



IntechOpen

Robot Manipulators Trends and Development

Edited by Agustín Jiménez and Basil M Al Hadithi



**ROBOT MANIPULATORS,
TRENDS AND DEVELOPMENT**

Edited by
**PROF. DR. AGUSTÍN JIMÉNEZ
AND DR. BASIL M. AL HADITHI**

Robot Manipulators Trends and Development

<http://dx.doi.org/10.5772/218>

Edited by Agustin Jimenez and Basil M Al Hadithi

© 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

Robot Manipulators Trends and Development

Edited by Agustin Jimenez and Basil M Al Hadithi

p. cm.

ISBN 978-953-307-073-5

eBook (PDF) ISBN 978-953-51-5901-8

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

4,200+

Open access books available

116,000+

International authors and editors

125M+

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



Meet the editors



Prof. Agustín Jiménez is Catedrático de Universidad (Full Professor) in the area of Systems Engineering at the Departamento de Automática, Ingeniería Electrónica e Informática Industrial of the Universidad Politécnica de Madrid. He was born in Toledo, Spain, in 1952, became Industrial Engineer at the Escuela Técnica Superior de Ingenieros Industriales of the U.P.M, and got the PhD degree at the U.P.M. in 1978. His main activities have been in the development and application of advanced control technologies in the area of continuous process control. In particular the main effort is in the integration of artificial intelligence techniques with conventional control applications. He has published several papers and books, being the latest State Space Control (in Spanish). His experience in R&D projects is very wide, having participated in several industrial projects, in ESPRIT projects (623, 2202 PLATO, 6447 HINT and 22130 DIXIT) IST projects (10258 DOTS and 29456 WEBFAIR), RTD-Inco 962074 PROQUS and in national founded research projects, most of them in the field of design and implementation of Intelligent Process Control Systems



Basil M. Al Hadithi , Associate professor and researcher in the electronic and control engineering field. Professor of Analog & Digital Electronic circuits, Design of electronic circuits, control systems, Advanced control techniques, Optimization and simulation

Preface

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

This book is the result of the effort by a number of contributors involved in robotics fields. The aim is to provide a wide and extensive coverage of all the areas related to the most up to date advances in robotics.

The authors have approached a good balance between the necessary mathematical expressions and the practical aspects of robotics. The organization of the book shows a good understanding of the issues of high interest nowadays in robot modelling, simulation and control. The book demonstrates a gradual evolution from robot modelling, simulation and optimization to reach various robot control methods. These two trends are finally implemented in real applications to examine their effectiveness and validity.

Editors:

Prof. Dr. Agustín Jiménez and Dr. Basil M. Al Hadithi

Contents

Preface	IX
1. Optimal Usage of Robot Manipulators Behnam Kamrani, Viktor Berbyuk, Daniel Wappling, Xiaolong Feng and Hans Andersson	001
2. ROBOTIC MODELLING AND SIMULATION: THEORY AND APPLICATION Muhammad Ikhwan Jambak, Habibollah Haron, Helmee Ibrahim and Norhazlan Abd Hamid	027
3. Robot Simulation for Control Design Leon lajpah	043
4. Modeling of a One Flexible Link Manipulator Mohamad Saad	073
5. Motion Control Sangchul Won and Jinwook Seok	101
6. Global Stiffness Optimization of Parallel Robots Using Kinetostatic Performance Indices Dan Zhang	125
7. Measurement Analysis and Diagnosis for Robot Manipulators using Advanced Nonlinear Control Techniques Amr Pertew, Ph.D, P.Eng., Horacio Marquez, Ph. D, P. Eng and Qing Zhao, Ph. D, P. Eng	139
8. Cartesian Control for Robot Manipulators Pablo Sanchez-Sanchez and Fernando Reyes-Cortes	165
9. Biomimetic Impedance Control of an EMG-Based Robotic Hand Toshio Tsuji, Keisuke Shima, Nan Bu and Osamu Fukuda	213
10. Adaptive Robust Controller Designs Applied to Free-Floating Space Manipulators in Task Space Tatiana Pazelli, Marco Terra and Adriano Siqueira	231
11. Neural and Adaptive Control Strategies for a Rigid Link Manipulator Dorin Popescu, Dan Seliteanu, Cosmin Ionete, Monica Roman and Livia Popescu	249
12. Control of Flexible Manipulators. Theory and Practice Pereira, E.; Becedas, J.; Payo, I.; Ramos, F. and Feliu, V.	267

13. Fuzzy logic positioning system of electro-pneumatic servo-drive Jakub E. Takosoglu, Ryszard F. Dindorf and Pawel A. Laski	297
14. Teleoperation System of Industrial Articulated Robot Arms by Using Forcefree Control Satoru Goto	321
15. Trajectory Generation for Mobile Manipulators Foudil Abdessemed and Salima Djebrani	335
16. Trajectory Control of Robot Manipulators Using a Neural Network Controller Zhao-Hui Jiang	361
17. Performance Evaluation of Autonomous Contour Following Algorithms for Industrial Robot Anton Satria Prabuwno, Samsi Md. Said, M.A. Burhanuddin and Riza Sulaiman	377
18. Advanced Dynamic Path Control of the Three Links SCARA using Adaptive Neuro Fuzzy Inference System Prabu D, Surendra Kumar and Rajendra Prasad	399
19. Topological Methods for Singularity-Free Path-Planning Davide Paganelli	413
20. Vision-based 2D and 3D Control of Robot Manipulators Luis Hernández, Hichem Sahli and René González	441
21. Using Object's Contour and Form to Embed Recognition Capability into Industrial Robots I. Lopez-Juarez, M. Peña-Cabrera and A.V. Reyes-Acosta	463
22. Autonomous 3D Shape Modeling and Grasp Planning for Handling Unknown Objects Yamazaki Kimitoshi, Masahiro Tomono and Takashi Tsubouchi	479
23. Open Software Structure for Controlling Industrial Robot Manipulators Flavio Roberti, Carlos Soria, Emanuel Slawiński, Vicente Mut and Ricardo Carelli	497
24. Miniature Modular Manufacturing Systems and Efficiency Analysis of the Systems Nozomu Mishima, Kondoh Shinsuke, Kiwamu Ashida and Shizuka Nakano	521
25. Implementation of an Intelligent Robotized GMAW Welding Cell, Part 1: Design and Simulation I. Davila-Rios, I. Lopez-Juarez, Luis Martinez-Martinez and L. M. Torres-Treviño	543
26. Implementation of an Intelligent Robotized GMAW Welding Cell, Part 2: Intuitive visual programming tool for trajectory learning I. Lopez-Juarez, R. Rios-Cabrera and I. Davila-Rios	563

27. Dynamic Behavior of a Pneumatic Manipulator with Two Degrees of Freedom 575
Juan Manuel Ramos-Arreguin, Efren Gorrostieta-Hurtado, Jesus Carlos Pedraza-Ortega,
Rene de Jesus Romero-Troncoso, Marco-Antonio Aceves and Sandra Canchola
28. Dexterous Robotic Manipulation of Deformable Objects with 587
Multi-Sensory Feedback - a Review
Fouad F. Khalil and Pierre Payeur
29. Task analysis and kinematic design of a novel robotic chair for 621
the management of top-shelf vertigo
Giovanni Berselli, Gianluca Palli, Riccardo Falconi, Gabriele Vassura
and Claudio Melchiorri
30. A Wire-Driven Parallel Suspension System with 8 Wires (WDPSS-8) 647
for Low-Speed Wind Tunnels
Yaqing ZHENG, Qi LIN1 and Xiongwei LIU

Optimal Usage of Robot Manipulators

Behnam Kamrani¹, Viktor Berbyuk², Daniel Wäppling³,
Xiaolong Feng⁴ and Hans Andersson⁴

¹*MSC.Software Sweden AB, SE-42 677, Gothenburg*

²*Chalmers University of Technology, SE-412 96, Gothenburg*

³*ABB Robotics, SE-78 168, Västerås*

⁴*ABB Corporate Research, SE-72178, Västerås
Sweden*

1. Introduction

Robot-based automation has gained increasing deployment in industry. Typical application examples of industrial robots are material handling, machine tending, arc welding, spot welding, cutting, painting, and gluing. A robot task normally consists of a sequence of the robot tool center point (TCP) movements. The time duration during which the sequence of the TCP movements is completed is referred to as cycle time. Minimizing cycle time implies increasing the productivity, improving machine utilization, and thus making automation affordable in applications for which throughput and cost effectiveness is of major concern. Considering the high number of task runs within a specific time span, for instance one year, the importance of reducing cycle time in a small amount such as a few percent will be more understandable.

Robot manipulators can be expected to achieve a variety of optimum objectives. While the cycle time optimization is among the areas which have probably received the most attention so far, the other application aspects such as energy efficiency, lifetime of the manipulator, and even the environment aspect have also gained increasing focus. Also, in recent era virtual product development technology has been inevitably and enormously deployed toward achieving optimal solutions. For example, off-line programming of robotic work-cells has become a valuable means for work-cell designers to investigate the manipulator's workspace to achieve optimality in cycle time, energy consumption and manipulator lifetime.

This chapter is devoted to introduce new approaches for optimal usage of robots. Section 2 is dedicated to the approaches resulted from translational and rotational repositioning of a robot path in its workspace based on response surface method to achieve optimal cycle time. Section 3 covers another proposed approach that uses a multi-objective optimization methodology, in which the position of task and the settings of drive-train components of a robot manipulator are optimized simultaneously to understand the trade-off among cycle time, lifetime of critical drive-train components, and energy efficiency. In both section 2 and 3, results of different case studies comprising several industrial robots performing different

tasks are presented to evaluate the developed methodologies and algorithms. The chapter is concluded with evaluation of the current results and an outlook on future research topics on optimal usage of robot manipulators.

2. Time-Optimal Robot Placement Using Response Surface Method

This section is concerned with a new approach for optimal placement of a prescribed task in the workspace of a robotic manipulator. The approach is resulted by applying response surface method on concept of path translation and path rotation. The methodology is verified by optimizing the position of several kinds of industrial robots and paths in four showcases to attain minimum cycle time.

2.1 Research background

It is of general interest to perform the path motion as fast as possible. Minimizing motion time can significantly shorten cycle time, increase the productivity, improve machine utilization, and thus make automation affordable in applications for which throughput and cost effectiveness is of major concern.

In industrial application, a robotic manipulator performs a repetitive sequence of movements. A robot task is usually defined by a robot program, that is, a robot path consisting of a set of robot positions (either joint positions or tool center point positions) and corresponding set of motion definitions between each two adjacent robot positions. *Path translation* and *path rotation* terms are repeatedly used in this section to describe the methodology. Path translation implies certain translation of the path in x, y, z directions of an arbitrary coordinate system relative to the robot while all path points are fixed with respect to each other. Path rotation implies certain rotation of the path with θ, φ, ψ angles of an arbitrary coordinate system relative to the robot while all path points are fixed with respect to each other. Note that since path translation and path rotation are relative concepts, they may be achieved either by relocating the path or the robot.

In the past years, much research has been devoted to the optimization problem of designing robotic work cells. Several approaches have been used in order to define the optimal relative robot and task position. A *manipulability measure* was proposed (Yoshikawa, 1985) and a modification to Yoshikawa's manipulability measure was proposed (Tsai, 1986) which also accounted for proximity to joint limits. (Nelson & Donath, 1990) developed a gradient function of manipulability in Cartesian space based on explicit determination of manipulability function and the gradient of the manipulability function in joint space. Then they used a modified method of the steepest descent optimization procedure (Luenberger, 1969) as the basis for an algorithm that automatically locates an assembly task away from singularities within manipulator's workspace.

In aforementioned works, mainly the effects of robot kinematics have been considered. Once a robot became employed in more complex tasks requiring improved performance, e.g., higher speed and accuracy of trajectory tracking, the need for taking into account robot dynamics becomes more essential (Tsai, 1999).

A study of time-optimal positioning of a prescribed task in the workspace of a 2R planar manipulator has been investigated (Fardanesh & Rastegar, 1988). (Barral et al., 1999) applied the simulated annealing optimization method to two different problems: robot placement and point-ordering optimization, in the context of welding tasks with only one restrictive

working hypothesis for the type of the robot. Furthermore, a state of the art of different methodologies has been presented by them.

In the current study, the dynamic effect of the robot is considered by utilizing a computer model which simulates the behavior and response of the robot, that is, the dynamic models of the robots embedded in ABB's IRC5 controller. The IRC5 robot controller uses powerful, configurable software and has a unique dynamic model-based control system which provides self-optimizing motion (Vukobratovic, 2002).

To the best knowledge of the authors, there are no studies that directly use the response surface method to solve optimization problem of optimal robot placement considering a general robot and task. In this section, a new approach for optimal placement of a prescribed task in the workspace of a robot is presented. The approach is resulted by path translation and path rotation in conjunction with response surface method.

2.2 Problem statement and implementation environment

The problem investigated is to determine the relative robot and task position with the objective of time optimality. Since in this study a relative position is to be pursued, either the robot, the path, or both the robot and path may be relocated to achieve the goal. In such a problem, the robot is given and specified without any limitation imposed on the robot type, meaning that any kind of robot can be considered. The path or task, the same as the robot, is given and specified; however, the path is also general and any kind of path can be considered. The optimization objective is to define the optimal relative position between a robotic manipulator and a path. The optimal location of the task is a location which yields a minimum cycle time for the task to be performed by the robot.

To simulate the dynamic behavior of the robot, RobotStudio is employed, that is a software product from ABB that enables offline programming and simulation of robot systems using a standard Windows PC. The entire robot, robot tool, targets, path, and coordinate systems can be defined and specified in RobotStudio. The simulation of a robot system in RobotStudio employs the ABB Virtual Controller, the real robot program, and the configuration file that are identical to those used on the factory floor. Therefore the simulation predicts the true performance of the robot.

In conjunction with RobotStudio, Matlab and Visual Basic Application (VBA) are utilized to develop a tool for proving the designated methodology. These programming environments interact and exchange data with each other simultaneously. While the main dataflow runs in VBA, Matlab stands for numerical computation, optimization calculation, and post processing. RobotStudio is employed for determining the *path admissibility* boundaries and calculating the cycle times. Figure 1 illustrates the schematic of dataflow in the three computational environments.

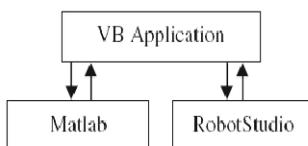


Fig. 1. Dataflow in the three computational tools

2.3 Methodology of time-optimal robot placement

Basically, the path position relative to the robot can be modified by translating and/or rotating the path relative to the robot. Based on this idea, translation and rotation approaches are examined to determine the optimal path position. The algorithms of both approaches are considerably analogous. The approaches are based on the response surface method and consist of following steps. First is to pursue the *admissibility boundary*, that is, the boundary of the area in which a specific task can be performed with the same robot configuration as defined in the path instruction. This boundary is obviously a subset of the general robot operability space that is specified by the robot manufacturer. The computational time of this step is very short and may take only few seconds. Then experiments are performed on different locations of admissibility boundary to calculate the cycle time as a function of path location. Next, optimum path location is determined by using constrained optimization technique implemented in Matlab. Finally, the sensitivity analysis is carried out to increase the accuracy of optimum location.

Response surface method (Box et al., 1978; Khuri & Cornell, 1987; Myers & Montgomery, 1995) is, in fact, a collection of mathematical and statistical techniques that are useful for the modeling and analysis of problems in which a response of interest is influenced by several decision variables and the objective is to optimize the response. Conventional optimization methods are often cumbersome since they demand rather complicated calculations, elaborate skills, and notable simulation time. In contrast, the response surface method requires a limited number of simulations, has no convergence issue, and is easy to use.

In the current robotic problem, the decision variables consist of x , y , and z of the reference coordinates of a prescribed path relative to a given robot base and the response of interest to be minimized is the task cycle time. A so-called full factorial design is considered by 27 experiment points on the path admissibility boundaries in three-dimensional space with original path location in center. Figure 2 graphically depicts the original path location in the center of the cube and the possible directions for finding the admissibility boundary.

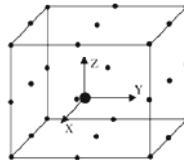


Fig. 2. Direction of experiments relative to the original location of path

Three-dimensional bisection algorithm is employed to determine the path admissibility region. The algorithm is based on the same principle as the bisection algorithm for locating the root of a three-variable polynomial. Bisection algorithm for finding the admissibility boundary states that each translation should be equal to half of the last translation and translation direction is the same as the last translation if all targets in the path are admissible; otherwise, it is reverse. Herein, targets on the path are considered admissible if the robot manipulator can reach them with the predefined configurations. Note that in this step the robot motion between targets is not checked.

Since the target admissibility check is only limited to the targets and the motion between the targets are not simulated, it has a low computational cost. Additionally, according to practical experiments, if all targets are admissible, there is a high probability that the whole

path would also be admissible. However, checking the target admissibility does not guarantee that the whole path is admissible as the joint limits must allow the manipulator to track the path between the targets as well. In fact, for investigating the path admissibility, it is necessary to simulate the whole task in RobotStudio to ascertain that the robot can manage the whole task, i.e., targets and the path between targets.

To clarify the method, an example is presented here. Let's assume an initial translation by 1.0 m in positive direction of x axis of reference coordinate system is considered. If all targets after translation are admissible, then the next translation would be 0.5 m and in the same (+ x) direction; otherwise in opposite (- x) direction. In any case, the admissibility of targets in the new location is checked and depending on the result, the direction for the next translation is decided. The amount of new translation would be then 0.25 m. This process continues until a location in which all targets are admissible is found such that the last translation is smaller than a certain value, that is, the considered tolerance for finding the boundary, e.g., 1 mm.

After finding the target admissibility boundary in one direction within the decided tolerance, a whole task simulation is run to measure the cycle time. Besides measuring the cycle time, it is also controlled if the robot can perform the whole path, i.e., investigating the *path admissibility* in addition to targets admissibility. If the path is not admissible in that location, a new admissible location within a relaxed tolerance can be sought and examined. The same procedure is repeated in different directions, e.g. 27 directions in full-factorial method, and by that, a matrix of boundary coordinates and vector of the corresponding cycle times are casted.

A quadratic approximation function provides proper result in most of response surface method problems (Myers & Montgomery, 1995), that is:

$$f(x,y,z) = b_0 + b_1x + b_2y + b_3z + \dots \quad (\text{linear terms}) \\ b_4xy + b_5yz + b_6xz + \dots \quad (\text{interaction terms}) \\ b_7x^2 + b_8y^2 + b_9z^2 \quad (\text{quadratic terms}) \quad (1)$$

By applying the following mapping:

$$x = x_1 ; \quad y = x_2 ; \quad z = x_3 \\ xy = x_4 ; \quad xz = x_5 ; \quad yz = x_6 \\ x^2 = x_7 ; \quad y^2 = x_8 ; \quad z^2 = x_9 \quad (2)$$

Eq. 1 can be expressed in linear form and by matrix notation as:

$$Y = XB + e \quad (3)$$

where Y is the vector of cycle times, X is the design matrix of boundaries, B is the vector of unknown model coefficients of $\{b_0, b_1, b_2, \dots, b_9\}$, and e is the vector of errors. Finally, B can be estimated using the least squares method, minimizing of $L=e^Te$, as:

$$B = (X^TX)^{-1} X^TY \quad (4)$$

In the next step of the methodology, when the expression of cycle time as a function of a reference coordinate (x, y, z) is given, the minimum of the cycle times subject to the determined boundaries is to be found. The *fmincon* function in Matlab optimization toolbox is used to obtain the minimum of a constrained nonlinear function. Note that, since the cycle time function is a prediction of the cycle time based on the limited experiments data, the obtained value (for the minimum of cycle time) does not necessarily provide the global minimum cycle time of the task. Moreover, it is not certain yet that the task in optimum location is kinematically admissible. Due to these reasons, the minimum of the cycle time function can merely be considered as an '*optimum candidate*.'

Hence, the optimum candidate must be evaluated by performing a confirmatory task simulation in order to, first investigate whether the location is admissible and second, calculate the actual cycle time. If the location is not admissible, the closest location in the direction of the translation vector is pursued such that all targets are admissible. This new location is considered as a new optimum candidate and replaced the old one. This procedure may be called sequential backward translation.

Due to the probability of inadmissible location and as a work around, the algorithm, by default, seeks and introduces several optimum candidates by setting different search areas in *fmincon* function. All candidate locations are examined and cycle times are measured. If any location is inadmissible, that location is removed from the list of optimum candidate. After examining all the candidates, the minimum value is selected as the final optimum. If none of the optimum candidates is admissible, the shortest cycle time of experiments is selected as optimum. In fact, and in any case, it is always reasonable to inspect if the optimum cycle time is shorter than all the experiment cycle times, and if not, the shortest cycle time is chosen as the local optimum.

As the last step of the methodology the sensitivity analysis of the obtained optimal solution with respect to small variations in x, y, z coordinates can be interesting to study. This analysis can particularly be useful when other constraints, for example space inadequacy, delimit the design of robotic cell. Another important benefit of this analysis is that it usually increases the accuracy of optimum location, meaning that it can lead to finding a precise local optimum location.

The sensitivity analysis procedure is generally analogous to the main analysis. However, herein, the experiments are conducted in a small region around the optimum location. Also, note that since it is likely that the optimum point, found in the previous step, is located on (or close to) the boundary, defining a cube around a point located on the boundary places some cube sides outside the boundary. For instance, when the shortest cycle time of the experiments is selected as the local optimum, the optimum location is already on the admissibility boundary. In such cases, as a work around, the nearest admissible location in the corresponding direction is considered instead.

Note that the sensitivity analysis may be repeated several times in order to further improve the results. Figure 3 provides an overview of the optimization algorithm.

As was mentioned earlier, the path position relative to the robot can be modified by translating as well as rotating the path. In path translation, the optimal position can be achieved without any change in path orientation. However, in path rotation, the optimal path orientation is to be sought. In other words, in path rotation approach the aim is to obtain the optimum cycle time by rotating the path around the $x, y,$ and z axes of a local frame. The local frame is originally defined parallel to the axes of the global reference frame

on an arbitrary point. The origin of the local reference frame is called the rotation center. Three sequential rotation angles are used to rotate the path around the selected rotation center. To calculate new coordinates and orientations of an arbitrary target after a path rotation, a target of T on the path is considered in global reference frame of X - Y - Z which is demonstrated in Fig. 4. The target T is rotated in local frame by a rotation vector of (θ, ϕ, ψ) which yields the target T' .

If the targets in the path are not admissible after rotating by a certain rotation vector, the boundary of a possible rotation in the corresponding direction is to be obtained based on the bisection algorithm. The matrices of experiments and cycle time response are built in the same way as described in the path translation section and the cycle time expression as a function of rotation angles of (θ, ϕ, ψ) is calculated. The optimum rotation angles are obtained using Matlab *fmincon* function. Finally, sensitivity analyses may be performed. A procedure akin to path translation is used to investigate the effect of path rotation on the cycle time.

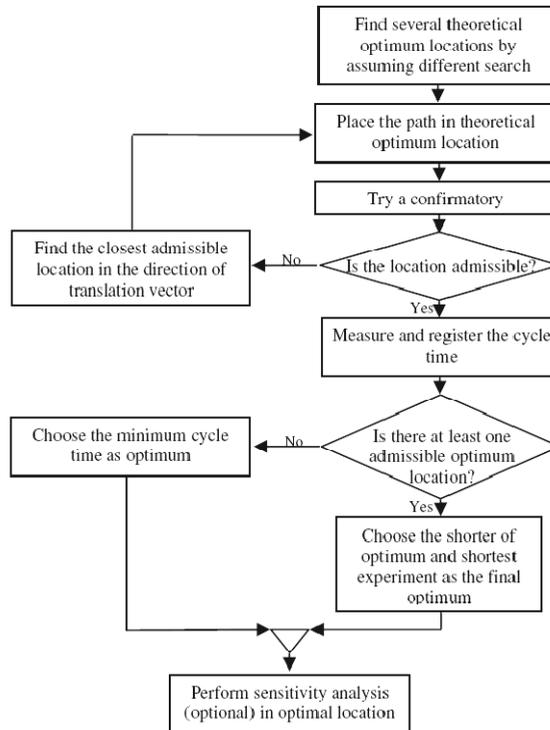


Fig. 3. Flowchart diagram of the optimization algorithm

Although the algorithm of path rotation is akin to path translation, two noticeable differences exist. First, in the rotation approach, the order of rotations must be observed. It can be shown that interchanging orders of rotation drastically influences the

resulting orientation. Thus, the order of rotation angles must be adhered to strictly (Haug, 1992). Consequently, in the path rotation approach, the optimal rotation determined by sensitivity analysis cannot be added to the optimal rotation obtained by the main analysis, whereas in the translation approach, they can be summed up to achieve the resultant translation vector. Another difference is that, in the rotation approach, the results logically depend on the selection of the rotation center location, while there is no such dependency in the path translation approach. More details concerning path rotation approach can be found in (Kamrani et al., 2009).

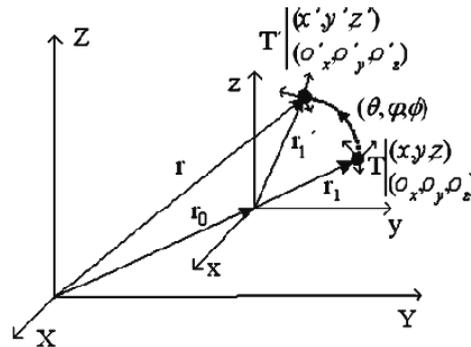


Fig. 4. Rotation of an arbitrary target T in the global reference frame

2.4 Results on time-optimal robot placement

To evaluate the methodology, four case studies comprised of several industrial robots performing different tasks are proved. The goal is to optimize the cycle time by changing the path position. A coordinate system with its origin located at the base of the robot, x-axis pointing radially out from the base, z-axis pointing vertically upwards, is used for all the cases below.

2.4.1 Path Translation

In this section, obtained by path translation approach are presented.

2.4.1.1 Case 1

The first test is carried out using the ABB robot IRB6600-225-175 performing a spot welding task composed of 54 targets with fixed positions and orientations regularly distributed around a rectangular placed on a plane parallel to the x - y plane (parallel to horizon). A view of the robot and the path in its original location is depicted in the Fig. 5. The optimal location of the task in a boundary of $(\pm 0.5 \text{ m}, \pm 0.8 \text{ m}, \pm 0.5 \text{ m})$ is calculated using the path translation approach to be as $(\Delta x, \Delta y, \Delta z) = (0 \text{ m}, 0.8 \text{ m}, 0 \text{ m})$. The cycle time of this path is reduced from originally 37.7 seconds to 35.7 seconds which implies a gain of 5.3 percent cycle time reduction. Fig. 6 demonstrates the robot and path in the optimal location determined by translation approach.

2.4.1.2 Case 2

The second case is conducted with the same ABB IRB6600-225-175 robot. The path is composed of 18 targets and has a closed loop shape. The path is shown in the Fig. 7 and as can be seen, the targets are not in one plane. The optimal location of the task in a boundary of $(\pm 1.0\text{ m}, \pm 1.0\text{ m}, \pm 1.0\text{ m})$ is calculated using the path translation approach to be as $(\Delta x, \Delta y, \Delta z) = (-0.104\text{ m}, -0.993\text{ m}, 0.458\text{ m})$. The cycle time of this path is reduced from originally 6.1 seconds to 5.6 seconds which indicates 8.3 percent cycle time reduction.

2.4.1.3 Case 3

In the third case study, an ABB robot of type IRB4400L10 is considered performing a typical machine tending motion cycle among three targets which are located in a plane parallel to the horizon. The robot and the path are depicted in the Fig. 8. The path instruction states to start from the first target and reach the third target and then return to the starting target. A restriction for this case is that the task cannot be relocated in the y -direction relative to the robot. The optimal location of the task in a boundary of $(\pm 1.0\text{ m}, 0\text{ m}, \pm 1.0\text{ m})$ is calculated using the path translation approach to be as $(\Delta x, \Delta y, \Delta z) = (0.797\text{ m}, 0\text{ m}, -0.797\text{ m})$. The cycle time of this path is reduced from originally 2.8 seconds to 2.6 seconds which evidences 7.8 percent cycle time reduction.

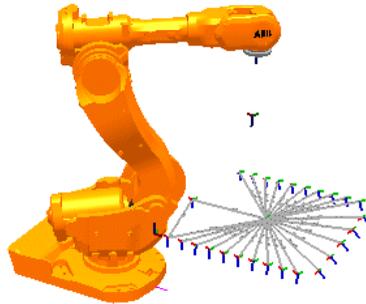


Fig. 5. IRB6600 ABB robot with a spot welding path of case 1 in its original location

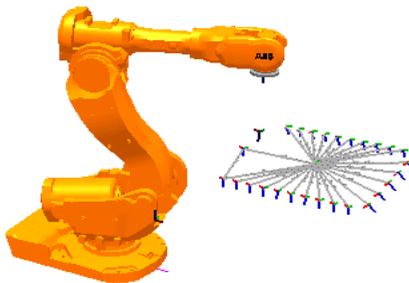


Fig. 6. IRB6600 ABB robot with a spot welding path of case 1 in optimal location found by translation approach

2.4.1.4 Case 4

The fourth case is carried out using an ABB robot of IRB640 type. In contrast to the previous robots which have 6 joints, IRB640 has merely 4 joints. The path is shown in the Fig. 9 and comprises four points which are located in a plane parallel to the horizon. The motion instruction requests the robot to start from first point and reach to the fourth point and then return to the first point again. The optimal location of the task in a boundary of $(\pm 1.0\text{ m}, \pm 1.0\text{ m})$ is calculated using the path translation approach to be as $(\Delta x, \Delta y, \Delta z) = (0.2\text{ m}, 0.2\text{ m}, -0.8\text{ m})$. The cycle time of this path is reduced from originally 3.7 seconds to 3.5 seconds which gives 5.2 percent cycle time reduction.

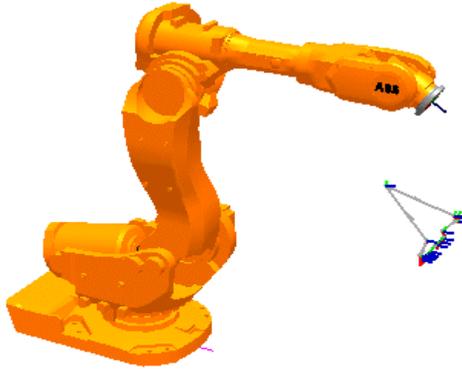


Fig. 7. IRB6600 ABB robot with the path of case 2 in its original location

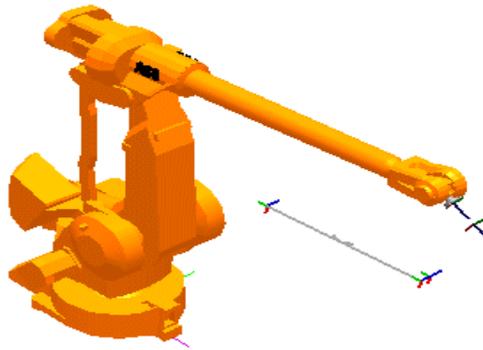


Fig. 8. IRB4400L10 ABB robot with the path of case 3 in its original location

2.4.2 Path Rotation

In this section, results of path rotation approach are presented for four case studies. Herein the same robots and tasks investigated in path translation approach are studied so that comparison between the two approaches will be possible.

2.4.2.1 Case 1

The first case is carried out using the same robot and path presented in section 2.4.1.1. The central target point was selected as the rotation center. The optimal location of the task in a boundary of $(\pm 45^\circ, \pm 45^\circ, \pm 30^\circ)$ is calculated using the path rotation approach to be as $(\Delta\theta, \Delta\phi, \Delta\psi) = (45^\circ, 0^\circ, 0^\circ)$. The path in the optimal location determined by rotation approach is shown in Fig. 10. The task cycle time was reduced from originally 37.7 seconds to 35.7 seconds which implies an improvement of 5.3 percent compared to the original path location.

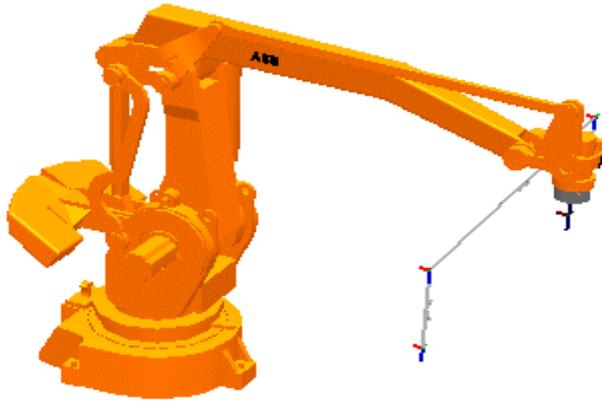


Fig. 9. IRB640 ABB robot with the path of case 4 in its original location

2.4.2.2 Case 2

The second case study is conducted with the same robot and path presented in 2.4.1.2. An arbitrary point close to the trajectory was selected as the rotation center. The optimal location of the task in a boundary of $(\pm 45^\circ, \pm 45^\circ, \pm 30^\circ)$ is calculated using the path rotation approach to be as $(\Delta\theta, \Delta\phi, \Delta\psi) = (45^\circ, 0^\circ, 0^\circ)$. The cycle time of this path is reduced from originally 6.0 seconds to 5.5 seconds which indicates 8.3 percent cycle time reduction.

2.4.2.3 Case 3

In the third example the same robot and path presented in section 2.4.1.3 are studied. The middle point of the long side was selected as the rotation center. To fulfill the restrictions outlined in section 2.4.1.3, only rotation around y-axis is allowed. The optimal location of the task in a boundary of $(0^\circ, \pm 90^\circ, 0^\circ)$ is calculated using the path rotation approach to be as $(\Delta\theta, \Delta\phi, \Delta\psi) = (0^\circ, -60^\circ, 0^\circ)$. Here the sensitivity analysis was also performed. The cycle time of this path is reduced from originally 2.8 seconds to 2.2 seconds which evidences 21 percent cycle time reduction.

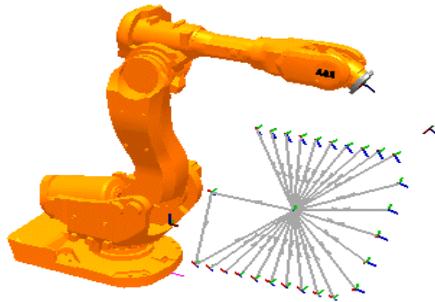


Fig. 10. IRB6600 ABB robot with a spot welding path of case 1 in optimal location found by rotation approach

2.4.2.4 Case 4

The fourth case study is carried out with the same robot presented in 2.4.1.4. The point in the middle of a line which connects the first and fourth targets was chosen as the rotation center. Due to the fact that the robot has 4 degrees of freedom, only rotation around the z-axis is allowed. The optimal location of the task in a boundary of $(0^\circ, 0^\circ, \pm 45^\circ)$ is calculated using the path rotation approach to be as $(\Delta\theta, \Delta\phi, \Delta\psi) = (0^\circ, 0^\circ, 16^\circ)$. In this case the sensitivity analysis was also performed. The cycle time of this path is reduced from originally 3.7 seconds to 3.6 seconds which gives 3.5 percent cycle time reduction.

2.4.3 Summary of the Results of Section 2

The cycle time reduction percentages that are achieved by translation and rotation approaches compared to longest and original cycle time are demonstrated in Fig. 11. The longest cycle time which corresponds to worst performance location is recognized as an existing admissible location that has the longest cycle time, i.e., the longest cycle time among experiments. As can be perceived, a cycle time reduction in range of 8.7 - 37.2 percent is achieved as compared to the location with the worst performance.

Results are also compared with the cycle time corresponding to original path location. This comparison is of interest as the tasks were programmed by experienced engineers and had been originally placed in proper position. Therefore this comparison can highlight the efficiency and value of the algorithm. The results demonstrate that cycle time is reduced by 3.5 - 21.1 percent compared with the original cycle time.

Fig. 11 indicates that both translation and rotation approaches are capable to noticeably reduce the cycle time of a robot manipulator.

A relatively lower gain in cycle time reduction in case four is related to a robot with four joints. This robot has fewer joint than the other tested robots with six joints. Generally, the fewer number of joints in a robot manipulator, the fewer degrees of freedom the robot has. The small variation of the cycle time in the whole admissibility area can imply that this robot has a more homogeneous dynamic behavior. Path geometry may also contribute to this phenomenon.

Also note that cycle time may be further reduced by performing more experiments. Although doing more experiments implies an increase in simulation time, this cost can

reasonably be neglected by noticing the amount of time saving, for instance 20 percent in one year. In other word, the increase in productivity in the long run can justify the initial high computational burden that may be present, noting that this is a onetime effort before the assembly line is set up.

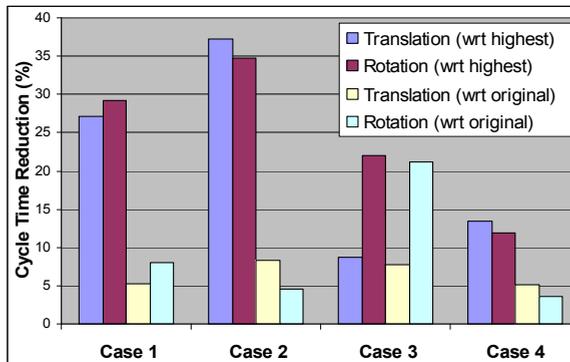


Fig. 11. Comparison of cycle time reduction percentage with respect to highest and original cycle time in four case studies

3. Combined Drive-Train and Robot Placement Optimization

3.1 Research background

Offline programming of industrial robots and simulation-based robotic work cell design have become an increasing important approach for the robotic cell designers. However, current robot programming systems do not usually provide functionality for finding the optimum task placement within the workspace of a robot manipulator (or relative placement of working stations and robots in a robotic cell). This poses two principal challenges: 1) Develop methodology and algorithms for formulating and solving this type of problems as optimization problems and 2) Implement such methodology and algorithms in available engineering tools for robotic cell design engineers.

In the past years, much research has been devoted to the methodology and algorithm development for solving optimization problem of designing robotic work cells. In Section 2, a robust and sophisticated approach for optimal task placement problem has been proposed, developed, and implemented in one of the well-known robot offline programming tool RobotStudio from ABB. In this approach, the cycle time is used as the objective function and the goal of the task placement optimization is to place a pre-defined task defined in a robot motion path in the workspace of the robot to ensure minimum cycle time.

In this section, firstly, the task placement optimization problem discussed in Section 2 will be extended to a multi-objective optimization problem formulation. Design space for exploring the trade-offs between cycle time performance and lifetime of some critical drive-train component as well as between cycle time performance and total motor power consumption are presented explicitly using multi-objective optimization. Secondly, a combined task placement and drive-train optimization (combined optimization will be termed in following texts throughout this chapter) will be proposed using the same multi-

objective optimization problem formulation. To authors' best knowledge, very few literature has disclosed any previous research efforts in these two types of problems mentioned above.

3.2 Problem statement

Performance of a robot may be modified by re-setting robot drive-train configuration parameters without any need of modification of hardware of the robot. Performance of a robot depends on positioning of a task that the robot performs in the workspace of the robot. Performance of a robot may therefore be optimized by either optimizing drive-train of the robot (Pettersson, 2008; Pettersson & Ölvander, 2009; Feng et al., 2007) or by optimizing positioning of a task to be performed by the robot (Kamrani et al., 2009).

