyperbolic paraboloid, for the special orientations: $\phi = \pm 30^\circ$, $\pm 90^\circ$, or $\pm 150^\circ$. However, the singularity locus expression of degree three is irresolvable, and the locus graph in infinite parallel principal sections includes a parabola, four pairs of intersecting straight lines and infinite hyperbolas for the general orientations: $\phi \neq \pm 30^\circ$, $\pm 90^\circ$, and $\pm 150^\circ$.

For the singularity loci of the 6/6-GSP which is a more general structure form and widely used in practice, its graphical representations of the singularity loci for different orientations are quite various and complex. Huang and Cao (2005) analyzed the singularity loci both in 3-D space and in the principal-section on which the moving platform lies. The singularity locus equation of this class of Gough–Stewart manipulators in three-dimensional space is also irresolvable, and the curves in infinite parallel principal sections of the singularity loci also contains one parabola, four pairs of intersecting straight lines, and infinite hyperbolas. We also found out an incredible phenomenon, in that special configuration six lines associated with the six extensible links of the 6/6-Gough-Stewart manipulator can intersect the same common line and the remnant instantaneous motion of the manipulator is a pure rotation.

All the above-mentioned analyses are only about positional singularity when the orientation of the moving platform is specified and invariable. On the other hand, there is a need to further find out the orientation-singularity space when the position of the moving platform is specified and invariable. Some researchers began to study the issue, such as Pernkopf and Husty (2002); Cao, Huang & Ge (2006). Of course, for this topic there is still much work to be done in depth.

2. The kinematics principle and linear-complex classification

2.1 The classification of singularity by linear-complex

A general algebraic equation for a linear complex (Hunt 1978; Ball 1900) is:

$$a_1P + a_2Q + a_3R + a_4L + a_5M + a_6N = 0 \quad (1)$$

where the six coefficients denote a twist screw

$$\mathcal{S}^m = (a_1 \ a_2 \ a_3; a_4 \ a_5 \ a_6) \quad (2)$$

Its pitch is

$$h^m = \frac{a_1a_4 + a_2a_5 + a_3a_6}{a_1^2 + a_2^2 + a_3^2} \quad (3)$$

Its reciprocal screw satisfying Eq. (1) is

$$\mathcal{S} = (L \ M \ N; P \ Q \ R) \quad (4)$$

and we have

$$LP + MQ + NR = 0 \quad (5)$$

where \mathcal{S} denotes a line vector. The infinite line vectors satisfying Eq. (1) composed a line complex.

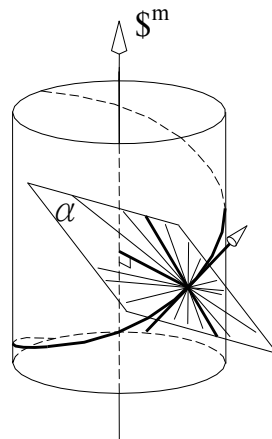


Fig. 1. Linear complex

In a linear complex (Hunt 1978), those lines that pass through any pole must all lie in the same polar plane; those lines that lie in any polar plane must all intersect the same point. Fig. 1 shows the pole and polar plane of a linear complex with the pencil of lines in α . All the lines that pass through the pole are normal to the helix. The linear complex can be divided into three parts according to its pitch h^m : when h^m is finite and nonzero, it is a general linear complex; when $h^m=0$, this is the first special linear complex, in which all the coaxial helices collapse into homocentric circles with a common axis S^m and all the lines of the complex intersect S^m or parallel to it; and when $h^m=\infty$, this is the second special linear complex, in which all the lines of the complex comprise planar fields of lines in all planes normal to the direction S^m , and S^m is no longer occupying a specific line. The last two forms are associated respectively with pure rotation and pure translation.

All singularities of the Gough-Stewart parallel mechanism belong to the linear-complex singularity. From this point of view, the singularity can be divided into three kinds with different instant output motion:

- (1) The general Linear-Complex Singularity. The possible motion of end-effector is a twist with h^m is finite and nonzero;
- (2) The First Special Linear-Complex Singularity. The possible motion of end-effector is a pure rotation with $h^m=0$;
- (3) The Second Special Linear-Complex Singularity. The possible motion of end-effector is a pure translation with $h^m=\infty$.

2.2 The kinematic principle of singularity

First of all, let us discuss the velocity relationship of three points in a moving body. The following issue is to introduce the principle of a novel method analyzing the singularities of parallel manipulators (Huang et al. 1999; 2003; Ebert-Uphoff et al., 2000; Kong and Gosselin 2001). Let us consider any non-collinear three points in a rigid body, and then we may deduce the following theorem:

Theorem 1: Three velocities of three points in a moving body have three normal planes at the corresponding three points. In general, the three planes intersect at a common point, and the intersecting point necessarily lies in the plane determined by the three points.

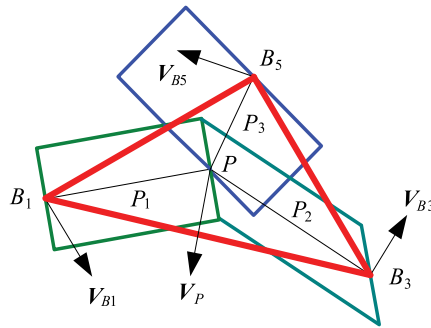


Fig. 2 The velocity relationship of three non collinear points in a moving body

Theorem 2: When three velocity directions of three points in a rigid body are given, then three normal planes of the three velocities are determined. If the intersecting point of the three planes lies in the plane determined by the three points, the three velocities can determine a twist; otherwise, the given velocities are improper and cannot determine a twist of that body.

The thinking of the velocity analysis in the proof of Theorem 2 itself is also useful for singularity study of the 3/6-GSP.

The 3/6-GSP is a typical manipulator which many authors paid attention to. The 3/6-GSP is represented schematically in Fig.3. It consists of a mobile platform \$B_1B_3B_5\$, equilateral triangle; a base platform \$C_1...C_6\$, semi-regular hexagon; and they are connected via six extensible prismatic actuators.

When all the legs of 3/6-GSP are locked, the three normal planes of three velocities \$V_{B_1}, V_{B_3}\$ and \$V_{B_5}\$ are respectively \$B_1C_1C_2, B_3C_3C_4\$ and \$B_5C_5C_6\$ (Fig.3). According to Theorem 2, we may educe the following deduction to determine the singularity of 3/6-GSP. Let us firstly define a "Star-frame \$C-B_1B_3B_5\$" in the moving platform. It is constructed by using three ray lines passing three points, \$B_1, B_2\$ and \$B_3\$, of the triangle \$B_1B_3B_5\$ and intersecting at a common point \$C\$ called the center of Star-Frame.

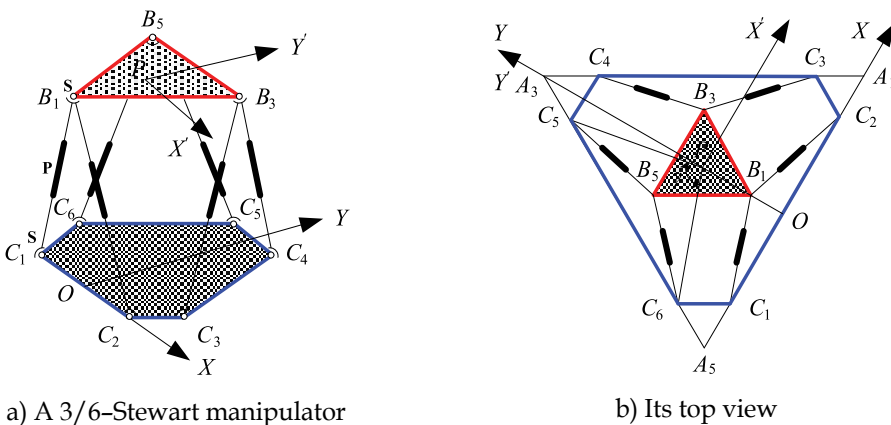


Fig. 3. A 3/6-Stewart parallel manipulator

Theorem 3: A necessary and sufficient condition that the three velocities of three points in a rigid body can express that the body has a possible twist motion is that the intersecting point of three normal planes of the three velocities lies in the plane determined by the three points.

3. Structure and property of singularity loci of 3/6-Gough-Stewart for special orientations

The kinematics method can determine the singularity of the manipulator. If the six extensible legs of the GSP are locked and the mechanism has an instantaneous freedom, the manipulator is singular. Now, let us firstly discuss the kinematics properties of the typical singularity structures including singularities: 3c, 4b, 4d, 5b (Merlet 1988; 1989) and others.

3.1 Singularity hyperbola equation derived in an oblique plane

Our task is to find the whole singularity loci of the GSP and identify their structure and property. It is of an important and difficult issue. Here three Euler angles ϕ , θ and ψ are used to represent orientation of the mobile in terms of a rotation ϕ about Z -axis, then a rotation θ about the new Y' -axis, and finally a rotation ψ about the new Z'' -axis.

In order to find the whole singularity loci and solve the issue, we first study the singularity equation in a special plane (Huang et al. 2003). The issue is divided into two parts:

(1) When the first Euler angle ϕ is equal to one of the following values, $\pm 30^\circ$, $\pm 90^\circ$, and $\pm 150^\circ$, it is a special orientation cases and easier to analyze.

(2) When ϕ is any value with the exception of $\pm 30^\circ$, $\pm 90^\circ$, and $\pm 150^\circ$, this is the general case.

Now, we solve the equation for singularity curve of the 3/6-GSP in a certain plane while the orientation of the mobile is provisionally set to $\phi=90^\circ$, $\psi=0$ and θ is any finite nonzero value.

The parameters of the parallel manipulator are as follows. The circumcircle radius of the basic hexagon platform is R_a , and the one of the triangle mobile is R_b ; β_0 denotes the central angle of the circumcircle of the basic hexagon corresponding to side C_1C_2 . Point P is the geometric center of the mobile (Fig. 3). The stationary frame O - XYZ is fixed to the base and the moving frame P - $X'Y'Z'$ is attached to the mobile.

Fig. 4 shows the position after the mobile rotates $(90^\circ, \theta, 0)$. The oblique plane in which the moving platform lies intersects the basic plane at line UV , which is parallel to axis X . For the orientation, $B_1P(Y')$ is parallel to $A_5A_1(X)$. At first, providing that point P is located at a special point C_0 in the perpendicular bisector of UV , and the distance from O_2 to point C_0 is equal to that between point O_2 and A_3 , then we deduce that C_0B_3 and A_3A_1 intersect at point V , and C_0B_5 and A_3A_5 intersect at U . In that case, the mechanism is singular according to Deduction 2. The included angle between the oblique plane and the basic one is θ . In order to conveniently express the oblique plane below, we call it θ -plane. Let us suppose that the mobile translates to the position $B_{11}B_{31}B_{51}$ in θ -plane and line $B_{11}P$ intersects line O_2C_0 at C . If line $B_{31}C$ intersects A_1A_3 at point V , and line C_1B_{51} intersects A_3A_5 at point U , the mechanism is also singular (Deduction 2). We can prove that the center of Star-frame always lies in line O_2C_0 for the orientation. In general, the singularity is a general-linear-complex singularity. Based on the analysis above, we study the singularities of 3/6-GSP when the mobile translates arbitrarily in θ -plane. The coordinates of point C_0 and O_2 with respect to the fixed frame are $(0, Y_0, Z_0)$ and $(0, u, 0)$, respectively. The frame O_2 - xyz is attached to θ -plane. It should be noticed that angle θ as shown in Fig. 5 about the Y' -axis is negative.

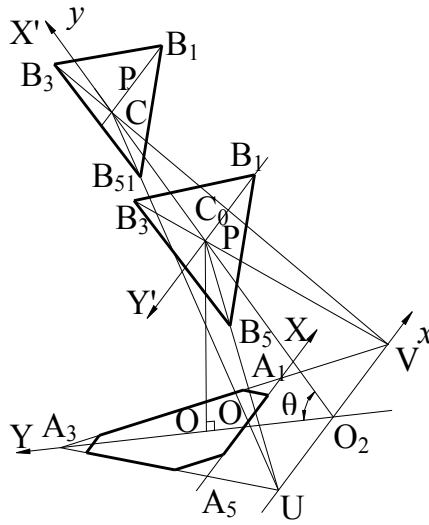


Fig. 4. θ - Oblique plane for the orientation $(90^\circ, \theta, 0)$

The coordinates of points P, C, B_{31} and V with respect to O_2 - xyz are

$$\begin{aligned}
 P &: (x, y, 0) \\
 C &: (0, y, 0) \\
 B_{31} &: \left(x - \frac{R_b}{2}, y + \frac{\sqrt{3}}{2}R_b, 0\right) \\
 V &: \left(-\frac{\sqrt{3}}{3} \frac{Z_0}{\sin \theta}, 0, 0\right)
 \end{aligned} \tag{6}$$

Considering $O_2C_0=O_2A_3$, we can obtain

$$O_2O_1 - OO_1 + OA_3 = O_2C_0 \tag{7}$$

namely

$$3R_a \cos(\beta_0 / 2) - Z_0 \frac{\cos \theta}{\sin \theta} - Y_0 = -\frac{Z_0}{\sin \theta} \tag{8}$$

In the right-angled triangle $\Delta O_1O_2C_0$, we obtain

$$Y_0 - u = -Z_0 \frac{\cos \theta}{\sin \theta} \tag{9}$$

Solving Eqs. (8) and (9) for Y_0 and Z_0 , we obtain

$$\begin{aligned}
 Y_0 &= u(1 - \cos \theta) + 3R_a \cos(\beta_0 / 2) \cos \theta \\
 Z_0 &= u \sin \theta - 3R_a \cos(\beta_0 / 2) \sin \theta
 \end{aligned} \tag{10}$$

Provided that the coordinates of an arbitrary point in line $B_{31}V$ are $(x_x, y_y, 0)$, its equation is written as

$$\frac{y_y - y - \frac{\sqrt{3}}{2} R_b}{-y - \frac{\sqrt{3}}{2} R_b} = \frac{x_x - x + \frac{R_b}{2}}{-\frac{\sqrt{3}}{3} \frac{Z_0}{\sin \theta} - x + \frac{R_b}{2}} \tag{11}$$

Since point C lies in line $B_{31}V$, substitute the coordinates of point C ($x_x = 0$ and $y_y = y$) into Eq. (9) and simplify as

$$xy - \frac{R_b}{2} y - \frac{Z_0 R_b}{2 \sin \theta} = 0 \tag{12}$$

Substituting Eq. (10) into Eq. (11) and eliminating Z_0 , yield

$$xy - \frac{R_b}{2} y + \frac{(3R_a \cos(\beta_0 / 2) - u)R_b}{2} = 0 \tag{13}$$

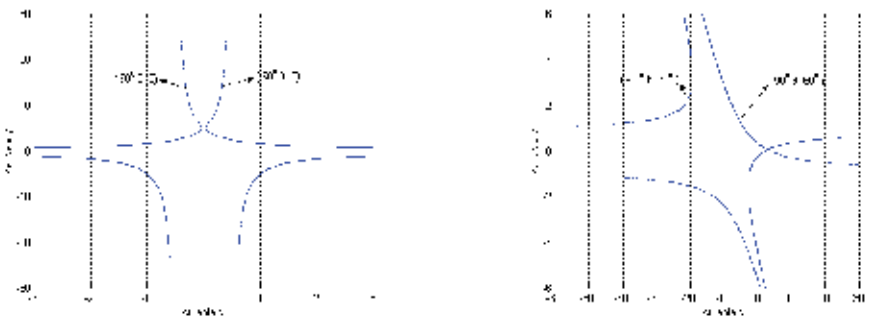
Eq. (13) denotes a hyperbola and is independent of the Euler angle θ . The coordinates of its center are $(R_b/2, 0)$, and its vertical and horizontal asymptotes are $x = R_b/2, y = 0$

This is an important conclusion, as we have known that the singularity equation of GSP in 3-dimension space is a polynomial expression of degree three. However, equation (13) is only a quadratic equation in the special θ -plane. Eq. (13) only contains variables x and y , so it denotes the positions of point p when the mechanism is singular. The equation is termed the equation of the singularity curve in θ -plane.

When orientation of the mobile is given by three Euler angles $(-90^\circ, \theta, 0)$, the singularity equation can also be obtained in θ -plane with respect to the frame O_{2-xyz} , the same as in Fig. 4.

$$xy + \frac{R_b}{2} y - \frac{(3R_a \cos(\beta_0 / 2) - u)R_b}{2} = 0 \tag{14}$$

When parameters of the mechanism are set to $R_a = \sqrt{2}, R_b = 1, \beta_0 = 90^\circ$ and $u = -2$, the hyperbolas denoted by Eqs. (13) and (14) are illustrated in Fig. 6(a). Since the result comes from the above-mentioned Theorem and satisfies the necessary and sufficient condition of singularity, so that there is no any singularity except the points on hyperbolas in that θ -plane.



(a) For the orientation $(\pm 90^\circ \theta 0)$

(b) For the orientation $(\pm 90^\circ \theta 60^\circ)$

Fig. 5. The singularity curve in θ plane

3.2 The singularity equation derived in three-dimensional space

Eqs. (13) and (14) are deduced by geometric method in the oblique plane. By Theorem 3, we can analyze the distribution properties of the singularities of 3/6-GSP in three-dimensional space.

The coordinates of point B_i ($i=1, 2, 3$) of the mobile are denoted as B_i' :(B_{ix}' , B_{iy}' , B_{iz}') in the moving frame, and B_i :(B_{ix} , B_{iy} , B_{iz}) in the fixed frame; the coordinates of point C_j are denoted as (C_{jx} , C_{jy} , C_{jz}) in the fixed frame.

The transformation matrix \mathbf{T} of the moving frame with respect to the fixed one can be written using Euler angles ϕ , θ and ψ as

$$[\mathbf{T}] = \begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta & X \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta & Y \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where(X , Y , Z) are the coordinates of point p with respect to the fixed frame. The coordinates of point B_i in the mobile with respect to the fixed frame are

$$\begin{Bmatrix} B_{ix} \\ B_{iy} \\ B_{iz} \\ 1 \end{Bmatrix} = [\mathbf{T}] \begin{Bmatrix} B_{ix}' \\ B_{iy}' \\ B_{iz}' \\ 1 \end{Bmatrix}, \quad i = 1, 2, 3 \quad (16)$$

3.2.1 Singularity equation for orientation (90° , θ , 0)

When three Euler angles are 90° , θ and 0 , respectively, from Eq. (16), we can obtain coordinates of three points B_i ($i = 1, 2, 3$) in the mobile with respect to the fixed frame. Thus three equations of three normal planes $B_1C_1C_2$, $B_3C_3C_4$ and $B_5C_5C_6$ and the one that the mobile belongs to can be written by the coordinates of the three corresponding points. The equation of plane $B_1C_1C_2$ is

$$\begin{vmatrix} x - B_{1x} & y - B_{1y} & z - B_{1z} \\ C_{1x} - B_{1x} & C_{1y} - B_{1y} & C_{1z} - B_{1z} \\ C_{2x} - B_{1x} & C_{2y} - B_{1y} & C_{2z} - B_{1z} \end{vmatrix} = 0 \quad (17)$$

where x , y and z are the coordinates of moving point in plane $B_1C_1C_2$ with respect to the fixed frame. Substituting coordinates of points B_1 , C_1 and C_2 into the above equation, we obtain

$$Zy - Yz = 0 \quad (18)$$

Similarly, the equation of plane $B_3C_3C_4$ can be obtained

$$\begin{aligned} & (-3R_b \sin\theta + 2\sqrt{3}Z)x + (2Z - \sqrt{3}R_b \sin\theta)y + (-2Y + 6R_a \cos(\beta_0/2) - \sqrt{3}R_b \cos\theta \\ & + \sqrt{3}R_b - 2\sqrt{3}X)z - 6ZR_a \cos(\beta_0/2) + 3\sqrt{3}R_a R_b \sin\theta \cos(\beta_0/2) = 0 \end{aligned} \quad (19)$$

The one of plane $B_5C_5C_6$ is:

$$(-3R_b \sin \theta - 2\sqrt{3}Z)x + (2Z + \sqrt{3}R_b \sin \theta)y + (-2Y + 6R_a \cos(\beta_0 / 2) + \sqrt{3}R_b \cos \theta - \sqrt{3}R_b + 2\sqrt{3}X)z - 6Z R_a \cos(\beta_0 / 2) - 3\sqrt{3}R_a R_b \sin \theta \cos(\beta_0 / 2) = 0 \quad (20)$$

The one of plane $B_1B_3B_5$ is

$$(\sin \theta)y + (\cos \theta)z - (\sin \theta)Y - (\cos \theta)Z = 0 \quad (21)$$

Note that, the equations of these planes are on the same condition that point $P(X, Y, Z)$ is located at some point and the orientation is denoted by three Euler angles $(90^\circ, \theta, 0)$.

Solving Eqs. (18), (19) and (20) for x, y and z , then substituting them into Eq. (21) and eliminating x, y and z , we obtain

$$[(\sin \theta)Y + (\cos \theta)Z][2XZ + R_b(\sin \theta)Y + R_b(\cos \theta)Z - R_b Z - 3R_b R_a \sin \theta \cos(\beta_0 / 2)] = 0 \quad (22)$$

According to Theorem 3, Eq. (22) denotes the singularity locus of point P for the orientation $(90^\circ, \theta, 0)$. Obviously, it includes a plane and a conicoid. The plane equation is

$$(\sin \theta)Y + (\cos \theta)Z = 0 \quad (23)$$

Eq. (23) denotes that singularity locus of point P is a plane containing line C_1C_2 or A_5A_1 , namely, X -axis. As the plane and plane $B_1B_3B_5$ denoted by Eq. (23) have the same normal vector, and when plane $B_1B_3B_5$ translates and coincides with plane expressed by Eq.(23), the configuration is singular. The case belongs to the Hunt's singularity and is the first special-linear-complex singularity explained in Case 5. Eq. (23) shows that the mechanism is singular, wherever point P locates in the plane.

The conicoid equation is

$$2XZ + R_b(\sin \theta)Y + R_b((\cos \theta) - 1)Z - 3R_b R_a \sin \theta \cos(\beta_0 / 2) = 0 \quad (24)$$

When θ is constant, Eq. (24) denotes a hyperbolic paraboloid and we will explain later.

Eq. (28) also represents a hyperbolic paraboloid.

3.2.2 Singularity equation for the orientation $(\pm 90^\circ, \theta, \psi)$

3.2.2.1. The Derivation of the Equation

For the orientation $(\pm 90^\circ, \theta, \psi)$, the transformation matrix \mathbf{T} is

$$[\mathbf{T}] = \begin{bmatrix} -c & -d & 0 & X \\ bd & -bc & a & Y \\ -ad & ac & b & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

where

$$a = \sin \theta ; b = \cos \theta ; c = \sin \psi ; d = \cos \psi \quad (26)$$

Using the same method above, the equations of the three normal planes can be obtained.

$$(acR_b - Z)y + (bcR_b + Y)z = 0 \quad (27)$$

$$\begin{aligned} &\sqrt{3}(\sqrt{3}adR_b - acR_b - 2Z)x + (\sqrt{3}adR_b - 2Z - acR_b)y + (2Y + 2\sqrt{3}X + \sqrt{3}bdR_b - 3cR_b - \\ &\sqrt{3}dR_b - bcR_b - 6R_a \cos(\beta_0 / 2))z + 3R_a \cos(\beta_0 / 2)(2Z + acR_b - \sqrt{3}adR_b) = 0 \end{aligned} \quad (28)$$

$$\begin{aligned} &\sqrt{3}(acR_b + 2Z + \sqrt{3}adR_b)x - (2Z + \sqrt{3}adR_b + acR_b)y + (2Y - bcR_b - 6R_a \cos(\beta_0 / 2) - \\ &\sqrt{3}bdR_b - 3cR_b + \sqrt{3}dR_b - 2\sqrt{3}X)z + 3R_a \cos(\beta_0 / 2)(2Z + acR_b + \sqrt{3}adR_b) = 0 \end{aligned} \quad (29)$$

The equation of plane $B_1B_3B_5$ is

$$ay + bz - aY - bZ = 0 \quad (30)$$

Solving Eqs. (27), (28) and (29) for x , y and z , and then substituting them into Eq. (30), the singularity equation is

$$[(\sin\theta)Y + (\cos\theta)Z](eZ^2 - fXZ + gYZ + hX - iY + jZ + k) = 0 \quad (31)$$

Eq. (31) shows that the singular loci include a plane and a conicoid. The plane equation is the same as Eq. (25). It also represents that in this case all the six lines cross a common line. This case belongs to the first special-linear-complex singularity. The quadratic equation is

$$eZ^2 - fXZ + gYZ + hX - iY + jZ + k = 0 \quad (32)$$

Eq. (32) is a singularity equation with respect to the fixed frame $O-XYZ$. When the mobile shown in Fig. 5 rotates an angle ψ about Z' -axis again, its orientation is $(90^\circ, \theta, \psi)$.