Two problems will be investigated: 1) Can the task placement optimization problem described in Section 2 be extended to a multi-objective optimization problem by including both cycle time performance and lifetime of some critical drive-train component in the objective function and 2) What significance can be expected if a combined optimization of a robot drive-train and robot task positioning (**simultaneously optimize a robot drive-train and task positioning**) is conducted by using the same multi-objective optimization problem formulation.

In the first problem, additional aspects should be investigated and quantified. These aspects include 1) How to formulate multi-objective function including cycle time performance and lifetime of critical drive-train component; 2) How to present trade-off between the conflicting objectives; 3) Is it feasible and how efficient the optimization problem may be solved; and 4) How the solution space would look like for the cycle time performance vs. total motor power consumption.

In the second problem investigation, in addition to those listed in the problem formulation for the first type of problem discussed above, following aspects should be investigated and quantified: 1) Is it meaningful to conduct the combined optimization? A careful benchmark work is requested; 2) How efficient the optimization problem may be solved when additional drive-train design parameters are included in the optimization problem? Will it be applicable in engineering practice?

It should be noted that, focus of this work presented in Section 3 is on methodology development and validation. Therefore implementation of the developed methodology is not included and discussed. However, the problem and challenge for future implementation of the developed methodology for the combined optimization will be clarified.

3.3 Methodology

3.3.1 Robot performance simulation

A special version of the ABB virtual controller is employed in this work. It allows access to all necessary information, such as motor and gear torque, motor and gear speed, for design use. Based on the information, total motor power consumption and lifetime of gearboxes may be calculated for used robot motion cycle. The total motor power is calculated by summation of power of all motors present in an industrial robot. The individual motor power consumption is calculated by sum of multiplication of motor torque and speed at each simulation time step. The lifetime of gearbox is calculated based on analytical formula normally provided by gearbox suppliers.

3.3.2 Objective function formulation

The task placement optimization has been formulated as a multi-objective design optimization problem. The problem is expressed by

$$\min F(\mathbf{DV}) = w_1 \times CT_{norm}(\mathbf{DV}) + w_2 \times 1/LT_{norm}(\mathbf{DV}) \quad (5)$$

where CT_{norm} is a normalized cycle time, calculated by

$$CT_{norm} = CT/CT_{original} \quad (6)$$

CT is the cycle time at each function evaluation in the optimization loop. $CT_{original}$ is the cycle time of the robot motion cycle with original task placement and original drive-train parameter setup for combined optimization. LT_{norm} is a normalized lifetime of gearbox of some selected critical axis. It is calculated by

$$LT_{norm} = LT/LT_{original} \quad (7)$$

LT is the lifetime of some critical gearbox selected based on the actual usage of the robot at each function evaluation in the optimization loop. $LT_{original}$ is the lifetime of the selected gearbox of the robot motion cycle with original task placement and original drive-train parameter setup for combined optimization. w_1 and w_2 are two weighting factors employed in the weighted-sum approach for multi-objective optimization (Ölvander, 2001). \mathbf{DV} is a design variable vector.

Two optimization case studies have been conducted. Robot task placement optimization with the design variable vector defined as

$$\mathbf{DV} = [\Delta X, \Delta Y, \Delta Z]^T \quad (8)$$

and combined optimization with the design variable vector defined as

$$\mathbf{DV} = [\Delta X, \Delta Y, \Delta Z, DV_1, DV_2, \dots, DV_n]^T \quad (9)$$

where $DV_1, DV_2, DV_3, \dots, DV_n$ are the drive-train configuration parameters, while $\Delta X, \Delta Y, \Delta Z$ are the *change* in translational coordinates of all robot targets defining the position of a task.

3.3.3 Optimizer: ComplexRF

The optimization algorithm used in this work is the Complex method proposed by Box (Box, 1965). It is a non-gradient method specifically suitable for this type of simulation-based optimization. Figure 12 shows the principle of the algorithm for an optimization problem consisting of two design variables. The circles represent the contour of objective function values and the optimum is located in the center of the contour. The algorithm starts with randomly generating a set of design points (see the sub-figure titled "Start"). The number of the design points should be more than the number of design variables. The worst design point is replaced by a new and better design point by reflecting through the centroid of the remaining points in the complex (see the sub-figure titled "1. Step"). This procedure repeats until all design points in the complex have converged (see last two sub-figures from left). This method does not guarantee finding a global optimum. In this work, an improved version of the Complex, or normally referred to as ComplexRF, is used, in which a level of

randomization and a forgetting factor are introduced for improvement of finding the global optimum (Krus et al., 1992; Ölvander, 2001).

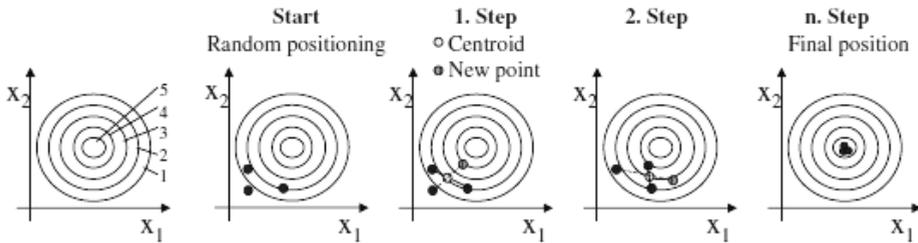


Fig. 12. The progress of the Complex method for a two dimensional example, with the optimum located in the center of the circles (Reprinted with permission from Dr. Johan Ölvander)

3.3.4 Workflow

The workflow of the proposed methodology starts with an optimizer generating a set of design variables. The variables defining robot task placement are used to manipulate the position of the robot task. The variables defining robot drive-train parameters are used to manipulate the drive-train parameters. The ABB robot motion simulation tool is run using the new task position and new drive-train setup parameters. Simulation results are used for computing objective function values. A convergence criterion is evaluated based on the objective function values. This optimization loop is terminated when either the optimization is converged or the limit for maximum number of function evaluations is reached. Otherwise, the optimizer analyzes the objective function values and proposes a new trial set of design variable values. The optimization loop continues until the convergence criterion is met.

3.4 Results on combined optimization

3.4.1 Case-I: Optimal robot usage for a spot welding application

In this case study, an ABB IRB6600-255-175 robot is used. The robot has a payload handling capacity of 175 kg and a reach of 2.55 m. A payload of 100 kg is defined in the robot motion cycle. The robot motion cycle used is a design cycle for spot welding application. The motion cycle consists of about 50 robot tool position targets. Maximum speed is programmed between any adjacent targets. A graphical illustration of the robot motion cycle is shown in Figure 5.

3.4.1.1 Task placement optimization

Only path translation is employed in the task placement optimization. Three design variables ΔX , ΔY , and ΔZ are used. They are added to all original robot targets so that the original placement of the robot task may be manipulated by ΔX in X coordinates, by ΔY in Y coordinates, and by ΔZ in Z coordinates. The limits for the path translation are

$$\begin{aligned}
\Delta X &\in (-0.1 \text{ m}, 0.1 \text{ m}) \\
\Delta Y &\in (0 \text{ m}, 0.8 \text{ m}) \\
\Delta Z &\in (-0.1 \text{ m}, 0.1 \text{ m})
\end{aligned}
\tag{10}$$

The weighting factors w_1 and w_2 in objective function (5) are set to $w_1 = 150$ and $w_2 = 100$ in this task placement optimization.

The convergence curve of the task placement optimization is shown in Figure 13(a). The optimization is well converged after about 100 function evaluations. The total optimization time is about 15 min on a portable PC with Intel(R) Core(TM) 2 Duo CPU T9600 @ 2.8 GHz.

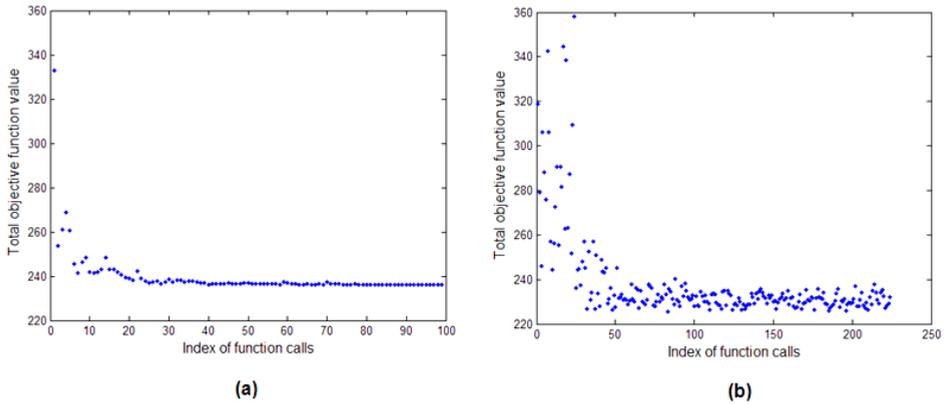


Fig. 13. Convergence curve. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

Figure 14(a) shows the solution space of normalized lifetime of a critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure suggest one solution point with 8% reduction in cycle time (or improved cycle time performance) on the cost of about 50% reduction in the lifetime (point A1 in the figure 14(a)). Another interesting result disclosed in the figure is solution points in region A2, where about 20% increase in lifetime may be achieved with the same or rather similar cycle time performance. Figure 15(a) shows the solution space of normalized total motor power consumption as function of cycle time. The normalized total motor power consumption is obtained by actual total motor power consumption at each function evaluation in the optimization loop divided by the total motor power consumption obtained on the robot motion cycle programmed at original task placement. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure disclose that the ultimate performance improvement point suggested by point A1 in figure 14(a) results in an increase of about 20% in total motor power consumption (point B1 in the figure 15(a)). Another interesting result disclosed in the figure is solution points in region B2, where about 5% saving of total motor power consumption

may be achieved for the solution points presented in region A2 in figure 14(a). In other words, the solution points in region A2 in figure 15(a) suggest not only increase in lifetime but also saving of total motor power consumption.

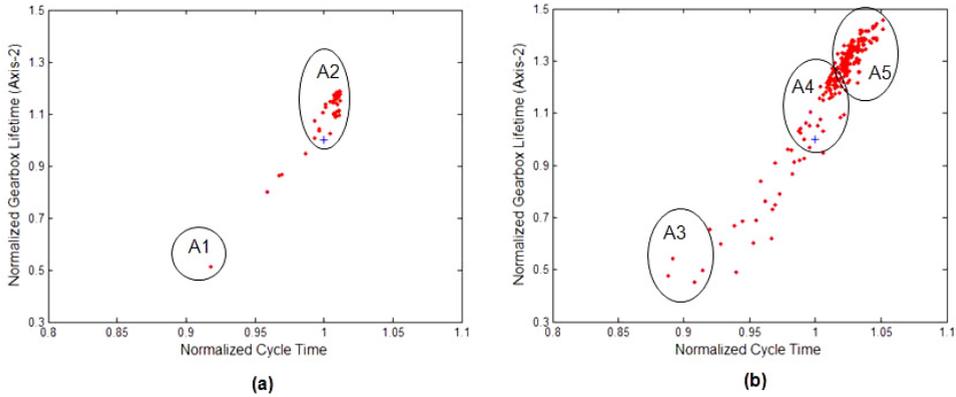


Fig. 14. Solution space of normalized lifetime of gearbox of axis-2 vs. normalized cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

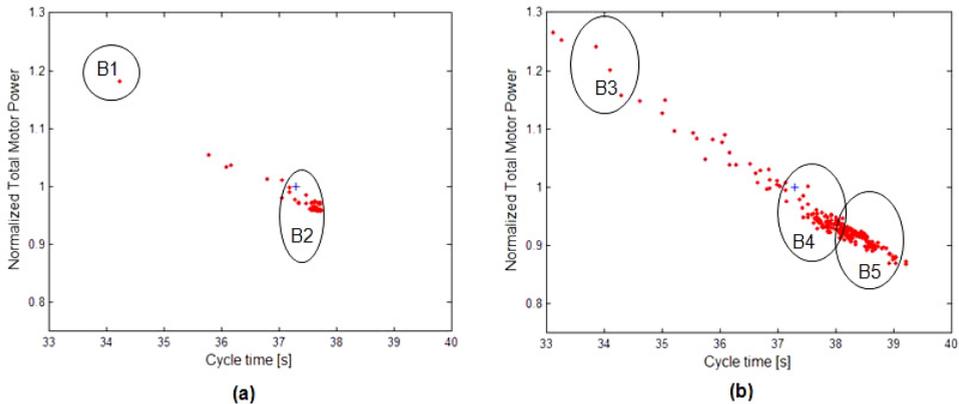


Fig. 15. Solution space of normalized total motor power vs. cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6600-255-175 robot)

3.4.1.2 Combined task placement and drive-train optimization

The combined optimization involves both path translation for robot task placement and change of robot drive-train parameter setup. Two sets of design variables are used, the first set includes ΔX , ΔY , and ΔZ described in the task placement optimization; the second set includes nine design variables DV_1, DV_2, \dots, DV_9 which are scaling factors to be multiplied to the original drive-train parameters of the three main axes (axes 1-3). The limits for the path

translation are the same as those used in the task placement optimization, i.e., the same as in (10).

The limits for the DV_1, DV_2, \dots, DV_9 are

$$DV_i \in (0.9, 1.2), \text{ where } i = 1, 2, \dots, 9 \quad (11)$$

The weighting factors w_1 and w_2 are also set to $w_1 = 150$ and $w_2 = 100$ in this combined optimization.

To ease the benchmark work of task placement optimization and the combined optimization, the results of the combined optimization are presented in the same figures as those of task placement optimization. In addition, the figures are carefully prepared at the same scale.

Figure 13(b) shows the convergence curve of the combined optimization. The maximum limit of function evaluations for the optimizer is set to be 225. Optimization is interrupted after the maximum number of function evaluation limit is reached. The total optimization time is about 45 min on the same portable PC used in this work.

Figure 14(b) shows the solution space of normalized lifetime of the same critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in the figure suggest one solution point with more than 10% reduction in cycle time (or improved cycle time performance) on the cost of about 50% reduction in the lifetime (point A3 in the figure). Another result set disclosed in region A4 in the figure indicates up to 25% increase in lifetime that may be achieved with the same or rather similar cycle time performance. When a cycle time increase of up to 5% is allowed in practice, the lifetime of the critical gearbox may be increased by as much as close to 50% (region A5).

Figure 15(b) shows the solution space of normalized total motor power consumption as function of cycle time. The normalized total motor power consumption is obtained by actual total motor power consumption at each function evaluation in the optimization loop divided by the total motor power consumption obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in the figure disclose that the ultimate performance improvement point suggested by point A3 in figure 14(b) results in an increase of about 20% in total motor power consumption (point B3 in the figure 15(b)). Another interesting result set disclosed in the figure is solution points in region B4, where about 5% saving of total motor power consumption may be achieved for the solution points presented in region A4 in figure 14(b). In other words, the solution points in region A4 in figure 14(b) suggest not only increase in lifetime but also saving of total motor power consumption. When a cycle time increase of up to 5% is allowed, not only the lifetime of the critical gearbox may be increased by as much as close to 50% (region A5) but also the total motor power consumption may be reduced by more than 10%.

3.4.1.3 Comparison between task placement optimization and combined optimization

When comparing the task placement optimization with combined optimization, it is evident that the combined optimization results in much large solution space. This implies in practice that robot cell design engineers would have more flexibility to place the task and setup drive-train parameters in more optimal way. However, the convergence time is also longer, due to the increase in number of design variables introduced in the combined optimization. In addition, changing drive-train parameters in robot cell optimization may pose additional consideration in robot design, so that the adaptation of drive-train in cell optimization would not result in unexpected consequence for a robot manipulator.

3.4.2 Case-II: Optimal robot usage for a typical material handling application

In this case study, an ABB IRB6640-255-180 robot is used. The robot has a payload handling capacity of 180 kg and a reach of 2.55 m. The payload used in the study is 80 kg. The robot motion cycle used is a typical pick-and-place cycle with 400 mm vertical upwards - 2000mm horizontal - 400mm vertical downwards movements - then reverse trajectory to return to the original position. Maximum speed is programmed between any adjacent targets.

3.4.2.1 Task placement optimization

Only path translation is employed in the task placement optimization. Three design variables, ΔX , ΔY , and ΔZ are used to manipulate the task position in the same manner as discussed in the Case-I. The limits for the path translation are

$$\begin{aligned}\Delta X &\in (-0.1 \text{ m}, 0.1 \text{ m}) \\ \Delta Y &\in (-0.1 \text{ m}, 0.1 \text{ m}) \\ \Delta Z &\in (-0.2 \text{ m}, 0.5 \text{ m})\end{aligned}\tag{12}$$

The weighting factors w_1 and w_2 are set to $w_1 = 100$ and $w_2 = 100$ in this task placement optimization.

The convergence curve of the task placement optimization is shown in Figure 16(a). The optimization is converged after 290 function evaluations. The total optimization time is about 40 min on the same portable PC used in this work.

Figure 17(a) shows the solution space of normalized lifetime of a critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure suggest one set of solution points with close to 6% reduction in cycle time (or improved cycle time performance) with somehow improved lifetime of the critical axis under study (region A6 in the figure). Another interesting result set disclosed in the figure is solution points in region A7, where about 20% increase in lifetime may be achieved with 3-4% improvement of cycle time performance. In engineering practice, 3-4% cycle time improvement can imply rather drastic economic impacts.

Figure 18(a) shows the solution space of normalized total motor power consumption as function of cycle time. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement. The results presented in the figure disclose

that the solution points with more than 4% cycle time performance improvement (region B6) result in at least 20% increase in total motor power consumption.

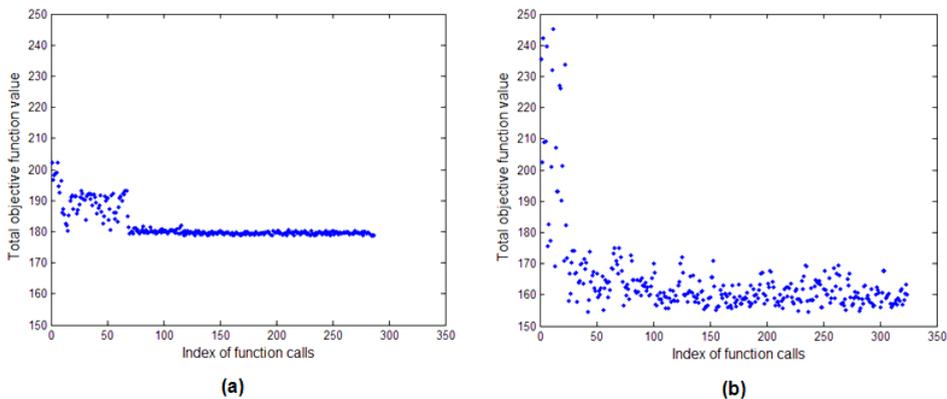


Fig. 16. Convergence curve. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

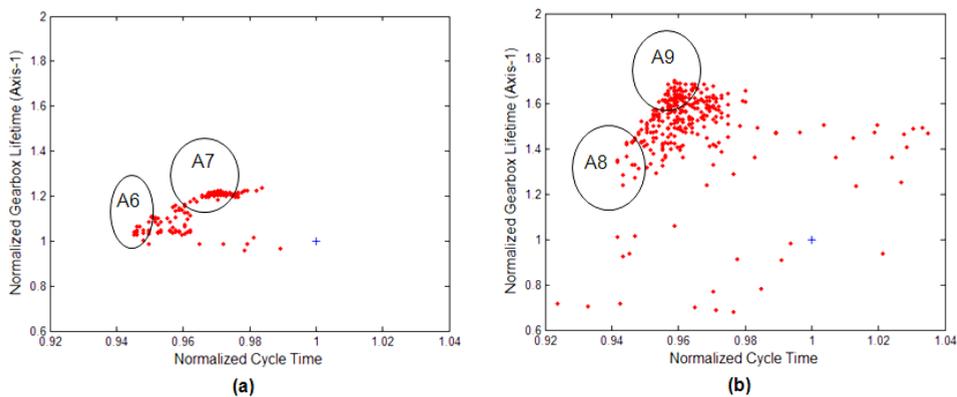


Fig. 17. Solution space of normalized lifetime of gearbox of axis-2 vs. normalized cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

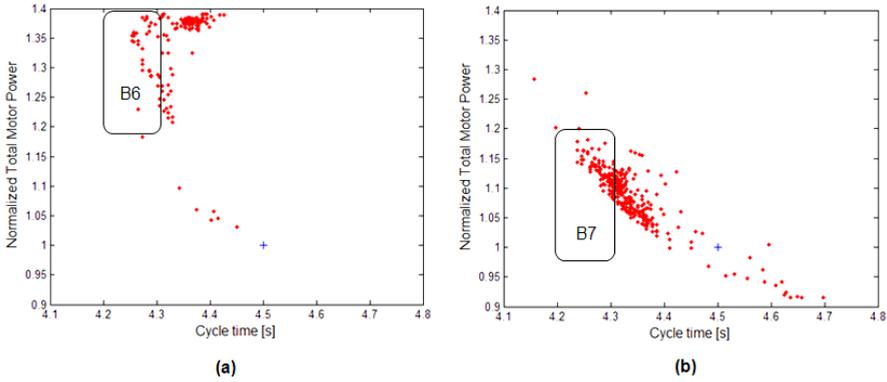


Fig. 18. Solution space of normalized total motor power vs. cycle time. (a) for optimal task placement and (b) for combined optimization, (ABB IRB6640-255-180 robot)

3.4.2.2 Combined task placement and drive-train optimization

As discussed in Case-I, the combined optimization involves both path translation for robot task placement and change of robot drive-train parameter setup. The same two sets of design variables are used. The limits for the path translation are the same as those used in the task placement optimization which are defined by (13).

The limits for the DV_1, DV_2, \dots, DV_9 are

$$DV_i \in (0.9, 1.2), \text{ where } i = 1, 2, \dots, 9 \quad (13)$$

The weighting factors w_1 and w_2 are also set to $w_1 = 100$ and $w_2 = 100$ in this combined optimization.

For the same reason, the results of the combined optimization are presented in the same figures as those of task placement optimization. In addition, the figures are carefully prepared at the same scale.

Figure 16(b) shows the convergence curve of the combined optimization. The maximum limit of function evaluations for the optimizer is set to be 325. Optimization is interrupted after the maximum number of function evaluation limit is reached. The total optimization time is about 65 min on the same portable PC used in this work.

Figure 17(b) shows the solution space of normalized lifetime of the same critical gearbox as function of normalized cycle time. The cross symbol in blue color indicates the coordinate representing normalized lifetime and normalized cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup values. The results presented in region A8 in the figure suggest a set of solution points with close to 6% reduction in cycle time but with clearly more than 20% increase in the lifetime. Another result set disclosed in region A9 in the figure indicates more than 60% increase in lifetime and with 3-4% improved cycle time performance!

Figure 18(b) shows the solution space of normalized total motor power consumption as function of cycle time. The cross symbol in blue color indicates the coordinate representing normalized total motor power consumption and cycle time obtained on the robot motion cycle programmed at original task placement and with original drive-train parameter setup

values. The solution points disclosed in region B9 indicate that the solution points with more than 4% cycle time performance improvement result in maximum 20% increase in total motor power consumption.

3.4.2.3 Comparison between task placement optimization and combined optimization

Compared to task placement optimization, it is evident that the combined optimization results in much large solution space. This implies in practice that robot cell design engineers would have more flexibility to place the task and setup drive-train parameters in more optimal way. Even more significantly, the optimization results obtained on this typical pick-and-place cycle reveals more interesting observations. When the same cycle time improvement may be achieved, much more significant lifetime improvement may be achieved by combined optimization and the same is true for the total motor power consumption.

However, the convergence time is longer and optimization has to be interrupted using pre-defined maximum number of function evaluations, due to the increase in number of design variables introduced in the combined optimization. In addition, the same consequence is evident: changing drive-train parameters in robot cell optimization may pose additional consideration in robot design, so that the adaptation of drive-train in cell optimization would not result in unexpected consequence for a robot manipulator.

3.5 Summary of the Results of Section 3

Multi-objective robot task placement optimization shows obvious advantage to understand the trade-off between cycle time performance and lifetime of critical drive-train component. Sometimes, it may be observed that the cycle time performance and lifetime can be simultaneously improved. When task placement optimization involving only path translation is conducted, reasonable optimization time can be achieved.

The combined optimization of a robot drive-train and robot task placement, in comparison with task placement optimization, has disclosed even more advantages in achieving 1) wider solution space and 2) even more simultaneously improved cycle time performance and lifetime. Benefit of the combined optimization has been evident. Even though the optimization time can be nearly 2-3 times longer than task placement optimization, it can still be justified to be used in engineering practice; namely, earning from longer lifetime of a robot installation is greater than the calculation costs. Furthermore, this suggests that more efforts should be devoted in the future to; 1) better understanding of the multi-objective combined optimization problem and its impact on simulation-based robot cell design optimization; 2) improving efficiency of the optimization algorithms; 3) including collision-free task placement; and finally 4) sophisticated software implementation for engineering usage.

The plots of lifetime of critical component as function of cycle time performance and that of total motor power consumption as function of cycle time performance are also suggested in this work. This graphical representation of the solution space can further ease robot cell design engineers to better understand the trade-off between lifetime of critical drive-train component or total motor power consumption to cycle time performance and therefore choose better design solution that meets their goal.

4. Conclusions and Outlook

4.1 Single Objective Optimization

The results confirm that the problem of path placement in a robot work cell is an important issue in terms of manipulator cycle time. Cycle time greatly depends on the path position relative to the robot manipulator. Up to the 37.2% variation of cycle time has been observed which is remarkably high. In other words, the cycle time is very sensitive to the path placement. Algorithm and tool were developed to determine the optimal robot position by path translation and path rotation approaches. Several case studies were considered to evaluate and verify the developed tool for optimizing the robot position in a robotic work cell. Results disclose that an increase in productivity up to 37.2% can be achieved which is profoundly valuable in industrial robot application. Therefore, using this tool can significantly benefit the companies which have similar manipulators in use.

It is certain that employing this methodology has many important advantages. First, the cycle time reduces significantly and, therefore, the productivity increases. The method is easy to implement and the expense is only simulation cost, i.e., not any extra equipment is needed to be designed or purchased. The solution coverage is considerably broad, meaning that any type of robots and paths can be optimized with the proposed methodology. Another merit of the algorithm is that convergence is not an issue, i.e., reducing the cycle time can be assured. However, a disadvantage is that a global optimum cannot be guaranteed. The importance of the developed methodology is not confined only to the robot end-user application. Robot designers can also take advantage of the proposed methodology by optimizing the robot parameters such as robot structure and drive-train parameters to improve robot performance. As a design application example, the idea of optimum relative position of robot and path can be applied to the design of a tool such as welding device or glue gun which is erected on the mounting flange of the robot. The geometry of the tool can be optimized by studying design parameters to achieve shorter cycle time. Another possibility can be to use the developed methodology for optimal robot placement to realize other optimization objective in robots such as minimizing the torque, energy consumption, and component wear.

One interesting issue that can be investigated is to consider the general problem of finding the optimum by translation and rotation of the path simultaneously. What has been demonstrated in section 2 of the current chapter is to find the optimum path location by either translation or rotation of the path. Obviously, it is also possible to apply both these approaches at the same time. This would probably further shorten the cycle time in comparison to the case when only one approach is used. However, developing an optimal strategy for concurrently applying both approaches is an interesting challenge for future research.

Another important subject to be investigated is to take into account constraints for avoiding collisions. In a real application, a robot is not alone in the work cell as other cell equipments can exist in the workspace of the robot. Hence, in real robot application it is important to avoid collision.

4.2 Multi-Objective Optimization

It is noteworthy that although the methodology is implemented in RobotStudio, the algorithm is general and not dependent on RobotStudio. Therefore, the same methodology

and algorithm can be implemented in any other robotic simulation software for achieving time optimality.

Multi-objective robot task placement optimization shows obvious advantage to understand the trade-off between cycle time performance and lifetime of critical drive-train components. The combined optimization of a robot drive-train and robot task placement, in comparison with task placement optimization, discloses even more advantages in achieving wider solution space and even more simultaneously improved cycle time performance and lifetime.

However, weighted-sum approach for formulating the multi-objective function has experienced difficulties in this work, since the weighting factors have been observed to significantly affect the final solution. Hence, an advanced formulation of multi-objective function and algorithms for multi-objective optimization need to be investigated.

In combined optimization, the reachability is presumed to be satisfied as the purpose of this work is to rather explore the effect and feasibility of the method. Nevertheless, advanced and practical solutions exist for reachability checking that need to be implemented in the future work. In this study, while the task placement defined in a robot program is manipulated, the relative placements among sub-tasks (representing in practice the relative placements among different robotic stations in a robot cell) are kept unchanged. In the future work, relative placements of sub-tasks in a robot cell can also be optimized using the proposed methodologies.

5. References

- Barral, D. & Perrin, J-P. & Dombre, E. & Lie'geois, A. (1999). Development of optimization tools in the context of an industrial robotic CAD software product, *International Journal of Advanced Manufacturing Technology*, Vol. 15(11), pp. 822-831, doi: 10.1007/s001700050138
- Box, G.E.P. & Hunter, W.G. & Hunter, J.S. (1978). Statistics for experimenters: an introduction to design, data analysis and model building, Wiley, New York
- Box, M. J., (1965). A New Method of Constrained Optimization and a Comparison with Other Methods, *Computer Journal*, Vol 8, pp. 42-52
- Fardanesh, B. & Rastegar, J. (1988). Minimum cycle time location of a task in the workspace of a robot arm, *Proceeding of the IEEE 23rd Conference on Decision and Control*, pp. 2280-2283
- Feng, X. & Sander, S.T. & Ölvander, J. (2007). Cycle-based Robot Drive Train Optimization Utilizing SVD Analysis, *Proceedings of the ASME Design Automation Conference*, Las Vegas, September 4-7, 2007
- Haug, E.J. (1992). Intermediate dynamics, Prentice-Hall, Englewood Cliffs, NJ
- Kamrani, B. & Berbyuk, V. & Wäppling, D. & Stickelmann, U. & Feng, X. (2009). Optimal Robot Placement Using Response Surface Method, *International Journal of Advanced Manufacturing Technology*, Vol. 44, pp. 201-210
- Khuri, A.I. & Cornell, J.A. (1987). Response surfaces design and analyses, Dekker, New York
- Krus, P. & Jansson, A. & Palmberg, J-O. (1992). Optimization Based on Simulation for Design of Fluid Power Systems, *Proceedings of ASME Winter Annual Meeting*, Anaheim, USA

- Luenberger, D.G. (1969). Optimization by vector space methods, Wiley, New York
- Myers, R.H. & Montgomery, D. (1995). Response surface methodology: process and product optimization using designed experiments, Wiley, New York
- Nelson, B. & Donath, M. (1990). Optimizing the location of assembly tasks in a manipulator's workspace, *Journal of Robotic Systems*, Vol 7(6), pp. 791-811, doi:10.1002/rob.4620070602
- Pettersson, M. & Ölvander, J. (2009). Drive Train Optimization for Industrial Robots, *IEEE Transactions on Robotics*, to be published
- Pettersson, M. (2008). A PhD Dissertation, Linköping University, Linköping, Sweden
- Tsai, L.W. (1999). Robot analysis, Wiley, New York
- Tsai, M.J. (1986). Workspace geometric characterization and manipulability of industrial robot. Ph.D. Thesis, Department of Mechanical Engineering, Ohio State University
- Vukobratovic, M. (2002). Beginning of robotics as a separate discipline of technical sciences and some fundamental results – a personal view, *Robotica*, Vol. 20(2), pp. 223-235
- Yoshikawa, T. (1985). Manipulability and redundancy control of robotic mechanisms, *Proceeding of the IEEE Conference on Robotics and Automation*, pp 1004-1009, St. Louis
- Ölvander J. (2001). Multiobjective Optimization in Engineering Design - Applications to Fluid Power Systems, A PhD Dissertation, No. 675 at Linköping University

ROBOTIC MODELLING AND SIMULATION: THEORY AND APPLICATION

¹Muhammad Ikhwan Jambak, ¹Habibollah Haron,

²Helmee Ibrahim and ²Norhazlan Abd Hamid

¹*Soft Computing Research Group, Universiti Teknologi Malaysia,
Malaysia*

²*Department of Modeling & Industrial Computing, Faculty of Computer Science &
Information System, Universiti Teknologi Malaysia,
Malaysia*

1. Introduction

The employment of robots in manufacturing has been a value-adding entity for companies in gaining a competitive advantage. Zomaya (1992) describes some features of robots in industries, which are decreased cost of labour, increased flexibility and versatility, higher precision and productivity, better human working conditions and displaced human working in hazardous and impractical environments.

Farrington *et al.* (1999) states that robotic simulation differs from traditional discrete event simulation (DES) in five ways in terms of its features and capabilities. Robotic simulation covers the visualization of how the robot moves through its environment. Basically, the simulation is based heavily on CAD and graphical visualization tools. Another type of simulation is numerical simulation, which deals with the dynamics, sensing and control of robots. It has been accepted that the major benefit of simulation is reduction in cost and time when designing and proving the system (Robinson, 1996).

Robotic simulation is a kinematics simulation tool, whose primary use is as a highly detailed, cell-level validation tool (Farrington *et al.*, 1999), and also for simulating a system whose state changes continuously based on the motion(s) of one or more kinematic devices (Roth, 1999). It is also used as a tool to verify robotic workcell process operations by providing a “mock-up” station of a robots application system, in order to check and evaluate different parameters such as cycle times, object collisions, optimal path, workcell layout and placement of entities in the cell in respect of each other.

This paper presents the methodology in modelling and simulating a robot and its environment using Workspace and X3D software. This paper will discuss the development of robotic e-learning to improve the efficiency of the learning process inside and outside the class.

This paper is divided into five sections. Section 2 discusses the robotic modelling method. Section 3 discusses robotic simulation. Its application using Workspace and X3D is presented in Section 4, and a conclusion is drawn in Section 5.

2. Robotic Modelling Method

This section presents the methodology in modelling and simulating the robot and its environment. There are two types of methodology being applied, which are the methodology for modelling the robot and its environment proposed by Cheng (2000), and the methodology for robotic simulation proposed by Grajo *et al.*(1994). The methodologies have been customised to tailor the constraints of the Workspace software. Section 4 presents experimental results of the project based on the methodologies discussed in this section.

2.1 Robotic Modelling

Robotic workcell simulation is a modelling-based problem solving approach that aims to sufficiently produce credible solutions for a robotic system design (Cheng, 2000). The methodology consists of six steps, as shown in Figure 1.

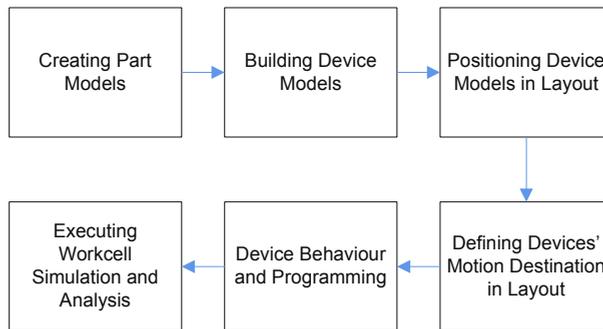


Fig. 1. A methodology for robotic modelling

2.1.1 Creating part models

Part model is a low-level or geometric entity. The parts are created using basic elements of solid modelling features of Workspace5. These parts consist of the components of the robot and the devices in its workcell, such as the conveyor, pallet and pick up station.

2.1.2 Building device models

Device models represent actual workcell components and are categorized as follows: robotic device model and non-robotic device model. The building device model starts by positioning the base of the part model as the base coordinate system. This step defines links with the robot. These links are attached accordingly to its number by using the attachment feature of Workspace5. Each attached link is subjected to a Parent/Child relationship.

2.1.3 Positioning device models in layout

The layout of the workcell model refers to the environment that represents the actual workcell. As in this case, the coordinates system being applied is the Hand Coordinates System of the robot. Placement of the model and devices in the environment is based on the actual layout of the workcell.

2.1.4 Defining devices motion destination in layout

The motion attributes of the device model define the motion limits of the joints of the device model in terms of home, position, speed, accelerations and travel. In Workspace5, each joint is considered part of the preceding link. A joint is defined by linking the Link or Rotor of the robot, for example Joint 1 is a waist joint which links the Base and the Rotor/Link. Each joint has its own motion limits. Once the joints have been defined, Workspace5 will automatically define the kinematics for the robot.

2.1.5 Device behaviour and programming

Device motion refers to the movement of the robot's arm during the palletizing process. The movement is determined by a series of Geometry Points (GPs) that create a path of motion for the robot to follow. Positioning the GP and the series is based on the movement pattern and the arrangement of bags. The GP coordinates are entered by using the Pendant features of Workspace5. There are three ways to create the GP: by entering the value for each joint, by entering the absolute value of X, Y and Z, and by mouse-clicking.

2.1.6 Executing workcell simulation and analysis

The simulation focuses only on the position of the robot's arm, not its orientation. After being programmed, the device model layout can be simulated over time. Execution of the simulation and analysis is done using the features of Workspace5. The simulated model is capable of viewing the movement of the robot's arm, layout checking, the robot's reachabilities, cycle time monitoring, and collision and near miss detection.

3. Robotic Simulation

3.1 Definition of Simulation

Shannon (1998) offered a good definition of simulation: "We will define simulation as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system". Robot simulation software or simulator is a computer program which mimics the elements of both the internal behaviour of a real-world system and the input processes which drive or control the simulated system. There are a few reasons why the simulation approach became the main option in real-world robotic related activity. Typically, most users make simulations because the experiment with the real world still does not yet exist, and experimentation with the robot's hardware is expensive, too time-consuming and too dangerous.

3.2 Type of Simulation

There are two types of simulation (F.E Cellier, 2006): *discrete event* simulation and *continuous system* simulation. *Discrete event* simulation divides a system into individual events that have their own specific start time and duration. The overall behaviour of a complex system of a real-world object will be determined from the sequencing and interactions of each event. This technique usually focuses on modelling the control logic for the routing of material and interaction of equipment. It also typically applies statistics to the system to simulate things like equipment breakdown or mixtures of different product models.

Continuous system simulation (F.E Cellier, 2006) describes systematically the mathematical models used in dynamic systems, and is usually done so using sets of either ordinary or partially different equations, possibly coupled with numerical integration, differential equation solvers or other mathematical approaches that can be simulated on a digital computer. Often, electrical circuits, control systems or similar mechanical systems are simulated in this way. Specific applications like thermal dynamics, aerodynamics, aircraft control systems or automobile crashes are commonly simulated with continuous system simulation.

3.3 Robot Programming

The main importance of robot functionality is its flexibility and ability to rearrange the new production, and its movement range. The flexibility of the robot depends on presupposed effective programming. Principally, the robot programming can take place in two different ways: on-line or off-line. In on-line programming, the use of the robot and equipment is required, whereas off-line programming is based on computer models of the production equipment. Both these methods have advantages and disadvantages. In this section we will look at how the two methods can be combined.

3.3.1 On-Line Programming

Currently, the operation of robot programming is through either on-line programming or off-line programming. The definition of on-line programming (Kin-Hua Low [10]) is a technique to generate a robot program using real robot systems. An on-line programming robot may be suitable to implement robot use by repeating a monotonous motion. The advantage of on-line programming is that it is easy to access. Its most significant advantage is that the robot is programmed in concordance with the actual position of the equipment and pieces.

However, the most significant disadvantage of on-line programming is the slow movement of the robot, the program implementation, program logic and calculations being hard to programme, the suspension of production whilst programming, poor documentation and costs equivalent to production value. The differentiation between on-line and off-line programming is shown clearly in the picture below.



Fig. 2. On-line robot programming

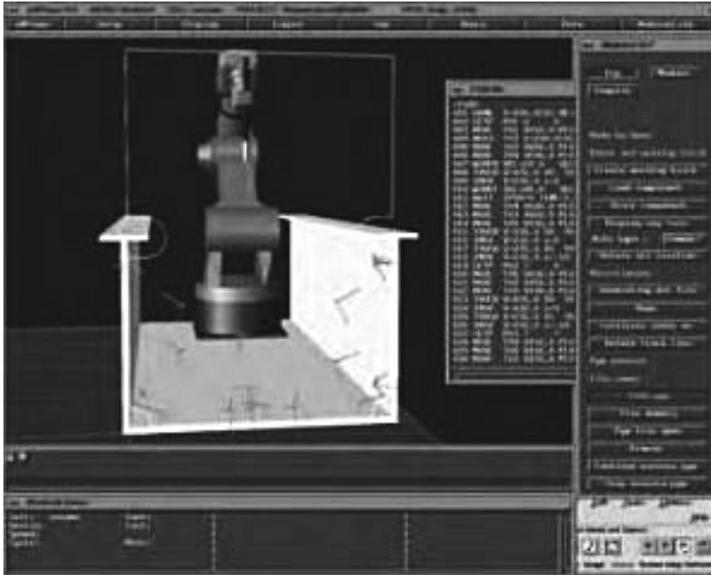


Fig. 3. Off-line robot programming

3.3.2 Off-Line Programming

The definition of off-line programming is that it is the technique used to generate a robot program without using a real machine. There are several advantages to this that have been cited by Bien (1998) resulting in a reduction in robot downtime, programs being prepared without interruptions to robot operation, and removal of the human from the potentially dangerous environment. There are several types of programming language used in off-line robotic programming, but the most popular off-line programming software was built using the JAVA language. Off-line programming using JAVA language become famous because it was easy to use, could be integrated with UML, supported C/C++ languages, had architecture independence and had an advanced network support.

The main advantage of off-line programming is that it does not occupy production equipment, and in this way, production can continue during the programming process. But on-line programming is the largest proportion of robot programming today due to the fact that off-line programming has had a very high burden rate and has demanded the need of expert users.

Advanced off-line programming tools contain facilities for debugging and these assist in effective programming. The programming tools support the utilization of supporting tools for the programming process, for instance optimization of the welding process.

However, off-line programming also has its disadvantages such as the fact that it demands investing in an off-line programming system. Most simulation tools/simulators are implemented with four characteristic: workspace visualization, trajectory planning, communication with robot control and system navigation.

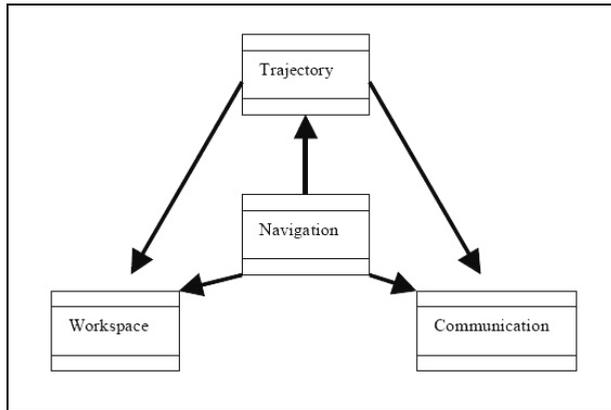


Fig. 4. Simulation tool modules

3.3.3 Off-Line Programming

Hybrid programming is a combination of both of the robotic programming methodology advantages shown above. By using both advantages, the programming technique can be optimized. A robot program consists mainly of two parts: locations (position and alignment) and program logics (controller structures, communication and calculations).

The program logics, debugging and simulation facilities are effectively developed on off-line programming. The main part of the movement can be created off-line by reusing the availability of CAD data and by programmer interaction.

Commands for movement to locating the piece placement in the robot's workcell can be more properly programmed on-line. In this situation, the advantages of both programming method can be utilized, indirectly increasing the flexibility in production.

The usage of hybrid programming is a very practical way of increasing flexibility in production and thereby increasing the effect of robot manufacturing. In the same way, rearrangement time can be substantially reduced, allowing for cost effectiveness even in the production of small batches.

3.4 Simulation Packages

The robotic simulation package is a tool which is used to create embedded applications for a specific (or not) robot without depending "physically" on the actual robot, thus saving cost and time. In some cases, the applications that were developed with the simulation package can be transferred to the real robot without modifications. This application allows the user to create a simple world and to programme this robot to interact with these worlds.

Most robotic simulation packages have their own unique features, but the main features for 3D modelling are robot rendering and environment. This type of robotics software has a simulator that is a "virtual" robot, which is capable of emulating the motion of an actual robot in a real work envelope. Some robotic simulation tools such as Matlab-Simulink can be used significantly in robot simulation, providing an interesting environment. Matlab-Simulink is an interactive robot simulation software that can be used as an interface of the system so that users can communicate with the system. This robotic simulation tool gives alternatives to minimize the limitation of Web Programming Language (WPL) and

Structured Programming Language (SPL). M. I. Jambak *et al.* (2008) state that Matlab-Simulink has been used in their previous research to model the graphical design of the Mitsubishi RV-2AJ robots and is dynamic in a 3D virtual reality (VR) environment, and uses the V-Realm Builder virtual programming language to apply the virtual reality modelling language (VRML).

Nathan *et al.* (2006) describe Virtual Reality Modelling Language (VRML) currently, as the de facto standard for web based 3D visualizations, which allows for easy definition of geometric shapes and provides many advanced 3D graphical functions such as lighting models and surface materials. VRML allows for simple interactions between a user of a virtual world and various objects within the world. Currently, VRML has been supported with various user browser and modelling programs.

Java3D (Nathan *et al.* 2006) is a simulation package which provides an object-oriented language-based approach for designing a 3D system. Java3D offers a high-level Application Programming Interface (API) for 3D scene description and graphical control. Besides that, it also allows for a fully object-oriented approach to define and control the virtual agent and its environment. Java3D is also designed to take advantage of multi-threaded programming techniques, allowing for better performance from the implementation.

Webots (Michel, 2004) is one of most popular mobile robot simulations and is widely used for educational purposes. Webots uses the ODE (Open Dynamics Engine) for collision detection and simulating rigid body dynamics. It contains a rapid prototyping tool, allowing the user to create a 3D virtual world. Webots runs on Windows, Linux and Mac OS X.

Microsoft Robotics Studio (Eric Colon and Kristel Verbiest, 2008) is a 3D modelling and simulation environment for mobile robots operating in real-world conditions, which respects the law of physics and runs on top of DirectX.

3.5 Robotic Simulation

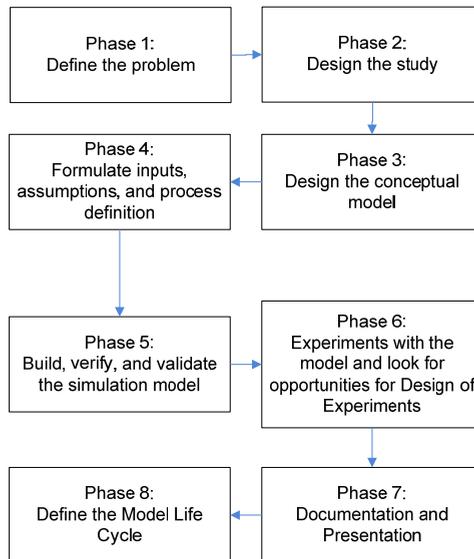


Fig. 5. A methodology for robotic simulation

The methodology consists of eight phases but the discussion only executes up to eight phases, as shown in Figure 5.

3.5.1 Define the problem

Problem identification is defined during the preliminary analysis of the problem's background. If the current system has no computer-based model that represents the robotic application, it is impossible to monitor and evaluate the performance of the robotic palletizing system. In contrast, the definition and analysis of the current system are easier to implement.

3.5.2 Design the study

The study is limited to the scope of the project. This phase acquires appropriate decisions for the tools and methodology to be used. Besides, proper planning and milestones need to be developed.

3.5.3 Design the conceptual model

The conceptual model is using the current application of the robotic system. This phase acquires collection of data of the parameters for the robotic workcell development. These data include layout of the robotic application, geometry configuration of the robot, robot motion parameters and the robot cycle time.

3.5.4 Formulate inputs, assumptions, and process definition

Modelling the robot application focuses on three activities: building the robot, motion path programming of the palletizing process, and running the simulation. Building the robot model is based heavily on the geometrical data of the robot using the CAD features of Workspace5. The dimension refers to the CAD drawing of the robot. Spatial data need to be considered in determining the motion path, such as the point of the pick up station where the robot will do the pick and place operation, the points that represent an arrangement and layer of the item to be picked, and the position of points in x , y and z coordinates.

3.5.5 Build, verify and validate the simulation model

During this phase, development of the robotic workcell is based on the methodology proposed by Cheng (2000). This is an interactive phase which aims to improve the model's precision and motion. Validation towards the model is based on the visualization of the system layout and robot cycle time in completing a task. The layout is generated using Workspace5 and compared to the actual system layout. During the gathering of preliminary data, a movie that shows the actual robot performing a task in a one-day operation is recorded. The model is assumed to represent the actual system once operated at the same movement of the actual system and is capable of performing at a similar cycle time as in the movie.

3.5.6 Experiments with the model and look for opportunities for Design of Experiments

This phase is similar to step six in the methodology by Cheng (2000). A simulation is run in order to visualize the arm movement and an analysis on collision detection is provided. Execution of the simulation is done using the features of Workspace5 for simulation. Workspace5 allows layout checking in order to prepare other devices within the robot's reachability. It is also capable of generating a working envelope for namely, two joints. During simulation, a cycle time is displayed. The simulation allows collision and near-miss detection among robot joints, and between the joints and any object within the workcell. The result is displayed in a report. This project is off-line programming. Neither robot language is generated or implemented at the actual workcell.

3.5.7 Documentation and presentation

This phase gathers and documents all the results generated from the simulation. A written report provides a better understanding of the experiment's executions and analysis.

There are advantages and disadvantages for this methodology (Mohd Johari *et al.*, 2008). The advantage of using this methodology is that it saves costs, avoiding designing, building, testing, redesigning, rebuilding and retesting which would be an expensive project. Simulations take the building or rebuilding phase out of the loop by using the model that has already been created in the design phase. Usually, the simulation test is cheaper and faster than performing multiple tests of the design each time.

The second advantage of using this methodology is the level of detail that we can get from the simulation. A simulation can give results that are not experimentally measurable with our current level of technology. Results such as time taken to complete the simulation and the details of collision detection of the simulation are not measurable by any current device.

There are also disadvantages to performing this methodology for robotic simulation. The first is simulation errors. Any incorrect key store for the value of the robot's details has the potential to alter the result of the simulation or give the wrong result. To get an accurate result, we must first run a baseline to prove that it works. In order for the simulation to be accepted in the general community, the experimental result is taken and simulates them. If the two data sets are compared, then any simulation of the design will have some credibility.

4. Application

This section describes two of the several projects that are related to modelling and simulation. The first is building robot simulation using Workspace5 and the second is robot simulation using X3D for e-learning. Below is an explanation of both of these:

4.1 Building Robot Simulation Using Workspace5

The experimental results presented in this section are based on authors' experience in supervising undergraduate and postgraduate final project works reported (Mohd Johari, 2008; Ariffin, 2007; Mohd Salih, 2008; Abdul Rahim, 2008; Muhammad Noor, 2005; Arifin, 2007; Zainal, 2008; Shafei, 2008, and Sukimin, 2007). Different types of robots were involved

in the experiments, which are situated in Universiti Teknologi Malaysia and other institutions.

Basic elements of solid modelling features in Workspace5 have been used to develop the robot and device models. Figures 3(a) and (b) show the development of the robot gripper and screwdriver device (Ariffin, 2008). Some solid modelling methods, such as union, subtract, or both, were applied in the models' development. Eventually, these models were compared with the actual robot for visual validation, as depicted in Figure 4.

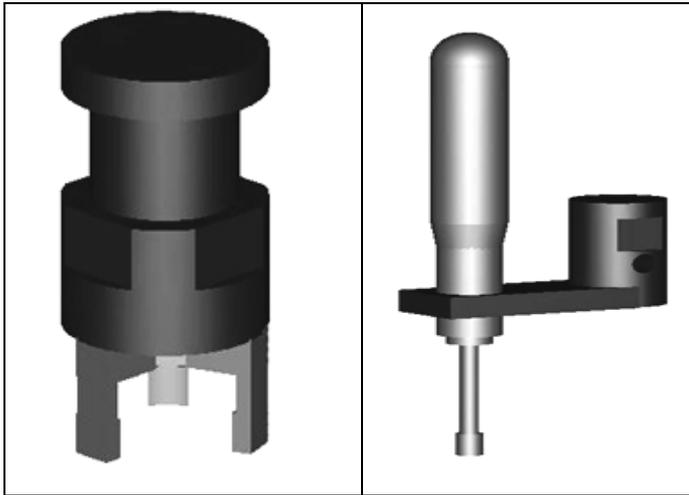


Fig. 6(a). Robot gripper and screwdriver model

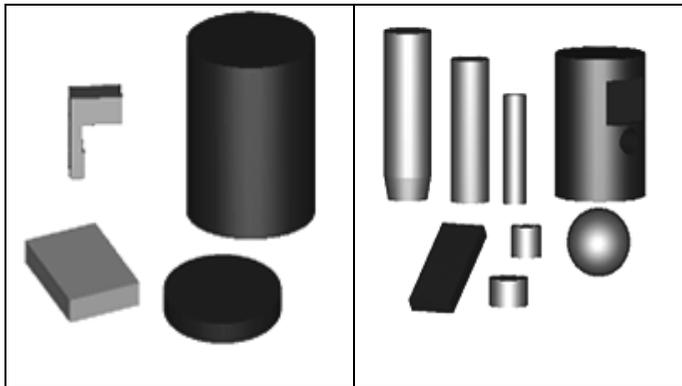


Fig. 6(b). Elements of robot gripper and screw driver model

Prior to simulating the robot movement and validating the simulation created in Workspace5, the actual robot movements first have to be specified and recorded. The cycle time of the actual robot completing time of certain tasks then has to be defined and compared with the cycle time of model simulation.

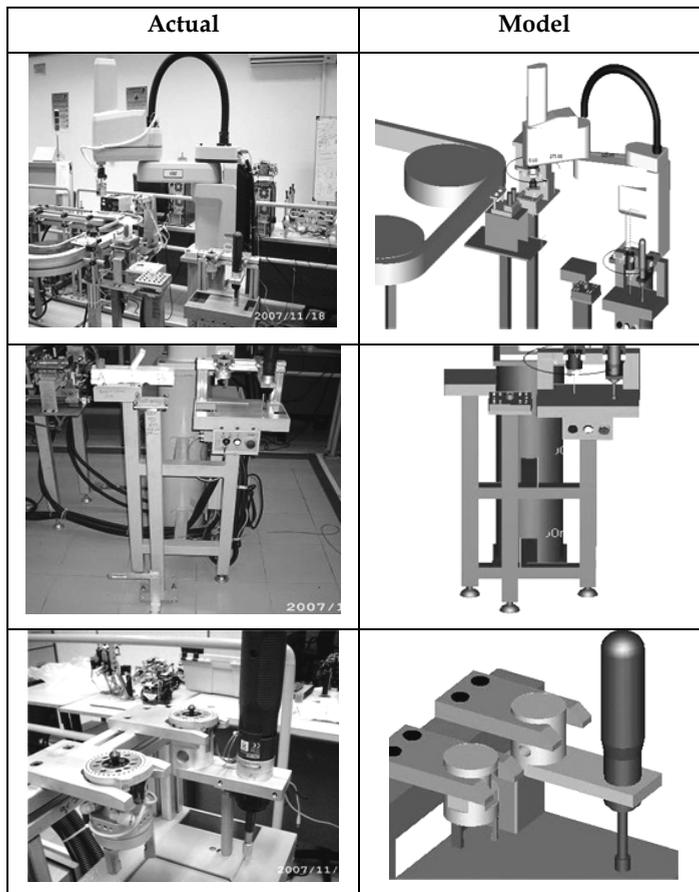


Fig. 7. Visual validation

Another project-work is reported in Nepal, R., and Baral, M. (2004), which is located in St. Cloud State University. Figures 5(a) and (b) show the development of the vacuum gripper attached to the Kawasaki 06L robot.

At the end of this project, the simulation is ready to grasp the object as depicted in Figures 6(a) and (b). When the cell reaches the bottom of the sooth, the robot grasps the object by its vacuum gripper and un-grasps the cell on the table, and moves back to its home position. Similarly, the remaining seven cells slide down the sooth in sequence and the robot picks and arranges the cells into a block on the table. The time taken for the complete simulation is 123.40 sec. There is no collision detected during the simulation.

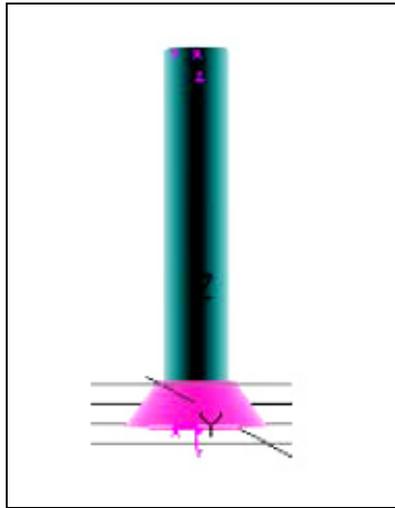


Fig. 8(a). Model of vacuum gripper

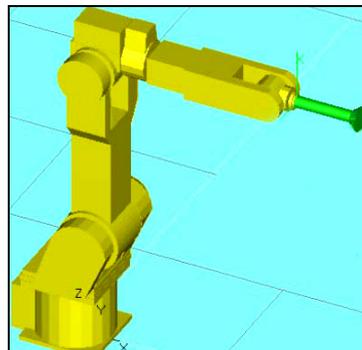


Fig. 8(b). Vacuum gripper is attached to Kawasaki 06L

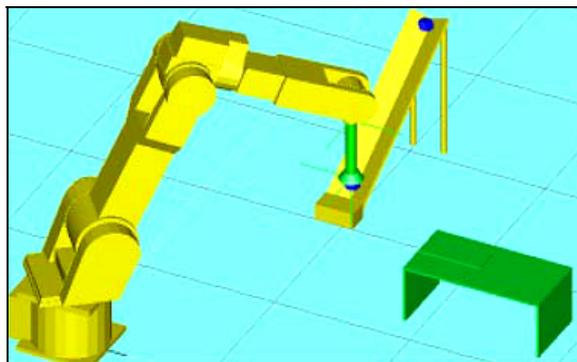


Fig. 9(a). Robot-picking cell

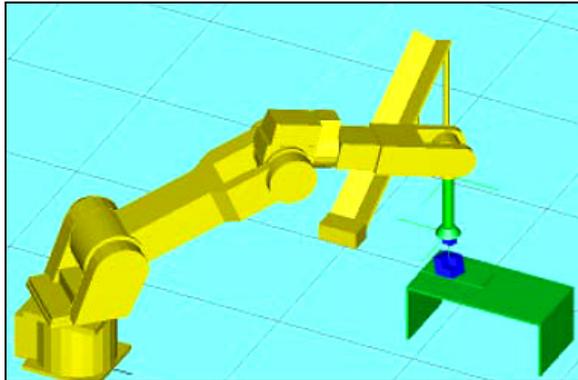


Fig. 9(b). Robot placing cells on the table

4.2 Robot Simulation Using X3D for E-Learning

This section will show the initial results based on the authors’ experience in developing the X3D model. Figure 4 shows the development of a virtual robot arm using the X3D programming written in X3D Edit 3.2 software. The X3D programming is similar to XML programming. Figure 5 shows some of the development programming.

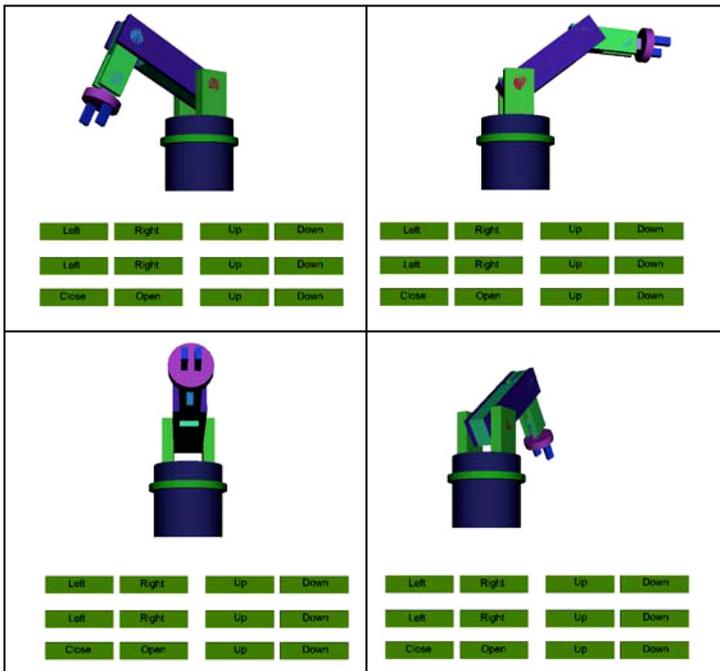


Fig. 10. Example movement of robotic simulation

At the end of this project, the virtual robot arm simulation is ready to capture the point of each movement and is also expected to generate the code based on Melfa Basic. To validate the robot simulation, the generated Melfa Basic code from the virtual robot simulation will be tested in the real environment of Melfa Basic Software and executed to the real robot. The virtual environment can also perhaps be simulated based on the input of the Melfa Basic code.

The virtual robot simulation will be embedded into an Internet web-server on a high-end server, and will be managed by content management tools. This phase also includes reliability and security testing. A simulation is run using the virtual tech pendant in order to visualize the arm movement using the client computer through web based. To simulate the virtual robot arm, we have to install a plug-in for the web browser such as Octaga Player or Cortona 3D. Different browsers will be used to make sure that the system is compatible with the browser to simulate the virtual robotic simulation.

The system can hopefully give students the realistic experience of simulation and modelling using this virtual robot arm through the e-learning portal. The information can be accessed simultaneously by users and they would not have to wait to seek the virtual robot arm simulation as this can be achieved by many users at the same time.

```

<TimeSensor DEF="TimerKanan" cycleInterval="5" loop="false"/>
  <OrientationInterpolator DEF="MuterKanan"
    key =      "0, 0.05, 0.1, 0.15,
                0.2, 0.25, 0.3, 0.35,
                0.4, 0.45, 0.5, 0.55,
                0.6, 0.65, 0.7, 0.75,
                0.8, 0.85, 0.9, 0.95,
                1"
    keyValue = "0 0 1 0,
                0 1 0 0.314159,
                0 1 0 0.628318,
                0 1 0 0.942478,
                0 1 0 1.25664,
                0 1 0 1.5708,
                0 1 0 1.88496,
                0 1 0 2.19911,
                0 1 0 2.51327,
                0 1 0 2.82743,
                0 1 0 3.14159,
                0 1 0 3.45575,
                0 1 0 3.76991,
                0 1 0 4.08407,
                0 1 0 4.39823,
                0 1 0 4.71239,
                0 1 0 5.02655,
                0 1 0 5.34071,
                0 1 0 5.65487,
                0 1 0 5.96903,
                0 0 1 0 "/>
<TimeSensor DEF="TimerKiri" cycleInterval="5" loop="false"/>

```

Fig. 11. Example movement robotic simulation programming

5. Discussion

The system development of virtual robotics simulation will be the alternative for the robotics learning process. This project will effort the virtual robotics simulation that will combine all the information about the robot arm. This system can also hopefully be the training place for students to simulate the robot's movement virtually and remotely for learning purposes.

6. References

- Abdul Rahim, N. (2008), Modelling and Simulation of FARA Robot RSM7 Movement and Its Environment (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Ariffin, N. S. (2008), Modelling and Simulation of SCARA Adept Cobra i600 Robot Arm Movement and Its Environment (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Arifin, S. M. (2007), Modelling and Simulation of Mitsubishi RV-2AJ Robot Arm Movement (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Bien, C. (1998) Simulation A Necessity In Safety Engineering. *Robot World*, Vol.10, No.4, pp.22-27.
- Cheng, F. S. (2000). A Methodology for Developing Robotic Workcell Simulation Models. *Proceedings of the 2000 Winter Simulation Conference*.
- Eric Colon and Kristel Verbiest (2008). 3D Mission Oriented Simulation. Royal Military School.
- Farrington, P.A., Nembhard, H.B., Sturrock, D. T. and Evans, G. W. eds. (1999). Increasing the Power and Value of Manufacturing Simulation Via Collaboration with Other Analytical Tools: A Panel Discussion. *Proceedings of the 1999 Winter Simulation Conference*.
- F.E Cellier. (2006). *Continuous System Simulation*. Argentina: Springer Science Business Media.
- Grajo, E. S., Gunal, A., SathyaDev, D. And Ulgen, O.M. (1994). A Uniform Methodology for Discrete-event and Robotic Simulation. *Proceeding of the Deneb Users Group Meeting*. Deneb Robotic, Inc. 17-24.
- Kin-Hua Low. (2008) *Industrial Robotics: Programming, Simulation And Applications: Germany*, Advanced Robotics Systems International.
- Michel, O/ Cyberbotics Ltd (2004). *Webot: Professional Mobile Simulation Robot*. *International Journal of Advance Robotic System*. Volume 1, Number 1.
- Mohd Johari, N. A. and Haron, H. (), *Robotic Modeling and Simulation of Palletizer Robot Using Workspace5*, Master Thesis, Universiti Teknologi Malaysia
- Mohd Salih, N. H. (2008), Modelling and Simulation of Adept Viper S650 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Muhammad Ikhwan Jambak, Habibollah Haron, Dewi Nasien. (2008) *Development of Robot Simulation Software For Five Joints Mitsubishi RV-2AJ Robot Using MATLAB/Simulink And V-Realm Builder*. *Fifth International Conference on Computer Graphics, Imaging And Visualization*.

- Muhammad Noor, N. F. (2005), Mitsubishi RV-2AJ Robot Arm Basic Movement Simulation Using Workspace 5 Software (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Nathan Smith, Cristopher Egert, Elisabeth Cuddihy, Deborah Walters (2006). Implementing Virtual Robots in Java3D Using a Sudsumption Architecture. Proceedings from the Association for the Advancement of Computing in Education.
- Nepal, R., and Baral, M. (2004). Simulation of Kawasaki 06L Robot in Workspace 5.0. Bachelor Thesis. St. Cloud State University.
- R.D. Kriz, D. Farkas, A.A. Ray, J.T. Kelso, and R.E. Flanery, Jr. (), Visual Interpretation and Analysis of HPC Nanostructure Models using Shared Virtual Environments, Conference Proceedings, High Performance Computing: Grand Challenges in Computer Simulations 2003, The Society for Modeling and Simulation International (SCS), San Diego, California.
- Robinson, P. (1996). Robotics Education and Training: A Strategy for Development. *Industrial Robot*. 23(2): 4-6.
- Robotic Simulation (2006), KUKA Robotic Corporation.
- Roth. N. (1999). *The International Journal of Robotics Research*. On the Kinematic Analysis of Robotic Mechanisms. 18(12): 1147-1160.
- Shafei, S. A. (2008), Modelling and Simulation SCORBOT-ER 4u Robot Arm Movement Using Workspace5 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Shannon, Robert E. (1998). Introduction to the art and science of simulation, Proceedings of the 1998 Winter Simulation Conference.
- Sukimin, Z. (2007) Design, Visualization and Simulation of Automatic Chopping Process (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Zainal Abidin, M. A. (2008), Modelling and Simulation Kawasaki FS03N Robot Arm Movement Using Workspace5 (in Malay), Bachelor Thesis, Universiti Teknologi Malaysia.
- Zomaya, A. Y. (1992). Modeling and Simulation of Robot Manipulators: A Parallel Processing Approach. Singapore: World Scientific Publishing Co. Pte. Ltd.

Robot Simulation for Control Design

Leon Žlajpah
Jožef Stefan Institute
Slovenia

Abstract

Research in the field of robotics is tightly connected to simulation tools for many reasons. On one side, simulation supports the development of new advanced control algorithms and on the other side, it is always not feasible to build a whole robot system to test some algorithms or it is not safe to perform tests on a real system (at least in the first design stages). The simulation has also a very important role for off-line programming, to design mechanical structure of robots, to design robotic cells and production lines, etc.

In the paper, an overview of the simulation in robotics is given and some topics like: how simulation makes things easier, advantages and backdraws of the simulation in robotics, virtual and real world, are pointed out. The scope of the paper is the role of the simulation in different fields of robotics, especially the dynamic simulation of robot manipulators. We present an integrated environment for the design and testing of advanced robot control schemes. The main capabilities of such environment are: the simulation of the kinematics and dynamics of manipulators, the integration of different sensor systems like vision and force sensors, scenarios for complex robot tasks, the visualization of robots and their environment and the integration of real robots in the simulation loop. We give an overview of simulation and visualization tools suitable for the simulation of robot systems using general dynamic engines and graphic languages. Finally, we present some typical simulation examples in different fields of robotics from offline programming, mobile robots to space robotics.

1. Introduction

Simulation has been recognized as an important research tool since the beginning of the 20th century. In the beginning, simulation was first of all an academic research tool. The "good times" for simulation started with the development of computers. First, the analog computers and later the digital computers have boosted simulation to new levels. So, the simulation is now a powerful tool supporting the design, planning, analysis, and decisions in different areas of research and development. Simulation has become a strategic tool in many fields, used by many researchers, developers and by many manufacturers. Of course, robotics as a modern technological branch is no exception. Actually, in robotics simulation plays a very important role, perhaps more important than in many other fields and we like to present in the following some insight in the robotics from the simulation point of view.

1.1 The role of simulation

Being able to simulate opens a wide range of options for solving many problems creatively. You can investigate, design, visualize, and test an object or even if it does not exist. You can

see the results of a system yet to be built. It is possible that your solutions may fail or even blow up, but only in simulation. So, using the simulation tools one can avoid injuries and damages, unnecessary changes in design after the production of parts has already started, to long cycle times in manufacturing process, and even unnecessary paper work. Simulation enables us to work even in four dimensions. For example, one can observe within a few minutes how a planned production will be realized in next month, or a fast process can be slowed down to observe all details in "slow motion". All these make things easier and cheaper. One of the problems in classical design and planning are "what-if" questions. Due to the system complexity many of them are often unasked or not answered. With up-to-date simulation tools one can deal with exact geometry, consider the dynamic characteristics of a system, include the man-machine interfaces, and visualize the object in 3D in detail. Having all these in mind there is no reason for avoiding any "what-if" question. The boundaries for what is possible or not are pushed far away especially in advanced virtual reality tools. Using simulator researchers may build experimental environments according to their own imagination. Complexity, reality, specificity can be gradually increased to a level where virtual systems can head to real challenges of the physical world and even beyond.

Simulation is a highly interdisciplinary field since it is widely used in all fields of research from engineering and computer science to economics and social science, and at different levels from academic research to manufactures. Of course, simulation has been also recognized as an important tool in robotics: in designing new products, investigating its performances and in designing applications of these products. Simulation allows us to study the structure, characteristics and the function of a robot system at different levels of details each posing different requirements for the simulation tools. As the complexity of the system under investigation increases the role of the simulation becomes more and more important.

2. Simulation of robot manipulators

The ways and methods in robotics research and development have always been influenced by the tools used. This is especially true when one considers the profound impact of recent technologies on robotics, especially the development of computers which have become indispensable when designing the complex systems like robots. Not many years ago, computing cost was still a significant factor to consider when deriving algorithms and new modeling techniques (Fenton & Xi, 1994; Latombe, 1995; Zhang & Paul, 1988). Nowadays, distributed computing, network technology and the computing power developed by commercial equipment open new possibilities for doing systems design and implementation. However, in spite of all that, the creativity of a human designer can not be left out in the design process. The best solution seems to be to provide the designer with proper tools which significantly increase his efficiency. Among them, the simulation has been recognized as an important tool in designing the new products, investigating their performances and also in designing applications of these products. For complex systems as robots, the simulation tools can certainly enhance the design, development, and even the operation of the robotic systems. Augmenting the simulation with visualization tools and interfaces, one can simulate the operation of the robotic systems in a very realistic way.

A large amount of simulation software is available for robot systems, and it is already being used extensively. The majority of the robot simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems they all include the kinematic or dynamic models of robot manipulators. Which type of models will be used depends on the objective of the simulation system. For

example, trajectory planning algorithms rely on kinematic models. Similarly, the construction of a robotized cell can be simulated efficiently by using only kinematic models of robot manipulators, without considering the dynamics or drives. On the other hand, dynamic models are needed to design the actuators. For example, modern control systems of robotic manipulators use internally different robot kinematic and dynamic models to improve the performance.

To model and simulate a robot manipulator different approaches are possible. They can differ in the way the user builds the model. Block diagram oriented simulation software requires that the user describes the system by combining the blocks, and there are other packages requiring the manual coding. To overcome the problems which arise when the system is very complex (and the robots usually are) several approaches exist to automatically generate the kinematic and/or dynamic models of robots.

The simulation tools for robotic systems can be divided into two major groups: the tools based on general simulation systems and special tools for robot systems. The tools based on general simulation systems are usually special modules, libraries or user interfaces which simplify the building of robot systems and environments within these general simulation systems. One of the advantages of such integrated toolboxes is that they enable you to use other tools available in the simulation system to perform different tasks. For example, to design control system, to analyse simulation results, to visualize results, etc. There exist several general simulation tools which are used for simulation of robot systems like MATLAB/Simulink, Dymola/Modelica, 20-sim, Mathematica, etc. Special simulation tools for robots cover one or more tasks in robotics like off-line programming, design of robot work cells, kinematic and dynamic analysis, mechanical design. They can be specialized for special types of robots like mobile robots, underwater robots, parallel mechanisms, or they are assigned to predefined robot family.

Simulation tools for robotic systems differ from each other regarding the aspect of the robot research they support, how open they are or on which platforms they work. However, many tools are not always fulfilling all the requirements of the research activities in robotic laboratories like reconfigurability, openness and ease of use, etc.

Reconfigurability and openness are features already recognized by many as essential in the development of advanced robot control algorithms (Alotto et al., 2004; Lambert et al., 2001; Lippiello et al., 2007). Not only is it important to have easy access to the system at all levels (e.g. from high-level supervisory control all the way down to fast servo loops at the lowest level), but it is a necessity to have open control architectures where software modules can be modified and exteroceptive sensors like force/torque sensors and vision systems can be easily integrated. Reconfigurability should also be reflected when more fundamental changes to the controller architecture are required, in the necessity of quickly being able to make modifications in the original design and verify the effect of these modifications on the system. In other words, the user should be able to quickly modify the structure of the control without having to alter the simulation system itself.

In the last decade the software has become more and more easy to use. This is still one of the main major issues when selecting a software tool. First of all, the tools are used by many users in a laboratory and not all of them have the same expertise. To boost the knowledge exchange, it is of benefit that they work with the same tools. Next, testing of different control algorithms on real robotic systems is in general not very user friendly: the algorithms usually have to be rewritten for the real-time execution and the different implementation details have to be considered (Lambert et al., 2001; Žlajpah, 2001). This forces the user to devote a large part of the design time to topics not connected with the main issues of the control de-

sign, especially when he is not interested in software implementation issues. The ease of use becomes even more important when students are working with robots. In most cases they work in a laboratory for a shorter period, they are focused on their projects and they could become frustrated if they have to learn a lot of things not directly connected to their tasks. Finally, in research laboratories different robot systems are used equipped with more or less open proprietary hardware and software architecture. Therefore, it is much desired that the control design environment is unified, i.e. the same tools can be used for all robot systems.

The simulation tools for robotic systems can be divided into two major groups: tools based on general simulation systems and special tools for robot systems. Tools based on general simulation systems are usually represented as special modules, libraries or user interfaces which simplify the building of robot systems and environments within these general simulation systems (e.g. SolidWorks (RobotWorks, 2008)). On the other hand, special simulation tools for robots cover one or more tasks in robotics like off-line programming and design of robot work cells (e.g. Robcad (RobCAD, 1988)) or kinematic and dynamic analysis (Corke, 1996; SimMechanics, 2005). They can be specialized for special types of robots like mobile robots, underwater robots, parallel mechanisms, or they are assigned to predefined robot family. Depending on the particular application different structural attributes and functional parameters have to be modelled.

For the use in research laboratories, robot simulation tools focused on the motion of the robotic manipulator in different environments are important, especially those for the design of robot control systems (Corke, 1996; MSRS, 2008; SimMechanics, 2005; Webots, 2005). Recently, Microsoft Robotics Studio (MSRS, 2008) has been launched with a general aim to unify robot programming for hobbyist, academic and commercial developers and to create robot applications for a variety of hardware platforms. The system enables both remotely connected and robot-based scenarios using .NET and XML protocols. The simulation engine enables real-time physics simulation and interaction between simulated entities. Each part of the control loop can be substituted with the real or simulated hardware. Although the system is still under development, it is not easy to add new entity, for example a new robot or a new sensor. One of the major drawbacks seems to be the low data throughput rate, which does not allow the realization of complex control laws at high sampling frequency. Therefore, it is not clear yet if MSRS is appropriate for research robotics, especially for complex systems. Real time requirements are better solved in another programming/simulation framework, MCA2 (MCA2, 2008). MCA is a modular, network transparent and realtime capable C/C++ framework for controlling robots and other hardware. The main platform is Linux/RTLinux, but the support for Win32 and MCA OS/X also exists. However, it is still a complex system and therefore less appropriate for education and students projects.

2.1 MATLAB based tools

MATLAB is definitely one of the most used platforms for the modelling and simulation of various kind of systems and it is not surprising that it has been used intensively for the simulation of robotics systems. Among others the main reasons for that are its capabilities of solving problems with matrix formulations and easy extensibility. As an extension to MATLAB, SIMULINK adds many features for easier simulation of dynamic systems, e.g. graphical model and the possibility to simulate in real-time. Among special toolboxes that have been developed for MATLAB we have selected four: (a) Planar Manipulators Toolbox (Zlajpah, 1997), (b) Planar Manipulators Toolbox with SD/FAST (SD/FAST, 1994), (c) "A Robotic

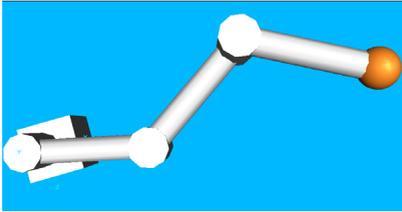


Fig. 1. Simple 3-R planar manipulator

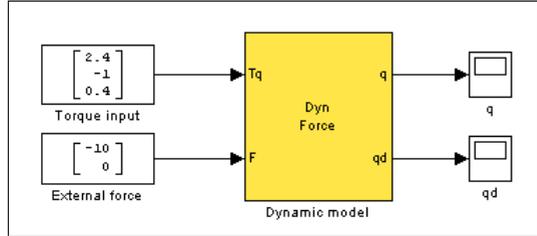


Fig. 2. Top level block scheme

Toolbox” (Corke, 1996), (d) “SimMechanics Toolbox” (SimMechanics, 2005) and (e) “20-sim” (Kleijn, 2009).

To illustrate different approaches to the dynamic simulation of robot manipulators we have selected as an object a simple planar manipulator which has 3 revolute joints acting in a plane as shown on Fig. 1. The main part of any simulation is the dynamic model. To focus on it, we simulate only the dynamics, without any task controller.