The plane in which the mobile lies is still θ -plane. After the coordinate transformation, the equation of the singularity curve in θ -plane with respect to the frame O_2-xyz is

$$\begin{cases} 2(\sin\psi)y^2 + 2(\cos\psi)xy + R_b \sin(2\psi)x + (-2u \sin\psi + 6R_a \sin\psi \cos(\beta_0 / 2) \\ -R_b \cos(2\psi))y - R_b^2 \sin\psi + R_b \cos(2\psi)(3R_a \cos(\beta_0 / 2) - u) = 0 \\ z = 0 \end{cases} \quad (33)$$

It is also a hyperbola. In addition, Eq. (33) is independent of the Euler angle θ .

3.2.2.2. Analysis of the Singularity Property

The four invariants Δ, D, I and J of Eq. (32) are

$$\Delta = \begin{vmatrix} 0 & 0 & -\frac{f}{2} & \frac{h}{2} \\ 0 & 0 & \frac{g}{2} & -\frac{i}{2} \\ -\frac{f}{2} & \frac{g}{2} & e & \frac{j}{2} \\ \frac{h}{2} & -\frac{i}{2} & \frac{j}{2} & k \end{vmatrix} = \frac{R_b^2 \sin^6 \theta \cos^2 3\psi}{4} \geq 0 \quad (34)$$

$$D = \begin{vmatrix} 0 & 0 & -\frac{f}{2} \\ 0 & 0 & \frac{g}{2} \\ -\frac{f}{2} & \frac{g}{2} & e \end{vmatrix} = 0 \quad (35)$$

$$I = 2 \sin \psi (1 + \cos \theta), \quad J = -\sin^2 \theta \quad (36)$$

The following cases are discussed according to its invariants, in which D is always zero whatever θ and ψ are.

1. If $\theta \neq 0$, $\psi \neq \pm 30^\circ, \pm 90^\circ$, and $\pm 150^\circ$, then $D=0$, $\Delta > 0$, the singular locus denoted by Eq. (32) is a hyperbolic paraboloid. Generally, the six lines 1, 2, ..., 6 belong to a general linear complex when point P locates at the surface.
2. If $\theta=0$, Eq. (32) can be written as

$$4(\sin \psi)Z^3 = 0 \quad (37)$$

- a. When $\psi=0$ and $Z \neq 0$, namely, the orientation is $(90^\circ, 0, 0)$, Eq. (37) is an identical equation and the mechanism is singular whatever the position of point P in three-dimensional space is. This is the Fichter's singular configuration and all the six lines belong to a general linear complex.
- b. When $Z=0$, the moving platform and the base are coplanar. The mechanism is also singular whatever Euler angle ψ is. The mechanism holds three remnant freedoms when all the legs are locked. In this case, there exist the first and the second special-linear-complex singularities.
3. If $\theta \neq 0$, $\psi = \pm 30^\circ, \pm 90^\circ$, or $\pm 150^\circ$, then $D=0$, $\Delta=0$ and $J \neq 0$, and the conicoid degenerates into a pair of intersecting planes. For instance, when $\psi = \pm 30^\circ$, two equations are

$$2Z - R_b \sin \theta = 0 \quad (38)$$

$$\sqrt{3}(\sin \theta)X - (\sin \theta)Y - (1 + \cos \theta)Z - R_b \sin \theta + 3R_a \sin \theta \cos(\beta_0 / 2) = 0 \quad (39)$$

When $\psi = -30^\circ, \pm 90^\circ$, or $\pm 150^\circ$, the conicoid also degenerates into two planes. The singularity cases are similar to the above.

3.2.2.3. Analysis of Other Singularities

The singularities discussed above are all for the orientations, $(\pm 90^\circ, \theta, \psi)$, of the mobile. In these cases, the intersecting lines between the oblique moving plane and the basic one are parallel to line C_1C_2 or A_1A_5 , one of the three sides of the triangle $A_1A_3A_5$.

The similar singularities with a plane equation and a quadratic one can also occur when the orientations are as follows

1. The Euler angles are

$$(-150^\circ, \theta, \psi) \text{ or } (30^\circ, \theta, \psi)$$

All the intersecting lines between the oblique mobile and the base are parallel to line C_3C_4 or A_1A_3 .

2. The Euler angles are

$$(150^\circ, \theta, \psi) \text{ or } (-30^\circ, \theta, \psi)$$

All the intersecting lines between the oblique mobile and the base are parallel to line C_5C_6 or A_3A_5 .

For the two cases, the singularity equation can also resolve into two parts: one is a plane equation containing the corresponding side C_iC_j , another is a hyperbolic paraboloid equation, too. When $\psi = \pm 30^\circ, \pm 90^\circ$, or $\pm 150^\circ$, the hyperbolic paraboloid also degenerates into two planes.

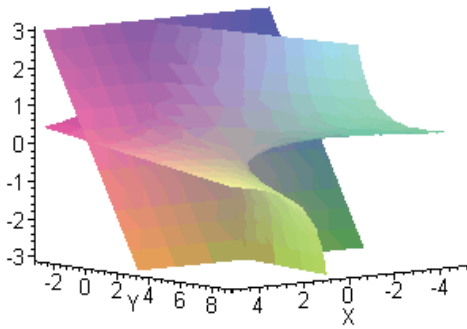
However, when the orientation is

$$(\phi, \theta, \pm 30^\circ), (\phi, \theta, \pm 90^\circ) \text{ or } (\phi, \theta, \pm 150^\circ)$$

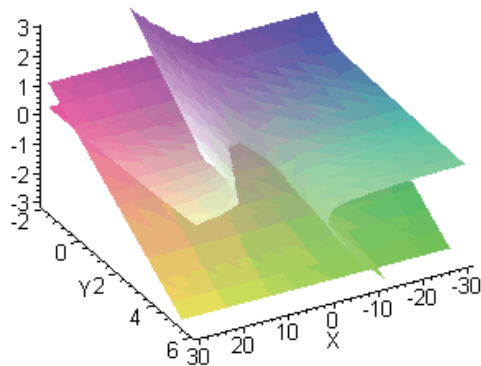
in which ϕ and θ , can be arbitrary values, the singularity locus also consists of two parts: One is a plane; another is also a hyperbolic paraboloid. When point P translates in the plane, two of three points B_1, B_3 and B_5 lie in the basic plane.

3.3 Singularity distribution in three-dimensional space

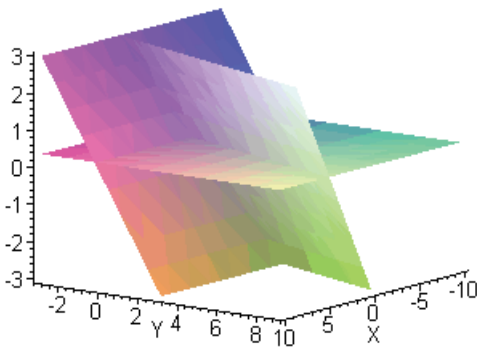
According to the analysis method above, we may easily know the distribution characteristics of the singularity loci of the 3/6-Gough-Stewart manipulator, and draw their singularity surface in three-dimensional space for some different orientations of the mobile in frame $O-XYZ$. Here, the parameters of the mechanism are set to $R_a = \sqrt{2}$, $R_m = 1$ m and $\beta_0 = 90^\circ$, and the surfaces are shown in Fig. 6.



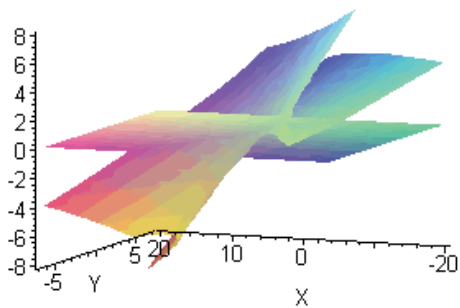
(a) The orientation (90° 45° 0°)



(b) The orientation (90° 30° 60°)



(c) The orientation (90° 45° 30°)



(d) The orientation (45° 25° 30°)

Fig. 6. The singularity loci for 3/6—Stewart parallel manipulator

The readers may wonder that the singularity loci are so huge and completed and ask how can the GSP work? In practice, if you notice the position of the origin point of the O-XYZ system in Figures, and the magnitudes of the parameters, $R_a = \sqrt{2}$ and $R_m=1m$, you can find that the workspace of the manipulator is smaller relative the singularity loci shown in figures. You can easily design the manipulator making its workspace locate over the singularity loci and avoiding singularity.

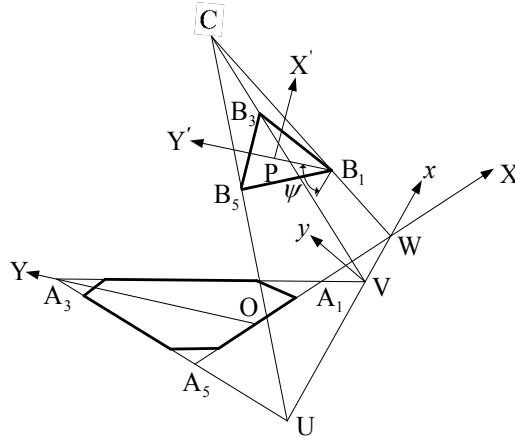


Fig. 7. The general case

4. Structure and property of the singularity loci of the 3/6-Gough-Stewart for general orientations

When ϕ takes any value with the exception of $\pm 30^\circ$, $\pm 90^\circ$, or $\pm 150^\circ$, this is the general orientation case of the mobile of the GSP, and the analysis of the singularity loci is more difficult. In this case, UV is not parallel to any side of triangle $A_1A_3A_5$, as shown in Fig. 7.

4.1 Singularity equation based on Theorem 3 for general orientations

For the most general orientations of the mobile, $\phi \neq \pm 30^\circ$, $\pm 90^\circ$, and $\pm 150^\circ$, the singularity equation can be directly obtained by using Theorem 3. The equation of normal plane $B_1C_1C_2$ is

$$\begin{vmatrix} x' - B_{1x} & y' - B_{1y} & z' - B_{1z} \\ C_{1x} - B_{1x} & C_{1y} - B_{1y} & C_{1z} - B_{1z} \\ C_{2x} - B_{1x} & C_{2y} - B_{1y} & C_{2z} - B_{1z} \end{vmatrix} = 0 \tag{40}$$

where (x', y', z') denotes coordinates of the moving point on plane $B_1C_1C_2$ in the fixed frame. This gives

$$Fy' + Gz' = 0 \tag{41}$$

Similarly, equations of three planes $B_3C_3C_4$, $B_5C_5C_6$ and $B_1B_3B_5$ can be obtained as well. According to Theorem 3, solving the linear equation system of the four planes for intersecting point C, the singularity locus equation for general orientations is as follows

$$\begin{aligned} & f_1 Z^3 + f_2 XZ^2 + f_3 YZ^2 + f_4 X^2 Z + f_5 Y^2 Z + f_6 XYZ + f_7 Z^2 + f_8 X^2 \\ & + f_9 Y^2 + f_{10} XY + f_{11} XZ + f_{12} YZ + f_{13} Z + f_{14} X + f_{15} Y + f_{16} = 0 \end{aligned} \quad (42)$$

where (X, Y, Z) are the coordinates of center point P . It is a polynomial expression of degree three. The equation is still very complicated and difficult to further analyze, but it is very simple in the following special cases.

When $\phi \neq \pm 30^\circ, \pm 90^\circ$, and $\pm 150^\circ$ and ψ is one of the values $\pm 30^\circ, \pm 90^\circ$, or $\pm 150^\circ$, Eq. (42) degenerates into a plane and a hyperbolic paraboloid as well. For example, when $\psi = 90^\circ$, the singularity equation is

$$\begin{aligned} & (2Z + R_b \sin \theta)(a_{11} X^2 + a_{22} Y^2 + a_{33} Z^2 + 2a_{23} YZ + 2a_{31} ZX + 2a_{12} XY + \\ & 2a_{14} X + 2a_{24} Y + 2a_{34} Z + a_{44}) = 0 \end{aligned} \quad (43)$$

where these coefficients are listed in the Appendix 2. Eq. (43) indicates a plane and a hyperbolic paraboloid. The first factor forms a plane equation

$$2Z + R_b \sin \theta = 0 \quad (44)$$

which is parallel to the basic plane. When point P lies in the plane, the mechanism is singular for orientation $(\phi, \theta, 90^\circ)$, because points B_3 and B_5 lie in the basic plane. This is similar to Case 6. All the six lines cross the same line $C_1 C_2$.