Let the configuration of the manipulator be represented by the vector q of n joint positions, and the end-effector position (and orientation) by m -dimensional vector x of task positions. The joint and task coordinates are related by the following expressions

$$x = p(q), \quad \dot{x} = J(q)\dot{q}, \quad \ddot{x} = J\ddot{q} + \dot{J}\dot{q} \tag{1}$$

where J is the Jacobian matrix, and the overall dynamic behaviour of the manipulator is described by the following equation

$$\tau = H(q)\ddot{q} + h(\dot{q}, q) + g(q) - \tau_F \tag{2}$$

where τ is the vector of control torques, H is the symmetric positive-definite inertia matrix, h is the vector of Coriolis and centrifugal forces, g is the vector of gravity forces, and vector τ_F represents the torques due to the external forces acting on the manipulator.

Fig. 2 shows the top level block scheme of the system. This scheme is the same in all cases, only the *Dynamic model* block is changed.

(a) Planar Manipulators Toolbox

Planar Manipulators Toolbox is intended for the simulation of planar manipulators with revolute joints and is based on Lagrangian formulation. Planar Manipulators Toolbox can be used to study kinematics and dynamics, to design control algorithms, for trajectory planning. It enables also real time simulation. Due to its concept it is a very good tool for education. To gain the transparency, special blocks have been developed to calculate the kinematic and dynamic models. These blocks are then used to build the desired model. Fig. 3 shows the dynamic model where an external force acts on the end-effector. The block *dymodall* which calculates the system vectors and matrices x, J, \dot{J}, H, h and g and then joint accelerations are calculated using Lagrangian equation.

(b) Planar Manipulators Toolbox with SD/FAST

In this case we use Planar Manipulators Toolbox but the dynamic model is calculated SD/FAST library. SD/FAST can be used to perform analysis and design studies on any mechanical system which can be modelled as a set of rigid bodies interconnected by joints, influenced by forces, driven by prescribed motions, and restricted by constraints (SD/FAST, 1994). The dynamic model has the same structure as given in Fig. 3 except that the block *dymodall*

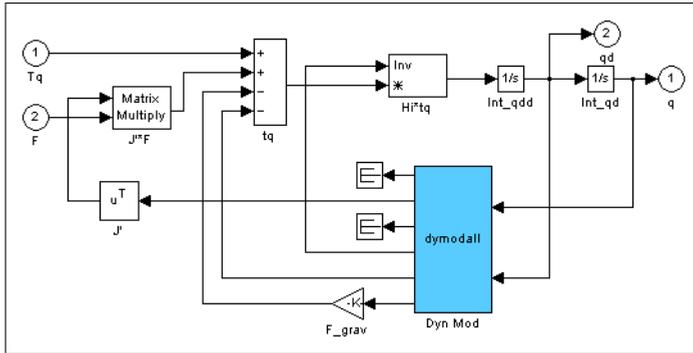


Fig. 3. Dynamic model (Planar Manipulators Toolbox)

is now a special S-function interfacing SD/FAST procedures and Simulink. The robot kinematics (geometry) and link mass properties are passed to SD/FAST in the System Description file (Fig.4). Then using the SD/FAST compiler the dynamic model is generated which is then called in S-function. To calculate the dynamics SD/FAST uses the advanced Kane's formulation and Order(n) formulation.

```
# model of a planar manipulator with 4dof
language = c
gravity = 0 -9.81 0
#link1
  body = link1 inb = $ground
  joint = pin prescribed = ?
  mass = 1 inertia = 0 0 1
  bodytojoint = 0.5 0 0
  inbtojoint = 0.5 0 0
  pin = 0 0 1
#link2
  body = link2 inb = link1
  joint = pin prescribed = ?
  mass = 1 inertia = 0 0 1
  bodytojoint = 0.5 0 0
  inbtojoint = 0.5 0 0
  pin = 0 0 1
#link3
  body = link3 inb = link2
  joint = pin prescribed = ?
  mass = 1 inertia = 0 0 1
  bodytojoint = 0.5 0 0
  inbtojoint = 0.5 0 0
  pin = 0 0 1
```

Fig. 4. System Description file for 3R planar manipulator (SDFAST)

(c) Robotics Toolbox

The Robotics Toolbox provides many functions that are required in robotics and addresses areas such as kinematics, dynamics, and trajectory generation. The Toolbox is useful for the simulation as well as for analysing the results from experiments with real robots, and can be a powerful tool for education. The Toolbox is based on a general method of representing the kinematics and dynamics of serial-link manipulators by description matrices. The inverse dynamics is calculated using the recursive Newton-Euler formulation. Although it was initially meant to be used with MATLAB, it can be also used with Simulink. Fig. 5 shows the definition of the robot model and the block scheme of the dynamic model using Robotics Toolbox.

(d) SimMechanics Toolbox

SimMechanics extends Simulink with the tools for modelling and simulating mechanical systems. With SimMechanics, you can model and simulate mechanical systems with a suite of tools to specify bodies and their mass properties, their possible motions, kinematic constraints,

```

%% Definition of the R3 planar robot
for i=1:nj
    LR{i}=link([0 L(i) 0 0 0],'standard');
    LR{i}.m=m(i);
    LR{i}.r=[-Lc(i),0,0];
    LR{i}.I=[1 1 1 0 0 0]*II(i);
    LR{i}.Jm=0;
    LR{i}.G=1;
    LR{i}.B=Bv(i);
    LR{i}.Tc=[0 0];
end R3=robot(LR);
R3.name='R3';
R3.gravity=[0 9.81 0];
R3.q=q0';
    
```

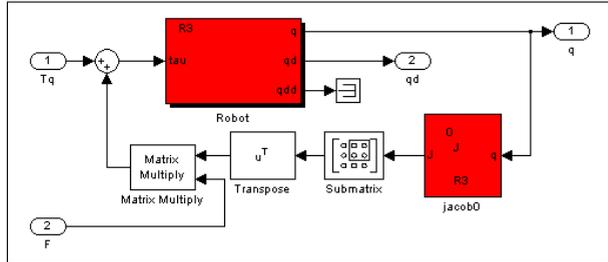


Fig. 5. Dynamic model (Robotics toolbox)

and coordinate systems, and to initiate and measure body motions (SimMechanics, 2005). To get a dynamic model of a robot manipulator we have first to build the link model, i.e. to connect link masses with joints as it is shown on Fig. 7. All link models are then connected together to the complete model (Fig. 6).

(e) 20-sim

Although 20-sim is a stand-alone simulation system (described later), it has a possibility to export the model to Simulink blocks as C-mex function. For comparison, we have modelled our robot manipulator using the 3D Mechanic Editor where you can model mechanical systems by specifying bodies, joints, sensors and actuators (Kleijn, 2009). To get a dynamic model of a robot manipulator we have first defined the links and then we have connected links with joints as it is shown on Fig. 8. Adding the trajectories generator, controllers and power amplifiers with gears a complete model of the system can be built (Fig. 9). Using the C code generator in 20-sim we have generated a Simulink block of the manipulator subsystem (R3). This block is then used in Simulink simulation scheme as shown in Fig. 2.

In all five cases it has been very easy to build the robot system. One of the differences between these tools is that special toolboxes for robot modelling have predefined more specific functions and blocks as the general toolboxes. The other difference is the execution time. In Fig. 10 we give the calculation time for the dynamic model for all five approaches. First we can see that SD/FAST is significantly faster than other and is increasing more slowly versus the

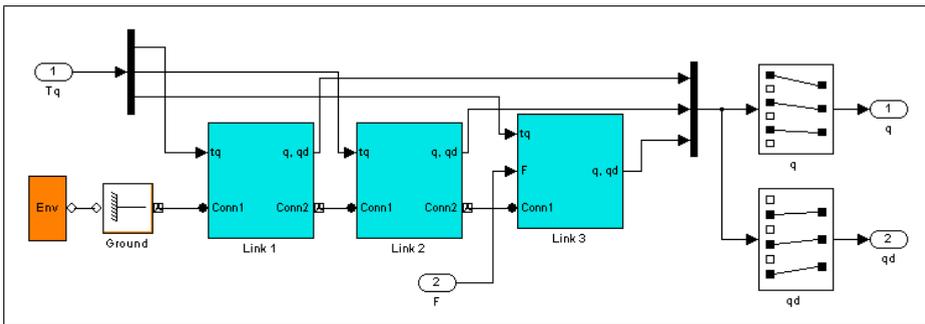


Fig. 6. Dynamic model of 3R manipulator (SimMechanics toolbox)

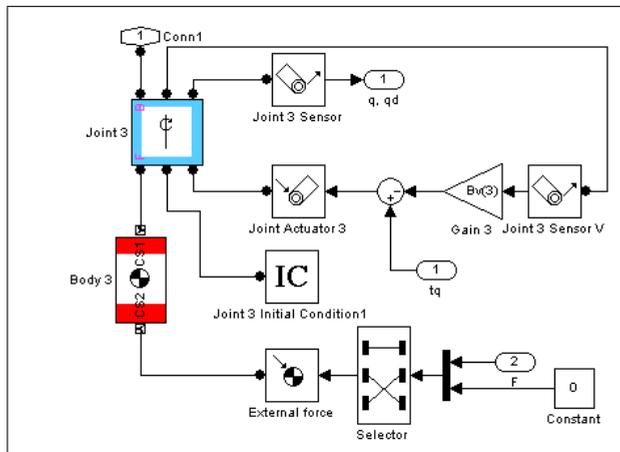


Fig. 7. Model of one link (SimMechanics toolbox)

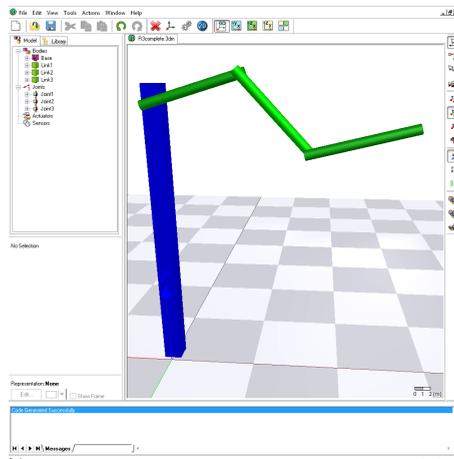


Fig. 8. Modelling robot manipulator using 20-sim 3D Mechanic Editor

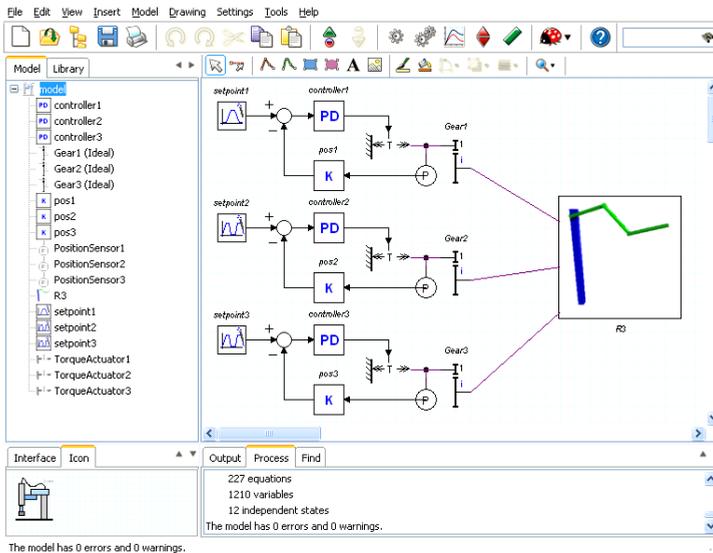


Fig. 9. Complete model of 3R manipulator (20-sim)

degrees-of-freedom than other. Next, Planar Manipulators Toolbox is fast for small number of degrees-of-freedom and the execution time increases fast with the number of degrees-of-freedom. The Robotics Toolbox is relatively fast as long as we use only the inverse dynamics (Note that in Fig. 10 only the calculation time for the inverse dynamic model is shown). Otherwise, e.g. for the calculation of the Jacobian matrix, it is significantly slower, because the calculation is based on M-functions. Also, the model generated in 20-sim is fast (simulation within 20-sim environment is even faster). A little slower is the SimMechanics Toolbox. In both cases the execution time versus the number of degrees-of-freedom increases similarly. However, if the models of robot manipulators should be used in the controller (e.g. the Jacobian matrix), then SimMechanics Toolbox and 20-sim are not appropriate.

2.2 Other general simulation systems

Similarly as in MATLAB the robot system can be simulated in Dymola and Modelica, or 20-sim. Here, the MultiBody library provides 3-dimensional mechanical components to model rigid multibody systems, such as robots. The robot system is built by connecting the blocks representing parts of the robot like link bodies, joints, actuators, etc. Fig. 11 shows the block scheme of a complete model of the KUKA robot including actuators, gears and the controller (Kazi & Merk, 2002). Fig. 12 shows the simulation of a parallel robot manipulator with 20-sim (3D Mechanics Toolbox) (Kleijn, 2009).

Robotica is a computer aided design package for robotic manipulators based on Mathematica (Nethery & Spong, 1994). It encapsulates many functions into a Mathematica package allowing efficient symbolic and numeric calculation of kinematic and dynamic equations for multi-degree-of-freedom manipulators. Robotica is intended, first of all, for model generation and analysis of robotic systems and for simulation.

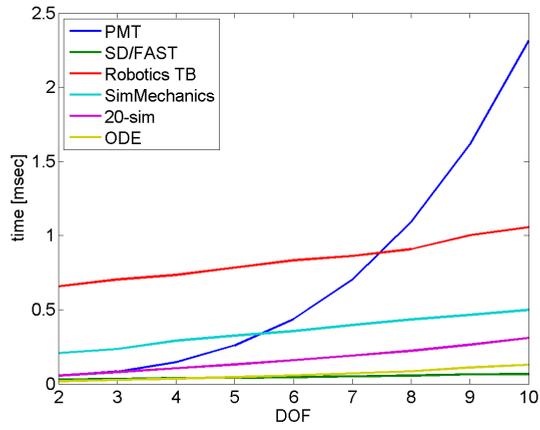


Fig. 10. Comparison of the calculation time versus number of DOF for the dynamic model of n -R planar robot manipulator

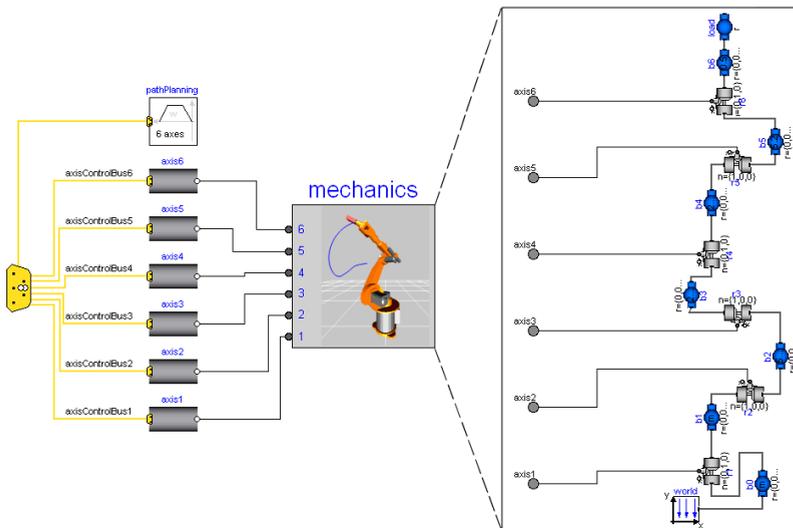


Fig. 11. Simulation of a robot with Modelica

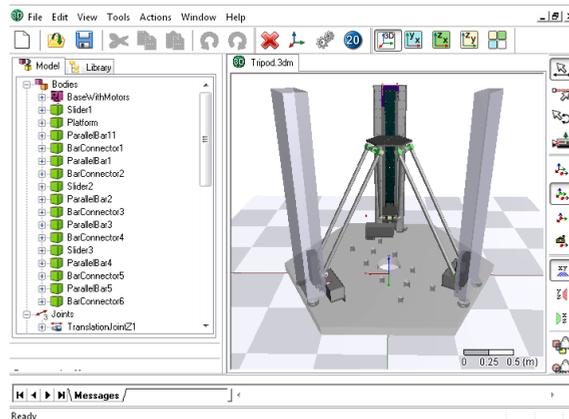


Fig. 12. Simulation of a Tripod with 20-sim 3D Mechanics Toolbox

2.3 Multibody dynamic engines

In the last years new simulation tools have been available based on the general engines for the simulation of physics environments (NGD, 2008; ODE, 1994; SD/FAST, 1994). These engines provide libraries for simulating the multi-body dynamics, i.e. the physics of motion of an assembly of constrained or restrained bodies. As such they encompass the behaviour of nearly every object and among them are, of course, also robot manipulators. These dynamic engines have beside the dynamics simulation engine also a collision detection engine. The collision engine is given information about the shape of each body and then it figures out which bodies touch each other and passes the resulting contact point information to the user. The user can then take the proper actions.

As an example we have selected the Open Dynamic Engine (ODE, 1994). Building the model of a robot is straightforward. First you have to create all bodies and connect them if desired with proper joints. For example, the 3DOF model as shown before can be defined as shown in Fig. 13. For comparison with MATLAB based tools the computational time for nR planar manipulators ($n=2, \dots, 10$) is shown in Fig. 10. It can be seen that the computational efficiency of ODE is comparable to the SD/FAST library.

Unfortunately, most of dynamics engines do not support functionality necessary to include robot models in the control algorithms. Advanced control algorithms including robot models include Jacobian matrices, inertia matrices, gravity forces, etc., and they are not explicitly defined. The user can use some implicit algorithms or other tools to get these parameters.

The dynamic simulation of multibody systems becomes very important when introducing robotics into human environments (Go et al., 2004; Khatib et al., 2002; Miller & Christensen, 2003) where the success will not depend only on the capabilities of the real robots but also on the simulation of such systems. For example, in applications like virtual prototyping, teleoperation, training, collaborative work, and games, physical models are simulated and interacted with both human users and robots.

For example, the dynamics engine within a robotic grasping simulator known as GraspIt! (Miller & Allen, 2004) computes the motions of a group of connected robot elements, such as an arm and a hand, under the influence of controlled motor forces, joint constraint forces, contact forces and external forces. This allows a user to dynamically simulate an entire grasping

```

// create world
contactgroup.create (0);
world.setGravity (9.81,0,0);
dWorldSetCFM (world.id(),1e-5);
dPlane plane(space,0,0,1,0);

// fixed robot base
xbody[0].create(world);
xbody[0].setPosition(0,0,SIDE/2);
box[0].create(space,SIDE,SIDE,SIDE);
box[0].setBody(xbody[0]);
bjoint = dJointCreateFixed (world,0);
dJointAttach (bjoint,xbody[0],0);
dJointSetFixed (bjoint);
// robot links
for (i=1; i<=NUM; i++) {
    xbody[i].create(world);
    xbody[i].setPosition(0,(i-0.5)*LENG,(i-0.5)*SIDE);
    m.setBox(1,SIDE,LENG,SIDE);
    m.adjust(MASS);
    xbody[i].setMass(&m);
    box[i].create(space,SIDE,LENG,SIDE);
    box[i].setBody(xbody[i]);
}
// robot joints
for (i=0; i<NUM; i++) {
    joint[i].create(world);
    joint[i].attach(xbody[i],xbody[i+1]);
    joint[i].setAnchor(0,(i)*LENG,(i+1)*SIDE);
    joint[i].setAxis(0,0,1);
}

```

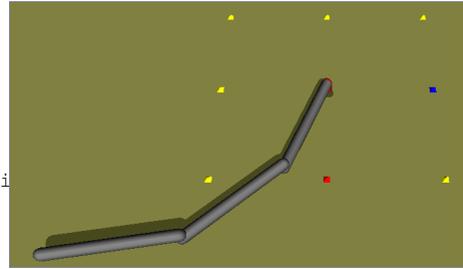


Fig. 13. Definition of 3R planar manipulator in ODE

task, as well as test custom robot control algorithms. Fig. 14 shows how a robot hand can grasp a mug (Miller & Allen, 2004). In this example, all contacts between the fingers and the mug and related forces are analysed.

3. Control design and integrated environment

A very important part of the robotic system is the control system. In the process of controller design different steps have to be performed. First of all, the system has to be modelled. In the next step, the control algorithm is developed. The first results are then obtained by the simulation. If the results are satisfactory, then in the final stage the control algorithms are tested on a real system. For this, a real-time code should be generated and implemented on the real system. The integration of all these steps, although essential, is very difficult. Namely, the different steps in the development of the controller require the use of different methods for which different tools are needed. Hence, the results from one step to another have to be transferred often by hand. This bottleneck can be overcome if control design and testing are done in an integrated environment.

The importance of simulation tools in the development of robot control systems has been recognized by researchers very early. We have been using different simulation tools for over 20 years and many of them have been developed in our laboratory. In the last decade we have been using for the control design MATLAB/Simulink based integrated environment based

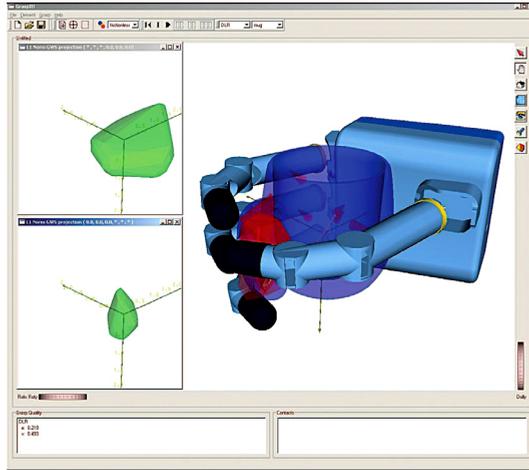


Fig. 14. A force-closure grasp of the mug using the DLR hand, which has a metal palm, inner link surfaces, and rubber fingertips.

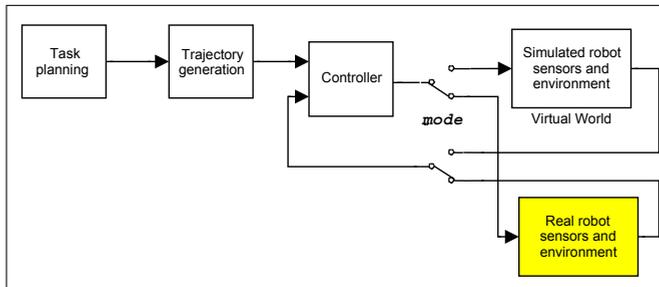


Fig. 15. A block diagram of the integrated environment

on Planar Manipulators Toolbox for dynamic simulation of redundant planar manipulators Žlajpah (2001). It enables the use of different sensors in the control loop and also the real-time implementation of the controller and hardware-in-the-loop simulation. Figure 15 shows the general simulation scheme in this environment. A crucial feature inherited in this scheme is indicated by the mode switches. Namely, the user can easy switch between using model or a real system in the simulation loop. This is one of the main features which we need for development of the robot control systems.

For example, Fig. 16 shows the dynamic model of a manipulator and a sensor detecting the object in the neighbourhood of the manipulator. When the developed controller is tested on a real system we substitute the manipulator with our experimental robot, i.e. the dashed blocks in Fig. 16 are replaced with the interface blocks as shown in Fig. 17a. Fig. 17b shows our laboratory manipulator with four revolute DOF acting in a plane, which has been developed specially for testing the different control algorithms for redundant robotic manipulators, performing an obstacle avoidance task.

The integration of the two modes is the most important feature of the integrated environment. This has been recognized also by many other researchers. For example, one of the goals of the

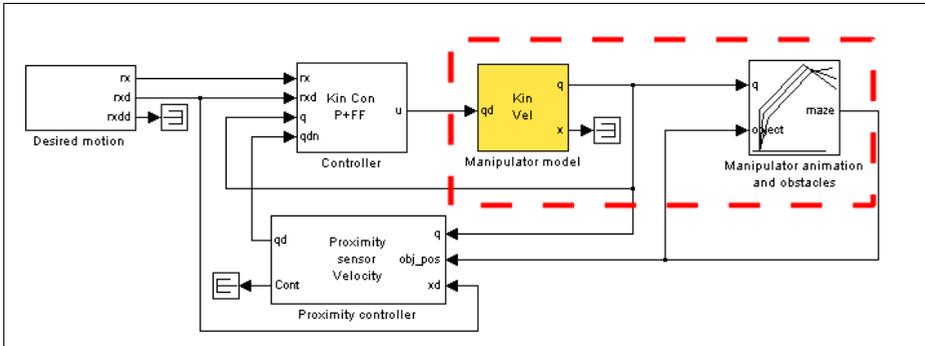
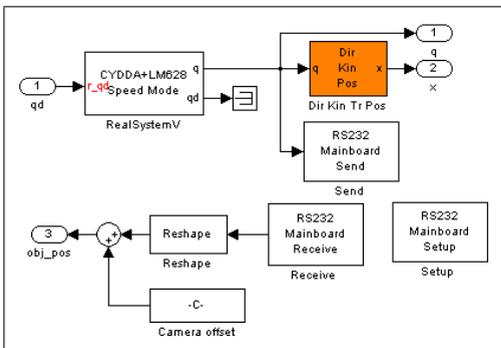


Fig. 16. A block diagram including the dynamic model of a manipulator and a sensor detecting the object in the neighbourhood of the manipulator



a) Robot interface block scheme

b) Experimental 4-R manipulator

Fig. 17. Avoiding obstacles - Hardware-in-the loop simulation

IST project "RealSim" was to develop an efficient tool for modelling, simulating and optimising industrial robots (Kazi & Merk, 2002). Fig. 18 shows the structure of the simulation system where a real control unit is connected to the simulator of an industrial robot. Using such system the controller can be tested without a real robot, e.g. before even the robot has been built.

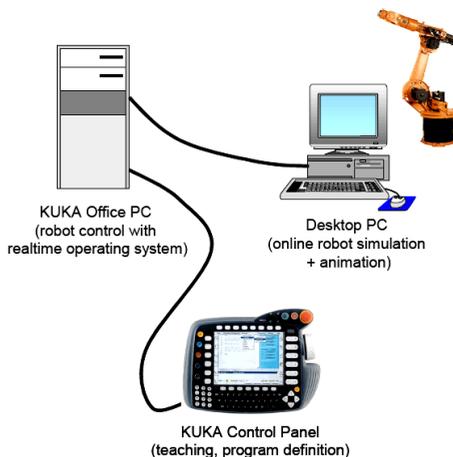


Fig. 18. Real time optimisation system (Project **RealSim** (Kazi & Merk, 2002))

3.1 The concept of distributed environment

The Planar Manipulators Toolbox has proved to be a very useful and effective tool for many purposes, but it has been primarily designed for the kinematic and dynamic simulation of the planar manipulators and to develop and test control algorithms on the lower control level, especially for redundant manipulators. In the last years, the scope of our research is oriented more in the development of control systems for humanoid and service robots (Gams et al., 2009; Omrčen et al., 2007). These robots have in general a more complex mechanical structure with many degrees-of-freedom. So, complex kinematic and dynamic models are necessary to simulate them. Furthermore, the control methods and algorithms are now usually a part of the higher robot control levels and the low level close-loop control algorithms are assumed to be a solved issue. These high level control algorithms can become very complex and may even require parallel computation distributed over more computers.

Considering all new requirements, which are:

to simulate the kinematics and dynamics of arbitrary chosen kinematic chain describing different manipulators,

to enable integration of different sensor systems like vision and force sensors,

to enable simulation of scenarios for complex robot tasks,

to include the model the robots' environments,

to visualize the robots and their environment and

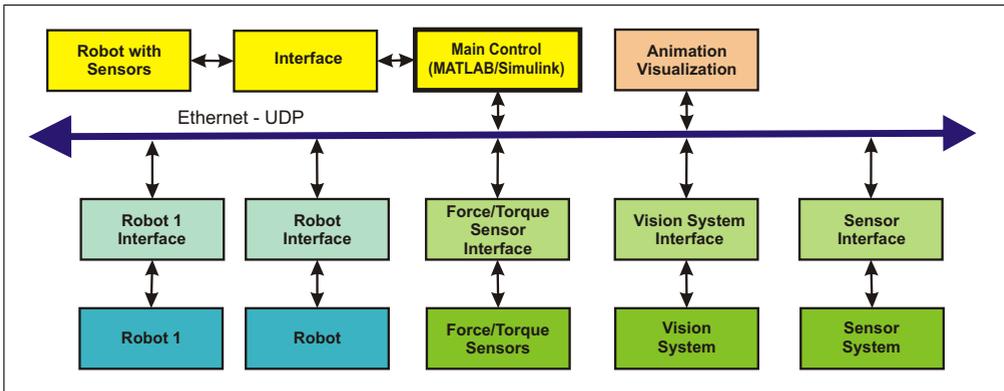


Fig. 19. A functional block diagram of the robot integrated environment in Robotics Laboratory including the robot PA10, mobile platform Nomad XR 4000 and sensor systems

to enable integration of real robots in the simulation loop, we have to reconsider the concept of the control design environment we will use in future. Based on our good experience with MATLAB/Simulink we have decided that this environment will be the kernel of our simulation tools. However, some of the above requirements can be easier fulfilled by using other tools. For example, the visualization of the robot and the environment can be easily done by dedicated graphics tools. Furthermore, advanced robot control strategies rely intensively on feedback sensor information. The most complex sensor system is the vision system, which can have several configurations and can be implemented on a single computer or on a computer cluster composed of many computers running different operating systems. To integrate such a diversity of hardware components in a unique framework we have decided to use the ethernet communication and the UDP protocol. In this way, we have maximal possible "degree-of-openness" of the system. Fig. 19 shows a typical scheme of our robot integrated environment.

In this scheme, each block can represent a real system or a model of that system. Note that because of using ethernet communication between the blocks, different software tools on different platforms can be used to simulate specific parts of the system. Consequently, the simulation environment can consist of several interacting applications, each representing a part of the system (Petrič et al., 2009).

3.2 Simulink block library

In Simulink, a system is modelled by combining input-output blocks. To gain the transparency, we try to represent a system by the block structure with several hierarchical levels, i.e. by combining different basic blocks, subsystems are built which become a single block at the higher level. In Figure 15 the typical robot subsystems can be seen: the trajectory generation, the controller, the model of the manipulator and the environment and the animation of manipulator motion. Figure 20 shows the Robot systems block library. The goal of the library is to provide blocks which are needed to simulate robotic systems and can not be modelled with standard blocks. First of all, these are the blocks for robot kinematic and dynamic models, the blocks for sensors systems, the typical transformations present in robot systems and the special interface blocks for robots, sensors and all other communications. Additionally, the

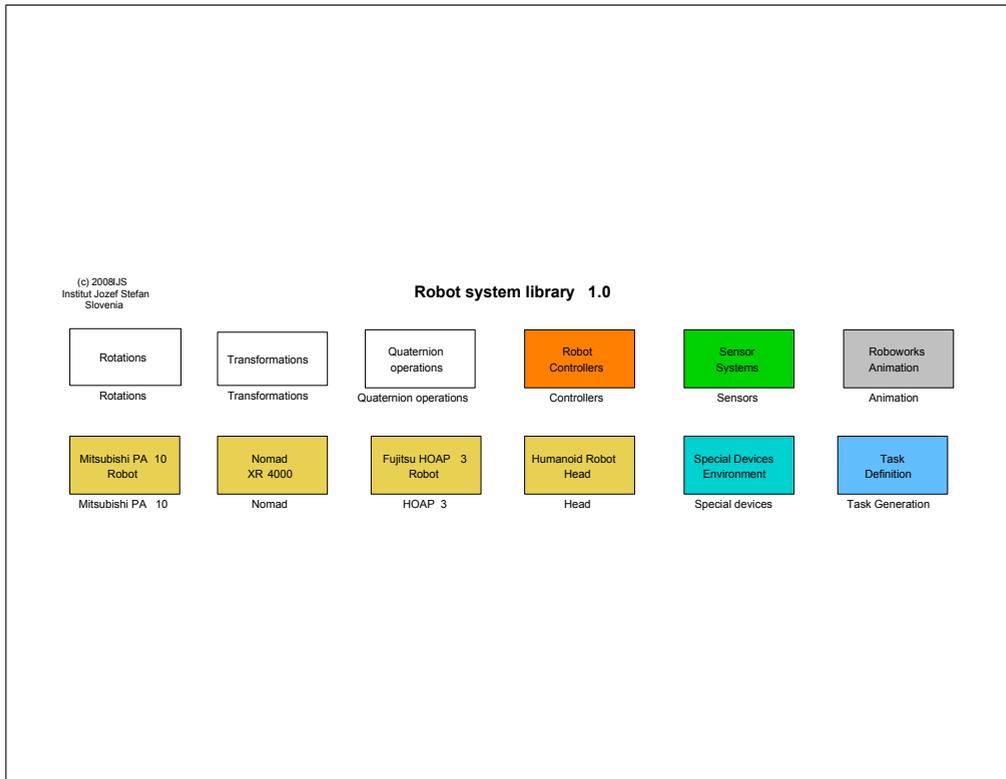


Fig. 20. Simulink Robot system library

library includes some blocks with standard subsystems like task space controllers, trajectory generation modules, etc.

3.3 Integration of sensors

Advanced robotics is characterized by the variety of complex sensory systems, e.g. vision sensors, force sensors, acoustic sensors, laser scanners, proximity sensor, etc. Therefore it is extremely important to apply as accurately as possible the sensor models into the simulation environment. The models of sensors are completely transparent to the design environment, i.e. real sensor can be substituted with the simulated one and vice versa in the control loop. The integration of sensors depends on their characteristics. Complex sensor systems like vision and acoustic sensors, or more advanced laser proximity sensors require relatively high computational power for signal processing. In many cases, it is difficult to accomplish all required data processing on the local computer. Often we have to apply a remote computer or even a remote computer cluster in order to obtain required computational power. In such a case, the subsystems are connected through ethernet with UDP protocol. We have developed a special protocol classes for different sensors, actuators and other subsystems. However, the performance is also affected by the communication delays. Therefore, it is favourable to pro-

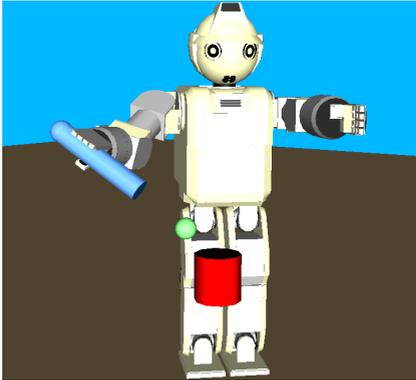


Fig. 21. Animation of the HOAP 3 humanoid robot using RoboWorks™



Fig. 22. Animation of the PA10 robot in Blender

cess signals of high frame-rate sensor, such as joint encoders, tachometers, force sensors, etc. on the local computer.

3.4 Visualization and animation

It is very important to visualize the simulation results. Especially in robotics it is necessary to “see” the motion of the robot and objects in the working environment. In our system we rely on external software for the visualization and animation of robots. In general, joint angles of robotic manipulators as well as the position and orientation of the other simulated objects in the scene are passed to the visualization tools using TCP/IP or UDP protocol. Currently, we have integrated into our simulation environment two visualization software packages - RoboWorks (RoboWorks, 2008) and Blender (Blender, 2008).

Roboworks incorporates simple, but efficient modeler. Because of its simplicity the RoboWorks package is the favourable tool for the visualization of simpler systems, i.e. one or two robots in non-complex environment. Figure 21 shows the animation of our HOAP 3 humanoid robot and also in the following examples the RoboWorks environment has been used for the visualization.

For more complex scenes we use Blender, an open source multi-platform 3D computer animation program, which has a lot of features that are potentially interesting for engineering purposes, such as the simulation and programming of robots, machine tools, humans and animals, and the visualization and post-processing of all sorts of data that come out of such biological or artificial “devices”. Blender supports also scripts (via Python interfaces to the core C/C++ code), hence it can be extended in many different ways. Among others, Blender has the capability of placing moving cameras at any link of the kinematic chain, it supports the real time photo realistic rendering for the virtual reality simulation and has also a physics engine for the simulation of the interactions between entities.

3.5 Real-time simulation

The real-time performance of the control algorithm is very important when dealing with low-level control. However, when developing higher level control algorithms, the real-time may be also important especially when high sample frequency improves the performance of the

system. Therefore, when manipulator-in-the loop simulation is performed, the simulation system which controls the robot system has to provide the real-time capabilities and enable high sample frequencies. There are many real-time operating systems as Real Time Linux, QNX, EYRX, SMX, etc. Disadvantages of these operational systems are time-consuming software development and incompatibility with other systems. The algorithms are usually written in C or some other low-level programming language, where more sophisticated control algorithms require more time and increase the chance of error. Due to the above mentioned disadvantages of some real-time operation systems, we use the Matlab/Simulink and the xPC Target operation system whenever possible (Omrčen, 2007). xPC Target enables the real-time simulation and hardware-in-the-loop simulation using corresponding interfaces. It is a good prototyping tool that enables connecting Matlab/Simulink models to physical systems and executing simulation in real-time on PC-compatible hardware. As xPC Target supports also UDP communication, this was also one of the reasons to select the UDP for the communication between different applications in the simulation environment (Omrčen, 2007). Nevertheless, using Matlab/Simulink and xPC Target environment brings some disadvantages. Most of the hardware used for a robot control, which is available on the market, does not provide drivers for xPC Target. Therefore, we had to develop drivers for our robots and sensors.

3.6 Case study

To show the efficiency, flexibility and usability of our control design environment we outline a typical experimental example using the Mitsubishi PA robot. The robot task is to play yo-yo, i.e. to keep the amplitude of the yo-yo at a desired level (Žlajpah, 2006). The yo-yo is tied to the tip of the robot. To be able to play the yo-yo it is necessary to know the position of the yo-yo and the force in the string or the velocity of the yo-yo (depending on the control algorithm). A WebCam has been used to measure the position of the yo-yo. To measure the string force a JR3 force/torque sensor mounted on the end-effector of the robot has been used. The experimental setup is shown in Fig. 23. The control should be implemented on PC's in MATLAB/SIMULINK environment and we wanted to use the PA10 motion control board which allows to control the end-effector positions of the robot.

In the first step of the control design when different control strategies have to be tested, we simulated the whole system in Simulink. We used the PA10 kinematic model and we had to develop a Simulink model of the yo-yo. The top level simulation scheme is shown in Figure

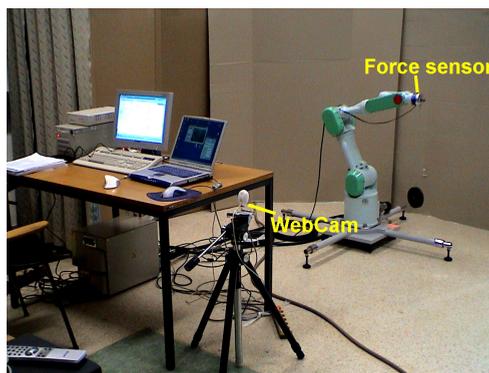


Fig. 23. Experimental setup (Mitsubishi PA10, vision system and force sensor)

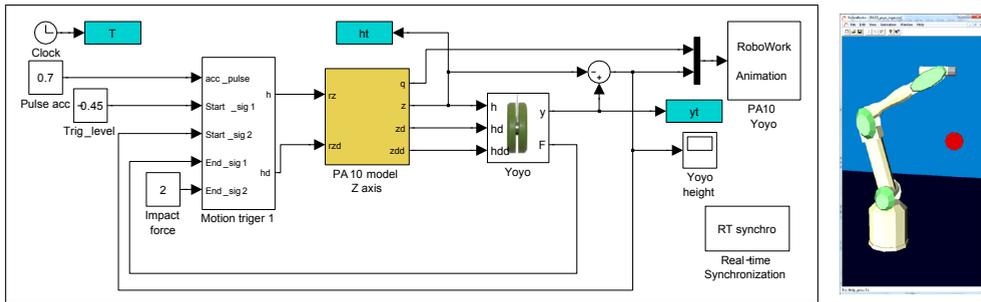


Fig. 24. Yoyo simulation: top level block scheme in Simulink and animation of the PA10 robot and yo-yo in RoboWorks

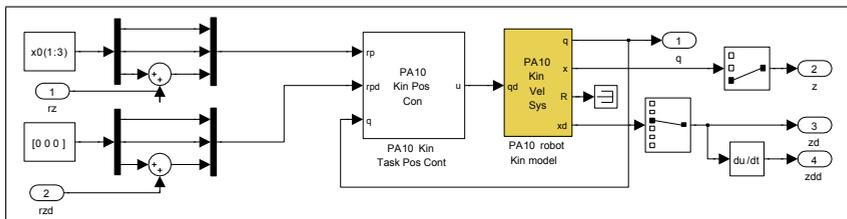


Fig. 25. PA10 model with kinematic task space position controller

24. The main three blocks are the controller, the robot model and a special model of the yo-yo (Žlajpah, 2006). As we want to move the robot end-effector only in the vertical direction the z -axis motion (x and y positions are fixed to the initial values), we have to use a kinematic task space controller. This subsystem can be easily composed by combining blocks in our Simulink library as it is shown in Figure 25.

After the best control strategy has been verified using this simulation scheme, the next step is to test the control when the sensor systems information is obtained via ethernet connection. Therefore, we have developed a special yo-yo simulator, which receives the hand position and sends the position of the yo-yo and the string force via ethernet connection using UDP protocol (see Figure 28). The simulation scheme is the same except that instead of yo-yo Simulink model, the corresponding UDP interface blocks are used (see Figures 26 and 27).

As the external yo-yo simulator is a real time simulator, also in Simulink real-time simulation should be used. As the sampling frequency in this case is rather low (100 Hz for robot control and 25Hz for vision system) and the computation time of the Simulink model is small enough, we can use a special block for real-time synchronization.

Finally, when the designed control algorithms give satisfactory simulation results, we can test the control strategy on a real system. In manipulator-in-the-loop simulation, the model of the PA10 robot is replaced by the corresponding interface blocks. The position of the yo-yo and the string force are now obtained from the vision system and force sensor using the same interface as when the yo-yo simulator has been used. The corresponding scheme is shown in Figures 29. From the top level scheme it can easily be seen that the controller part of the system has not been changed and is the same as in the previous simulation schemes.

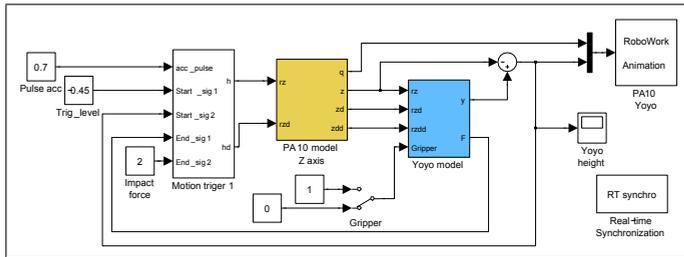


Fig. 26. The case with kinematic PA10 robot model and external yo-yo simulator

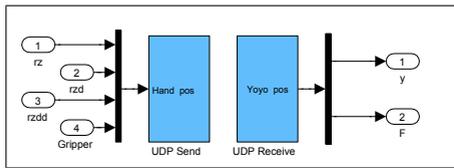


Fig. 27. Interface for external yo-yo simulator (Yoyo model block)

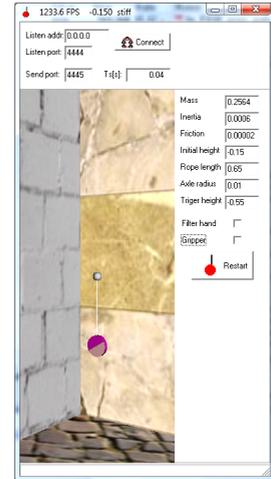


Fig. 28. External yo-yo simulator

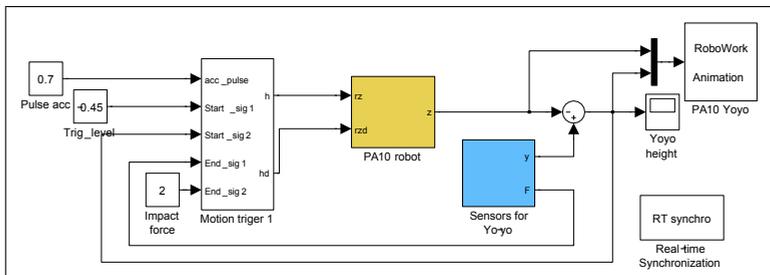


Fig. 29. Hardware-in-the-loop simulation (real PA10 robot, force sensor and vision systems are in the simulation loop)

4. Overview of simulation in different robotic fields

Robotics is very interdisciplinary, fast growing research field. Robots are introduced in various areas and used for different tasks also the requirements for the simulation tools depend on the particular application. However, all new applications, methodologies and technologies in robotics share the requirement to simulate robot systems and the environment with sufficient sophistication and accuracy. In the following, the role of the simulation tools in some robotic fields is presented.

4.1 Off-line programming

The greatest advantage of robots is their flexibility, i.e. their ability to be rearranged for new production tasks. Utilization of the robot's flexibility presupposes the effective programming. Robot can be programmed directly using the robot controller and other required equipment. However, to overcome the limitation that requires floor presence for programming and if we do not want that production equipment (robot and auxiliary devices) is not occupied during

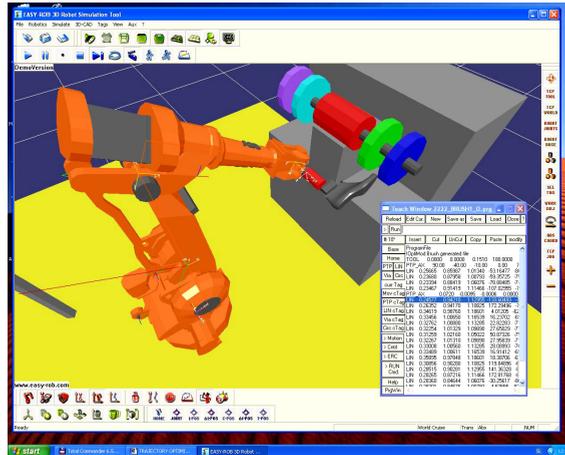


Fig. 30. Simulation of robot cell for the shoe finishing operations

programming the alternative is off-line programming. Off-line programming takes place on a separate computer, usually in the office and it uses the models of the workcell including robots and other devices that are used in the cell. Almost all commercially available industrial robots are supported by the off-line programming systems (ABB, 2008; Kuka, 2008; MotoSim, 2008). However, many other more general off-line programming systems are available for off-line programming and simulation of workcells which are not dedicated to a certain robot family (Easy-Rob, 2008; RobCAD, 1988; RobotWorks, 2008).

Using off-line programming systems robot programs can in most cases be created by the reuse of existing CAD data so that the programming will be quick and effective. Additionally, the robot programs are verified in simulation and any errors are corrected. For example, we have developed a robot cell for shoe finishing operation (creaming, brushing, etc.) where the trajectories have been generated using CAD data without considering the kinematic limitations of the robot and collisions with the objects in the workspace of the robot. Using off-line simulation these trajectories have been then verified and optimized before applied to the real robot (Fig. 30).

4.2 Humanoid robots

Advances in humanoid robotics open new possibilities of introducing humanoid robots into human environments. The goals of this emerging technology is that robots will work, assist, entertain and cooperate with humans or do certain jobs instead of a human. The foundation for these applications are control strategies and algorithms, sensory information and appropriate models of robot and environment. Virtual worlds, dynamic walking and haptic interaction are the topics addressed by researchers in this field (Hirukawa et al., 2003; Khatib et al., 2002; NASA, 2008; Stilman & Kuffner, 2003).

As an example we show a simulation environment which allows the user to interactively control a virtual model of a humanoid (Stilman & Kuffner, 2003). Fig. 31 shows a humanoid robot which is permitted to reconfigure the environment by moving obstacles and creating free space for a path. The software components include modelling of the robot geometry and



Fig. 31. Humanoid robot navigating among movable obstacles

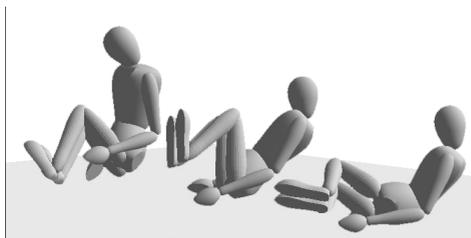


Fig. 32. Humanoid body falling due to gravity.

kinematics, simulated stereo visual sensing, and motion planning. In addition, the simulation environment serves as a graphical user interface for controlling the real robot hardware interactively.

The next example considers the whole-body behaviours (Khatib et al., 2005). Fig. 32 illustrates a virtual real-time simulation where a humanoid is falling under the effects of the gravity and colliding with the floor at multiple contact points. The proposed control was implemented and verified in a virtual environment that integrates multi-body dynamics, multi-robot control, multi-contact multi-body resolution and haptic interaction for robot teleoperation.

4.3 Robotics in medicine

The modelling of deformable organs, planning and simulation of robotics procedures, safe and real-time integration with augmented reality are the key topics in the field of robot assisted surgery (Ève et al., 2004). Fig. 33 shows the interface of the software designed at INRIA (Chir Robotics Medical Team) for the planning and simulation of robot assisted surgical interventions. The simulator has the double aim of offering the surgeon a realistic environment to develop good control over the robot, and of validating the suggested incision ports.

4.4 Mobile robotics

Mobile robotics is a complex research area in which many advanced technologies and open research issues are combined all together. It is often difficult to master perfectly every technology, and hence realistic simulations and fast prototyping of mobile robots help to reduce the amount of time and hardware spent in developing mobile robotics applications. Moreover, the simulation tools allow the researchers to focus on the most interesting parts of their robotics projects and hence to achieve more advanced results. Especially, as mobile robots move out of laboratories and into the hands of users it is required that the simulation tools provide an integrated development environment for specifying, evaluating and deploying robot missions and it should allow non-expert users to specify robot missions in an easy way, e.g. using a visual programming paradigm. Furthermore, integrated support for evaluating solutions via simulation and finally deploying them on robots must also be available.

The Webots mobile robotics simulation software (Webots, 2005) provides you with a rapid prototyping environment for modelling, programming and simulating mobile robots. The included robot libraries enable you to transfer your control programs to many commercially available real mobile robots. Figure 34 shows the simulation of Sony AIBO robot.

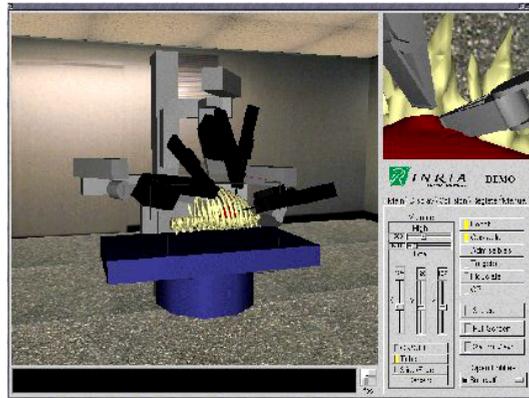


Fig. 33. Simulation of robot assisted surgical interventions (INRIA — ChIR)



Fig. 34. Simulation of Sony Aibo ERS-7 in Webots simulator

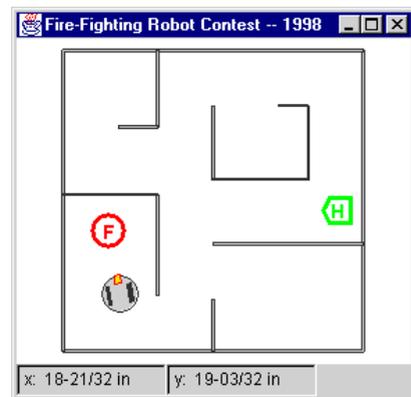


Fig. 35. Simulated competition arena from the Trinity College Fire Fighting Home Robot Contest

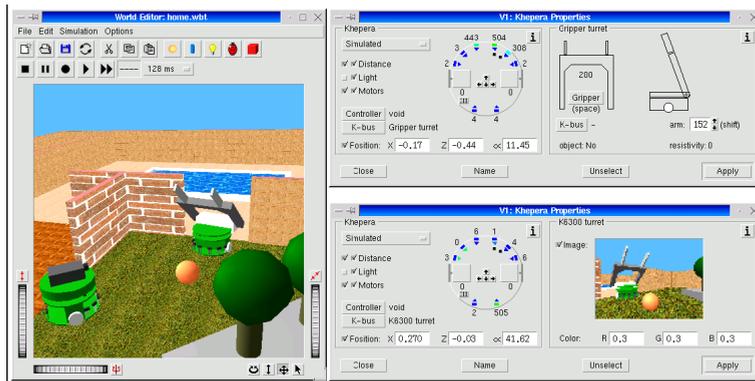


Fig. 36. Webots simulator: Simulation of Khepera robots

Rossum's Playhouse is a modest, two-dimensional robot simulation intended to aid developers implementing control and navigation logic. It allows applications to build a data-configurable robot which can interact with a simulated landscape or solve a virtual maze. It is primarily a tool for programmers and other developers who are writing the software for robotic applications. It can also be used as a testbed for algorithms and control logic. A number of developers are currently using the simulator to test and improve competition strategies for small robots. Fig. 35 shows a maze-solving application in a known environment.

Some of the simulation programs are designed for special robots. A very popular mobile robot used in research is Khepera and there are many simulation packages which support this robot, like Webots (Michel, 2008) (Fig. 36) or Matlab based toolbox KiKS (Storm, 2008). Another example is the EyeSim simulator (EyeSim, 2008). EyeSim is a multiple mobile robot simulator that allows experimenting with EyeBot mobile robots. The user can test the same unchanged programs that run on the real robots. In Fig. 37 a 3D scene representation of the environment and robots in it is being shown, together with the views of active robots.

A special kind of mobile robots are biomimetic robots which borrow their structure and senses from animals, such as insects. The most well-known early biomimetic robots were a cockroach and a lobster. The research is aimed at developing new mobile robots that exhibit much greater robustness in performance in unstructured environments than mobile robots with wheels. These new robots will be substantially more compliant and stable than current robots, and will take advantage of new developments in materials, fabrication technologies, sensors and actuators. Applications will include autonomous or semi-autonomous tasks in an obstacle-strewn ground or a sloshy ocean bay. Also here simulation plays an important role.

Fig. 38 shows a walking bug (Reichler & Delcomyn, 2000). The simulation system is able to simulate anything from a single leg segment to an entire walking insect, including muscles, sense organs, and the nervous control system. The user can enter the physical dimensions of a single leg of an insect, place muscles and sense organs where they are known to be located on the leg of a living insect, and study how the nervous system might use these components to generate reflex movements produced by a stimulus applied to one of the sense organs. By entering the physical dimensions of the entire body as well as all six legs, it is possible to study the way in which the nervous system might generate coordinated walking under various

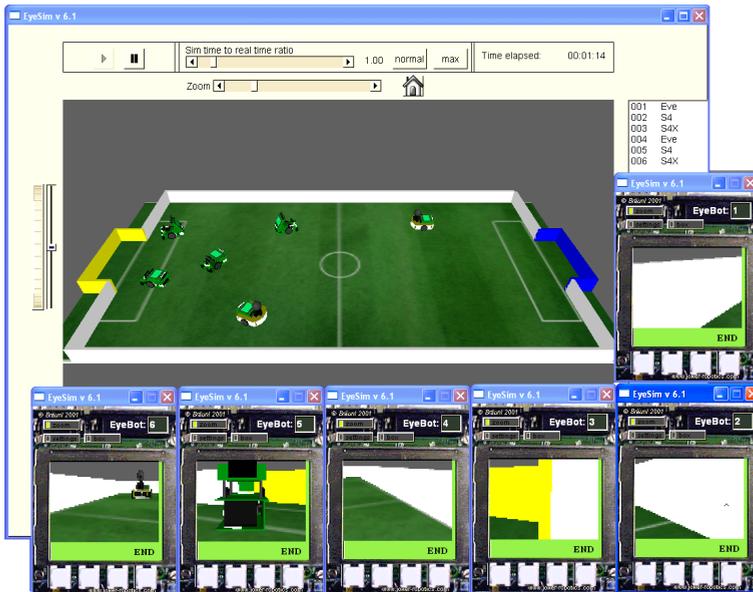


Fig. 37. EyeSim simulator: EyeBots playing football

circumstances. The nervous system can also be emulated flexibly – as a single "controller" or as a hierarchical chain of controllers, as desired.

4.5 Nanorobotics

The automation, control, and manufacturing of nanorobots is a challenging and very new field. The design and the development of complex nanomechatronic systems with high performance should be addressed via simulation to help pave the way for future applications of nanorobots in biomedical engineering problems. Successful nanorobotic systems must be able to respond efficiently in real time to change the aspects of microenvironments. Hence, simulation tools should not provide only animation or visualization, but should encounter also physical characteristics of nanorobots and environment (Cavalcanti & Jr., 2005). Fig. 39 shows the virtual environment which is inhabited by nanorobots, biomolecules, obstacles, and organ inlets.

4.6 Space robotics

The behaviour of free-floating manipulators in space is different compared to the manipulators "on ground" because they are not fixed, the gravity is negligible and due to the conservation of linear and angular momentum. This makes control and planning of robotic end effector trajectories highly complicated (Dubowsky & Papadopoulos, 1993). The simulation plays an important role in complex trajectory planning task especially as the space manipulator can have more than conventional number of six degrees of freedom. Additionally, the robotics simulation tools for space applications are designed also to meet the challenges of astronaut and ground personnel training as well as providing a valuable tool for space operations support. The simulator supports the critical tasks to be performed by astronauts,

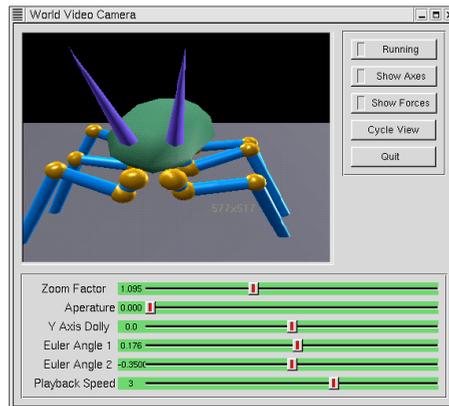


Fig. 38. A simulation of a walking insect



Fig. 39. View of simulator workspace showing the vessel wall with a grid texture, cells and nanorobots

including payload handling, berthing and de-berthing. For that it has to provide real-time high-fidelity simulation of the flexible dynamics performance of robotic arms and if necessary also of astronauts, contact dynamics models, and 3D visual models have to support realistic views generated by cameras in an operational and dynamic lighting environment that includes the production of split screen views. Fig. 40 shows a simulation of a humanoid robot Robonaut in space. The Robonaut simulation has been developed to bridge a gap between operations and development activities (NASA, 2008). The simulator matches the appearance, kinematics and dynamics of Robonaut and serves as a platform to test new control theories and configurations without having to use the real Robonaut or to construct new expensive hardware. Another benefit of the simulation regards path planning. Now operators can test arm motions on difficult tasks before actual operation of the robot, thus minimizing the risk to the hardware.

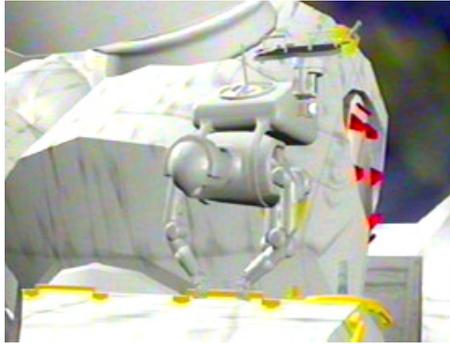


Fig. 40. The Robonaut simulator

5. Conclusion

The simulation is widely used in all fields of robotics from kinematics and dynamics to industrial applications. Actually, advanced robot systems require sophisticated simulation tools which can model accurately enough the physical world at a sufficient speed and allow the user interaction. New challenges in the simulation of the robotic systems are multi-body dynamics that computes robot and object motions under the influence of external forces, fast collision detection and contact determination, realistic visualization of the robot and environment, and haptic interaction. The advanced simulation tools are the foundation for the design of sophisticated robot systems, for the application of robots in complex environments and for the development of new control strategies and algorithms. The simulation being once a tool for the analysis of a robot system and task planning, has become an open platform for developing new robot systems. Not only that the modern simulation tools can simulate and visualize the real world in a very realistic way, they allow to go beyond the reality. Namely, the researchers may build experimental environments according to their own imagination, using robots and technologies which are not available yet. In the end, we believe that the simulation in robotics has reached a very important role and by using different simulation software, the current and future capabilities of complex robotic systems can be significantly improved.

6. References

- ABB (2008). RobotStudio 5, ABB, <http://www.robotstudio.com/rs5/>.
- Alotto, G., Bona, B. & Calvelli, T. (2004). Prototyping Advanced Real-Time Robotic Controllers on Linux RTAI Systems with Automatic Code Generation, *Proceedings of International Conference Mechatronics and Robotics 2004*, Aachen, Germany.
- Blender (2008). Blender: <http://www.blender.org/>.
- Cavalcanti, A. & Jr., R. A. F. (2005). Nanorobotics Control Design: A Collective Behavior Approach for Medicine, *IEEE Transactions on NanoBioScience* (2): 133 – 140.
- Corke, P. I. (1996). A Robotics Toolbox for MATLAB, *IEEE Robotics & Automation Magazine* 3(1): 24 – 32.
- Dubowsky, S. & Papadopoulos, E. (1993). The Kinematics, Dynamics, and Control of Free-Flying and Free-Floating Space Robotic Systems, *IEEE Trans. on Robotics and Automation, Special Issue on Space Robotics* 9(5): 531 – 543.
- Easy-Rob (2008). Easy-Rob, 3D Robot Simulation Tool, <http://www.easy-rob.de/>.

- Ève, C.-M., Adhami, L., Mourgues, F. & Bantiche, O. (2004). Optimal planning of robotically assisted heart surgery: Transfer precision in the operating room, *International Journal of Robotics Research* 4(4): 539–548.
- EyeSim (2008). EyeSim, EyeBot Simulator, <http://robotics.ee.uwa.edu.au/eyebot/index.html>.
- Fenton, R. & Xi, F. (1994). Computational analysis of robot kinematics, dynamics, and controlling the algebra of rotations, *IEEE Trans. on Systems, Man, Cybernetics* 6(6): 936 – 942.
- Gams, A., Ijspeert, A. J., Schaal, S. & Lenarčič, J. (2009). On-line learning and modulation of periodic movements with nonlinear dynamical systems, *Autonomous Robots* 27(1): 3 – 23.
- Go, J., Browning, B. & Veloso, M. (2004). Accurate and flexible simulation for dynamic, vision-centric robots, *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*, New York.
- Hirukawa, H., Kanehiro, F., Kajita, S., Fujiwara, K., Yokoi, K., Kaneko, K. & Harada, K. (2003). Experimental Evaluation of the Dynamic Simulation of Biped Walking of Humanoid Robots, *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, pp. 1640 – 1645.
- Kazi, A. & Merk, G. (2002). Experience with RealSim for Robot Applications, *Technical report*, KUKA Roboter GmbH, Augsburg.
- Khatib, O., Brock, O., Chang, K.-S., Conti, F. & Ruspini, D. (2002). Human-Centered Robotics and Interactive Haptic Simulation, *Int. J. of Robotic Research* 23(2): 167 – 178.
- Khatib, O., Sentis, L., Park, J.-H. & Warren, J. (2005). Whole Body Dynamic Behavior and Control of Human-Like Robots, *International Journal of Humanoid Robotics* 1(1): 29–43.
- Kleijn, C. (2009). *Getting Started with 20-sim 4.1*, Controllab Products B.V.
- Kuka (2008). KUKA.Sim, KUKA Industrial robots, http://www.kuka-robotics.com/en/products/software/kuka_sim/.
- Žlajpah, L. (2006). Robotic yo-yo : modelling and control strategies, *Robotica* 24(2): 211 – 220.
- Lambert, J. M., Moore, B. & Ahmadi, M. (2001). Essential Real-Time and Modeling tools for Robot Rapid Prototyping, *Proceeding of the 6 th International Symposium on Artificial Intelligence and Robotics & Automation in Space i-SAIRAS 2001*, Quebec, Canada.
- Latombe, J.-C. (1995). Controllability, recognizability, and complexity issues in robot motion planning, *Proc. of the 36th Annual Symposium on Foundations of Computer Science*, Los Alamitos, CA, USA, pp. 484 – 500.
- Lippiello, V., Villani, L. & Siciliano, B. (2007). An open architecture for sensory feedback control of a dual-arm industrial robotic cell, *An International Journal Industrial Robot* 34(1): 46–53.
- MCA2 (2008). Modular Controller Architecture 2 — MCA2: <http://mca2.org/>.
- Michel, O. (2008). Webots: A Powerful Simulator for the Khepera Robot, http://www.ccnacht.de/_ccnacht2/media/pdf/khepera_webots.pdf.
- Miller, A. & Allen, P. K. (2004). Graspit!: A versatile simulator for robotic grasping, *IEEE Robotics and Automation Magazine* 11(4): 110 – 122.
- Miller, A. T. & Christensen, H. I. (2003). Implementation of multi-rigid-body dynamics within a robotic grasping simulator, *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, pp. 2262 – 2268.
- MotoSim (2008). Solutions in Motion - MotoSim, Motoman Yaskawa Company, <http://www.motoman.com>.

- MSRS (2008). Microsoft Robotics Studio. <http://msdn2.microsoft.com/en-us/robotics/default.aspx>.
- NASA (2008). Robonaut, NASA, http://vesuvius.jsc.nasa.gov/er_er/html/robonaut/robonaut.html.
- Nethery, J. & Spong, M. (1994). Robotica: a Mathematica package for robot analysis, *IEEE Robotics & Automation Magazine* **1**(1).
- NGD (2008). Newton Game Dynamics, <http://physicsengine.com/>.
- ODE (1994). Open Dynamics Engine, <http://ode.org/ode.html>.
- Omrčen, D. (2007). Developing matlab simulink and xpc target real-time control environment for humanoid jumping robot, *16th Int. Workshop on Robotics in Alpe-Adria-Danube Region - RAAD 2007*, Ljubljana, Slovenia, pp. 18–23.
- Omrčen, D., Žlajpah, L. & Nemec, B. (2007). Compensation of velocity and/or acceleration joint saturation applied to redundant manipulator, *Robot. auton. syst.* **55**(4): 337 – 344.
- Petrič, T., Gams, A. & Žlajpah, L. (2009). Controlling yo-yo and gyroscopic device with non-linear dynamic systems, *Proceedings of 18th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, Brasov, Romania, p. 6.
- Reichler, J. A. & Delcomyn, F. (2000). Dynamics Simulation and Controller Interfacing for Legged Robots, *Int. J. of Robotic Research* (1): 41 – 57.
- RobCAD (1988). *ROBCAD/Workcell, User's manual*, Tecnomatix.
- RobotWorks (2008). RobotWorks - a Robotics Interface and Trajectory generator for SolidWorks, <http://www.robotworks-eu.com/>.
- RoboWorks (2008). RoboWorks™: http://www.newtonium.com/public_html/Products/RoboWorks/RoboWorks.htm.
- SD/FAST (1994). *SD/FAST User's Manual*, Symblic Dynamics, Inc.
- SimMechanics (2005). *SimMechanics, User's Guide*, The Mathworks.
- Stilman, M. & Kuffner, J. (2003). Navigation Among Movable Obstacles: Real-time Reasoning in Complex Environments, *Proc. IEEE Int. Conf. on Humanoid Robotics*, Los Angeles, CA, USA.
- Storm, T. (2008). KiKS is a Khepera Simulator, user guide, <http://www.tstorm.se/projects/kiks/>.
- Žlajpah, L. (1997). Planar Manipulators Toolbox: User's Guide, *Technical Report DP - 7791*, Jožef Stefan Institute. URL: <http://www2.ijs.si/leon/planman.html>.
- Žlajpah, L. (2001). Integrated environment for modelling, simulation and control design for robotic manipulators, *Journal of Intelligent and Robotic Systems* **32**(2): 219 – 234.
- Webots (2005). *Webots User Guide*, Cyberbotics Ltd.
- Zhang, Y. & Paul, R. P. (1988). Robot Manipulator Control and Computational Cost, *Technical Report MS-CIS-88-10*, University of Pennsylvania Department of Computer and Information Science. <http://repository.upenn.edu/cis-reports/621>.

Modeling of a One Flexible Link Manipulator

Mohamad Saad

Department of Applied Sciences

Université du Québec en Abitibi-Témiscamingue

Rouyn-Noranda, Québec, Canada

mohamad.saad@uqat.ca

1. Introduction

Dynamics and control of flexible robot manipulators have received wider attention during the last two decades. In manufacturing and space applications, the use of lightweight structures in robot manipulators is motivated by their capacity for high speed maneuvers, their high payload to arm weight ratio, higher mobility, reduced energy consumption, and lower inertia forces for accurate positioning. To insure satisfactory performances of such systems, their flexibility should be included in modeling and in control design. This flexibility becomes more significant in cases of larger structures and more stringent performance demands.

In modeling flexible link manipulators, the most widely used methods to generate spatially discrete models are the Assumed-Mode Method (AMM), and the Finite Element Method (FEM). The accuracy of the dynamical model obtained from the analytical formulation is highly dependent on the adopted mode shapes of the link deflection and their number.

In the AMM, the shape functions are typically eigenfunctions of a closely related simpler problem with standard boundary conditions (BCs). For example, the Euler-Bernoulli beam in one of the following configurations (Mirovitch, 1967): clamped-free, pinned-free, clamped-mass, or pinned-mass. In the FEM, the shape functions, known as interpolation functions, are simple polynomials that verify the continuity conditions between two adjacent elements or nodes. Examples of interpolation functions are Hermite cubics (Chen & Menq, 1990), cubic splines (Cho et al., 1991; Saad et al., 2006) and cubic B-splines (Saad et al., 2006).

In the literature, most of the comparison studies that have been done are for clamped versus pinned mode shapes (Barbieri & Ozguner, 1988; Cetinkunt & Yu, 1991; Hastings and Book, 1987). The general conclusion is that for a slewing beam, clamped modes are more appropriate than pinned modes. Meirovitch and Kwak (Mirovitch & Kwak, 1990) compared the convergence rate of a clamped-free assumed-mode model vs. a linear interpolation finite elements model in estimating the frequencies of a horizontal beam with longitudinal deformation. The convergence rate was slow in both cases. To accelerate the rate of convergence, assumed-modes that take into account the natural BCs and interpolation functions that have the ability to satisfy the differential equation of the system were introduced. Buffinton and Lam (Buffinton & Lam, 1992) compared a lumped-parameter model versus an AMM in modeling and control of a one-link flexible manipulator in the horizontal plane. They concluded that from a control viewpoint, the AMM based model yields better performances when compared to lumped-parameter model. In comparing two clamped-mass assumed-modes and two Hermite cubic finite elements, Theodore and Ghosal (Theodore & Ghosal, 1995) concluded that

fewer mathematical operations are required for inertia matrix computation with the finite element model in comparison with the assumed-modes formulation. However, the number of equations of motion is greater for the finite element model. Junkins and Kim (Junkins & Kim, 1993) found that, for a flexible appendage with rigid body motion, the convergence rate of the natural frequencies of clamped-free assumed-modes is much slower than the convergence rate of Hermite cubic finite elements. This is because the shape functions used in the AMM satisfy only the essential BCs of the given problem. Tokhi and Mohamed (Tokhi & Mohamed, 1999) evaluated the computational requirements of FEM in characterizing the behavior of a flexible link. They compared the convergence rate of the first three frequencies while increasing the number of elements. In contrast with most studies, the frequencies were under-estimated because the model did not include the base and load inertias.

Parametric and simulation studies were also performed to investigate the influence of different system parameters on the natural frequencies (Li et al., 1998) or on the zeros and the poles of the transfer functions of a flexible link (Parks & Pak, 1991). To validate the models of multi-link flexible manipulators, experimental frequency response results are generally compared to linearized model responses (Nicosia et al., 1996) or to simplified model responses (Book & Oberfell, 2000; Stanway et al., 1998). A reduced finite element model is used in (Xia & Menq, 1992) to study the effect of robot configurations on the elastic mode shape functions.

In control design, the issue of the number of flexible modes used becomes important, especially when using the mass matrix in the control law. As the number of modal variables is increased in the model, more elastic modes can be obtained and more accurate estimation of the dominant modes of the system can be achieved. However, the computational burden makes dynamical simulations and control very difficult (Xia & Menq, 1992). In particular, the mass matrix becomes more ill-conditioned. This can lead to loss of performance or even instability (Moallem et al., 2002). A trade-off is often required between model accuracy and avoiding the loss of reliability due to increased numerical ill-conditioning that is inherent to using a large number of flexible modes.

These studies show the importance of a good representation of the flexibility in order to get an accurate and precise model and point to the lack of systematic methods to determine the appropriate choice of shape functions or their number. Two important questions come to mind in particular: does the convergence rate of the eigenfrequencies represent a good performance index? How accurately do different choices of shape functions represent the eigenmodes of a flexible link?

The objective of this work is to develop the dynamical model of one flexible link under the assumption of small deformations using Lagrange equations and taking into account the foreshortening effect of the link. A compact dynamical model is given for different shape functions. The orthogonality relations between shape functions are presented. The objective is also to compare the performance of the different shape functions with respect to their ability to accurately represent the dynamic and the static behavior of a flexible rotating beam system. Based on the mathematical model of this system, an analytical solution for the exact shape functions describing the link's flexibility is developed. These functions are then substituted in the spatially discrete model to obtain a reference model with ideal performances for all cases presented in the following sections. Therefore, the performances of the different shape functions based models are compared to the reference one.

Several candidates based on the AMM and FEM are selected to develop the approximated models. In this paper, however, the FEM is not used in the formal sense. The interpolation functions are generated locally for an element but are represented globally over the entire

length of the link. This representation allows the use of locally defined functions in the AMM as will be seen in the sequel. The selected shape functions are: the eigenfunctions of a beam rotating in the horizontal plane, a clamped-payload beam, and a clamped-free beam, together with the polynomial functions, the cubic splines, and the cubic B-splines. The comparison is done for the eigenfrequencies, the eigenmodes and their derivatives, and for the static deformation under gravity. The second derivatives are particularly important because they are related to the link curvature. The latter is widely used in modeling and control experiments to measure and estimate flexibility using strain gauges (Piedboeuf & Miller, 1994). In addition, load parameters are changed from their nominal values to test sensitivity of the shape functions based models. In all cases, we assume that the beam parameters are well known. Section 2 describes the features of the flexible link under consideration and presents the continuous model of this system. The eigenvalues problem and the discretized model are presented in sections 3 and 4 respectively. Section 5 deals with the analytical aspects of the shape functions. A comparison of the different assumed-modes is presented in Section 6 in which the merits and demerits of each shape function are examined.

2. The Continuous Model

Figure 1 shows the test bed system. It consists of a motor, a flexible beam, and a payload. The system parameters and their nominal values are given in Table 1. The motor applied

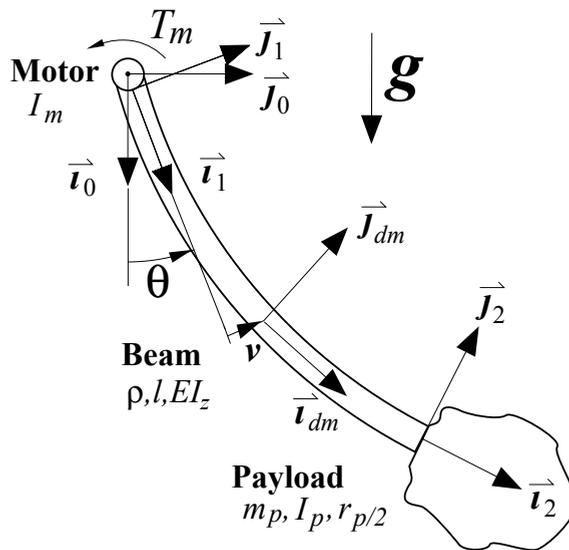


Fig. 1. A flexible rotating beam

torque is T_m . Gravity acts along the x -axis of frame (\vec{i}_0, \vec{j}_0) . The motor angular position is $\theta(t)$ and the link deformation is $v(x, t)$. For simplification, x and t are omitted. When writing the dynamical equations, we make the following assumptions. The flexible link is modeled as a Euler-Bernoulli beam and the deformation is assumed to be small. We also take into account the foreshortening effect of the link by assuming a second order kinematic. However, when comparing the assumed modes models, we eliminate the foreshortening effect by assuming

Parameter Name		Nominal Value
Motor inertia	(I_m)	10 kg m^2
Beam length	(l)	1.4 m
Beam linear density	(ρ)	1 kg/m
Beam rigidity	(EI_z)	500 N m^2
Payload mass	(m_p)	2 kg
Payload inertia	(I_p)	0.36 kg m^2
Payload center of mass	(r_p)	0 m

Table 1. Nominal parameters.

small motor angular velocity and by linearizing the dynamical equations at the motor position $\theta = \pi/2$. The joint and the beam internal damping are neglected.

The continuous model consists of one ordinary differential equation (ODE) for the motor's motion, and a partial differential equation (PDE) with four associated boundary conditions (BCs) for the flexible link.

2.1 Kinematics

The position of reference frame \mathcal{R}_1 relative to the inertial reference system \mathcal{R}_0 is:

$${}^0P_1 = [0 \quad 0 \quad 0]^T$$

The rotation matrix of \mathcal{R}_1 relative to \mathcal{R}_0 is :

$${}^0R_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let dm be an element of the link with x is the distance from the base along the link's neutral axis, \bar{i}_1 . The position of \mathcal{R}_{dm} relative to \mathcal{R}_1 is:

$${}^1P_{dm} = \begin{bmatrix} x - \underbrace{\frac{1}{2} \int_0^x v'^2(s,t) ds}_{\mathcal{O}_2(v)} \\ v(x,t) \\ 0 \end{bmatrix}^T$$

The $\mathcal{O}_2(v)$ term is a 2nd order term in v . This term is to be neglected if first order kinematic was considered. The rotation matrix of \mathcal{R}_{dm} relative to \mathcal{R}_1 is (Piedboeuf, 1998):

$${}^1R_{dm} = \begin{bmatrix} 1 - \frac{1}{2}v'^2 & -v' & 0 \\ v' & 1 - \frac{1}{2}v'^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that ${}^1R_{dm}$ verify the following: ${}^1R_{dm} {}^1R_{dm}^T = I$, where I is the identity matrix.

The position vector of the payload center of mass relative to \mathcal{R}_1 is:

$$\begin{aligned} {}^1P_c &= {}^1P_2 + {}^1R_2 {}^2P_c \\ &= \begin{bmatrix} L + r_p + \frac{1}{2} \int_0^L v'^2 dx - \frac{1}{2} r_p v'_L{}^2 \\ v_L + r_p v'_L \\ 0 \end{bmatrix} \end{aligned}$$

The 1R_2 matrix is obtained from ${}^1R_{dm}$ by substituting x by L . If the payload is concentrated at the links extremity, $r_c = 0$. The rotation matrix of the payload center of mass relative to \mathcal{R}_1 is:

$${}^1R_c = {}^1R_2 {}^2R_c = \begin{bmatrix} 1 - \frac{1}{2} v'_L{}^2 & -v'_L & 0 \\ v'_L & 1 - \frac{1}{2} v'_L{}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where 2R_c is the identity matrix.