4.2 Singularity analysis using singularity-equivalent-mechanism

The singularity locus expression (Eq. (43)) for general orientations has been derived by Theorem 3. But it is still quite complicated, and we are not sure whether it consists of some typical geometrical figures. Meanwhile the property of singularity loci is unknown yet. In order to reply this question, a "Singularity-Equivalent-Mechanism" which is a planar mechanism is proposed. Thus the troublesome singularity analysis of the GSP can be transformed into a position analysis of the simpler planar mechanism.

4.2.1 The parallel case

4.2.1.1 The Singularity-Equivalent-Mechanism

In the parallel case, the three Euler angles of the mobile platform are $(90^\circ, \theta, \psi)$, while θ and ψ can be any nonzero value. The mobile plane of the mechanism lies on θ -plane (Fig. 5).

The corresponding imaginary planar singularity-equivalent-mechanism is illustrated in Fig. 8. Where R denotes a revolute pair and P a prismatic pair, triangle $B_1 B_3 B_5$ is connected to ground by three kinematic chains, RPP, PPR and RPR. The latter two pass through two points U and V , respectively, while the first one slides along the vertical direction and keeps $L_1 C // UV$. Three slotted links, L_1, L_2 and L_3 , intersect at a common point C . In order to keep the three links always intersecting at a common point and satisfying Deduction 2, a concurrent kinematic chain PRPRP is used. It consists of five kinematic pairs, where two R pairs connect three sliders. The three sliders and three slotted links form three P pairs. The PRPRP chain coincides with a single point C from top view. Based on the Grübler-Kutzbach criterion, the mobility of the mechanism is two.

It is evident that the planar mechanism can guarantee that the three lines passing through three vertices intersect at a common point, and these three lines can always intersect the

corresponding sides of the basic triangle. From Deduction 2, every position of the planar mechanism corresponds to a special configuration of the original GSP. So we call it a “singularity-equivalent-mechanism”. Thus the position solution of the planar mechanism expresses the singularity of the original mechanism.

4.2.1.2 Forward Position Analysis of the Singularity -Equivalent-Mechanism

The frames are set as the same as in Fig. 5 and Fig. 10. The coordinates of point P in frame O_2-xy are (x, y) . ψ indicates the orientation of the triangle $B_1B_3B_5$ in θ -plane. In order to obtain the locus equation of point P, firstly we can set three equations of three lines passing through the three vertices, and substitute the coordinates of points B_1, B_3 and B_5 into the equations, then (x, y) and ψ can be obtained.

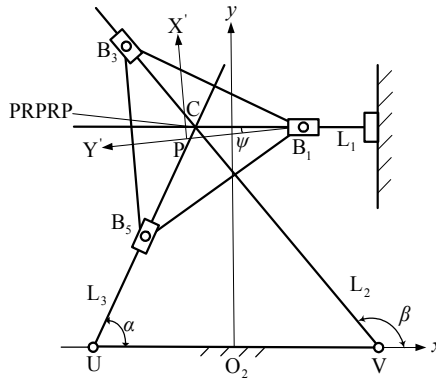


Fig. 8. The singularity-equivalent-mechanism for $(90^\circ, \theta, \psi)$

Considering that the mobility of this mechanism is two, there need two inputs α and β . Three equations of three lines CU, CV and CB₁ in reference frame O_2-xy are respectively

$$Y = (\tan \alpha)(X + a / 2) \tag{45}$$

$$Y = (\tan \beta)(X - a / 2) \tag{46}$$

and

$$Y = -\frac{a \tan \alpha \tan \beta}{\tan \alpha - \tan \beta} \tag{47}$$

Solving Eqs. (52), (53) and (54) yields

$$x = \frac{R_b \cos \psi J_1 - (\sqrt{3}R_b \sin \psi + a)J_3}{2(\tan \alpha - \tan \beta)} \tag{48}$$

$$y = \frac{R_b \sin \psi J_2 - \sqrt{3}R_b J_3 \cos \psi - 2a \tan \alpha \tan \beta}{2(\tan \alpha - \tan \beta)} \tag{49}$$

and

$$\tan \psi = \frac{(\tan \beta + \tan \alpha)}{\sqrt{3}\tan \alpha - \sqrt{3}\tan \beta - 2\tan \alpha \tan \beta} \tag{50}$$

where $J_1 = \tan a - \tan \beta - 2\sqrt{3}$, $J_2 = \tan a - \tan \beta - 2\sqrt{3} \tan a \tan \beta$, $J_3 = \tan \alpha + \tan \beta$, Eqs. (48), (49) and (50) denote direct kinematics of the mechanism.

4. 2. 1. 3. Singularity Equation in the θ plane

Once the orientation (90° , θ , ψ) of the mobile platform is specified, in Fig.10, Euler angle ψ is an invariant. So it only needs to choose one input in this case. From Eq. (65) one obtains

$$\tan \beta = \frac{\tan \alpha (\sqrt{3} \tan \psi - 1)}{\sqrt{3} \tan \psi + 2 \tan \alpha \tan \psi + 1} \quad (51)$$

So the singularity equation in θ plane for the orientation (90° , θ , ψ) can be obtained from Eqs. (48), (49) and (50) by eliminating parameters α and β

$$\begin{aligned} & 2(\sin \psi)y^2 + 2(\cos \psi)xy + R_b \sin(2\psi)x + \\ & (\sqrt{3}a \sin \psi - R_b \cos(2\psi))y - R_b^2 \sin \psi + \sqrt{3}aR_b \cos(2\psi) / 2 = 0 \end{aligned} \quad (52)$$

where $a = 2(3R_a \cos(\beta_0 / 2) - u) / \sqrt{3}$. Eq. (52) denotes a hyperbola. Especially, when $\psi = \pm 90^\circ$, Eq. (52) degenerates into a pair of intersecting straight lines respectively. Two of the four equations are

$$y - R_b/2 = 0, \quad y + R_b/2 = 0 \quad (53)$$

In both cases, two points B_3 and B_5 lie in line UV. So that four lines are coplanar with the base plane. This is the singularity of Case 6. The similar situation is for $\psi = 30^\circ, \psi = -150^\circ, \psi = -30^\circ$ and $\psi = 150^\circ$.

4.2.2 The general case

When $\psi \neq \pm 30^\circ, \pm 90^\circ$, and $\pm 150^\circ$, the intersecting line UVW between θ -plane and the base one is not parallel to any side of triangle $A_1A_3A_5$. This is the most general and also the most difficult case.

4.2.2.1 The Singularity-Equivalent-Mechanism

Fig. 11 shows the singularity-equivalent-mechanism. The triangle $B_1B_3B_5$ is connected to the ground passing through three points W, V and U by three RPR kinematic chains. The three points U, V and W, as shown in Fig. 9, are three intersecting points between θ -plane and sides A_3A_5 , A_1A_3 and A_1A_5 , respectively. Three slotted links L_1 , L_2 and L_3 intersect at a common point C. In order to keep the three links always intersecting at a common point, a concurrent kinematic chain, PRPRP, is used as well. Therefore, all the configurations of the equivalent mechanism satisfying Deduction 2 are special configurations of the Gough-Stewart manipulator. So we can analyze direct kinematics of the equivalent mechanism to find singularity loci of the manipulator.

Similarly the mobility of the equivalent mechanism is two, and it needs two inputs when analyzing its position.

4.2.2.2 Forward Position Analysis of the Singularity-Equivalent -Mechanism

The frames are set as shown in Fig. 11. Similar to section 4.2.1.2, we may set three equations of three straight lines passing through three vertices, and substitute the coordinates of points B_1 , B_3 and B_5 into the equations, then solutions, (x, y) and ψ , can be obtained

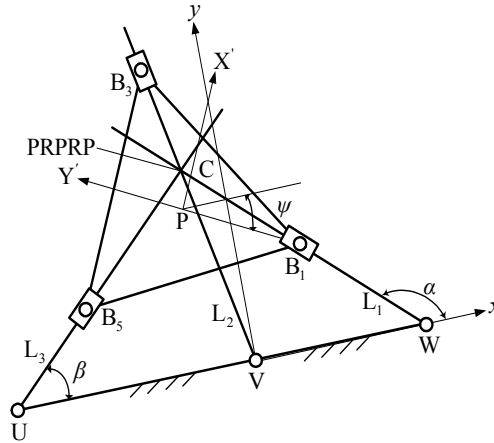


Fig. 9. The singularity-equivalent-mechanism for general case

$$\begin{aligned} x = & -(3R_b \sin \psi - 2R_b \tan \alpha \cos \psi + 2w \tan \alpha + \sqrt{3}R_b \cos \psi \\ & - R_b \tan \beta \cos \psi + \sqrt{3}R_b \tan \beta \sin \psi - 2u \tan \beta) / (2 \tan \beta - 2 \tan \alpha) \end{aligned} \quad (54)$$

$$\begin{aligned} y = & (-R_b \tan \alpha \sin \psi - \sqrt{3}R_b \tan \alpha \cos \psi + 3R_b \tan \alpha \tan \beta \cos \psi + 2u \tan \alpha \tan \beta \\ & - \sqrt{3}R_b \tan \alpha \tan \beta \sin \psi - 2R_b \tan \beta \sin \psi - 2w \tan \alpha \tan \beta) / (2 \tan \beta - 2 \tan \alpha) \end{aligned} \quad (55)$$

$$\tan \psi = \frac{2\sqrt{3} w \tan \alpha - 3u \tan \alpha \tan \beta - \sqrt{3}u \tan \beta}{\tan \beta (-2\sqrt{3} w \tan \alpha + \sqrt{3}u \tan \alpha - 3u)} \quad (56)$$

where u indicates the distance from point U to V, and w the distance from V to W. Substituting Eq. (56) into Eqs. (55) and (54), and eliminating ψ , the relations between (x, y) and the inputs α, β can be obtained. This is direct kinematics of the equivalent mechanism.

4.2.2.3 Singularity Equation in the θ plane

Under a general case, Euler angle ϕ can be any value with the exception of $\pm 30^\circ$, $\pm 90^\circ$, or $\pm 150^\circ$. From Eq (56) one obtains

$$\tan \beta = \frac{2\sqrt{3}w \tan \alpha}{-2\sqrt{3}w \tan \alpha \tan \psi + \sqrt{3}u \tan \alpha \tan \psi - 3u \tan \psi + 3u \tan \alpha + \sqrt{3}u} \quad (57)$$

In the case of some specified ψ , there are the same three particular situations that is B_1 and B_5 , B_1 and B_3 , or B_3 and B_5 lie in the line UV, respectively. The singularity loci are three pairs of intersecting straight lines.

In order to use the above-mentioned formulas, u and w in Eq. (57) should be calculated in advance. They depend on their relative positions in UV, as shown in Fig. 10.

The distance w between V and W is

$$|w| = |WV| = \left| \frac{3R_a \cos(\beta_0 / 2) - \sqrt{3}x_v}{\cos \phi} \right| \quad (68)$$

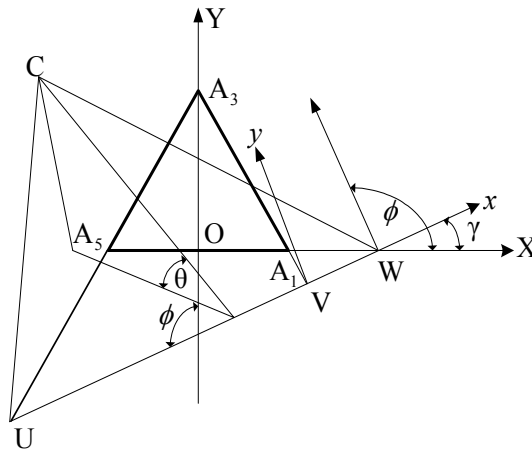


Fig. 10. The Intersecting Line UW of two planes

The distance u between U and V is

$$|u| = |UV| = \left| \frac{2\sqrt{3}x_v}{(\sqrt{3} + \cot \phi) \sin \phi} \right| \tag{59}$$

The sign of w is positive when point W is on the right side of V, and it is negative when W is on the left side of V. It is similar for the sign of u .