The angular velocity of \mathcal{R}_1 relative to \mathcal{R}_0 is:

$$\omega_1 = \dot{\theta} \vec{k}_1$$

Using the antisymmetric matrix to represent the velocity of an element dm of the link:

$$S = {}^1\dot{R}_{dm} {}^1R_{dm}^{-1} = \begin{bmatrix} 0 & -\dot{v}' & 0 \\ \dot{v}' & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

the angular velocity of \mathcal{R}_{dm} relative to \mathcal{R}_1 is then given by:

$${}^1\Omega_{dm} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{v}' \end{bmatrix}$$

and relative to \mathcal{R}_0 , it is given by:

$$\omega_{dm} = \omega_1 + {}^0R_1 {}^1\Omega_{dm} = [0 \quad 0 \quad \dot{\theta} + \dot{v}']^T$$

The linear velocity of the origin of \mathcal{R}_{dm} relative to \mathcal{R}_0 is:

$$v_{dm} = {}^0R_1 {}^1V_{dm} + \omega_1 \times ({}^0R_1 {}^1P_{dm})$$

When expressed in \mathcal{R}_1 , it becomes:

$$\begin{aligned} {}^1v_{dm} &= {}^0R_1^T v_{dm} = \frac{d {}^1P_{dm}}{dt} + \omega_1 \times {}^1P_{dm} \\ &= \begin{bmatrix} -\dot{\theta} v \\ \dot{v} + x \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} -\int_0^x v' \dot{v}' ds \\ -\frac{1}{2} \dot{\theta} \int_0^x v'^2 ds \\ 0 \end{bmatrix} \end{aligned}$$

The payload angular velocity is obtained from ω_{dm} by replacing x by L :

$$\omega_c = [0 \quad 0 \quad \dot{\theta} + \dot{v}'_L]^T$$

The absolute velocity of the payload, written in \mathcal{R}_1 , is:

$$\begin{aligned} {}^1v_c &= \frac{d}{dt}({}^1P_c) + \omega_1 \times {}^1P_c \\ &= \left[\begin{array}{c} -\dot{\theta}(v_L + r_p v'_L) - \int_0^L v' \dot{v}' dx - r_p v'_L \dot{v}'_L \\ (L + r_p)\dot{\theta} + \dot{v}_L + r_p \dot{v}'_L - \frac{1}{2}\dot{\theta} \int_0^L v'^2 dx - \frac{1}{2}r_p \dot{\theta} v'^2_L \end{array} \right] \end{aligned}$$

The gravity vector is represented in \mathcal{R}_0 by:

$${}^0g = [g \ 0 \ 0]^T$$

where g is the gravitational acceleration. In \mathcal{R}_1 , it is given by

$${}^1g = {}^1R_0 {}^0g = {}^0R_1^T {}^0g = \begin{bmatrix} g \cos \theta \\ -g \sin \theta \\ 0 \end{bmatrix}$$

2.2 Kinetic Energy

The kinetic energy of the given system is given by:

$$T = T_B + \int_0^L T_{dm} + T_C \quad (1)$$

where T_B is the kinetic energy of the base,

$$T_B = \frac{1}{2} I_m \dot{\theta}^2$$

T_{dm} is the kinetic energy of an element of the link,

$$T_{dm} = \frac{1}{2} \rho v_{dm}^T v_{dm} dx = \frac{1}{2} \rho {}^1v_{dm}^T {}^1v_{dm} dx$$

and T_C is the kinetic energy of the payload,

$$T_C = \frac{1}{2} I_p \omega_c^2 + \frac{1}{2} M_p v_c^T v_c = \frac{1}{2} I_p \omega_c^2 + \frac{1}{2} M_p {}^1v_c^T {}^1v_c$$

(See Appendix for kinetic energy expressions).

Linearizing the kinetic energy to the 2nd order in v give:

$$\begin{aligned} T &= \frac{1}{2} \left(I_m + \frac{1}{3} \rho L^3 + I_p + M_p (L + r_p)^2 \right) \dot{\theta}^2 \\ &+ \dot{\theta} \left(\rho \int_0^L x \dot{v} dx + M_p (L + r_p) \dot{v}_L + [I_p + M_p r_p (L + r_p)] \dot{v}'_L \right) \\ &+ \frac{1}{2} \rho \int_0^L v^2 dx + \frac{1}{2} M_p v_L^2 + \frac{1}{2} (I_p + M_p r_p^2) v'^2_L + M_p r_p v_L v'_L \\ &+ \frac{1}{2} \dot{\theta}^2 \left(\rho \int_0^L v^2 dx + M_p v_L^2 + 2M_p r_p v_L v'_L \right) \\ &- \frac{1}{2} \dot{\theta}^2 \left(\rho \int_0^L x \int_0^x v^2 ds + M_p (L + r_p) \int_0^L v'^2 dx \right) - \frac{1}{2} \dot{\theta}^2 \left(M_p r_p L v'^2_L \right) \end{aligned} \quad (2)$$

2.3 Potential Energy

The potential energy of the given system is given by:

$$V = \int_0^L V_{dm} + V_c \quad (3)$$

where V_{dm} is the potential energy of the links element dm ,

$$\begin{aligned} V_{dm} &= \frac{1}{2} E I_z v''^2 dx - \rho^1 g^T 1 P_{dm} dx \\ &= \frac{1}{2} E I_z v''^2 dx - \rho g x \cos \theta dx + \rho g v \sin \theta dx + \frac{1}{2} \rho g \cos \theta \int_0^x v'^2 ds dx \end{aligned}$$

and V_c is the potential energy of the payload:

$$\begin{aligned} V_c &= -M_p 1 g^T 1 P_c \\ &= -M_p g (L + r_p) \cos \theta + M_p g (v_L + r_p v'_L) \sin \theta + \frac{1}{2} M_p g \cos \theta \left(\int_0^L v'^2 dx + r_p v'^2_L \right) \end{aligned}$$

2.4 Rayleigh Dissipation Function

Friction is due to the motor's viscous friction and the link's internal damping. The Rayleigh dissipation function is then given by:

$$R = \frac{1}{2} b_m \dot{\theta}^2 + \frac{1}{2} k_e E I_z \int_0^L \dot{v}''^2 dx \quad (4)$$

where k_e is the links internal damping coefficient.

2.5 Dynamics

To develop the dynamics of the given system, we use Hamilton's principle associated with the kinetic and potential energy developed previously. The variational of these expressions are simplified using integration by parts. First we neglect damping forces. The only non conservative applied force is the motor torque T_m . Two dynamic equations are then obtained. The first one is associated with the motor's angle θ , and the second is associated with the deformation of the link v . Moreover, four boundary conditions (BCs) are associated to the dynamic equation of the deformation. These conditions describe the way in which the arm is attached to the base (geometrical BCs) and to the payload (natural BCs).

In the absence of viscous friction, the dynamic equation associated with θ is:

$$\begin{aligned} T_m &= \left(I_m + \frac{1}{3} \rho L^3 + I_p + M_p (L + r_p)^2 \right) \ddot{\theta} + \int_0^L \rho x \ddot{v} dx + M_p (L + r_p) \ddot{v}_L \\ &+ (I_p + M_p r_p (L + r_p)) \ddot{v}'_L + \left(\frac{1}{2} \rho g L^2 + M_p g (L + r_p) \right) \sin \theta \\ &+ \left[\int_0^L \rho g v dx + M_p g (v_L + r_p v'_L) \right] \cos \theta + \rho \int_0^L \frac{\partial}{\partial t} (\dot{\theta} v^2) dx \\ &- \frac{1}{2} \rho \int_0^L (L^2 - x^2) \frac{\partial}{\partial t} (\dot{\theta} v'^2) dx + M_p \frac{\partial}{\partial t} (\dot{\theta} v'^2_L) - M_p (L + r_p) \int_0^L \frac{\partial}{\partial t} (\dot{\theta} v'^2) dx \\ &+ 2M_p r_p \frac{\partial}{\partial t} (\dot{\theta} v_L v'_L) - M_p L r_p \frac{\partial}{\partial t} (\dot{\theta} v'^2_L) \end{aligned} \quad (5)$$

The equation associated with the link's deformation v is:

$$0 = -\rho(\ddot{v} + x\ddot{\theta}) - EI_z v'''' + g \cos \theta (\rho[-v' + (L-x)v''] + M_p v'') - \rho g \sin \theta + \theta^2 \left(\rho v + M_p[L + r_p]v'' + \frac{1}{2}\rho[-2xv' + (L^2 - x^2)v''] \right) \quad (6)$$

The BCs are given by:

$$\begin{aligned} 0 &= EI_z v''' \delta v|_0^L - M_p(L\ddot{\theta} + \ddot{v}_L)\delta v_L - \frac{1}{2}\rho\dot{\theta}^2(L^2 - x^2)v' \delta v|_0^L + M_p\dot{\theta}^2 v_L \delta v_L \\ &\quad - M_p L \dot{\theta}^2 v' \delta v|_0^L - \rho g \cos \theta (L-x)v' \delta v|_0^L + M_p r_p \dot{\theta}^2 v'_L \delta v_L - M_p r_p \ddot{\theta} \delta v_L \\ &\quad - M_p r_p \ddot{v}'_L \delta v_L - M_p r_p \dot{\theta}^2 v' \delta v|_0^L - M_p g \cos \theta v' \delta v|_0^L - M_p g \sin \theta \delta v_L \\ 0 &= -I_p(\ddot{\theta} + \ddot{v}'_L)\delta v'_L - EI_z v'' \delta v|_0^L + M_p r_p \dot{\theta}^2 v_L \delta v'_L - M_p L r_p \ddot{\theta} \delta v'_L \\ &\quad - M_p r_p \ddot{v}_L \delta v'_L + M_p r_p^2 \dot{\theta}^2 v'_L \delta v'_L - M_p r_p^2 \ddot{\theta} \delta v'_L - M_p r_p^2 \ddot{v}'_L \delta v'_L \\ &\quad - M_p r_p^2 \dot{\theta}^2 v'_L \delta v'_L - M_p L r_p \dot{\theta}^2 v'_L \delta v'_L - M_p r_p g \cos \theta v'_L \delta v'_L - M_p r_p g \sin \theta \delta v'_L \end{aligned}$$

To complete these dynamics, we add to the preceding equations the viscous friction at the base and the link's internal damping. To take into account the friction at the base in Eq. (5), the motor torque T_m is replaced by $T_m - b_m \dot{\theta}$. The internal damping of the link is considered using the model of Voigt-Kelvin. This model gives the following relation between the constraint (σ) and the deformation (ϵ):

$$\sigma = E(\epsilon + k_e \frac{d\epsilon}{dt})$$

In (6) and the associated BCs, Young modulus E is replaced by:

$$E_k = E(1 + k_e D) \quad (7)$$

where D is the partial derivatives operator, i.e. $Dv = \frac{\partial v}{\partial t}$.

Taking motor and internal damping into account, the dynamical equations (5)-(6) and the associated BCs are then replaced by:

Dynamic Equation of the Motor's Angle θ :

$$\begin{aligned} T_m &= \left(I_m + \frac{1}{3}\rho L^3 + I_p + M_p(L + r_p)^2 \right) \ddot{\theta} + b_m \dot{\theta} + \int_0^L \rho x \ddot{v} dx + M_p(L + r_p) \ddot{v}_L \\ &\quad + (I_p + M_p r_p(L + r_p)) \ddot{v}'_L + \left(\frac{1}{2}\rho g L^2 + M_p g(L + r_p) \right) \sin \theta \\ &\quad + \left[\int_0^L \rho g v dx + M_p g(v_L + r_p v'_L) \right] \cos \theta + \int_0^L \frac{\partial}{\partial t} (\rho \dot{\theta} v^2) dx \\ &\quad - \frac{1}{2} \int_0^L \rho(L^2 - x^2) \frac{\partial}{\partial t} (\dot{\theta} v'^2) dx + M_p \frac{\partial}{\partial t} (\dot{\theta} v_L^2) \\ &\quad - M_p(L + r_p) \int_0^L \frac{\partial}{\partial t} (\dot{\theta} v'^2) dx + 2M_p r_p \frac{\partial}{\partial t} (\dot{\theta} v_L v'_L) - M_p L r_p \frac{\partial}{\partial t} (\dot{\theta} v_L^2) \end{aligned} \quad (8)$$

Dynamic Equation of the Link's Deformation v :

$$0 = -\rho(\ddot{v} + x\ddot{\theta}) - EI_z v'''' - k_e EI_z \dot{v}'''' + g \cos \theta (\rho[-v' + (L-x)v''] + M_p v'') - \rho g \sin \theta + \dot{\theta}^2 \left(\rho v + M_p [L + r_p] v'' + \frac{1}{2} \rho [-2xv' + (L^2 - x^2)v''] \right) \quad (9)$$

Boundary Conditions:

The BCs are compatible with the geometric configuration of the given system. The link is clamped at the joint, i.e.

$$v(0, t) = v'(0, t) = 0,$$

and $\delta v(L, t)$ and $\delta v'(L, t)$ are arbitrary. The BCs are then given by:

$$\text{at } x = 0: v_0 = 0 \text{ and } v'_0 = 0 \quad (10)$$

at $x=L$:

$$E_k I_z v_L'''' = M_p (L\ddot{\theta} + \ddot{v}_L) + M_p r_p (\ddot{\theta} + \ddot{v}'_L) - M_p \dot{\theta}^2 (v_L + r_p v'_L) + M_p (L + r_p) \dot{\theta}^2 v'_L + M_p g v'_L \cos \theta + M_p g \sin \theta \quad (11)$$

$$-E_k I_z v_L'' = (I_p + M_p r_p^2) (\ddot{\theta} + \ddot{v}'_L) + M_p r_p (L\ddot{\theta} + \ddot{v}_L) - M_p r_p (v_L + r_p v'_L) \dot{\theta}^2 + M_p r_p (L + r_p) v'_L \dot{\theta}^2 + M_p r_p g \cos \theta v'_L + M_p r_p g \sin \theta \quad (12)$$

In the literature dynamics of flexible links manipulators with non linear kinematics were developed (Boyer et al., 2002; Piedboeuf, 1998). In (Boyer et al., 2002) the procedure of development of the dynamic equations of a robot with several flexible links is given. This procedure was applied to a robot having one flexible link. However, the analytical form of the model is not given.

3. Eigenvalues Problem

The eigenvalues problem (EVP) consists in solving, for deformation v , the dynamics of the system represented by ordinary differential equations (ODE) and partial differential equations (PDE) with associated BCs. Let us consider the case of a homogeneous problem where the motor's torque is null, i.e $T_m = 0$. We consider a solution for v separable in space and time. The deformation is then given by:

$$v(x, t) = \phi(x) q_f(t) \quad (13)$$

where $\phi(x)$ represents the link's configuration and is only dependent on the spatial variable x , and $q_f(t)$ indicates the nature of the movement carried out by the configuration and depends only on time t . By substituting Eq. (13) in the dynamic equations and the associated BCs, the PDE associated with v is transformed into ODE in $q_f(t)$. To have a stable solution, a harmonic movement is selected for $q_f(t)$ so that:

$$\ddot{q}_f(t) = -\omega^2 q_f(t)$$

where ω represents natural frequency of the system.

The EVP thus amounts finding one ω and a nontrivial solution $\phi(x)$ that verifies homogeneous discretized equations and the associated BCs. The corresponding ω are the characteristic values or the eigenvalues and the $\phi(x)$ are the eigenfunctions. EVP generally generates

the solution of a characteristic equation having an infinite countable solutions w_r . For each eigenvalue w_r corresponds an eigenfunction $\phi_r(x)$. In the general case where dynamics is represented by nonlinear equations and with nonuniform parameters, a solution of the EVP is practically not possible. For that, we use approximate methods to solve this kind of problems.

4. Spatial Discretization

The order of the solution is infinite. In order to analyze and control this system, a spatially discrete model of finite order is suitable. A finite number is then retained in the discretization of the deformation which is rewritten in the form:

$$v(x, t) = \sum_{i=1}^{\nu} \phi_i(x) q_{f_i}(t) = \phi^T(x) q_f(t) \quad (14)$$

where ν is the number of retained modes, $\phi = [\phi_1, \phi_2, \dots, \phi_\nu]^T$ et $q_f = [q_{f_1}, q_{f_2}, \dots, q_{f_\nu}]^T$. The discretization of the kinetic energy (2) gives:

$$T = \frac{1}{2} I_t \dot{\theta}^2 + \dot{\theta} \beta^T \dot{q}_f + \frac{1}{2} \dot{q}_f^T M_{ff} \dot{q}_f + \frac{1}{2} \dot{\theta}^2 q_f^T C_{rr} q_f \quad (15)$$

where I_t is the total inertia at the base, given by:

$$\begin{aligned} I_t &= I_m + \frac{1}{3} \rho L^3 + I_p + M_p(L + r_p)^2 \\ \beta &= \int_0^L \rho x \phi dx + M_p(L + r_p) \phi_L + [I_p + M_p r_p(L + r_p)] \phi'_L \\ M_{ff} &= \rho \int_0^L \phi \phi^T dx + M_p \phi_L \phi_L^T + (I_p + M_p r_p^2) \phi'_L \phi'^T_L + M_p r_p (\phi'_L \phi_L^T + \phi_L \phi'^T_L) \\ C_{rr} &= \rho \int_0^L \phi \phi^T dx + M_p \phi_L \phi_L^T + M_p r_p (\phi_L \phi'^T_L + \phi'_L \phi_L^T) - \rho \int_0^L x \int_0^x \phi' \phi' ds \\ &\quad - M_p(L + r_p) \int_0^L \phi' \phi'^T dx - M_p r_p L \phi'_L \phi'^T_L \end{aligned}$$

In a matrix form,

$$T = \frac{1}{2} \underbrace{\begin{pmatrix} \dot{\theta} & \dot{q}_f^T \end{pmatrix}}_{\dot{q}^T} \underbrace{\begin{pmatrix} I_t + q_f^T C_{rr} q_f & \beta^T \\ \beta & M_{ff} \end{pmatrix}}_{M(q)} \underbrace{\begin{pmatrix} \dot{\theta} \\ \dot{q}_f \end{pmatrix}}_{\dot{q}} \quad (16)$$

The $q_f^T C_{rr} q_f$ element in the mass matrix $M(q)$ is of second order in v . It will be neglected. The potential energy (3) discretization gives:

$$\begin{aligned} V &= \frac{1}{2} q_f^T \int_0^L EI_z \phi'' \phi''^T dx q_f - \left(\frac{1}{2} \rho g L^2 + M_p g(L + r_p) \right) \cos \theta \\ &\quad + \frac{1}{2} g \cos \theta q_f^T \left(\rho \int_0^L \int_0^x \phi' \phi'^T ds dx + M_p \left[\int_0^L \phi' \phi'^T dx + r_p \phi'_L \phi'^T_L \right] \right) q_f \\ &\quad + g \sin \theta \left(\int_0^L \rho \phi^T dx + M_p (\phi_L^T + r_p \phi'^T_L) \right) q_f \\ &= \frac{1}{2} q_f^T K_{ff} q_f - G_{rr} \cos \theta + \frac{1}{2} \cos \theta q_f^T G_{ff} q_f + \sin \theta G_{rf}^T q_f \end{aligned} \quad (17)$$

where

$$\begin{aligned}
 K_{ff} &= EI_z \int_0^L \phi'' \phi''^T dx \\
 G_{rr} &= \frac{1}{2} \rho g L^2 + M_p g (L + r_p) \\
 G_{rf} &= \int_0^L \rho g \phi dx + M_p g (\phi_L + r_p \phi'_L) \\
 G_{ff} &= \rho g \int_0^L (L-x) \phi' \phi'^T dx + M_p g \int_0^L \phi' \phi'^T dx + M_p g r_p \phi'_L \phi'^T_L
 \end{aligned}$$

The discretization of Rayleigh dissipation function (4) gives:

$$\begin{aligned}
 R &= \frac{1}{2} b_m \dot{q}_r^2 + \frac{1}{2} \dot{q}_f^T k_e \int_0^L EI_z \dot{v}''^2 dx \dot{q}_f \\
 &= \frac{1}{2} \underbrace{\begin{pmatrix} \dot{\theta} & \dot{q}_f^T \end{pmatrix}}_{\dot{q}^T} \underbrace{\begin{pmatrix} b_m & 0 \\ 0 & B_{ff} \end{pmatrix}}_B \underbrace{\begin{pmatrix} \dot{\theta} \\ \dot{q}_f \end{pmatrix}}_{\dot{q}}
 \end{aligned} \quad (18)$$

where $B_{ff} = k_e K_{ff}$.

We then introduce these expressions into the Lagrangian $\mathcal{L} = T - V$, and we apply Lagrange equations,

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} + \frac{\partial R}{\partial \dot{q}_j} = Q_j, \quad j = 1, 2, \dots, n \quad (19)$$

where q_j is the generalized coordinates, Q_j are the generalized forces, $j = 1, 2, \dots, n$ and $n = \nu + 1$ is the total number of generalized coordinates. In Lagrange equations, $q_f(t)$ is the generalized coordinate vector associated with the flexibility. Let q_r be the generalized coordinate associated with the base movement (rigid coordinate), i.e. $q_r(t) = \theta(t)$. The generalized coordinate vector is then given by:

$$q(t) = [q_r(t), q_f^T(t)]^T$$

The spatially discrete dynamical model is then given by:

$$T_m = I_t \ddot{q}_r + \beta^T \ddot{q}_f + b_m \dot{q}_r + G_{rr} \sin q_r + \cos q_r G_{rf} q_f + 2\dot{q}_r \dot{q}_f^T C_{rr} \dot{q}_f + \ddot{q}_r \dot{q}_f^T C_{rr} q_f \quad (20)$$

$$0 = \beta \ddot{q}_r + M_{ff} \ddot{q}_f + B_{ff} \dot{q}_f - \dot{q}_r^2 C_{rr} q_f + K_{ff} q_f + g \cos q_r G_{ff} q_f + g \sin q_r G_{rf} \quad (21)$$

Written in a matrix form, equations (20)-(21) become:

$$\begin{aligned}
 &\underbrace{\begin{pmatrix} I_t + q_f^T C_{rr} q_f & \beta^T \\ \beta & M_{ff} \end{pmatrix}}_{M(q)} \underbrace{\begin{pmatrix} \ddot{q}_r \\ \ddot{q}_f \end{pmatrix}}_{\ddot{q}} + \underbrace{\begin{pmatrix} b_m & 0 \\ 0 & B_{ff} \end{pmatrix}}_B \underbrace{\begin{pmatrix} \dot{q}_r \\ \dot{q}_f \end{pmatrix}}_{\dot{q}} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & K_{ff} \end{pmatrix}}_K \underbrace{\begin{pmatrix} q_r \\ q_f \end{pmatrix}}_q \\
 &+ \underbrace{\begin{pmatrix} G_{rr} & G_{rf}^T \\ G_{rf} & G_{ff} \end{pmatrix}}_{G_1} \underbrace{\begin{pmatrix} \sin q_r \\ q_f \cos q_r \end{pmatrix}}_{G(q)} + \underbrace{\begin{pmatrix} 2\dot{q}_r \dot{q}_f^T C_{rr} \dot{q}_f \\ -\dot{q}_r^2 C_{rr} q_f \end{pmatrix}}_{C(q,\dot{q})} = \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_L T_m
 \end{aligned} \quad (22)$$

or also in the compact form:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + B\dot{q} + Kq + G(q) = L T_m \quad (23)$$

where M , B , and K are respectively the mass, damping, and rigidity matrices, and $C(q, \dot{q})\dot{q}$, and $G(q)$ are the Coriolis and centrifugal, and gravity force vectors. M , B and K_{ff} are positive definite symmetric matrices. The associated matrix with the gravity, G_1 , is also symmetric.

Proposition 1. *The mass matrix, M , and the Coriolis and centrifugal force vector, $C(q, \dot{q})\dot{q}$, verify the following:*

$$\dot{M}(q) - 2C(q, \dot{q}) = S \quad (24)$$

where S is an antisymmetric matrix.

Proof. Let

$$C(q, \dot{q}) = \begin{pmatrix} \dot{q}_f^T C_{rr} \dot{q}_f & \dot{q}_r \dot{q}_f^T C_{rr} \\ -\dot{q}_r C_{rr} \dot{q}_f & 0 \end{pmatrix} \quad (25)$$

Then,

$$x^T (\dot{M}(q) - 2C(q, \dot{q})) x = 0 \quad \forall x \in \mathcal{R}^{(v+1)}$$

where v is the flexible modes number, and \mathcal{R} is the real set numbers. ∇

By neglecting the 2nd order elements (relative to v) in (23), i.e. in the mass matrix and the gravity element G_{ff} , (23) becomes:

$$M\ddot{q} + C(q, \dot{q})\dot{q} + B\dot{q} + Kq + G(q) = L T_m \quad (26)$$

5. Description of Admissible Functions

In this section, we describe two types of admissible functions, namely, global functions defined over the entire beam length (such as beam eigenfunctions and polynomial functions) and piecewise polynomial functions (such as cubic splines and cubic B-splines).

5.1 Global Admissible Functions

5.1.1 Beam Eigenfunctions

The analytical solution of the eigenfunctions of the system shown in Figure 1 are given by (Saad et al., 2006):

$$\phi_i(x) = A_i \left(\mu_3 \lambda_i^3 (\cos \beta_i x - \cosh \beta_i x) + (1 + c_i) \sin \beta_i x + (1 - c_i) \sinh \beta_i x - 2\beta_i x \right) \quad (27)$$

Here, $\beta_i = \frac{\lambda_i}{l}$ and λ_i is the solution of the following characteristic equation :

$$\begin{aligned} & (\mu_1 \mu_2 \mu_3 + \mu_2 \mu_3 \mu_5 - \mu_3 \mu_4^2) \lambda^7 (1 - C Ch) - \mu_3 (\mu_1 + \mu_5) \lambda^6 (S Ch + C Sh) - 2\mu_3 \mu_4 \lambda^5 S Sh \\ & + (\mu_2 \mu_5 + \mu_1 \mu_2 + \mu_2 \mu_3 - \mu_4^2) \lambda^4 (C Sh - S Ch) + \mu_3 \lambda^3 (1 + C Ch) + 2(\mu_1 + \mu_5) \lambda^3 C Ch \\ & + 2\mu_4 \lambda^2 (C Sh + S Ch) + 2\mu_2 \lambda S Sh + S Ch - C Sh = 0 \end{aligned} \quad (28)$$

where:

$$\mu_1 = \frac{I_p}{\rho l^3}, \mu_2 = \frac{m_p}{\rho l}, \mu_3 = \frac{I_m}{\rho l^3}, \mu_4 = \frac{m_p r_p}{\rho l^2}, \mu_5 = \frac{m_p r_p^2}{\rho l^3},$$

$$C = \cos \lambda, \quad S = \sin \lambda, \quad Ch = \cosh \lambda, \quad Sh = \sinh \lambda.$$

The admissible shape functions are generally the eigenfunctions of a simpler but related problem. The eigenfunctions of a rotating beam with a payload concentrated at its end are obtained from (27) and (28) by replacing the payload center of mass r_p by zero. The eigenfunctions of a clamped beam with or without payload are deduced by taking the motor inertia I_m as infinity, or the payload inertia I_p and mass m_p as zero, respectively. The first four eigenfunctions of a rotating beam are shown in Figure 2.

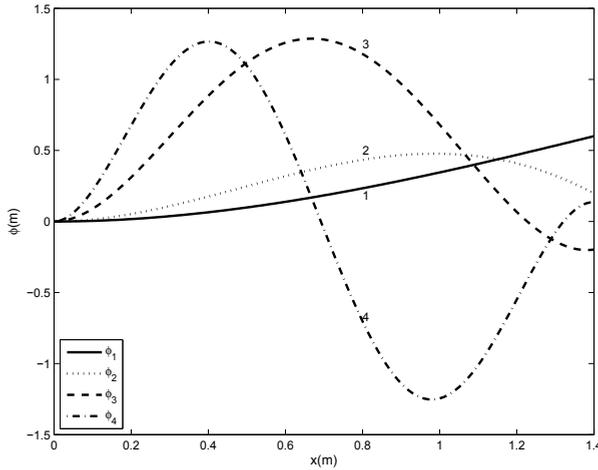


Fig. 2. First four eigenfunctions of a rotating beam

5.1.2 Polynomial functions

In this case, the beam deformation is given by

$$v(x, t) = \sum_{i=1}^v \left(\frac{x}{l}\right)^{i+1} \eta_i(t), \quad (29)$$

where, the BCs at $x = 0$ ($v_0 = v'_0 = 0$) are considered. The vector of the assumed-modes is

$$\boldsymbol{\phi}^T(x) = \left[\left(\frac{x}{l}\right)^2 \quad \dots \quad \left(\frac{x}{l}\right)^{i+1} \quad \dots \quad \left(\frac{x}{l}\right)^v \quad \left(\frac{x}{l}\right)^{v+1} \right]. \quad (30)$$

The polynomial functions are very simple. Figure 3 shows the first four polynomial functions. It is known that the sets of polynomial functions and beam eigenfunctions are complete, i.e., the error between the actual and the estimated eigenvalues can be rendered as small as desired by increasing the number of terms in the series (14) (Mirovitch, 1967). Therefore, the computed eigenvalues approach the actual ones from above as the number of admissible functions is increased.

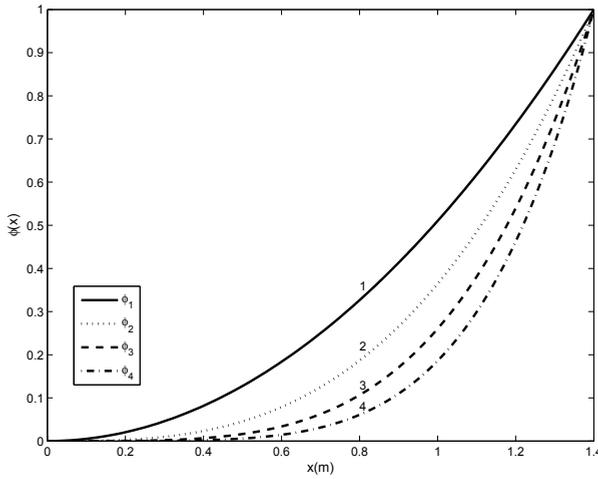


Fig. 3. First four polynomial functions

5.2 Piecewise admissible functions

The cubic splines and cubic B-splines are used as interpolation functions. In earlier studies, cubic splines were used in modeling specific types of robot manipulators by adjusting the BCs at both extremities (Cho et al., 1991). It is shown next that the cubic splines can be used simply as a set of assumed-modes by fixing the conditions at the link extremity, $x = l$, independently of the payload.

5.2.1 Description of the cubic splines

On the interval $x_i \leq x < x_{i+1}$, the deformation is approximated by a cubic spline function as (Gerald & Wheatley, 2004)

$$v_i(x, t) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \quad (31)$$

The deformation v and the curvature v'' at $x = x_i$ are y_i and y_i'' . The deformation and the curvature at $x = x_{i+1}$ are y_{i+1} and y_{i+1}'' . The continuity of the deformation and the curvature at the internal nodes gives the following relations:

$$\begin{aligned} a_i &= \frac{y_{i+1}'' - y_i''}{6h_i}, \\ b_i &= \frac{y_i''}{2}, \\ c_i &= \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i y_i'' + h_i y_{i+1}''}{6}, \\ d_i &= y_i, \end{aligned}$$

where $h_i = x_{i+1} - x_i$. The continuity condition of the slope at $x = x_i$ ($v'_i(x_i, t) = v'_{i-1}(x_i, t)$) gives the following equation:

$$\begin{aligned} h_{i-1} y''_{i-1} + 2(h_{i-1} + h_i) y''_i + h_i y''_{i+1} \\ = 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right). \end{aligned} \quad (32)$$

Equation (32) is applicable to each of the internal nodes, from $i = 2$ to $n - 1$, where n is the total number of nodes. This gives $n - 2$ equations relating the n values of y''_i and y_i . Two additional equations are obtained by specifying the conditions at both extremities. If the beam is assumed to be clamped into the base, then $v(0, t) = y_1 = 0$ and $v'(0, t) = 0$. It follows that:

$$2h_1 y''_1 + h_1 y''_2 = 6 \frac{y_2}{h_1}. \quad (33)$$

For the distal extremity $x = l$, the conditions are to some extent arbitrary. Four choices are possible:

1. Taking $y''_n = 0$, which amounts to assuming no bending moment;
2. Taking $y''_n = y''_{n-1}$, which is equivalent to a constant bending moment on the last element;
3. Taking y''_n as a linear extrapolation from y''_{n-1} and y''_{n-2} which yields

$$y''_n = \frac{(h_{n-2} + h_{n-1}) y''_{n-1} - h_{n-1} y''_{n-2}}{h_{n-2}};$$

4. Keeping the curvature y''_n at $x = l$ as an unknown. This choice is only applicable when the curvatures are taken as general coordinates as shown later.

An interesting feature of the cubic splines is the relation between the node curvatures and displacements. Indeed, combining (32) with the conditions for the extremities gives the following equation:

$$\mathbf{AY}'' = 6\mathbf{CY} \quad (34)$$

where, \mathbf{Y}'' and \mathbf{Y} are the vectors of node curvatures and displacements. Equation (34) can be solved either for the displacement \mathbf{Y} or the curvature \mathbf{Y}'' . Therefore, the generalized coordinates used in the discrete model can be either the node curvatures or the node displacements. The use of curvatures presents some advantages in controlling flexible manipulators since these are directly measurable using strain gauges.

For a flexible link that is part of a general manipulator, the load at the link extremity is not known a priori. Therefore, the bending moment, which is related to the curvature, is unknown. When (34) is solved in terms of the node displacement \mathbf{Y} , the curvature at $x = l$ may remain unknown (condition 4 above). It means that the $n - 1$ displacements y_i are expressed in terms of the n curvatures y''_i .

An additional manipulation is required to put the cubic splines in a form suitable for the assumed-modes. A cubic spline is defined on each individual interval. To obtain the deformation v at any point x between 0 and l , the Heaviside function $\mu(x)$ is used ($\mu(x) = 1$ when $x \geq 0$ and $\mu(x) = 0$ when $x < 0$). The deformation v is thus written as

$$\begin{aligned} v(x, t) = & \sum_{i=1}^{n-1} v_i (\mu(x - x_i) - \mu(x - x_{i+1})) \\ & + v_{n-1} \mu(x - x_{n-1}). \end{aligned} \quad (35)$$

It is then possible to express (35) in terms of the generalized flexible coordinates η (either the node displacements or curvatures) as follows:

$$v(x, t) = \underbrace{\left(\frac{\partial v(x, t)}{\partial \mathbf{Y}''(t)} \right)^T}_{\boldsymbol{\phi}^T(x)} \underbrace{\mathbf{Y}''(t)}_{\boldsymbol{\eta}(t)} \quad \text{or}$$

$$v(x, t) = \underbrace{\left(\frac{\partial v(x, t)}{\partial \mathbf{Y}(t)} \right)^T}_{\boldsymbol{\phi}^T(x)} \underbrace{\mathbf{Y}(t)}_{\boldsymbol{\eta}(t)}, \quad (36)$$

where each component of the vector $\boldsymbol{\phi}(x)$ is an assumed-mode function $\phi_i(x)$.

As an example, two cases are presented to illustrate the application of the cubic splines. The system used is the flexible rotating beam shown in Figure 1 where the link is divided into three finite elements. In the first case, the second condition ($Y_4'' = Y_3''$) is used to represent the curvature at the link extremity. Equation (34) becomes

$$\underbrace{\begin{bmatrix} 2h_1 & h_1 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 \\ 0 & h_2 & 2h_2 + 3h_3 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} y_1'' \\ y_2'' \\ y_3'' \end{bmatrix}}_{\mathbf{Y}''}$$

$$= 6 \underbrace{\begin{bmatrix} \frac{1}{h_1} & 0 & 0 \\ -\frac{1}{h_1} - \frac{1}{h_2} & \frac{1}{h_2} & 0 \\ \frac{1}{h_2} & -\frac{1}{h_2} - \frac{1}{h_3} & \frac{1}{h_3} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\mathbf{Y}}. \quad (37)$$

This equation can be solved either for the node curvatures \mathbf{Y}'' or for the node displacements \mathbf{Y} . If, in the second case, the curvature at the extremity, Y_4'' , is left as an unknown, then the matrix \mathbf{C} remains as in (37), while the matrix \mathbf{A} becomes

$$\mathbf{A} = \begin{bmatrix} 2h_1 & h_1 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 \end{bmatrix}. \quad (38)$$

Equation (37) can only be solved for the node displacements while the node curvatures are the generalized coordinates.

Figure 4a illustrates the three assumed-modes when the node displacements are used as coordinates. Figure 4b shows the four assumed-modes when the node curvatures are used as coordinates while the curvature at the last node y_4'' is kept unknown.

The cubic interpolation functions verify the completeness and the continuity requirements for convergence. Therefore, one should expect that the solution of the resulting eigenvalue problem is, in the limit as $h \rightarrow 0$, convergent. However, monotonic convergence cannot be guaranteed. This is because two coordinates, displacement and curvature, are added for each element with each mesh refinement. Indeed, Mirovitch and Silverberg (1983) proposed two bracketing theorems characterizing the non-monotonic convergence of the eigensolution of the h-version of the FEM using Hermite cubics. It is shown in (Saad, 2004) that the eigensolution for the h-version of the FEM using cubic splines is monotonically convergent.

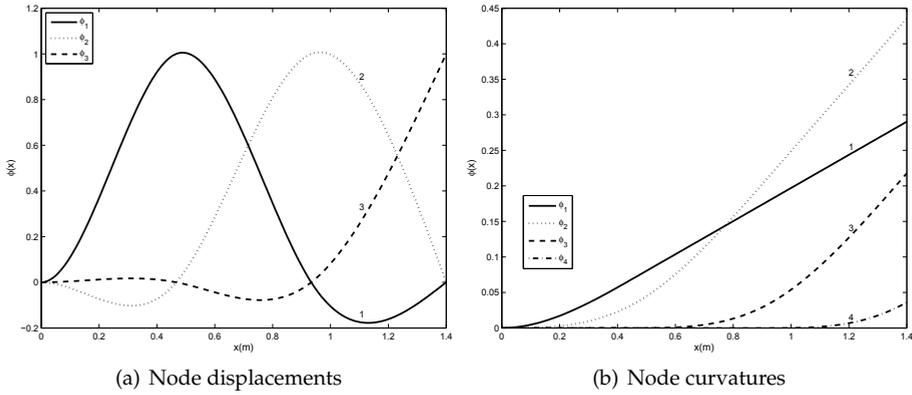


Fig. 4. Cubic splines defined on three intervals

5.2.2 Description of the cubic B-splines

The cubic B-splines are used as interpolation functions in the Rayleigh-Ritz method and as assumed modes for flexible manipulators (Yang & Gibson, 1989). A cubic B-spline is defined on four intervals with a cubic spline on each interval as follows:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad (39)$$

for $x_{i-1} \leq x < x_i$, $i = 1, \dots, 4$. Since each cubic spline requires four parameters, a total of 16 parameters are needed. If the B-splines are different from zero on each interval, then continuity of the splines and their first two derivatives yields nine separate conditions. Six out of the remaining seven conditions can be found by fixing the three BCs at each extremity (namely v , v' , and v''). The last condition can be fixed arbitrarily at the middle node, e.g., by specifying unit displacement. A cubic B-spline can be zero on some of the four intervals.