For a given x_v , the singularity equation in θ -plane can be obtained by eliminating the parameter α

$$bxy + cy^2 + dx + ey + f = 0 \tag{60}$$

The two invariants D, δ of Eq. (60) are

$$D = \begin{vmatrix} 0 & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{vmatrix} = -\frac{1}{4}(b^2f + d^2c - bde) \tag{61}$$

and

$$\delta = \begin{vmatrix} 0 & b/2 \\ b/2 & c \end{vmatrix} = -\frac{1}{4}b^2 < 0 \tag{62}$$

Generally, $D \neq 0$ and $\delta < 0$ for a general value of x_v , so Eq. (60) indicates a set of hyperbolas.

4.3 Five special cases of the singularity equation

There are five special cases. For the given parameters (R_a, R_b, β_0) and (ϕ, θ, ψ) , D is a quartic equation while δ a quadratic equation with respect to the single variable x_v , respectively. Generally, there are four real roots of x_v when $D=0$ and $\delta \neq 0$, and Eq. (60) degenerates into four pairs of intersecting straight lines. For the same reason, there is one real root of multiplicity 2 when $\delta=0$ and $D \neq 0$, and Eq. (60) degenerates into a parabola.

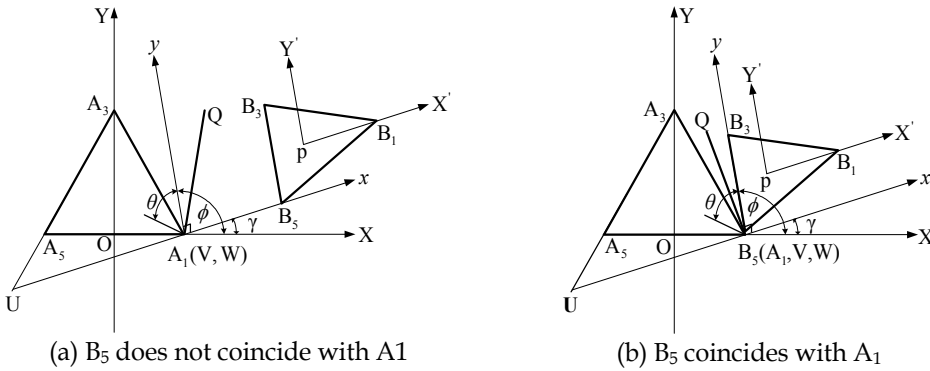


Fig. 11. UV passes through the points A₁

Case 1. The line UV passes through point A₁, as shown in Fig. 11. In this case $x_v = \sqrt{3}R_a \cos(\beta_0 / 2)$, two points W and V coincide with point A₁. The singularity equation denoted by Eq. (75) degenerates into a pair of intersecting straight lines

$$[y - R_b \sin(\psi + 60^\circ)][(-\sqrt{3} \sin(\psi) + \cos(\psi))x + (\sqrt{3} \cos(\psi) + \sin(\psi))y + R_b] = 0 \quad (63)$$

One of them is

$$y - R_b \sin(\psi + 60^\circ) = 0 \quad (64)$$

Case 2. UV passes through point A₃. In this case $x_v = 0$, two points U and V coincide with point A₃. Eq. (60) degenerates into a pair of intersecting straight lines either

$$[(y + R_b \sin(\psi))[x \cos(\psi) + y \sin(\psi) - R_b / 2] = 0 \quad (65)$$

The first part of Eq. (65) indicates a straight line parallel to x -axis. Similarly when B₁ coincides with point A₃, the singularity of this point is the first special-linear-complex singularity and the instantaneous motion is a pure rotation. When B₁ does not coincide with A₃, the singularities of points lying in this straight line are the general-linear-complex singularity and its instantaneous motion is a twist with $h^m \neq 0$.

The second part of Eq. (62) denotes another straight line. The singularities of points lying in this straight line are all the general-linear-complex singularity.

Case 3. UV passes point A₅. In this case

$$x_v = \sqrt{3}R_a \cos(\beta_0 / 2)(\sqrt{3} + \cot \phi) / (\sqrt{3} - \cot \phi)$$

two points U and W coincide with point A₅. Eq. (60) degenerates into a pair of intersecting straight lines.

$$[y - R_b \sin(\psi - 60^\circ)][(\sqrt{3} \sin(\psi) + \cos(\psi))x + (-\sqrt{3} \cos(\psi) + \sin(\psi))y + R_b - 2\sqrt{3} \cos(\beta_0 / 2) \sin(\psi + 60^\circ) / \sin(\phi - 60^\circ)] = 0 \quad (66)$$

The first factor indicates a straight line parallel to the x -axis. Similarly when B₃ coincides with A₅, the singularity of this point is the first special-linear-complex singularity. When B₃ does not coincide with A₅, the singularities of points lying in this straight line are the general-linear-complex singularity.

Similarly the second factor of Eq. (66) denotes another straight line. The singularities of points lying in this straight line are all the general-linear-complex singularity.

Case 4. When

$$x_v = \frac{(-1 + 2 \cos(2\phi))(R_b \cos \phi - 2R_a \cos(\beta_0 / 2) \cos \psi)}{(2(\sqrt{3} \sin \phi - \cos \phi) \sin(\phi + \psi))} \quad (67)$$

Eq. (67) degenerates into a pair of intersecting straight lines as well

$$\begin{aligned} & (R_a \cos((\beta_0 - 6\psi) / 2) - R_b \cos(\phi - 2\psi) + R_a \cos((\beta_0 + 6\psi) / 2) \\ & - 2y \sin(\phi + \psi))(ax + by + c) = 0 \end{aligned} \quad (68)$$

For the first straight line when $\beta_0=90^\circ$, $(\phi, \theta, \psi)=(60^\circ, 30^\circ, 0)$, and the coordinates of point P_6 are $x = R_b/2$, $y = (2\sqrt{2}R_a - R_b)/2\sqrt{3}$, point B_5 lies in the intersecting line of two normal planes $B_1A_1A_5$ and $B_3A_1A_3$. Therefore, the six lines associated with the six extensible links of the 3/6-GSP intersect a common line B_5A_1 . It is the first special-linear-complex singularity. The instantaneous motion is a pure rotation about line B_5A_1 . The singularities of points lying in the first line with the exception of the above-mentioned point and the points lying in the second line are all belong to the general-linear-complex singularity.

Case 5. When

$$x_v = R_a \cos(\beta_0 / 2) \cos \psi (\cos \phi + \sqrt{3} \sin \phi) / \sin(\phi + \psi), \quad \delta \neq 0 \text{ and } D \neq 0,$$

Eq. (60) degenerates into a parabola.

$$cy^2 + dx + ey + f = 0 \quad (69)$$

According to the analysis mentioned above, it is shown that the singularity expression in θ -plane is not cubic but always quadratic. This indicates the θ -plane is a very special cross section of the singularity surface, so the special θ -plane can be called the principal section. Generally speaking, the singularity loci of the 3/6-GSP for the most general orientations are different from those for some special orientations. The singularity loci in infinite parallel principal sections are all quadratic equations. The structure of the singularity loci in the principal sections of the cubic singularity surface includes a parabola, four pairs of intersecting straight lines and infinity of hyperbolas. The singularity loci in three-dimensional space are illustrated in Fig. 12.

In addition, it should be pointed out that once the mechanism is singular at the orientation (ϕ, θ, ψ) , any orientation with different variable θ is singular as well (Huang et al. 2003).

5. Structure and property of the singularity loci of the 6/6-Gough-Stewart

Base on the above-mentioned analysis of the 3/6-GSP, here we focus on the most difficult issue, the singularity locus analysis of the 6/6-GSP including the singularity equation and the structure of singularity surface. The 6/6-GSP is typical manipulator. The 6/6-GSP is represented schematically in Fig. 13. It consists of two semi-regular hexagons: a mobile platform $B_1B_3\dots B_6$ and a base platform $C_1\dots C_6$. They are connected via six extensible prismatic actuators.

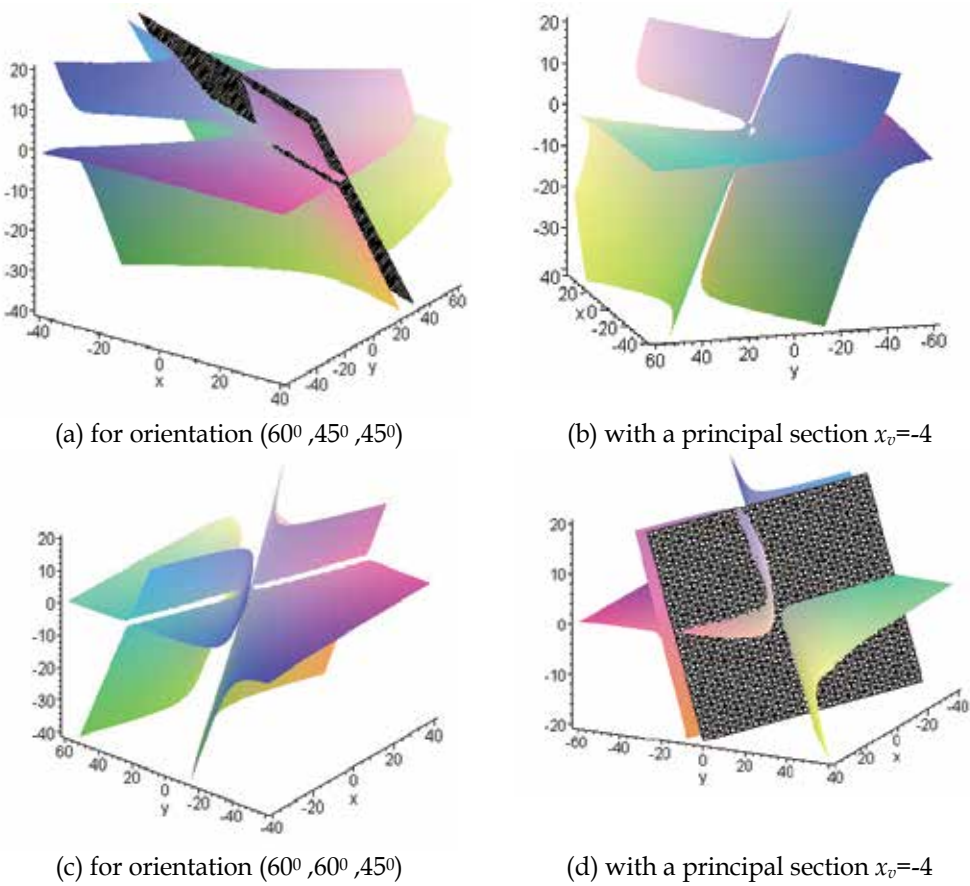


Fig. 12. The singularity loci in three-dimensional space for the general orientations

5.1 The Jacobian matrix

The Jacobian matrix of this class of the Gough-Stewart manipulators can be constructed as follows according to the theory of static equilibrium (Huang and Qu 1987)

$$\begin{aligned}
 [\mathbf{J}]^T &= [\mathcal{S}_1 \ \mathcal{S}_2 \ \mathcal{S}_3 \ \mathcal{S}_4 \ \mathcal{S}_5 \ \mathcal{S}_6] = \begin{pmatrix} \mathcal{S}_1 & \mathcal{S}_2 & \mathcal{S}_3 & \mathcal{S}_4 & \mathcal{S}_5 & \mathcal{S}_6 \\ \mathcal{S}_{O1} & \mathcal{S}_{O2} & \mathcal{S}_{O3} & \mathcal{S}_{O4} & \mathcal{S}_{O5} & \mathcal{S}_{O6} \end{pmatrix} \\
 &= \begin{pmatrix} \frac{(\mathbf{B}_1 - \mathbf{C}_1)}{\|\mathbf{B}_1 - \mathbf{C}_1\|} & \frac{(\mathbf{B}_2 - \mathbf{C}_2)}{\|\mathbf{B}_2 - \mathbf{C}_2\|} & \frac{(\mathbf{B}_3 - \mathbf{C}_3)}{\|\mathbf{B}_3 - \mathbf{C}_3\|} & \frac{(\mathbf{B}_4 - \mathbf{C}_4)}{\|\mathbf{B}_4 - \mathbf{C}_4\|} & \frac{(\mathbf{B}_5 - \mathbf{C}_5)}{\|\mathbf{B}_5 - \mathbf{C}_5\|} & \frac{(\mathbf{B}_6 - \mathbf{C}_6)}{\|\mathbf{B}_6 - \mathbf{C}_6\|} \\ \frac{(\mathbf{C}_1 \times \mathbf{B}_1)}{\|\mathbf{B}_1 - \mathbf{C}_1\|} & \frac{(\mathbf{C}_2 \times \mathbf{B}_2)}{\|\mathbf{B}_2 - \mathbf{C}_2\|} & \frac{(\mathbf{C}_3 \times \mathbf{B}_3)}{\|\mathbf{B}_3 - \mathbf{C}_3\|} & \frac{(\mathbf{C}_4 \times \mathbf{B}_4)}{\|\mathbf{B}_4 - \mathbf{C}_4\|} & \frac{(\mathbf{C}_5 \times \mathbf{B}_5)}{\|\mathbf{B}_5 - \mathbf{C}_5\|} & \frac{(\mathbf{C}_6 \times \mathbf{B}_6)}{\|\mathbf{B}_6 - \mathbf{C}_6\|} \end{pmatrix} \quad (70)
 \end{aligned}$$

where vectors, $\mathbf{B}_i, \mathbf{C}_i$ ($i=1, 2, \dots, 6$), respectively denote the vertex vectors of the moving and base platforms with respect to the fixed frame, Fig. 15; \mathcal{S}_i ($i=1, 2, \dots, 6$) is a line vector of the corresponding extensible link, and its Plücker coordinates are as follows $\mathcal{S}_i = (\mathcal{S}_i; \mathcal{S}_{O_i}) = (L_i, M_i, N_i; P_i, Q_i, R_i)$ where the subscript i ($i=1, 2, \dots, 6$) indicates the i th limb connected by two vertices $\mathbf{B}_i, \mathbf{C}_i$ of the moving and base platforms of the manipulator. \mathcal{S}_i is a unit vector

specifying the direction of line vector S_i , and S_{O_i} is a vector indicating the position of the line vector together with S_i .

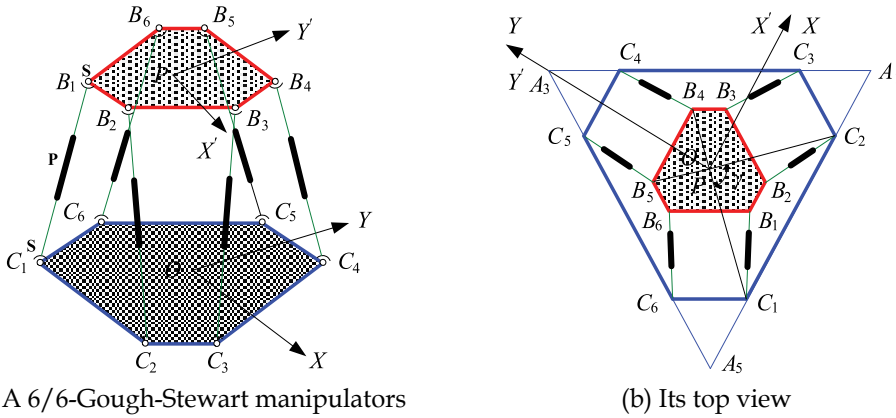


Fig. 13. Schematic of a class of the Gough-Stewart manipulators

5.2 Singularity analysis in three-dimensional space

A moving reference frame $P-X'Y'Z'$ and a fixed one $O-XYZ$ are respectively attached to the moving platform and the base platform of the manipulator, as shown in Fig. 15, where origins P and O are corresponding geometric center of the moving and base platforms. The position of the moving platform is given by the position of point P with respect to the fixed frame, designated by (X, Y, Z) , and the orientation of the moving platform is represented by the standard $Z-Y-Z$ Euler angles (ϕ, θ, ψ) . Furthermore, geometric parameters of the manipulator can be described as follows. The circumcircle radius of the base hexagon is R_a , and that of the mobile hexagon is R_b . β_0 denotes the central angle of circumcircles of the hexagons corresponding to sides C_1C_2 and B_1B_6 , as shown in Fig. 15. The coordinates of six vertices, B_i ($i=1, 2, \dots, 6$), of the moving platform are denoted by B'_i with respect to the moving frame, and B_i with respect to the fixed frame. Similarly, C_i and A_j represent coordinates of vertices, C_i ($i=1, 2, \dots, 6$) and A_j ($j=1, 3, 5$), of the base platform with respect to the fixed frame.

Gosselin and Angeles (1990) pointed out that singularities of parallel manipulators could be classified into three different types, i.e., inverse kinematic singularity, direct kinematic singularity and architecture singularity. Here we only discuss the direct kinematic singularity of this class of 6/6-Gough-Stewart manipulators, which occurs when the determinant of the Jacobian matrix of the manipulator is equal to zero, i.e., $\det(J)=\det(J^T)=0$. Expanding and factorizing the determinant of the Jacobian matrix, the singularity locus equation of the manipulator can be written as

$$\begin{aligned}
 & f_1Z^3 + f_2XZ^2 + f_3YZ^2 + f_4X^2Z + f_5Y^2Z + f_6XYZ + f_7Z^2 + f_8X^2 + \\
 & f_9Y^2 + f_{10}XY + f_{11}XZ + f_{12}YZ + f_{13}Z + f_{14}X + f_{15}Y + f_{16} = 0
 \end{aligned}
 \tag{71}$$

Eq. (71) represents the constant-orientation singularity locus of this class of the Gough-Stewart manipulators in the Cartesian space for a constant orientation (ϕ, θ, ψ) . It is a polynomial expression of degree three in the moving platform position parameters XYZ .

Coefficients of Eq. (71), f_i ($i=1, 2, \dots, 15, 16$), are all functions of geometric parameters, R_a , R_b and β_0 , and orientation parameters, (ϕ, θ, ψ) , of the manipulator.

Graphical representations of the constant-orientation singularity locus of the manipulator for different orientations are given to illustrate the result, as shown in Fig. 14. Geometric parameters used here are given as $R_b=2$, $R_a=1.5$, $\beta_0=\pi/2$.

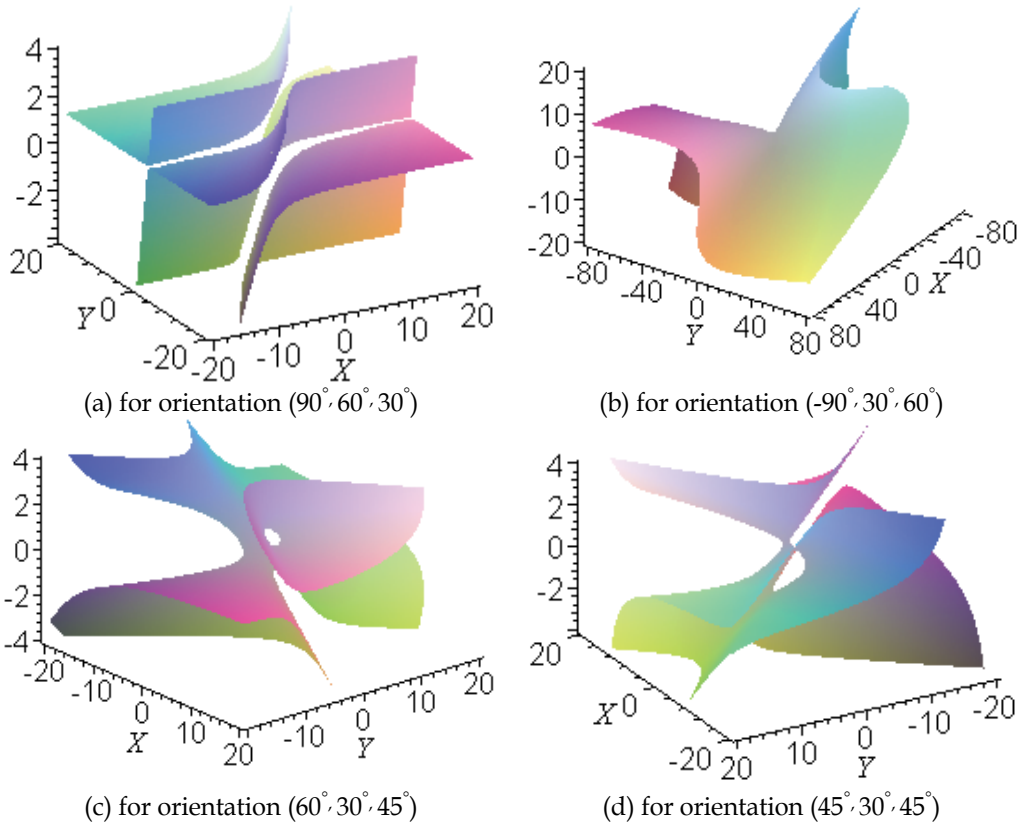


Fig. 14. Singularity loci for different orientations

From Figure 14, it can be clearly seen that the singularity loci for different orientations are quite different, and they are complex and various. Among them, the most complicated graph of the singularity loci looks like a trifoliate surface, whose two branches are of the shape of a horn with one hole (Figure 14 (c) and (d)).

5.3 Singularity analysis in parallel principal-sections

5.3.1 Singularity locus equation in θ -plane

Huang, Chen and Li (2003) pointed out that the cross-sections of the cubic singularity locus equation of the 3/6-GSP in parallel θ -planes are all quadratic expressions that include a parabola, four pairs of intersecting lines and infinite hyperbolas. This conclusion is of great importance for the property identification of the singularity loci of the 3/6-GSP. Similarly, in order to identify the characteristics of singularity loci of this class of the 6/6-GSP, singularity loci of the manipulator in parallel θ -planes will also be discussed in this section. Fig. 16

shows the position of the manipulator for orientation (ϕ, θ, ψ) . The oblique plane is θ -plane on which the moving platform lies.

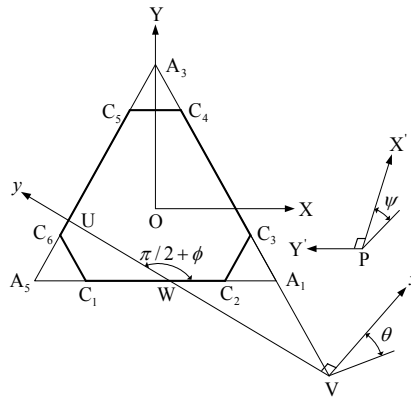


Fig. 15. The position of the manipulator for orientation (ϕ, θ, ψ)

When $\theta \neq 0$, the moving platform is not parallel to the base one. θ -plane intersects the base plane at a line UWV , where points U, W, V are intersecting points between θ -plane and three sides, $A_3A_5, A_1A_5,$ and A_3A_1 , of the base hexagon, as shown in Fig. 15. We set another moving reference frame V - xy in θ -plane, and the coordinates of point P in this moving frame V - xy are denoted by (x, y) .

Equations of three lines, $A_3A_5, A_1A_5,$ and A_3A_1 , in the fixed frame O - XYZ can be easily written. Owing to space limitations, we do not present these equations here. As point V , i.e., origin of the moving frame V - xy , lies on line A_3A_1 , and it can be assumed that the coordinates of point V with respect to the fixed frame O - XYZ are

$$V : (X_V, Y_V, 0)$$

where X_V is a variable indicating the position of θ -plane, i.e., the position of the moving platform for any given geometric and orientation parameters, and Y_V can be established by the following expression

$$Y_V = 2R_a \cos(\beta_0 / 2) - \sqrt{3}X_V \quad (72)$$

So, the equation of line UV can be written as follows

$$Y_{UV} - 2R_a \cos(\beta_0 / 2) + \sqrt{3}X_V = -\cot(\phi)(X_{UV} - X_V) \quad (73)$$

Therefore, coordinates of points U and W can be easily obtained. The coordinates of point P designated by (X, Y, Z) with respect to the fixed frame and (x, y) in the moving frame V - xy satisfy the following expression

$$\begin{aligned} X &= \cos \phi \cos \theta x - \sin \phi y + X_V \\ Y &= \sin \phi \cos \theta x + \cos \phi y + Y_V \\ Z &= -\sin \theta x \end{aligned} \quad (74)$$

Substituting Eq. (81) into Eq. (78) and after some rearrangements and factorizations, the singularity locus equation of the manipulator in θ -plane can be written as follows

$$\sin^3 \theta (ax^2 + 2bxy + cy^2 + 2dx + 2ey + f) = 0 \tag{75}$$

Since $\theta \neq 0$, the singularity locus equation of the manipulator with respect to θ -plane becomes

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \tag{76}$$

It can be proved that the coefficient c is always equal to zero, so Eq. (76) is a quadratic polynomial expression with respect to x and y , and the maximum degree of variable x is 2 and y is 1. Coefficients a, b, d, e, f of Eq. (76) are all functions of geometric parameters R_a, R_b and β_0 , Euler angles (ϕ, ψ) and X_V . They are all independent of Euler angle θ . Generally, the intersecting curve between a cubic surface and a plane is also a cubic expression that may also contain a closed-loop curve. For example, when $R_a=2, R_b=1.5, \beta_0=\pi/2, (\phi, \theta, \psi)=(\pi/3, \pi/6, \pi/4)$, intersecting curves between the corresponding singularity locus surface and the following two planes, $Z=-Y/3$ and $Z=-4(X-14)/45$, are respectively presented as follows