For n intervals, the required number N of cubic B-splines is given by the following equation:

$$N = n - \nu_0 - \nu_1 + 3, \quad (40)$$

where, ν_0 and ν_1 are the numbers of BCs for $S(x)$, $S'(x)$ or $S''(x)$ being equal to zero at $x = 0$ and at $x = l$. For example, a beam clamped into a base and having a payload at its extremity has two zero BCs at $x = 0$ ($S(0) = S'(0) = 0$) and none at $x = l$. Therefore, $\nu_0 = 2$ and $\nu_1 = 0$. If the beam is divided into two intervals ($n = 2$), then three cubic B-splines are required. As an example, we develop next a cubic B-spline defined on two intervals. The internal and BCs are:

$$\begin{aligned} S(x) = S'(x) = S''(x) = S'''(x) = 0 & \text{ for } x_0 \leq x < x_1, \\ S(x_1) = S'(x_1) = 0, \\ S(x), S'(x), S''(x) & \text{ are continuous at } x = x_2, \\ & \text{and } S(x_2) = 1, \\ S(x_3) = S''(x_3) = 0, \\ S(x) = S'(x) = S''(x) = S'''(x) = 0 & \text{ for } x_3 \leq x < x_4. \end{aligned}$$

It can be verified that 16 conditions are imposed. The Heaviside function is again used to insure that the functions are defined over the entire beam length. The B-spline verifying the above conditions is

$$\begin{aligned} \phi(x) = & \left(\frac{-11x^3 + 51x^2 - 69x + 29}{7} \right) \mu(x-1) \\ & + \left(\frac{16x^3 - 96x^2 + 192x - 128}{7} \right) \mu(x-2) \\ & + \left(\frac{-5x^3 + 45x^2 - 123x + 99}{7} \right) \mu(x-3). \end{aligned} \quad (41)$$

The choice of the B-splines must respect the geometric BCs. If any dynamic BC is zero, e.g., no inertial load, then taking this into account in the B-splines improves the solution convergence. Figure 5 shows the four cubic B-splines chosen as assumed modes when three intervals are used.

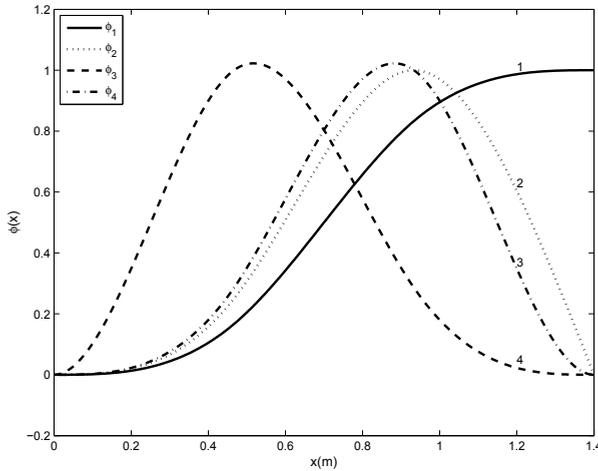


Fig. 5. Four cubic B-spline functions

6. Modal Analysis

The goal of this section is to study the accuracy of the different shape functions based models discussed in this paper in modeling the one-link robot system shown in Figure 1. First, a comparison in the frequency domain is carried out for frequencies, mode shapes and spatial derivative of mode shapes. Then, we compare the static mode shapes and their spatial derivatives (for $\theta = \pi/2$ as defined in the model assumptions). We also discuss the computational complexity to generate these functions.

Table 2 describes the shape functions used in the comparison. The determination of the beam eigenfunctions requires the knowledge of the system parameters. We are using the nominal parameters in these cases. On the other hand, the spline and polynomial shape functions require only the knowledge of the beam length. Still, for the cubic spline functions, we have

to make a choice for the BCs at the extremity. For the cubic B-splines, when there are less than four intervals, different B-splines can satisfy the geometric BCs (Saad et al., 2006).

<i>cf</i>	Eigenfunctions of a clamped-free beam using the nominal parameters
<i>cp</i>	Eigenfunctions of a clamped-payload beam using the nominal parameters
<i>bp</i>	Eigenfunctions of a rotating beam using the nominal parameters
<i>cbs</i>	Cubic B-splines
<i>cs-d</i>	Cubic splines with displacements as coordinates and with the curvature at the last node n being equal to the curvature at node $n - 1$
<i>cs-c</i>	Cubic splines with curvatures as coordinates and the curvature at the last node is kept unknown
<i>pol</i>	Polynomial functions

Table 2. Shape functions used in the model comparison

To test the robustness of the shape functions given in Table 2, we vary the system parameters as presented in Table 3. In Case 1, the payload is removed. In Case 2, the motor inertia is reduced by a factor of one thousand. This is equivalent to removing a 32:1 reduction gear. In Case 3, the payload mass and inertia are increased by factors of 2.5 and 2.8 respectively. Additionally, the payload center of mass is moved further from the base to a position equivalent to 50% of the beam length. This provides insight into the more general two-link problem as the payload may be viewed as a rigid, locked, second link.

Parameter	Nominal	Case 1	Case 2	Case 3
I_m ($kg\ m^2$)	10	10	0.01	10
m_p (kg)	2	0	2	5
I_p ($kg\ m^2$)	0.36	0	0.36	1
r_p (m)	0	0	0	0.7

Table 3. Variation of the nominal parameters

6.1 Performance indices for comparison

For each case, we compute the exact solution for the frequency comparison using the eigenfunctions of the rotating beam given by (27) and (28). For the static comparison, we compute the exact static deformation of a horizontal clamped-payload beam, by solving equations (9)-(12) with null time variations, which leads to:

$$s(x) = -\frac{g x^2}{24 E I_z} \times \left(\rho(x^2 - 4 x l + 6 l^2) + m_p(12 l + 12 r_p - 4 x) \right) \quad (42)$$

and get the estimated solutions using the spatially discretized model (23). The eigenfrequencies and eigenvectors are the solution of the following algebraic eigenvalue problem:

$$\mathbf{K}\mathbf{u} = \tilde{\omega}^2 \mathbf{M}\mathbf{u}, \quad (43)$$

where, \mathbf{M} and \mathbf{K} are the mass and the stiffness matrices given in (23), $\tilde{\omega}$ is the estimated frequency, and \mathbf{u} is the eigenvector associated with that frequency. Since damping is negligible, the rigid mode is always zero. It will not be included in the comparison results. We combine the eigenvectors corresponding to the flexible coordinates $\boldsymbol{\eta}$ in the following matrix:

$$\mathbf{U}_f = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_\nu],$$

where ν is the number of the generalized coordinates. The following equation gives the eigenmodes or mode shapes in terms of the eigenvectors:

$$\tilde{Y}(x) = \mathbf{U}_f^T \boldsymbol{\phi}(x), \quad (44)$$

with $\boldsymbol{\phi}(x)$ the vector of shape functions. The first three derivatives of the mode shapes are computed using $\boldsymbol{\phi}'(x)$, $\boldsymbol{\phi}''(x)$ and $\boldsymbol{\phi}'''(x)$. The modes and their first derivatives are normalized such that their maximum is equal to one. The second and the third mode derivatives are normalized to equal one at $x = 0$.

We compute the relative error between the exact and the estimated frequencies using the following expression:

$$\epsilon_{f,i} = 100 \frac{\tilde{f}_i - f_i}{f_i}, \quad (45)$$

with \tilde{f}_i and f_i the approximate and the exact values of the i th frequency in Hz.

To compare the mode shapes, we use the average of the absolute error divided by the average of the absolute value of the exact mode shape:

$$\epsilon_{m,d,i} = 100 \frac{\int_0^l |\tilde{Y}_{d,i}(x) - Y_{d,i}(x)| dx}{\int_0^l |Y_{d,i}(x)| dx}, \quad (46)$$

where $d = 0, \dots, 3$ denotes the mode shape derivative and i is the mode number. \tilde{Y} and Y are the approximated and the exact mode shapes.

The estimated static deformation is evaluated using (14) and (23) as follows:

$$\tilde{s}(x) = \boldsymbol{\phi}(x)^T \mathbf{K}_f^{-1} \mathbf{h}_f. \quad (47)$$

The first three derivatives of the static mode shapes are obtained using $\boldsymbol{\phi}'(x)$, $\boldsymbol{\phi}''(x)$ and $\boldsymbol{\phi}'''(x)$. The error is similar to the one calculated for the mode shapes and is given by

$$\epsilon_{s,d,i} = 100 \frac{\int_0^l |\tilde{s}_{d,i}(x) - s_d(x)| dx}{\int_0^l |s_d(x)| dx}. \quad (48)$$

6.2 Comparison results

For each of the shape functions given in Table 2, we generated models using one to eight shape functions (2 to 8 for the Splines) for each of the four cases for a total of 212 different models. These models are created with Maple using the symbolic modeling programs SYMOFROS (Piedboeuf, 1996) and the analysis is done using MATLAB. For all studied cases, the exact frequencies are generated symbolically and computed numerically in MAPLE with a 30 digits precision to avoid truncation in Matlab. In this paper, we study the improvement in the solution when using one to eight shape functions. We are presenting only the results for the first three modes but the discussion applies to higher modes as well.

6.2.1 Non-Concentrated Payload (Case 3)

Since Case 3 is the most challenging one, we discuss it in more details. Figure 6 gives the absolute eigenvalue relative errors in (45) for the shape functions presented in Table 2. The exact frequencies for the first three modes are 1.6, 9.3 and 45.3 Hz. On each graph, three curves appear for each shape function. The lowest one corresponds to the first mode and the highest one to the third mode. For example, three *cs-c* shape functions give an error of 0.05% on the second frequency (Figure 6b). The graphs show the solution improvement when the number of shape functions increases. The curves for *bp* and *cp* models are very close, especially for the second and third modes (Figure 6a). For the first frequency, the precision of *cp* and *bp* models does not improve after three shape functions owing to numerical errors. The curves for *cbs* and *cs-c* models are almost identical for the three modes. The *cp* and *bp* models give smaller errors for the first and second modes than the *cbs* and *cs-c* models but not for the third mode (Figure 6a vs. 6b). The error for the *cf* model barely goes under 1% (Figure 6a). The error for the *cs-d* first mode is higher than the second mode error of the *cs-c* model (Figure 6b). For the first mode, the *pol* model gives results similar to the *cbs* model up to five shape functions: the results are even identical when two shape functions are used (Figure 6c). Afterward, the precision diminishes drastically mainly because the powers $(\frac{x}{l})^k$ are nearly linearly dependent since they essentially have the same weight in the neighborhood of $x = l$.

Figure 7 shows the relative errors on the eigenmodes while Figure 8 gives the error on the second derivatives. The second derivative is important in experimentation since it is related to the curvature read by strain-gauges. The observations done for the frequencies hold for the eigenmodes and their second derivative. However, the error on the eigenmode approximation is higher than the one on the frequencies. The error on the second derivative is even higher. For example, the error on the first mode of the *cf* model for the second derivative (Figure 8a) goes barely under 10% even with eight shape functions. The errors on the *pol* model for the second derivative sometime increase when a shape function is added (Figure 8c). The error on the eigenmodes for the *pol* model are close to *cbs* and *cs-c* models up to four shape functions. Figure 9 shows the static shape errors while Figure 10 gives the errors on the second derivative of the static shape functions. Clearly, the *pol* model gives the best result, especially for three shape functions. This was expected since the exact solution (Equation 42) is a fourth order polynomial. The *cp*, *bp*, *cbs* and *cs-c* models have similar errors (*cbs* and *cs-c* are even super-imposed). The error for the *cf* model is important especially for the second derivative where the lowest error is around 6% with eight shape functions. The *cs-d* model does not perform as well as the two others spline models (*cbs* and *cs-c*).

6.2.2 Nominal, No-Payload (Case 1) and Reduced Inertia (Case 2)

Globally, the above observations for decentralized payload (Case 3) hold also for the Nominal, the No-Payload (Case 1), and for Reduced Inertia (Case 2) cases. In particular, for the Nominal Case, as expected the *bp* model gives the best results. It represents the exact solution of the system at hand. The eigenfrequencies and eigenmodes errors are due to numerical errors since the exact frequencies are computed symbolically in MAPLE with a 30 digits precision. The behavior of the other models are generally the same as for Case 3. Still, the convergence rate of the *cp* model is better in Case 3. Additionally, the error on the first frequency for the *cs-c* model is higher in Case 3. The case with no payload (Case 1) is the only one where the *cf* model gives good results. The other models that do not assume a zero bending torque at the extremity take longer to converge and the errors are larger than those in Case 3. For the

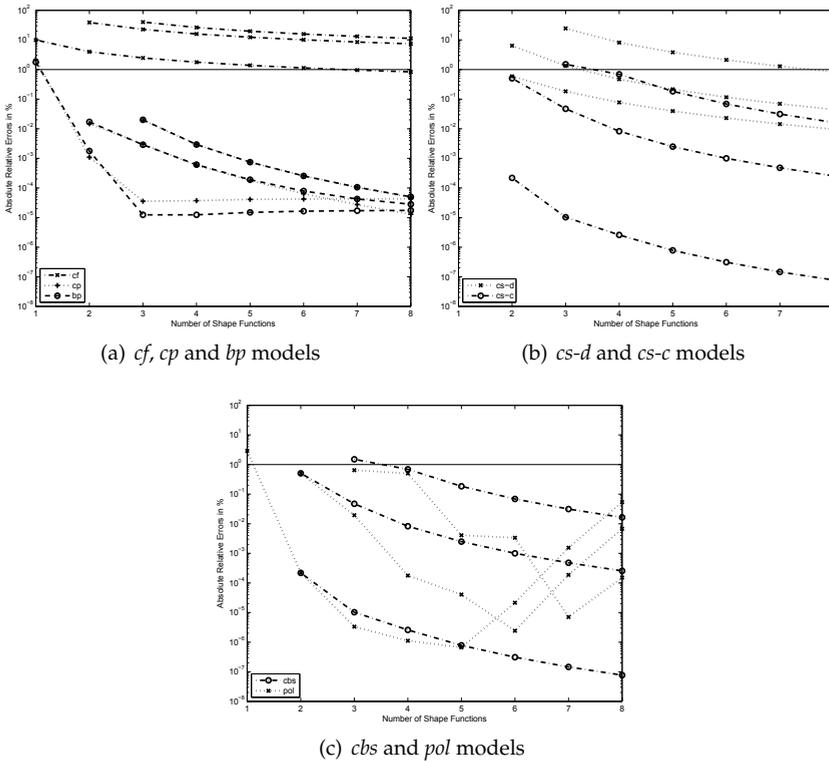


Fig. 6. Absolute relative errors on the eigenvalues in % for Case 3

spline models, it will be easy to generate a spline having zero moment at the extremity. Such splines improve the results. When the base inertia is reduced by a factor of 1000 (Case 2), the convergence rate of the selected parameters is slower and more shape functions are required to reach the same precision as compared to Case 3.

6.2.3 Sensitivity of the models

Table 4 gives the condition number of the eigenvector matrix, $CN_2(V)$ (Watkins, 1991), of the different models, from one to eight shape functions. These condition numbers measure the sensitivity of the eigenvalues to the system parameter variations. They indicate by how much a variation in the $M^{-1}K$ matrix will be amplified (a condition number of one being the best case). Table 4 reveals that the *pol* model becomes very sensitive when more shape functions are added. Therefore, while the solution should improve by adding shape functions, the sensitivity to numerical and parameter errors wipes off this improvement. The sensitivity of the *cf* and *cs-d* models and, to a less extent, of *cbs* model also increases when the number of shape functions increases. On the other hand, the sensitivities of the *cp*, *bp* and *cs-c* models stay almost constant. The *cs-c* model with the lowest condition number is then the less sensitive to model error.

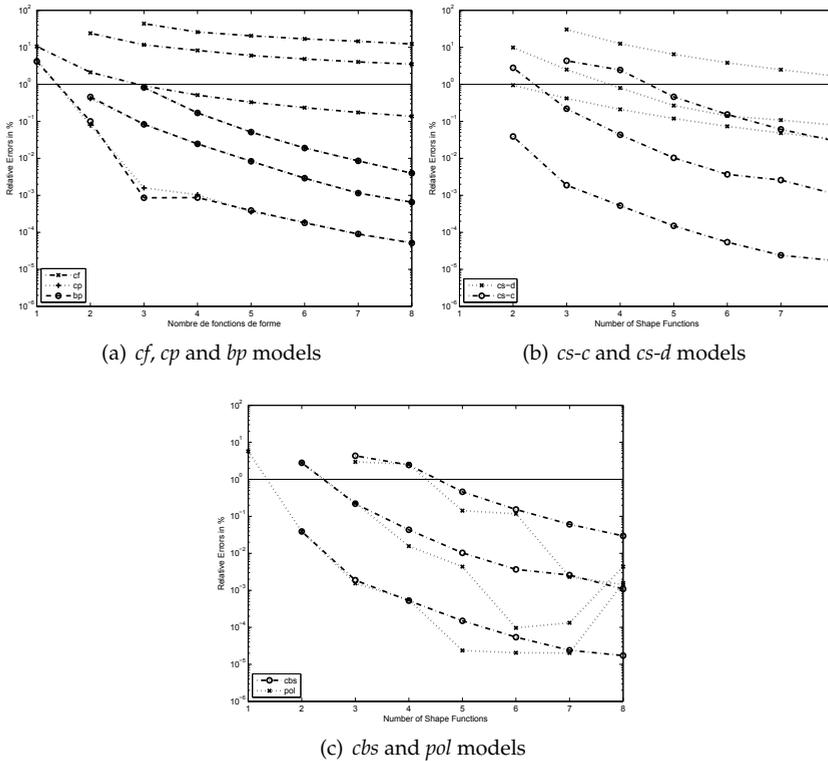


Fig. 7. Relative errors on the eigenmodes in % for Case 3

7. Conclusion

In this work, the mathematical model of a flexible link manipulator using the assumed-mode method to discretize the link flexibility was presented. A detailed comparison of several admissible shape functions in assumed-mode models was carried out on a flexible slewing beam for the eigenvalues, the eigenmodes and their derivatives, and the static deformations and their derivatives. Load parameters were changed from their nominal values to test the sensitivity of the shape functions based models. The study confirms the fact that the completeness of the admissible functions guarantees convergence of the approximate models but the convergence rate can be slow. In clamped-free case, it was revealed that the poor convergence is related to the inability of satisfying the natural BCs with a finite number of assumed-modes. The comparisons show that for the system of a one flexible link: (1) the clamped-free eigenfunctions are mostly inadequate. They only behave well when there is no payload; (2) the clamped-payload eigenfunctions are good candidates even when the payload parameters are changed; (3) in comparison to the clamped-payload eigenfunctions, the complexity of the rotating beam eigenfunctions did not translate into marked rate convergence improvement; (4) the polynomial functions are very attractive and are generally good for a small number of shape functions, but they are too sensitive to system parameters variations; (5) both the cu-

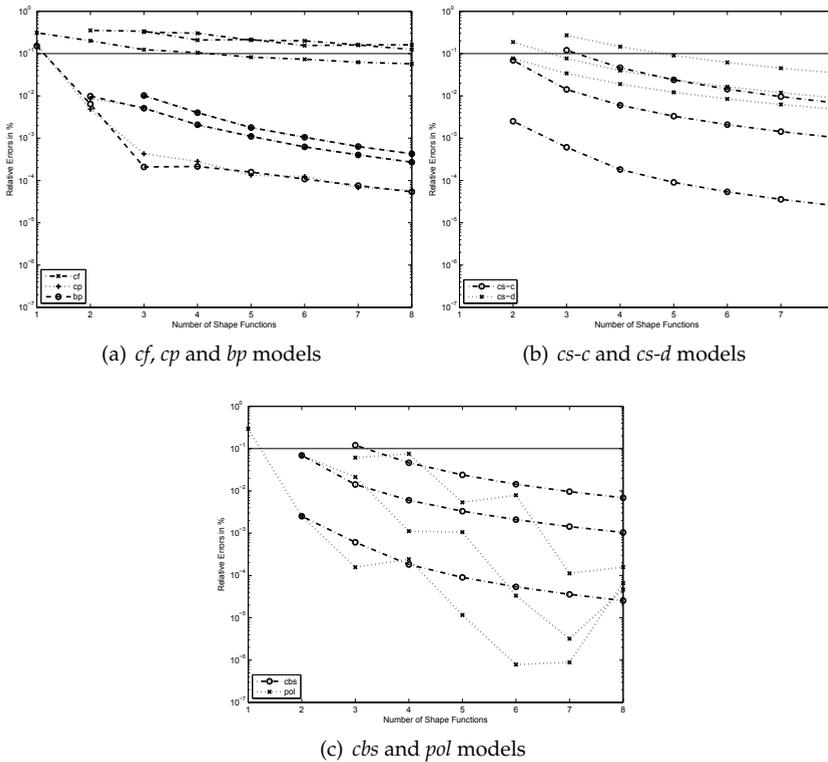


Fig. 8. Relative errors on the 2nd derivative of the eigenmodes in % for Case 3

bic B-splines and the cubic splines with curvatures as coordinates offer identical behavior; (6) the cubic splines with curvatures as coordinates provides better results than the cubic splines with displacements as coordinates for all the convergence criteria; (7) overall, the cubic splines using curvatures as generalized coordinates offer the best compromise between good precision and low calculation complexity. It is also worth to mention that based on the results of this paper, the cubic spline functions are now utilized in Symofros in the discretization of the deformation of Canada Flexible Arm at the Space Canadian Agency rather than the beam eigenfunctions or the polynomial functions that were previously used.

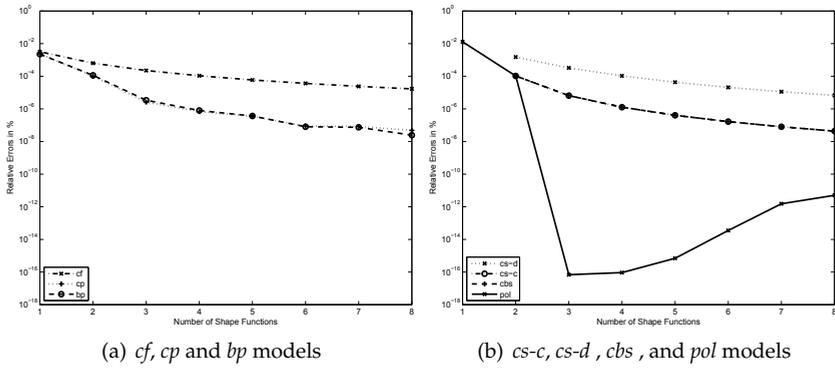


Fig. 9. Relative errors on the static shapes in % for Case 3

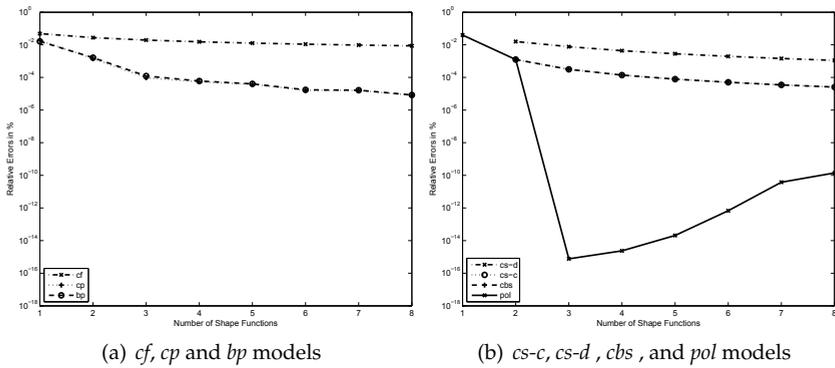


Fig. 10. Relative errors on the 2nd derivative of the static shape in % for Case 3

	1	2	3	4	5	6	7	8
<i>cf</i>	2.3	6.1	10.7	16.1	22.3	29.2	36.6	44.7
<i>cp</i>	1.4	2.5	2.5	2.5	2.5	2.5	2.5	2.5
<i>bp</i>	1.5	2.2	2.2	2.2	2.2	2.2	2.2	2.2
<i>cbs</i>	–	2.3	4.9	3.6	4.8	6.2	7.7	9.3
<i>cs-c</i>	–	1.7	1.7	1.7	1.6	1.6	1.6	1.6
<i>cs-d</i>	–	5.2	8.4	12.1	16.3	21.1	26.2	31.6
<i>pol</i>	1.9	4.8	11.6	24.4	54.5	253	1.3e3	7.3e3

Table 4. Condition number on the eigenvalue matrix for Case 3

8. References

- Barbieri, E. and Özgüner, U. (1988). Unconstrained and Constrained Mode Expansions for a Flexible Slewing Link, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 110, No. 4, 416–421.
- Book, W. and Oberfell, K. (2000). Practical Models for Practical Flexible Arms, *Proc. 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April, 835–842.
- F. Boyer, N. Glandais, and W. Khalil, (2002). Flexible Multibody Dynamics Based on a Non-Linear Euler-Bernoulli Kinematics. *International Journal for Numerical Methods in Engineering*, Vol. 54, No. 1, 27–59.
- Buffinton, K.W. and Lam, J. (1992). A Comparative Study of Simple Dynamic Models and Control Schemes for Elastic Manipulators, *American Control Conference*, Chicago, IL, USA, June, 3334–3339.
- Cetinkunt, S. and Yu, S. (1991). Closed-Loop Behavior of a Feedback-Controlled Flexible Arm: A Comparative Study, *International Journal of Robotics Research*, Vol. 10, No. 3, 263–275.
- Chen, C.H. and Menq, J.S. (1990). Experiments on the Payload-Adaptation of a Flexible One-Link Manipulator with Unknown Payload, *Proc. 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, May, 1614–1619.
- Cho, K. *et al.* (1991). On the Controllability and Observability of Flexible Beams Under Rigid-Body Motion, *IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON'91)*, Kobe, Japan, 455–460.
- Gerald, C.F. and Wheatley, P.O. (2004). *Applied Numerical Analysis*, Addison Wesley, Boston.
- Hastings, G.G. and Book, W.J. (1987). A Linear Dynamic Model for Flexible Robotic Manipulator, *IEEE Control Systems Magazine*, Vol. 7, No. 1, 61–64.
- Junkins, J.L. and Kim, Y. (1993). *Introduction to Dynamics and Control of Flexible Structure*, AIAA, Washington, D.C.
- Li, D. *et al.* (1998). Dynamic Modeling and Mode Analysis of Flexible-Link, Flexible-Joint Robots, *Mechanism and Machine Theory*, Vol. 33, No. 7, 1031–1044.
- Meirovitch, L. (1967). *Analytical Methods in Vibrations*, MacMillan, New York.
- Meirovitch, L. and Silverberg, L.M. (1983). Two Bracketing Theorems Characterizing the Eigensolution for the h -Version of the Finite Element Method, *International Journal for Numerical Methods in Engineering*, Vol. 19, 1691–1704.
- Meirovitch, L. and Kwak, M.K. (1990). Convergence of the Classical Rayleigh-Ritz Method and the Finite Element Method, *AIAA Journal*, Vol. 28, No. 8, 1509–1516.
- Moallem, M., Patel, R.V., and Khorasani, K., (2002). Nonlinear Tip-Position Tracking Control of a Flexible- Link Manipulator: Theory and Experiments, *Automatica*, Vol. 37, No. 11, 1825–1834.
- Nicosia, S., Valigi, P., and Zaccarian, L. (1996). Dynamic Modelling of a Two Link Flexible Robot Experimental Validation, *Proc. 1996 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, April, 1953–1958.
- Parks, T.R. and Pak, H.A. (1991). Effect of Payload on the Dynamics of a Flexible Manipulator - Modeling for Control, *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 113, No. 3, 409–418.
- Piedbœuf, J.C. (2001). Six Methods to Model a Flexible Beam Rotating in the Vertical Plane, *Proc. 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, May, 2832–2839.
- Piedbœuf, J.-C. (1996). Modeling of Flexible Robots with Maple, *The Maple Technical Newsletter*, Vol. 3, No. 1, 38–47.

- J.-C. Piedbœuf, (1998). Recursive Modeling of Serial Flexible Manipulators, *Journal of the Astronautical Sciences*, Vol. 47, No. 1, 1-24.
- Piedbœuf, J.-C. and Miller, S. (1994). Estimation of Endpoint Position and Orientation of a Flexible Link Using Strain Gauges, *4th IFAC Symposium on Robot Control*, Capri, Italy, September, 675–680.
- Saad M. (2004). Modélisation et commande par la passivité d'un système à un bras flexible, *Ph.D. dissertation*, École Polytechnique de Montréal, Montréal, Canada.
- Saad, M., Piedboeuf, J.-C., Akhrif, O, and Saydy, L. (2005). Modal analysis of assumed-mode models of a flexible slewing beam, *Int. J. Modelling, Identification and Control*, Vol. 1, No. 4, 325–337.
- Saad, M., Akhrif, O, and Saydy, L. (2000). Robust Noncollocated Passive Models of a Flexible Link with Uncertain Payload and Joint Inertia, *7th IEEE International Conference on Electronics, Circuits and Systems*, Beirut, Lebanon, December, 713–717.
- Stanway, J. *et al.* (1998). Comparison and Validation of Dynamics Simulation Models for a Structurally Flexible Manipulator, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 120, No. 3, 404–409.
- Theodore, R. and Ghosal, A. (1995). Comparison of the Assumed Modes and Finite Element Models for Flexible Multilink Manipulators, *International Journal of Robotics Research*, Vol. 14, No. 2, 91–111.
- Tokhi, M.O. and Mohamed, Z. (1999). Finite Element Approach to Dynamic Modelling of a Flexible Robot Manipulator: Performance Evaluation and Computational Requirements, *Communications in Numerical Methods in Engineering*, Vol. 15, No. 9, 669–678.
- Watkins, D.S. (1991). *Fundamentals of Matrix Computations*, John Wiley, New York.
- Xia, Z. and Menq, C.H. (1992). Modeling and Control of Flexible Manipulators: Part I - Dynamic Analysis and Characterization, *AMD, ASME, Dynamics of Flexible Multibody Systems: Theory and Experiment*, Vol. 141, 105–114.
- Yang, Y.P. and Gibson, J.S. (1989). Adaptive control of a flexible manipulator with a flexible link, *Journal of Robotic Systems*, Vol. 6, No. 3, 217–232.

Motion Control

Sangchul Won and Jinwook Seok
Pohang University of Science and Technology
South Korea

1. Introduction

This chapter presents an introduction to the various methods of controlling the motion of rigid manipulators. Motion control of robotic manipulators has been the subject of considerable research, and many control schemes have been evolved. Typical and recently proposed motion control strategies are introduced and the strengths and weaknesses of each control scheme are also described in this chapter. We assume that the robotic manipulators are rigid, that is, the manipulators do not have flexible links and elastic joints.

In this chapter, we discuss some useful properties of the robot dynamic equations, which are used in deriving robot control schemes in section 2. Proportional-integral-derivative (PID) control schemes, which are widely used in robotic manipulator control, are introduced in section 3. Computed torque control is described in section 4. A modified computed-torque control scheme which overcomes some disadvantages of the conventional one is also introduced in the section. To compensate for parametric uncertainties in the robot dynamic equations, various adaptive strategies for the control of robotic manipulators are introduced in section 5. We discuss the robust control that is capable of compensating for both structured and unstructured uncertainties in section 6 and conclude the chapter in section 7.

2. Robot Dynamic Equation

In the absence of friction and other disturbances, the dynamic equation of an n -link robot manipulator can be written as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \quad (1)$$

where q is the $n \times 1$ vector of joint variables, τ is the $n \times 1$ vector of input torques, $M(q)$ is the $n \times n$ symmetric positive-definite manipulator inertia matrix, $C(q, \dot{q})\dot{q}$ is the $n \times 1$ vector of centrifugal and Coriolis torques and $g(q)$ is the $n \times 1$ vector of gravitational torques. The control schemes that will be introduced in this chapter are based on some important properties of dynamic equation (1).

Property of Inertia Matrix:

The inertia matrix $M(q)$ is symmetric positive-definite and bounded as

$$\mu_1 I \leq M(q) \leq \mu_2 I, \quad (2)$$

where μ_1 and μ_2 are the positive scalars that may be computed for any given arm.

Property of Centrifugal and Coriolis Vector:

The matrix

$$\dot{M}(q) - 2C(q, \dot{q}) \quad (3)$$

is skew-symmetric. This property implies that $\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q})$. The matrix $C(q, \dot{q})$ is quadratic in \dot{q} and bounded so that $\|C(q, \dot{q})\| \leq c_0 \|\dot{q}\|$ for some positive constant c_0 .

Property of Linearity in the Parameters:

(Craig, 1988) exploited a property that the equation (1) is linear in the inertia parameters. This is important, because some or all of the parameters may be unknown; thus the dynamics are linear in the unknown terms:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = W(q, \dot{q}, \ddot{q})\phi, \quad (4)$$

where ϕ is a vector of unknown constant parameters and $W(q, \dot{q}, \ddot{q})$ is a known matrix of robot functions.

Those properties are used in deriving robot control schemes in subsequent sections.

3. PID Control

The conventional proportional-derivative (PD) and PID controllers are general feedback control mechanisms that are widely used in industrial control systems. These controllers have a strong point in that they are simple to implement and control.

3.1 PD control

PD control is useful for fast-response controllers that do not need a steady-state error of zero. Fundamentally, PD control is a position and velocity feedback control that gives good closed-loop properties when applied to a double integrator system.

First, consider the regulation problem of the robot manipulator described by (1). Because the desired joint velocity $\dot{q}_d = 0$, the control law of the PD controller with gravity compensation is

$$\tau = -K_v \dot{q} + K_p e + g(q), \quad (5)$$

where K_v and K_p are positive-definite gain matrices and $e = q_d - q$. Because this control law has no feed-forward term, it can never achieve zero steady-state error. A common

modification is to add an integral term to eliminate steady-state errors. This introduces additional complications because care must be taken to maintain stability and to avoid integrator windup. When the control law (5) is applied to (1), the closed-loop system becomes

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + K_v\dot{q} - K_p e = 0. \quad (6)$$

Now, we investigate the stability achieved by PD control with gravity compensation. We choose the Lyapunov function candidate,

$$V(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}e^T K_p e. \quad (7)$$

The function V is positive-definite; and has a derivative that is negative semi-definite using property (3):

$$\begin{aligned} \dot{V}(q, \dot{q}) &= -\dot{q}^T K_v \dot{q} + \frac{1}{2}\dot{q}^T (\dot{M}(q) - 2C(q, \dot{q}))\dot{q} \\ &= -\dot{q}^T K_v \dot{q} \\ &\leq -\lambda_{\min}\{K_v\} \|\dot{q}\|^2 \end{aligned} \quad (8)$$

where $\lambda_{\min}\{\cdot\}$ denotes the smallest eigenvalue. By the Lyapunov stability theory and LaSalle's theorem (Khalil, 2002), the regulation error converges to zero asymptotically. In case of PD control without gravity compensation

$$\tau = -K_v \dot{q} + K_p e, \quad (9)$$

and, the closed-loop dynamic equation becomes

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + K_v\dot{q} - K_p e = 0. \quad (10)$$

Let us choose Lyapunov function candidate,

$$V(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}e^T K_p e + U(q) + U_0 \quad (11)$$

where $U(q)$ is the potential energy generating gravity forces and U_0 is a suitable constant. Taking the time derivative of V along the closed-loop dynamics (10) gives the same result (8) as the PD control with gravity compensation. In this case, the control system must be stable in the sense of Lyapunov, but we cannot conclude that the regulation error will converge to zero by LaSalle's theorem.

Next, consider tracking control. The control law of PD control with gravity compensation is

$$\tau = M(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + g(q) + K_v\dot{e} + K_p e. \quad (12)$$

Then the closed-loop system is

$$M(q)\ddot{e} + C(q, \dot{q})\dot{e} - K_v\dot{e} - K_p e = 0. \quad (13)$$

To show the stability, we chose the Lyapunov function candidate

$$V(e, \dot{e}, t) = \frac{1}{2}\dot{e}^T M(q)\dot{e} + \frac{1}{2}e^T K_p e + \varepsilon e^T M(q)\dot{e}, \quad (14)$$

where ε is a positive small constant. The derivative of the function V becomes

$$\dot{V}(e, \dot{e}, t) = -\dot{e}^T (K_v - \varepsilon M)\dot{e} - \varepsilon e^T K_p e + \varepsilon e^T (-K_v + \frac{1}{2}\dot{M})\dot{e}. \quad (15)$$

Choosing ε sufficiently small insures that \dot{V} is negative-definite and hence that the system is exponentially stable by LaSalle's theorem. It is notable that asymptotic tracking requires exact cancelation of gravity and disturbance forces and relies on accurate models of these quantities as well as the manipulator inertia matrix. Therefore, in practical implementations, modeling errors and disturbances result in tracking errors.

3.2 PID control

We have seen that PD control makes the system exponentially stable. However, in practical implementation, in the presence of constant disturbance (from the local point of view), PD control gives a nonzero steady-state error. Consequently, adding an integral action to the controller can compensate for the constant disturbance. The PID controller has the form

$$\begin{aligned} \tau &= K_p e + K_i \int e \, dt + K_v \dot{e} \\ &= K_p e + K_v \dot{e} + v \\ \dot{v} &= K_i e, \quad v(0) = v_0, \end{aligned} \quad (16)$$

where K_i is a positive-definite gain matrix. Choose any positive diagonal matrix K'_p and let

$$K_p := K'_p + \frac{1}{\varepsilon} K_i, \quad (17)$$

where ε is a positive small constant to be determined. Then the error dynamics become

$$\begin{aligned} M(q)\ddot{q} + C(q, \dot{q})\dot{q} + K_v\dot{q} - K'_p e &= \frac{1}{\varepsilon} K_i e + \tilde{v} \\ \dot{\tilde{v}} &= -K_i e, \end{aligned} \quad (18)$$

where \tilde{v} denotes $v - g(q_d)$. To analyze the stability of the closed loop system, we choose the following Lyapunov function candidate with cross terms (Loria et al., 2000)

$$V = \frac{1}{2} \dot{q}^T M \dot{q} + U(q) - U(q_d) + e^T g(q_d) + \frac{1}{2} e^T K'_p e + \frac{\varepsilon}{2} \left(\frac{1}{\varepsilon} K_i e + \tilde{v} \right)^T K_i^{-1} \left(\frac{1}{\varepsilon} K_i e + \tilde{v} \right) - \varepsilon e^T M \dot{q}. \quad (19)$$

Using property of inertia matrix (2), rewriting $e^T K'_p e = (\lambda_1 + \lambda_2 + \lambda_3) e^T K'_p e$ and $\dot{q}^T M(q) \dot{q} = (\lambda_1 + \lambda_2 + \lambda_3) \dot{q}^T M(q) \dot{q}$ with $0 < \lambda_i < 1$, one can show that if

$$\lambda_{\min} \{K'_p\} \geq \max \left\{ \frac{c_g}{\lambda_1}, \frac{\varepsilon^2 \mu_2}{\lambda_1 \lambda_2} \right\}, \quad (20)$$

then the function V satisfies the inequality

$$V \geq \frac{\lambda_3}{2} e^T K'_p e + \frac{\lambda_2 + \lambda_3}{2} \dot{q}^T M(q) \dot{q}. \quad (21)$$

Hence, V is positive-definite and radially unbounded. Next, using the property of the matrix $C(q, \dot{q})$ and the inequality $\|g(q_d) - g(q)\| \leq c_g \|e\|$, the time derivative of V along the trajectories of (16) satisfies

$$\dot{V} \leq - \left(\lambda_{\min} \{K_v\} - \frac{\varepsilon}{2} \lambda_{\max} \{K_v\} - \varepsilon c_0 \|e\| - \varepsilon \mu_2 \right) \|\dot{q}\|^2 - \varepsilon \left(\lambda_{\min} \{K'_p\} - c_g - \frac{1}{2} \lambda_{\max} \{K_v\} \right) \|e\|^2, \quad (22)$$

which is negative semi-definite if

$$\begin{aligned} \lambda_{\min} \{K_v\} &> \varepsilon (\lambda_{\max} \{K_v\} + 2\mu_2) \\ \lambda_{\min} \{K_p\} &> c_g + \frac{1}{2} \lambda_{\max} \{K_v\} \\ \|e\| &\leq \frac{\lambda_{\min} \{K_v\}}{2\varepsilon c_0}, \end{aligned} \quad (23)$$

where $\lambda_{\max} \{\cdot\}$ denotes the largest eigenvalue. Then the local asymptotic stability of the origin $x=0$ follows by LaSalle's theorem.