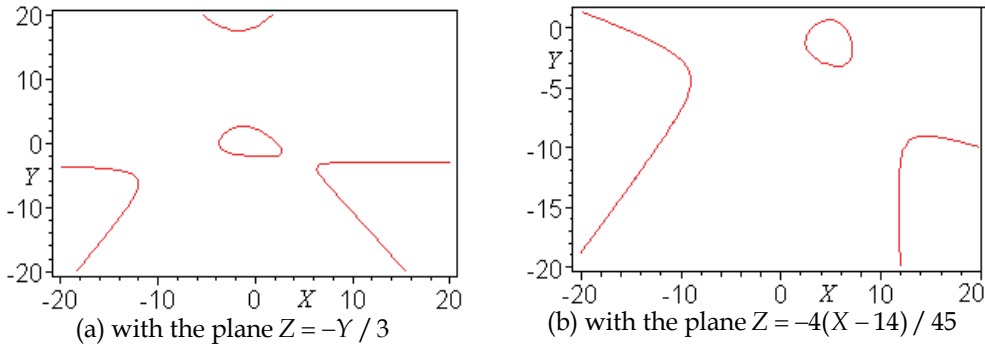


Fig. 16. Intersecting curves of closed-loop with different sections

Obviously, the intersecting curves between the singularity locus surface of the manipulator and the two aforementioned planes are actually cubic expressions, which contain a closed-loop curve, Fig.18. However, Eq. (76) is always a quadratic polynomial expression; it is worth noting the same conclusion presented above holds for any manipulator of this class of the Gough-Stewart manipulators considered in the current study. Therefore, θ -plane reflects characteristics of the singularity loci of this class of the Gough-Stewart manipulators, and that is why we call it the principal-section.

5.3.2 Property identification of the singularity loci in parallel principal-sections

The property of the singularity loci of the manipulator in parallel principal-sections can be analyzed by two invariants, D and δ , of Eq. (76)

$$\delta = \begin{vmatrix} a & b \\ b & c \end{vmatrix} = ac - b^2 = -b^2 \tag{77}$$

$$D = \begin{vmatrix} a & b & d \\ b & c & e \\ d & e & f \end{vmatrix} = -(ae^2 + fb^2 - 2bde) \tag{78}$$

For any given geometric parameters and orientation parameters, generally, $D \neq 0$ and $\delta \neq 0$ for general values of X_V , so Eq. (76) indicates a set of hyperbolas shown in Fig. 19. Geometric parameters and orientation parameters used in these examples are given as $R_a=2, R_b=1.5, \beta_0=\pi/2, (\phi, \theta, \psi)=(\pi/3, \pi/6, 0)$.

Further research shows that for any given geometric parameters, R_a, R_b and β_0 , and orientation parameters, (ϕ, θ, ψ) , of the manipulator, D is a quartic expression while δ is a quadratic expression with respect to the single variable X_V . Generally, there are four real roots when $D=0$ and $\delta \neq 0$, and, in each of the four cases, Eq. (76) degenerates into two intersecting lines. For the same reason, there is one real root of multiplicity two when $\delta=0$ and $D=0$; in this case, Eq. (83) degenerates into a parabola. In order to demonstrate the aforementioned theoretical results, a 6/6-GSP will be studied, whose geometric parameters and the orientation parameters are given as follows: $R_a=2, R_b=1.5, \beta_0=\pi/2, (\phi, \theta, \psi)=(\pi/3, \pi/6, 0)$. Please note that the following calculations are all based on these parameters.

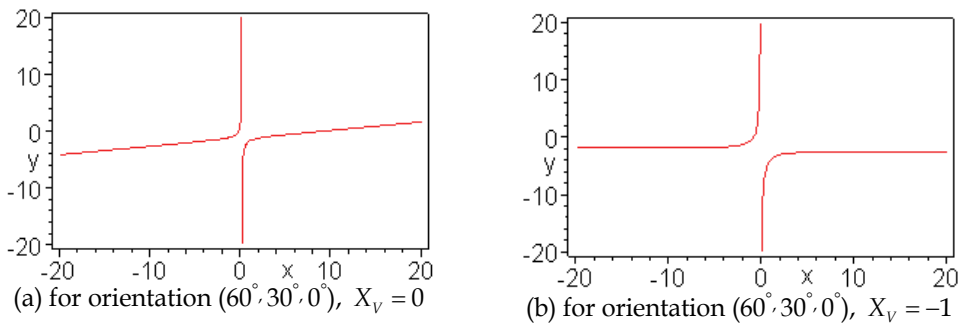


Fig. 17. Singularity loci in parallel principal-planes for general values of X_V

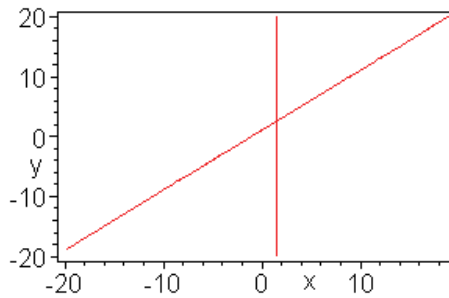


Fig. 18. The first case of two intersecting lines

5.3.2.1 First Case of Two Intersecting Lines

The intersecting line UV passes through point C_3 and then point V coincides with point C_3 . When $X_{V1} = (\sqrt{6} + \sqrt{2}) / 2$, Eq. (76) degenerates into two intersecting lines, as shown in Fig. 18.

$$(8x - 3(\sqrt{6} + \sqrt{2}))(x + k_1y + c_1) = 0 \tag{79}$$

5.3.2.2 Second Case of Two Intersecting Lines

The intersecting line UV passes through point C_4 and then point V coincides with point C_4 . When

$$X_{V_2} = (\sqrt{6} - \sqrt{2}) / 2 \quad (80)$$

Eq. (76) also degenerates into two intersecting lines

$$(8x - 3(\sqrt{6} - \sqrt{2}))(x + k_2y + c_2) = 0 \quad (81)$$

The first part of Eq. (81) is

$$8x - 3(\sqrt{6} - \sqrt{2}) = 0 \quad (82)$$

which is a line parallel to y -axis. Meanwhile, it can be proved that point B_1 is located on the base plane. Similarly, when B_1 coincides with C_4 , the singularity of this point is of the first special-linear-complex singularity. Similarly, when B_1 does not coincide with C_4 , singularities corresponding to points lying in line of Eq. (82) are of the general-linear-complex singularity.

The second part of Eq. (81) denotes another line. Singularities corresponding to points lying in this line are all of the general-linear-complex singularity.

5.3.2.3 Third Case of Two Intersecting Lines

When $X_{V_3} = 3(\sqrt{6} + \sqrt{2}) / 2$, $D=0$ and $\delta \neq 0$, Eq. (76) degenerates into two intersecting lines

$$(8x + (9\sqrt{2} + 3\sqrt{6}))(x + k_3y + c_3) = 0 \quad (83)$$

5.3.2.4 The Fourth Case of Two Intersecting Lines

When $X_{V_4} = (11\sqrt{6} - 3\sqrt{2}) / 12$, $D=0$ and $\delta \neq 0$, Eq. (76) degenerates into two intersecting lines

$$(24x - (15\sqrt{2} + 5\sqrt{6}))(x + k_4y + c_4) = 0 \quad (84)$$

The first part of Eq. (84) is

$$24x - (15\sqrt{2} + 5\sqrt{6}) = 0 \quad (85)$$

which is a line parallel to y -axis of frame \mathbf{V} - xy . In particular, there are three special points at which all the segments associated with the six extensible links of the manipulator intersect one common line, respectively

5.3.2.5 One Case of a Parabola

When $X_{V_5} = (7\sqrt{6} + 3\sqrt{2}) / 6$, $\delta=0$ and $D \neq 0$, Eq. (76) degenerates into a parabola, as shown in Fig. 23

$$(528\sqrt{6} - 912\sqrt{2})x^2 + (1423\sqrt{3} - 2472)x + (504 - 288\sqrt{3})y + 513\sqrt{6} - 909\sqrt{2} = 0 \quad (86)$$

The manipulator is always singular corresponding to points lying in the parabola. Similarly, there are three special points at which all the segments associated with the six extensible links of the manipulator intersect one common line.

Based on the analyses described above, it can be concluded that the singularity loci of this class of the 6/6-Gough-Stewart manipulators in parallel principal-sections are always

quadratic expressions that generally include infinite hyperbolas. However, for four parallel locations of the principal-section, the quadratic expression degenerates into two intersecting lines respectively, and in one location the quadratic expression is a parabola when $\theta \neq 0$.

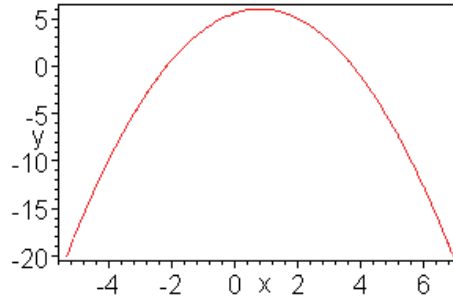


Fig. 19. One case of a parabola

5.3.3 Singularity analysis when $\theta \neq 0$

When $\theta=0$, the moving platform is parallel to the base one. Meanwhile, Eq. (94) can be reduced as follows

$$Z^3 \cos(\phi + \psi) = 0 \quad (87)$$

1. When $Z=0$, the moving and the base platforms are coincident. In this special configuration, the manipulator has three DOFs: two rotational freedoms and one translational freedom.
2. When $(\phi + \psi) = \pm 90^\circ$, for the 6/6-Gough-Stewart manipulator, it is the singularity proposed by Fichter (1986).

As we have discussed, the singularity loci of this class of the 6/6-Gough-Stewart manipulators in parallel principal-sections include infinite hyperbolas, four cases of two intersecting lines, and one case of a parabola when $\theta \neq 0$.

From analytic geometry, there are five different types of quadric surface with hyperbola sections: hyperbolic cylinder, hyperbolic paraboloid, hyperboloid of one sheet, hyperboloid of two sheets, and conic surface. However, none of these can contain infinite hyperbolas, one case of a parabola, and four cases of two intersecting lines simultaneously. Therefore, the singularity locus equation of this class of 6/6-Gough-Stewart manipulators considered in three-dimensional space is a special irresolvable polynomial expression of degree three, whose cross-sections in parallel principal-sections contain one case of a parabola, four cases of two intersecting lines and infinite hyperbolas.

6. Conclusion

6.1 A necessary and sufficient condition that the three velocities of three non-collinear points in a rigid body can reflect that the body has a possible twist motion is that the intersecting point of three normal planes of three velocities lies in the plane determined by the three points. This is also the necessary and sufficient condition of occurring singularity for a parallel mechanism, when all actuators are locked.

6.2 Based on the singularity kinematics principle and Singularity-Equivalent-Mechanism method, the structure and property of the singularity surface of 3/6-Gough-Stewart platform for all different orientations (ϕ, θ, ψ) can be finally concluded as follows

6.2.1 When $\theta=0$, the mobile platform and the base one are parallel with each other. The special configuration of the mechanism is $(90^\circ, 0, 0)$ which is proposed by Fichter (1986). The mechanism is singular whatever the position of the mobile platform is. It belongs to the general-linear-complex singularity.

6.2.2 When $\theta=0, Z=0$, the mobile platform and the base one are coincident. The mechanism is singular whatever the other two Euler angles are. The mechanism has three DOF: two rotational freedoms and one translational freedom. They belong to the first or the second special-linear-complex singularity, respectively (Huang; Chen; Li 2003).

6.2.3 When $\theta \neq 0, \phi \neq \pm 30^\circ, \pm 90^\circ, \text{ and } \pm 150^\circ$ and $\psi \neq \pm 30^\circ, \pm 90^\circ, \text{ and } \pm 150^\circ$, the singularity loci for this orientation include a plane (Hunt plane) and a hyperbolic paraboloid (Huang; Chen; Li 2003). The intersecting line UV between the mobile platform and the base one is parallel to some side of the basic triangle $A_1A_3A_5$.

6.2.4 When $\theta \neq 0, \phi \neq \pm 30^\circ, \pm 90^\circ, \text{ and } \pm 150^\circ$ and $\psi = \pm 30^\circ, \pm 90^\circ, \text{ or } \pm 150^\circ$, the singularity loci are three intersecting planes (Huang; Chen; Li 2003).

6.2.5 When $\theta \neq 0, \phi \neq \pm 30^\circ, \pm 90^\circ, \text{ and } \pm 150^\circ$ and $\psi = \pm 30^\circ, \pm 90^\circ, \text{ or } \pm 150^\circ$, the singularity loci for this orientation also include a plane and a hyperbolic paraboloid (Huang; Chen; Li 2003).

6.2.6 When $\theta=0$, neither ϕ nor ψ is equal to any one of the angles, $\pm 30^\circ, \pm 90^\circ, \pm 150^\circ$. This is the most general case for 3/6-GSP. The singularity equation for this orientation is a special irresolvable polynomial expression of degree three, and the structure of the singularity loci in infinite parallel principal sections includes a parabola, four pairs of intersecting straight lines and infinity of hyperbolas (Huang and Cao 2006).

6.2.7 There is only one instantaneous freedom forming a twist with $h^m \neq 0$, appearing when 3/6-GSP is singular for infinite general orientations. But there are seven special situations in which the instantaneous motion is a pure rotation with $h^m = 0$. However, under the case $\theta=0, Z=0$ the instantaneous motion is a pure translation with $h^m = \infty$.

6.2.8 According to the singularity-equivalent-mechanism, the singularity can occur at any point all over the θ -plane. That is to say, the singularity can occur everywhere in the workspace of the mechanism. But the singularity orientation of the platform in different points may be different.

6.2.9 The planar section parallel to the mobile platform is named principal section of the singularity surface. The singularity expressions in infinite parallel principal sections are always quadratic.

6.3 The structure and property of the singularity loci of this class of 6/6-Gough-Stewart manipulators for all different orientations can be finally concluded as follows (Huang and Cao 2005)

6.3.1 The singularity locus equation of degree three, is of a special irresolvable polynomial expression whose cross-sections in parallel principal-sections contain one parabola, four pairs of intersecting lines, and infinite hyperbolas.

6.3.2 The graphical representations of the singularity locus of this class of the 6/6-Gough-Stewart manipulators are quite complex and various for different orientations. The most complex graphic of the singularity loci looks like a trifoliate surface with two holes.

6.3.3 We find an incredible phenomenon, for this class of the 6/6-Gough-Stewart manipulators, there are also some special singularity cases where six lines associated with the six extensible links of the manipulator can intersect one common line and the remnant motion of the manipulator is a pure rotational motion. Even for the same orientation of the manipulator, there are two or more positions of the manipulator at which the six lines of the manipulator all intersect one common line simultaneously.

6.3.4 When $\theta=0$, $Z=0$, the manipulator also has three remnant DOFs: two rotational freedoms and one translational freedom.

7. Acknowledgment

The research work reported here is supported by the NSFC under Grant No. 59885006 and 50275129, 50905075 and Supported by the Self-determined Research Program of Jiangnan University under Grant No. JUSRP10908.

8. References

- Ball, R. S. (1900). *Theory of Screw*. Cambridge:Cambridge University Press, 1900.
- Cao, Y.; Huang, Z. & Ge, Q. J. (2005). Orientation-Singularity and Orientation Capability Analysis of The Stewart-Gough Manipulator, *ASME 2005 paper DETC2005-84556*, 2005.
- Chan, V. K. & Ebert-Uphoff, I. (2000). Investigation of the deficiencies of parallel manipulators in singular configurations through the Jacobian nullspace. *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2000.
- Collins, C. L. & Long, G. L. (1995). The Singularity Analysis of an In-Parallel Hand Controller for Force-Reflected Teleoperation," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, October, 1995.
- Collins, C. L. & McCarthy, J. M. (1997). The singularity loci of two triangular parallel manipulator. *IEEE ICAR 97'*, Monterey, CA, July 7:9, pp. 473-478, 1997.
- Collins, C. L. & McCarthy, J. M. (1998). The quatic singularity surfaces of planar platforms in the Clifford algebra of the projective plane, *Mechanism and Machine Theory*, 33:7, pp.931-944, 1998.
- Di Gregorio, R.(2001). Analytic Formulation of the 6-3 Fully-Parallel Manipulator's Singularity Determination," *Robotica*, 19, pp.663-667, 2001.
- Di Gregorio, R. (2002). Singularity-Locus Expression of a Class of Parallel Mechanisms, *Robotica*, 20, pp. 323-328, 2002.
- Di Gregorio, R. (2004). Properties of the SX-YS-ZS Structures and Singularity Determination in Parallel Manipulators which Generate Those Structures. *Proceedings of DETC'04: ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference* September 28 – October 2, 2004, Salt Lake City, Utah USA, 2004.
- Ebert-Uphoff, I.; Lee, J.-K. & Lipkin, H. (2000). Characteristic Tetrahedron of Wrench Singularities for Parallel Manipulators with Three Legs. *Proc. of A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of "A Treatise on the Theory of Screws"* University of Cambridge, Trinity College, 2000.
- Fichter, E. F. (1986). A stewart platform-based manipulator: General theory and practical construction, *Int. J. Rob. Res.* 5, pp.157-182, 1986.
- Gough, V. E. (1956-57). Contribution to discussion to papers on research in automobile stability and control and in tyre performance. *Proc. Auto. Div. Instn mech. Engrs*, p.392. by Cornell staff, pp. 392-397, 1956-57.
- Gosselin, C. & Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains, *IEEE Trans. Rob. Autom.* 6, pp. 281-190, 1990
- Hao, F. & McCarthy, J. M. (1998). Conditions for Line-Based Singularities in Spatial Platform Manipulator, *Journal of Robotic Systems*, 15(1), pp. 43-55, 1998.
- Huang Z. and Qu Y. Y. (1987). The analysis of the special configuration of the spatial parallel manipulators, *The 5th National Mechanism Conference*, Lu Shan, China, pp. 1-7, 1987.
- Huang, Z.; Kong, L.F. & Fang, Y.F. (1997). *Theory and Control of Parallel Robotic Mechanisms Manipulator*. Beijing, China. Publisher of Mechanical Industry, 1997.
- Huang, Z.; Zhao, Y.; Wang, J. & Yu, J. J. (1999). Kinematic principle and geometrical Condition of general-linear-complex special configuration of parallel manipulators, *Mechanism Machine Theory*, 34, pp. 1171-1186, 1999.
- Huang, Z.; Chen, L.H. & Li, Y. W. (2003). The Singularity Principle and Property of Stewart manipulator, *Journal of Robotic Systems*, 20:4, pp.163-176, 2003.
- Huang, Z. & Cao, Y. (2005). Property Identification of the Singularity Loci of a Class of Gough- Stewart Manipulators, *The International Journal of Robotics Research*, 24:8, pp. 675-685, 2005

- Huang, Z.; Cao, Y.; Li Y. W., & Chen L.H. (2006). Structure and Property of the Singularity Loci of the 3/6-Stewart-Gough Platform for General Orientations, *Robotica*, 2006, 24, pp. 75-84, 2006.
- Hunt, K.H. (1978). *Kinematic Geometry of Mechanisms*. Oxford, UK: Oxford University Press, 1978.
- Hunt, K.H. (1983). Structural kinematics of in-parallel-actuated robot-arms, *ASME J. Mech. Transmissions Autom. Design*, 105, pp. 705-712, 1983.
- Karger, A. & Husty, M., (1998). Architecture singular parallel manipulators, *Proceedings of ARK'98 (J. Lenarcic and M. Husty eds)*, pp. 445-454, 1998.
- Karger, A. (2001). Singularities and self-motions of equiform platforms. *Mechanism and Machine Theory*, 36:7, pp. 801-805, 2001.
- Kong, X W. (1998). Generation of 6-SPS Parallel Manipulators," *Proc. of the 1998 ASME Design Engineering Technical Conference*, Atlanta, USA, 98DETC/MECH-5952, 1998.
- Kong, X. & Gosselin, C. M., (2001). Uncertainty Singularity Analysis of Parallel Manipulators Based on the Instability Analysis of Structures, *The International Journal of Robotics Research*, Vol 20 :11, pp. 847-856, 2001.
- Kong, X. & Gosselin, C.M. (2002). Generation of Architecturally Singular 6-SPS Parallel Manipulators with Linearly Related Planar Platforms. *Electronic Journal of Computational Kinematics (EJCK)*, May 2002. 1(1), Paper No. 7, 2002
- Ma, O. & Angeles, J., (1991). Architecture Singularity of Platform Manipulators, *IEEE Int. Conf. on Robotics and Automation*. Sacramento, USA , pp. 1542-1547, 1991.
- Mayer St-Onge, B. & Gosselin, C. (2000). Singularity analysis and representation of the general Gough-Stewart platform, *Int. J. Rob. Res.* Vol.19, No. 3, pp. 271-288, 2000.
- McAree, P.R. & Daniel, R. W. (1999). An Explanation of never-special Assembly Changing Motions for 3-3 Parallel Manipulators, *Int. J. of Robotics Research*, 18:6, pp. 556-574, 1999.
- Merlet, J. P. (1988). Parallel manipulator part 2: Singular configurations and grassmann geometry. *Technical report, INRIA*, Sophia Antipolis, France, 1988.
- Merlet J.P. (1989). Singular configurations of parallel manipulators and Grassmann geometry, *Int. J. Rob. Res.* 8, pp. 45-56, 1989.
- Parenti-Castelli, V. & Innocenti, C., (1990). Direct displacement analysis for some classes of spatial parallel mechanisms, *Proc. of the 8th CISM-IFTOMM Symp. on Theory and Practice of Robots and Manipulators*, Cracow, Poland, pp. 126-133, 1990.
- Pernkopf, F. & Husty, M. L. (2002). Singularity Analysis of Spatial Stewart-Gough Platforms with Planar Base and Platform. *Proceedings of DETC.02 ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Montreal, Canada, September 29 - October 2, 2002, DETC2002/MECH-34267.
- Sefrioui, J. & Gosselin, C. (1994). Étude et représentation des lieux de singularité des manipulateurs parallèles sphériques à trois degrés de liberté avec actionneurs prismatiques, *Mechanism Machine Theory*, 29, pp. 559-579, 1994.
- Sefrioui, J. & Gosselin, C. (1995). On the quadratic nature of the singularity curves of planar three-degree-of-freedom parallel manipulators, *Mechanism Machine Theory*, 30, pp. 533-551, 1995.
- Sugimoto, K. & Duffy, J. (1981). Special Configuration of Industrial Robots, *Proc. of the 11th ISIR*, pp. 309-316, 1981.
- Stewart, D. (1965). A platform with six degrees of freedom, *Proc. Inst. Mech. Eng*, 180, pp. 371-378, 1965.
- Wang, G. Z. (1998). Singularity analysis of a class of the 6-6 Stewart platforms. *Proc. of DETC'98 1998 ASME Design Engineering Technical Conference*, Atlanta, Georgia, USA, 1998.
- Wang, J. & Gosselin, C.M. (1996). Kinematic analysis and singularity loci of spatial four-degree-of-freedom parallel manipulators. *Proc. of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, California, USA, 1996.
- Wang, J. & Gosselin, C. (1997). Kinematic analysis and singularity representation of spatial five-degree-of-freedom parallel mechanisms, *J Robotic Syst.* 14, pp. 851-869, 1997.
- Zlatanov, D., Bonev, I.A. & Gosselin, C.M., (2002), Constraint singularities of parallel mechanisms, *Proceedings of the 2002 IEEE International conference on Robotics and Automation*, Washington, DC, USA, pp. 496-502, 2002.
- Zlatanov, D.; Fenton, R.G. & Benhabib, B. (1994). Singularity analysis of mechanisms and robots via a motion-space model of the instantaneous kinematics, *Proc. of 1994 IEEE Int. Conf. On Rob. And Auto 2* pp. 980-991, 1994.
- Zlatanov, D, Fenton, RG & Benhabib, B. (1995). A Unifying Framework for Classification and Interpretation of Mechanism Singularities. *ASME J Mechanical Design*, 117, pp. 566-572, 1995.

Edited by S. Ehsan Shafiei

Amongst the robotic systems, robot manipulators have proven themselves to be of increasing importance and are widely adopted to substitute for human in repetitive and/or hazardous tasks. Modern manipulators are designed complicatedly and need to do more precise, crucial and critical tasks. So, the simple traditional control methods cannot be efficient, and advanced control strategies with considering special constraints are needed to establish. In spite of the fact that groundbreaking researches have been carried out in this realm until now, there are still many novel aspects which have to be explored.

Photo by IgorLitvinzev / iStock

IntechOpen