(Qu & Dorsey, 1991) proposed a similar proof for the uniform ultimate boundedness of the error in the trajectory tracking problem. (Rocco, 1996) proposed a stability analysis method different from other approaches. The proof is based on a formulation of the robot dynamic model where the nominal, decoupled and linear closed loop system is emphasized, whereas the nonlinear terms are split into terms dependent on the control parameters and other norm-bounded terms. However, PID control lacks a global asymptotic stability proof. Moreover, to ensure local stability, the gain matrices must satisfy complicated inequalities.

3.3 Saturated PID control

In implementing PID control on any actual robot manipulator, one effect can cause serious problems: any real robot arm will have limits on the voltages and torques of its actuators. These limits may or may not cause a problem with PD control, but are virtually guaranteed to cause problems with integral control due to a phenomenon known as integrator windup (Lewis, 1992).

To account for bounded control torques, i.e.,

$$|\tau_i| \leq \tau_{i,\max}, \quad i = 1, \dots, n, \quad (24)$$

the actual control torque equipped with a saturation function is defined as

$$\tau = \text{Sat}\left(\hat{g}(q_d) + K_p e + K_v \dot{e} + K_i \int e dt, \tau_{\max}\right), \quad (25)$$

where $\tau_{\max} = [\tau_{1,\max}, \dots, \tau_{n,\max}]^T$, and $\text{Sat}(\cdot, \tau_{\max})$ is a strictly increasing saturation function with upper limit $+\tau_{\max}$ and lower limit $-\tau_{\max}$. The assumption that the saturation function dominates over gravitational torques should be considered. The assumption becomes a necessary condition for the manipulator to be stabilizable at any desired equilibrium configuration $q_d \in \mathbb{R}^n$. In the presence of uncertainty in the gravitational force vector $g(q)$, $\tau_{i,\max}$ should be chosen such that it is acceptably lower than the maximum torque. Under this assumption, if $\|K_p\|$ and $\|K_v\|$ are large enough, and $\|K_i\|$ is small enough, the saturated PID control (25) yields semi-global asymptotic stabilization of the robot dynamics at any desired position $q_d \in \mathbb{R}^n$ (Alvarez-Ramirez et al., 2008). (Sun et al., 2009) presented global stability of a saturated nonlinear PID controller with a new class of saturated function.

3.4 Summary

In this section, we have presented various PID control methods. Although the success of industrial applications has proven the effectiveness of the PD and PID controllers for complex nonlinear robotic manipulators, PID control is cannot cope with highly nonlinear systems for tracking problems. To overcome these limitations, several types of modified PID controllers were introduced subsequently. These are described in the next section.

4. Computed-Torque Control

A special application of the feedback linearization of nonlinear systems is computed-torque control, which consists of an inner nonlinear compensation loop and an outer feedback loop (Fig. 1). In this section, we cover computed-torque and computed-torque with a compensation control scheme which is a dynamic controller.

4.1 Computed-torque control

The computed-torque control law with a PD outer-loop controller is given by

$$\tau = M(q)(\ddot{q}_d - u) + C(q, \dot{q})\dot{q} + g(q), \tag{26}$$

where the auxiliary control signal $u = -K_v \dot{e} - K_p e$, which is of the PD feedback. Computed-torque control is a model-based motion control approach created for manipulators, that is, in which one makes explicit use of the knowledge of the matrices $M(q)$, $C(q, \dot{q})\dot{q}$ and $g(q)$. Furthermore, the control action (26) is computed using the desired trajectory of motion $q_d(t)$, and its derivatives $\dot{q}_d(t)$ and $\ddot{q}_d(t)$, as well as the position and velocity measurements $q(t)$ and $\dot{q}(t)$.

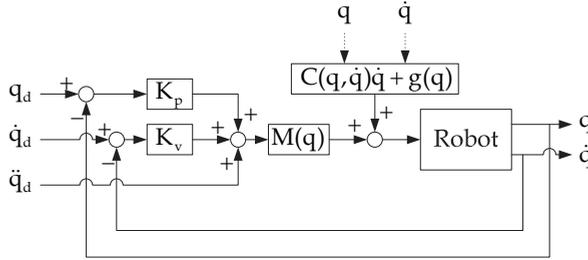


Fig. 1. Computed-torque control

The closed-loop error dynamics of the system (Fig. 1) have the form

$$\ddot{e} + K_v \dot{e} + K_p e = 0. \tag{27}$$

The error dynamics (27) can also be rewritten in state-space form as

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \triangleq A \begin{bmatrix} e \\ \dot{e} \end{bmatrix}. \tag{28}$$

Because the error dynamics (28) is linear, its solutions may be obtained in closed form and used to assess the stability of the origin.

We start by introducing the small constant ε satisfying

$$\lambda_{\min}\{K_v\} > \varepsilon > 0. \tag{29}$$

Multiplying by $x^T x$ where $x \in \mathbb{R}^{2n}$ is any nonzero vector yields $\lambda_{\min}\{K_v\} x^T x > \varepsilon x^T x$.

Because K_v is a symmetric and positive-definite matrix, $x^T K_v x \geq \lambda_{\min}\{K_v\} x^T x$ and therefore,

$$x^T [K_v - \varepsilon I] x > 0 \quad \forall x \neq 0 \in \mathbb{R}^{2n}. \tag{30}$$

This means that the matrix $K_v - \varepsilon I$ is positive-definite. Considering all this, we conclude that

$$K_p + \varepsilon K_v - \varepsilon^2 I > 0. \quad (31)$$

We choose the total energy of the system as Lyapunov function,

$$\begin{aligned} V(e, \dot{e}) &= \frac{1}{2} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T \begin{bmatrix} K_p + \varepsilon K_v & \varepsilon I \\ \varepsilon I & I \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \\ &= \frac{1}{2} \dot{e}^T \dot{e} + \frac{1}{2} e^T [K_p + \varepsilon K_v] e + \varepsilon e^T \dot{e}. \end{aligned} \quad (32)$$

Taking the derivative and applying (26) yields,

$$\begin{aligned} \dot{V}(e, \dot{e}) &= \ddot{e}^T \dot{e} + e^T [K_p + \varepsilon K_v] \dot{e} + \varepsilon \dot{e}^T \dot{e} + \varepsilon e^T \ddot{e} \\ &= -\dot{e}^T [K_v - \varepsilon I] \dot{e} - \varepsilon e^T K_p e. \end{aligned} \quad (33)$$

Because $K_v - \varepsilon I$ is positive-definite (30), the function $\dot{V}(e, \dot{e})$ is globally negative-definite. By the Barbashin-Krasovskii theorem (Khalil, 2002), we conclude that the origin $[e^T, \dot{e}^T]^T$ of the closed-loop equation is globally uniformly asymptotically stable.

4.2 Computed-torque control with compensation

The computed-torque method is an approach that makes direct use of the complete dynamic model of the manipulator. Therefore, we have to know accurate parameters of the model. To compensate modeling errors, we introduce a computed-torque controller with compensation which consists of the computed-torque control law (26), and dynamic terms. The control law (Kelly et al., 2005) is

$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + C(q, \dot{q})\dot{q} + g(q) - C(q, \dot{q})v, \quad (34)$$

where v represents the filtered errors of the position and velocity. We choose v as

$$v = -\frac{bp}{p+\lambda} \dot{e} - \frac{b}{p+\lambda} (K_v \dot{e} + K_p e), \quad (35)$$

where p is the differential operator d/dt ; λ and b are positive design constants. For simplicity, and without loss of generality, we take $b=1$. Due to the presence of the vector v the computed-torque with compensation control law is dynamic, that is, the control action τ depends not only on the actual values of the state vector formed by q and \dot{q} , but also on its past values. As a consequence of this fact, additional state variables are defined as (37) to characterize the control law completely. The state space realization of (35) is a linear autonomous system given by

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} -\lambda I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} K_p & K_v \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad (36)$$

$$v = \begin{bmatrix} -I & -I \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} - \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad (37)$$

where $\xi_1, \xi_2 \in \mathbb{R}^n$ are the new state variables.

To derive the closed-loop equation, we substitute the control law (34) into (1).

$$M(q)(\ddot{e} + K_v \dot{e} + K_p e) - C(q, \dot{q})v = 0 \quad (38)$$

In terms of the state vector $[e^T, \dot{e}^T, \xi_1^T, \xi_2^T]^T \in \mathbb{R}^{4n}$, equations (36) to (38) can be used to obtain the closed-loop equation

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \\ \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ -K_p & -M^{-1}(q)C(q, \dot{q}) - K_v & -M^{-1}(q)C(q, \dot{q}) & -M^{-1}(q)C(q, \dot{q}) \\ K_p & K_v & -\lambda I & 0 \\ 0 & -\lambda & 0 & -\lambda I \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \xi_1 \\ \xi_2 \end{bmatrix} \quad (39)$$

of which the origin $[e^T, \dot{e}^T, \xi_1^T, \xi_2^T]^T = 0$ is an equilibrium point.

To analyze the control system we first write it in a different but equivalent form. For this, notice that the expression for v given in (35) allows one to derive

$$\dot{v} + \lambda v = -(\ddot{e} + K_v \dot{e} + K_p e). \quad (40)$$

Substituting (40) into (38) yields

$$M(q)(\dot{v} + \lambda v) + C(q, \dot{q})v = 0. \quad (41)$$

Equation (40) is the starting point in the following stability analysis. Consider the Lyapunov function

$$V(v, e) = \frac{1}{2} v^T M(q) v. \quad (42)$$

The derivative of V with respect to time is given by

$$\begin{aligned} \dot{V}(v, e) &= v^T M(q) \dot{v} + \frac{1}{2} v^T \dot{M}(q) v \\ &= -v^T \lambda M(q) v \leq 0. \end{aligned} \quad (43)$$

Considering V and \dot{V}

$$\dot{V}(v,e) = -2\lambda V(v,e), \quad (44)$$

which implies that

$$V(v(t),e(t)) = V(v(0),e(0))\exp(-2\lambda t). \quad (45)$$

From the property of inertia matrix (2),

$$\begin{aligned} \mu_1 v^T v &\leq v^T M(q)v = 2V(v(t),e(t)) \\ v^T v &\leq \frac{2V(v(0),e(0))}{\mu_1} \exp(-2\lambda t). \end{aligned} \quad (46)$$

This means that $v(t)$ tends to zero exponentially as time t is increasing. Equation (40) may also be written as

$$e = -(p + \lambda)(p^2 I + pK_v + K_p)^{-1} v. \quad (47)$$

The input to the linear system (47) is v which tends to zero exponentially, and the output is e . Because system (47) is a strictly proper linear system which is asymptotically stable, we invoke the fact that a stable strictly proper filter with an exponentially decaying input produces an exponentially decaying output, that is,

$$\lim_{t \rightarrow \infty} e(t) = 0, \quad (48)$$

which means that the motion control objective is verified.

We need an accurate dynamic model or must calculate the control input in real time because computed-torque methods are an approach that makes direct use of the complete dynamic model of the manipulator. To avoid these conditions, various kind of modified computed-torque control schemes are introduced in the following section.

5. Adaptive Control

Adaptive controllers are formulated by updating controller parameters on-line and are adequate for systems that have structured uncertainties. Designing an adaptive controller is to develop an estimation algorithm, called the adaptation law, that guarantees convergence of the controller parameters as well as stability.

5.1 Adaptive computed-torque control

Because the computed-torque method needs exact dynamic model of the manipulator, performance and stability of the system cannot be guaranteed when parametric mismatches

exist. One way to solve the problem of parameter uncertainties is to use the computed-torque controller with estimates of the unknown parameters in place of the actual parameters. Based on computed-torque control law (26), the adaptive computed-torque controller has the form

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q), \quad (49)$$

where $\hat{M}(q)$, $\hat{C}(q, \dot{q})$ and $\hat{g}(q)$ are the estimations of $M(q)$, $C(q, \dot{q})$ and $g(q)$. The adaptive controller is based on the fact that the parameters appear linearly in the robot model as (4). By utilizing (4), control law (49) can be written as

$$\tau = \hat{M}(q)(\ddot{e} + K_v \dot{e} + K_p e) + W(q, \dot{q}, \ddot{q})\hat{\phi}, \quad (50)$$

where $\hat{\phi}$ is an $r \times 1$ vector that represents a time-varying estimate of the unknown constant parameters. Using (1) and (4), we have the tracking error system

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\phi}, \quad (51)$$

where $\tilde{\phi} = \phi - \hat{\phi}$ is the parameter error. To obtain an adaptive control law, (51) can be rewritten in the state-space form

$$\dot{e} = \begin{bmatrix} 0_n & I_n \\ -K_p & -K_v \end{bmatrix} e + \begin{bmatrix} 0_n \\ I_n \end{bmatrix} M^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\phi} \triangleq A e + B M^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\phi}, \quad (52)$$

where the tracking error vector $e = [e^T, \dot{e}^T]^T$. We select the Lyapunov function

$$V = e^T P e + \tilde{\phi}^T \Gamma^{-1} \tilde{\phi}, \quad (53)$$

where P is a $2n \times 2n$ positive symmetric matrix, and Γ is a diagonal positive-definite $r \times r$ matrix. The derivative of (53) is

$$\begin{aligned} \dot{V} &= e^T P \dot{e} + \dot{e}^T P e + 2\tilde{\phi}^T \Gamma^{-1} \dot{\tilde{\phi}} \\ &= e^T P (A e + B \hat{M}^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\phi}) + (A e + B \hat{M}^{-1}(q)W(q, \dot{q}, \ddot{q})\tilde{\phi})^T P e + 2\tilde{\phi}^T \Gamma^{-1} \dot{\tilde{\phi}} \\ &= -e^T Q e + 2\tilde{\phi}^T (\Gamma^{-1} \dot{\tilde{\phi}} + W^T(q, \dot{q}, \ddot{q})\hat{M}^{-1}(q)B^T P e) \end{aligned} \quad (54)$$

where Q is the positive-definite symmetric matrix that satisfies the Lyapunov equation

$$A^T P + P A = -Q. \quad (55)$$

To have \dot{V} negative semi-definite, the adaptive update rule is chosen as

$$\dot{\hat{\phi}} = -\Gamma W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) B^T P e, \quad (56)$$

which implies that $\dot{V} = -e^T Q e$. Equation (56) gives the adaptive update rule for the parameter estimate vector $\hat{\phi}$ because $\dot{\phi}$ is equal to zero. Substituting $\dot{\phi} = \phi - \hat{\phi}$ into (56) gives the adaptive update rule:

$$\dot{\hat{\phi}} = \Gamma W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) B^T P e \quad (57)$$

for the parameter estimate vector $\hat{\phi}$.

Detailed stability analysis (Craig, 1988) shows that the tracking error vector e approaches to zero asymptotically. The adaptive computed-torque controller has some restrictions required for the implementation. That is, the controller needs to measure accurate acceleration \ddot{q} and to ensure that $\hat{M}^{-1}(q)$ exists. To avoid these restrictions, other adaptive control schemes are introduced in following sections.

5.2 Adaptive inertia-related control

(Slotine & Li, 1987) proposed an adaptive inertia-related control scheme that does not need to measure joint acceleration and ensure inversion of the estimated inertia matrix.

Consider the control input

$$\tau = \hat{M}(q)(\ddot{q}_d + \Lambda \dot{e}) + \hat{C}(q, \dot{q})(\dot{q}_d + \Lambda e) + \hat{g}(q) + K_v r, \quad (58)$$

where the auxiliary signal r is defined as $r = \Lambda e + \dot{e}$, with Λ being an $n \times n$ positive-definite diagonal matrix. Using $\dot{q} = \dot{q}_d + \Lambda e - r$, $\ddot{q} = \ddot{q}_d + \Lambda \dot{e} - \dot{r}$ and property (4), the robot dynamic equation (1) can be rewritten as

$$\tau = Y(\cdot)\phi - M(q)\dot{r} - C(q, \dot{q})r, \quad (59)$$

where

$$Y(\cdot)\phi = M(q)(\ddot{q}_d + \Lambda \dot{e}) + C(q, \dot{q})(\dot{q}_d + \Lambda e) + g(q), \quad (60)$$

and $Y(\cdot)$ is an $n \times r$ matrix of known time functions. Equation (60) is the same type of parameter separation that was used in the formulation of the adaptive computed-torque controller. However, here $Y(\cdot)$ is independent of the joint acceleration \ddot{q} . Similar to the formulation (60), we also have

$$\hat{M}(q)(\ddot{q}_d + \Lambda \dot{e}) + \hat{C}(q, \dot{q})(\dot{q}_d + \Lambda e) + \hat{g}(q) = Y(\cdot)\hat{\phi}. \quad (61)$$

To form the error system, substituting the control input (58) into the equation of motion (1) yields

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \hat{M}(q)(\ddot{q}_d + \Lambda \dot{e}) + \hat{C}(q, \dot{q})(\dot{q}_d + \Lambda e) + \hat{g}(q) + K_v r. \quad (62)$$

Substituting $\ddot{q} = \ddot{q}_d - \ddot{e}$ and $\dot{q} = \dot{q}_d - \dot{e}$ into (62), and using (60) and (61), the equation (62) can be rewritten as

$$M(q)\dot{r} + C(q, \dot{q})r + K_v r = Y(\cdot)\tilde{\varphi}, \quad (63)$$

where $\tilde{\varphi} = \varphi - \hat{\varphi}$ is the parameter error. To show the convergence of the tracking error to zero, (Slotine & Li, 1987) selected the inertia-related Lyapunov-like function that is a function of the tracking error and the parameter error:

$$V = \frac{1}{2} r^T M(q) r + \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi}, \quad (64)$$

where Γ is defined as in (53). Differentiating (64) with respect to time yields

$$\begin{aligned} \dot{V} &= r^T (Y(\cdot)\tilde{\varphi} - K_v r) + r^T \left(\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right) r + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \\ &= r^T (Y(\cdot)\tilde{\varphi} - K_v r) + \tilde{\varphi}^T \Gamma^{-1} \dot{\tilde{\varphi}} \\ &= -r^T K_v r + \tilde{\varphi}^T (\Gamma^{-1} \dot{\tilde{\varphi}} + Y^T(\cdot)r). \end{aligned} \quad (65)$$

By selecting adaptive update rule as

$$\dot{\hat{\varphi}} = \Gamma Y^T(\cdot)(\Lambda e + \dot{e}), \quad (66)$$

(65) becomes $\dot{V} = -r^T K_v r$, which is negative semi-definite. Detailed analysis (Slotine & Li, 1987) shows that the tracking error e and \dot{e} are asymptotically stable.

5.3 Adaptive control based on passivity

To unify many adaptive control schemes that have different torque control laws or adaptive update rules, an adaptive control scheme has been developed based on the passivity approach. It requires neither feedback of joint accelerations nor inversion of the estimated inertia matrix. First, we define an auxiliary filtered tracking error variable $r(s)$ that is similar to that defined for the adaptive inertia-related controller:

$$r(s) = H^{-1}(s)e(s), \quad (67)$$

where

$$H^{-1}(s) = \left[sI_n + \frac{1}{s} K(s) \right], \quad (68)$$

and s is the Laplace transform variable. The $n \times n$ gain matrix $K(s)$ is chosen such that $H(s)$ is a strictly proper, stable transfer function matrix with relative degree 1. As in the preceding schemes, the adaptive control strategies require that the known time functions can be separated from the unknown constant parameters. Therefore, using $\dot{q} = \dot{q}_d + (1/s)K(s)e - r$, $\ddot{q} = \ddot{q}_d + K(s)e - \dot{r}$ and the property (4), the robot dynamic equation (1) can be rewritten as

$$\tau = Z(\cdot)\varphi - M(q)\dot{r} - C(q, \dot{q})r, \quad (69)$$

where

$$Z(\cdot)\varphi = M(q)(\ddot{q}_d + K(s)e) + C(q, \dot{q})\left(\dot{q}_d + \frac{1}{s}K(s)e\right) + g(q), \quad (70)$$

and $Z(\cdot)$ is a known $n \times r$ regression matrix. The equation (70) can be arranged such that Z and r do not depend on the measurements of the joint acceleration \ddot{q} . The adaptive control scheme given here is called the passivity approach because the mapping of $-r \rightarrow Z(\cdot)\tilde{\varphi}$ is constructed to be a passive mapping (Ortega & Spong, 1988). That is, we construct an adaptive update rule such that

$$\int_0^t -r^T(\sigma)Z(\sigma)\tilde{\varphi}(\sigma) d\sigma \geq -\beta \quad (71)$$

is satisfied for all time and for some positive scalar constant β . We use the concept of passivity to analyze the stability of a class of adaptive controllers. For this class of adaptive controllers, the torque control is given by

$$\begin{aligned} \tau &= \hat{M}(q)(\ddot{q}_d + K(s)e) + \hat{C}(q, \dot{q})\left(\dot{q}_d + \frac{1}{s}K(s)e\right) + \hat{g}(q) + K_v r \\ &= Z(\cdot)\hat{\varphi} + K_v r \end{aligned} \quad (72)$$

Similar to the formulation from (62) to (63), the tracking error system can be expressed in terms of the tracking error variable r and regression matrix $Z(\cdot)$ as

$$M(q)\dot{r} + C(q, \dot{q})r + K_v r = Z(\cdot)\hat{\varphi}. \quad (73)$$

To analyze the stability of this system, we choose the Lyapunov-like function

$$V = \frac{1}{2}r^T M(q)r + \beta - \int_0^t r^T(\sigma)Z(\sigma)\tilde{\varphi}(\sigma) d\sigma, \quad (74)$$

and note from (71) that $V \geq 0$. Differentiating V and substituting (73) into (75) give

$$\begin{aligned}
\dot{V} &= \mathbf{r}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{r}} + \frac{1}{2} \mathbf{r}^T \dot{\mathbf{M}}(\mathbf{q}) \mathbf{r} - \mathbf{r}^T \mathbf{Z}(\cdot) \hat{\phi} \\
&= -\mathbf{r}^T \mathbf{K}_v \mathbf{r} + \frac{1}{2} \mathbf{r}^T (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \mathbf{r} \\
&= -\mathbf{r}^T \mathbf{K}_v \mathbf{r},
\end{aligned} \tag{75}$$

which is negative semi-definite. The passivity approach gives a general class of torque control laws. (Lewis et al., 2003) showed the type of stability for the tracking error which is asymptotically stable and some examples that unify some of the research in adaptive control.

6. Robust Control

Robust control is a control of fixed structure that guarantees stability and performance in uncertain systems. Its design only requires some knowledge about bounding functions on the largest possible size of the uncertainties. This limited requirement implies that robust control is capable of compensating for both structured and unstructured uncertainties, and this is one of the major advantages of robust control over adaptive control. Compared to adaptive control, other advantages of robust control are computational simplicity in implementation, better compensation for time-varying parameters and for unstructured nonlinear uncertainties, and guaranteed stability.

6.1 Passivity-based approach

First, we present controllers that rely directly on the passive structure of rigid robots. Based on the passivity theorem (Ortega & Spong, 1988), if one can show the passivity of the system which maps control input τ to a new vector \mathbf{r} which is a filtered version of \mathbf{e} , then a controller which closes the loop between $-\mathbf{r}$ and τ will guarantee the asymptotic stability of both \mathbf{e} and $\dot{\mathbf{e}}$. Consider the following controller (Abdallah et al., 1991)

$$\tau = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{K}(s)\mathbf{e}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \left(\dot{\mathbf{q}}_d + \frac{1}{s} \mathbf{K}(s)\mathbf{e} \right) + \mathbf{g}(\mathbf{q}) + \mathbf{K}_v \mathbf{r}, \tag{76}$$

where $\mathbf{K}(s)$ and \mathbf{r} are defined in (67) and (68). Substituting (76) into (1), yields the tracking error system in terms of the tracking error variable \mathbf{r} as

$$\mathbf{M}(\mathbf{q}) \dot{\mathbf{r}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{r} + \mathbf{K}_v \mathbf{r} = 0. \tag{77}$$

Then it may be shown that both \mathbf{e} and $\dot{\mathbf{e}}$ are asymptotically stable. This passivity-based approach was introduced in section 5.3, but its modification in the design of robust controllers when $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q})$ are not exactly known is not obvious.

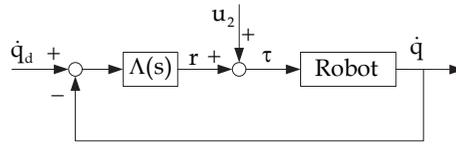


Fig. 2. Passivity-based control

The passivity-based control input is given by

$$\tau = \Lambda(s)\dot{e} + u_2, \quad (78)$$

where $\Lambda(s)$ is a strictly proper and stable transfer function, and the external input u_2 is bounded in the L_2 norm. Using the control law (78), we get from Fig. 2,

$$r = \Lambda(s)\dot{e}. \quad (79)$$

By an appropriate choice of $\Lambda(s)$ and u_2 , we can apply the passivity theorem and deduce that \dot{e} and r are bounded in the L_2 norm. Because $\Lambda^{-1}(s)$ is strictly proper and stable function, we can conclude that \dot{e} is asymptotically stable because

$$\dot{e} = \Lambda^{-1}(s)r. \quad (80)$$

This implies that the position error e is bounded but the asymptotic stability is not guaranteed in case of trajectory tracking problem. However, in the regulation problem, the asymptotic stability of e can be guaranteed using LaSalle's theorem.

6.2 Variable-structure controllers

The variable-structure theory has been applied to the control of many nonlinear processes. One of the main features of this approach is that one only needs to drive the error to a switching surface, after which the system is in sliding mode and will not be affected by any modeling uncertainties and disturbances. The first application of this theory to robot control seems to be in (Young, 1978), where the set-point regulation problem was solved using the following controller:

$$\tau_i = \begin{cases} \tau_i^+, & \text{if } r_i > 0 \\ 0, & \text{if } r_i = 0, \\ \tau_i^-, & \text{if } r_i < 0 \end{cases} \quad (81)$$

where $i = 1, \dots, n$ for an n -link robot. The switching planes r_i are defined as

$$r = \Lambda e + \dot{e}, \quad (82)$$

where $\Lambda = \text{diag}\{\lambda_1 \cdots \lambda_n\}$ with $\lambda_i > 0$. Let the control input be

$$\tau = M(q)(\lambda \dot{e} - \dot{q}_d + K \text{sgn}(r)) - C(q, \dot{q})\dot{q} - g(q), \quad (83)$$

where $K = \text{diag}\{k_1 \cdots k_n\}$ with $k_i > 0$ and

$$\text{sgn}(r_i) = \begin{cases} +1, & \text{if } r_i > 0 \\ -1, & \text{if } r_i < 0 \end{cases}. \quad (84)$$

Choose $V = (1/2)r^T r$ as a Lyapunov candidate. Differentiating V and using (1) and (82),

$$\begin{aligned} \dot{V}(r) &= r^T \dot{r} \\ &= r^T (M^{-1}(q)C(q, \dot{q})\dot{q} + M^{-1}(q)g - M^{-1}\tau + \dot{q}_d + \Lambda \dot{e}) \end{aligned} \quad (85)$$

Substituting the control input (83) into (85) yields

$$\begin{aligned} \dot{V}(r) &= r^T (-K \text{sgn}(r)) \\ &= -\sum_{i=1}^n k_i |r_i| \leq 0 \end{aligned} \quad (86)$$

According to the Lyapunov stability theorem, the origin is stable equilibrium point. When $r=0$ in the sliding mode, the tracking error e decays at an exponential rate. Therefore, the control system is asymptotically stable with the switching function (82) and the control law (83).

For most of these schemes, the control effort is discontinuous along $s_i = 0$; this causes chattering which may excite unmodeled high-frequency dynamics. (Slotine, 1985) modified the variable-structure controller. (Chen et al., 1990) introduced a variable-structure controller which avoided the need to invert of the inertia matrix.

6.3 Adaptive robust control

Robust controls ensure robust stability for robotic systems. Robust controls can be defined in terms of a bounding function; determination of this function requires information on the bound of the uncertainties, such as maximum load variation. Without specifying applications, this size information may be difficult to obtain. While under-estimation is not permitted when considering robustness, and over-estimating the maximum size of uncertainties can potentially give robust control an unnecessarily large magnitude and gain, and consequently put too many requirements on the actuators. One approach to maintaining robustness while reducing conservatism is to introduce an adaptive scheme into robust control, that is, to design a so-called adaptive robust control that estimates on-line the size of the uncertainties.

Suppose that the dynamic equation given by

$$\mathbf{w} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \dot{\mathbf{e}}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_d + \mathbf{e}) + \mathbf{g}(\mathbf{q}), \quad (87)$$

represents the uncertainty for a given robot controller. It is assumed that a positive scalar function ρ can be used to bound the uncertainty as follows (Lewis et al., 2003)

$$\rho = \delta_0 + \delta_1 \|\mathbf{e}\| + \delta_2 \|\mathbf{e}\|^2 \geq \|\mathbf{w}\|, \quad (88)$$

where $\mathbf{e} = [\mathbf{e}^T, \dot{\mathbf{e}}^T]^T$ is the tracking error vector and the values of δ_i represent the positive bounding constants that are based on such quantities as the largest possible payload mass, link mass and disturbances. The adaptive robust controller learns these bounding constants on-line as the manipulator moves. In the control implementation, knowledge of the bounding constants is not required; only the existence of the bounding constants defined in (88) is required.

The adaptive robust controller has the following form

$$\boldsymbol{\tau} = \mathbf{K}_v \mathbf{r} + \frac{\mathbf{r} \hat{\rho}^2}{\hat{\rho} \|\mathbf{r}\| + \varepsilon} \quad (89)$$

where the filtered tracking error $\mathbf{r} = \mathbf{e} + \dot{\mathbf{e}}$ and

$$\dot{\varepsilon} = -k_\varepsilon \varepsilon, \quad \varepsilon(0) > 0, \quad (90)$$

where k_ε is a positive scalar control constant. $\hat{\rho}$ is a scalar function defined as

$$\hat{\rho} = \hat{\delta}_0 + \hat{\delta}_1 \|\mathbf{e}\| + \hat{\delta}_2 \|\mathbf{e}\|^2, \quad (91)$$

and the values of $\hat{\delta}_i$ are the dynamic estimates of the corresponding bounding constants δ_i . (91) can be rewritten in the matrix form

$$\hat{\rho} = \mathbf{S} \hat{\boldsymbol{\theta}} = \begin{bmatrix} 1 & \|\mathbf{e}\| & \|\mathbf{e}\|^2 \end{bmatrix} \begin{bmatrix} \hat{\delta}_0 & \hat{\delta}_1 & \hat{\delta}_2 \end{bmatrix}^T. \quad (92)$$

Similar to (92), the actual bounding function ρ can be also be written as

$$\rho = \mathbf{S} \boldsymbol{\theta} = \begin{bmatrix} 1 & \|\mathbf{e}\| & \|\mathbf{e}\|^2 \end{bmatrix} \begin{bmatrix} \delta_0 & \delta_1 & \delta_2 \end{bmatrix}^T. \quad (93)$$

Then the bounding estimates defined in (92) are updated on-line by the update rule

$$\dot{\hat{\boldsymbol{\theta}}} = \gamma \mathbf{S}^T \|\mathbf{r}\|, \quad (94)$$

where γ is a positive scalar control constant and the filtered tracking error $\mathbf{r} = \mathbf{e} + \dot{\mathbf{e}}$.

Because δ_i are constants, (94) can be rewritten as

$$\dot{\tilde{\theta}} = -\gamma S^T \|r\|, \quad (95)$$

where $\tilde{\theta} = \theta - \hat{\theta}$.

Substituting the adaptive robust controller (89) into the robot dynamic equation (1) gives the error system:

$$M(q)\dot{r} + C(q, \dot{q})r + K_v r + \frac{r\hat{\rho}^2}{\hat{\rho}\|r\| + \varepsilon} - w = 0. \quad (96)$$

Now, let us investigate the stability of the corresponding error system (96) for the adaptive robust controller (89). We choose the Lyapunov function candidate as follows

$$V = \frac{1}{2} r^T M(q)r + \frac{1}{2} \tilde{\theta}^T \gamma^{-1} \tilde{\theta} + k_\varepsilon^{-1} \varepsilon. \quad (97)$$

The time derivative of (97) is

$$\begin{aligned} \dot{V} &= \frac{1}{2} r^T \dot{M}(q)r + r^T M(q)\dot{r} + \tilde{\theta}^T \gamma^{-1} \dot{\tilde{\theta}} + k_\varepsilon^{-1} \dot{\varepsilon} \\ &= -r^T K_v r - S\tilde{\theta}\|r\| + r^T \left(w - \frac{r\hat{\rho}^2}{\hat{\rho}\|r\| + \varepsilon} \right) + k_\varepsilon^{-1} \dot{\varepsilon}. \end{aligned} \quad (98)$$

Using (88), we can place an upper bound on \dot{V} in the following manner:

$$\begin{aligned} \dot{V} &\leq -r^T K_v r - S\tilde{\theta}\|r\| + S\theta\|r\| - \frac{r^T r \hat{\rho}^2}{\hat{\rho}\|r\| + \varepsilon} + k_\varepsilon^{-1} \dot{\varepsilon} \\ &= -r^T K_v r + S\hat{\theta}\|r\| - \frac{r^T r \hat{\rho}^2}{\hat{\rho}\|r\| + \varepsilon} - \varepsilon \\ &= -r^T K_v r - \varepsilon + \frac{\varepsilon S\hat{\theta}\|r\|}{S\hat{\theta}\|r\| + \varepsilon}. \end{aligned} \quad (99)$$

Because the sum of the last two terms in (99) is always less than zero, we can place the new upper bound on \dot{V} :

$$\dot{V} \leq -r^T K_v r, \quad (100)$$

which is negative semi-definite and used to know the type of stability for the tracking error which is asymptotically stable (Corless & Leitmann, 1983).

In the conventional adaptive robust control methods, however, the explicit quantitative

For low frequencies $Q(s) \approx 1$, $G_{rq}(s) \approx P_n(s)$, $G_{dq}(s) \approx 0$ and $G_{sq}(s) \approx 1$. For high frequencies $Q(s) \approx 0$, $G_{rq}(s) \approx P(s)$, $G_{dq}(s) \approx P(s)$ and $G_{sq}(s) \approx 0$. This implies that the disturbance observer rejects low-frequency disturbances and high-frequency measurement noise. Selection of a low-pass filter $Q(s)$ is an important factor for designing the disturbance observer, because this selection constitutes a design trade-off between disturbance rejection versus noise rejection and robust stability. Because the disturbance observer uses a low-pass-filter to reduce the measurement noise of the output and to make the transfer function $Q(s)P^{-1}(s)$ proper, the performance of the observer mainly depends on the designed filter. Hence, many studies have dealt with design methods of robust disturbance observer and of $Q(s)$. However most of these studies of design and analysis are based on linear system techniques (Umeno et al., 1993; Choi et al., 2003; Kobayashi et al., 2007). Those techniques are not applicable when the system does not work as a nominal linear plant. For this reason, nonlinear disturbance observers for nonlinear dynamics of the system have been proposed to overcome the limitation of analysis based on linear system (Chen, 2004; Liu & Svoboda, 2006).

7. Conclusion

In this chapter, various motion control schemes for rigid robotic manipulator were introduced. The first control schemes were conventional PD and PID control which have simple structures. However, PID control methods have limitations for nonlinear robotic manipulators. To overcome these disadvantages, modified PID control and computed-torque control were introduced as a special application of the feedback linearization of nonlinear systems.

To handle uncertainties in the robotic manipulator, adaptive and robust control methods were discussed. Adaptive controllers are formulated by updating controller parameters online; these controllers are suited for systems with structured uncertainties. In robust control of fixed structure, the stability and performance in uncertain systems is guaranteed. Robust control schemes can be combined with adaptive control techniques effectively. The bounds of uncertainties are estimated by adding an adaptive scheme to the robust controller. Some disturbance observer based control schemes are also shown to be robust control methods. By compensating for all disturbances which consist of system uncertainties and disturbances, DOB control can be applied to practical robot manipulators effectively.

8. References

- Alvarez-Ramirez, J.; Santibanez, V. & Campa, R. (2008). Stability of Robot Manipulators Under Saturated PID Compensation. *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 6, November 2008, pp. 1333-1341, ISSN: 1063-6536
- Abdallah, C.; Dawson, D.M.; Dorato, P. & Jamshidi, M. (1991). Survey of robust control for rigid robots. *IEEE Control Systems Magazine*, Vol. 11, No. 2, February 1991, pp. 24-30, ISSN: 0272-1708
- Chen, W-H. (2004). Disturbance observer based control for nonlinear systems. *IEEE/ASME Transactions on Mechatronics*, Vol. 9, No. 4, December 2004, pp. 706-710, ISSN: 1083-4435

- Chen, Y-F T.; Mita, T. & Wakui, S. (1990). A new and simple algorithm for sliding mode trajectory control of the robot arm. *IEEE Transactions on Automatic Control*, Vol. 33, No. 1, January 1988, pp. 118-122, ISSN: 0018-9286
- Choi, Y.; Yang, K.; Chung, W.K.; Kim, H.R. & Suh I.H. (2003). On the robustness and performance of disturbance observers for second-order systems. *IEEE Transactions on Automatic Control*. Vol. 48, No. 2, February 2003, pp. 315-320, ISSN: 0018-9286
- Corless, M. & Leitmann, G. (1983). Adaptive control of systems containing uncertain functions and unknown functions with uncertain bounds. *Journal of Optimization Theory and Applications*, Vol. 41, No. 1, September 1983, pp. 155-168, ISSN: 0022-3239
- Craig, J.J. (1988). *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, ISBN: 0201104903
- Imura, J.; Sugie, T. & Yoshikawa, T. (1994). Adaptive Robust Control of Robot Manipulators-Theory and Experiment. *IEEE Transactions on Robotics and Automation*. Vol. 10, No. 5, 1994, pp. 705-710, ISSN:1042-296X
- Kelly, R.; Santibanez, V. & Loria, A. (2005). *Control of Robot Manipulators in Joint Space*, Springer, ISBN: 18523399442
- Khalil, H.K. (2002). *Nonlinear Systems, Second Edition*, Prentice-Hall, ISBN: 0130673897, Upper Saddle River, NJ
- Kobayashi, H.; Katsura, S. & Ohnishi, K. (2007). An Analysis of Parameter Variations of Disturbance Observer for Motion Control. *IEEE Transactions on Industrial Electronics*. Vol. 54, No. 6, 2007, pp. 3413-3421, ISSN:0278-0046
- Lewis, F.L. (1992). *Applied Optimal Control and Estimation*, Prentice Hall, ISBN: 013040361X, Englewood Cliffs, NJ
- Lewis, F.L.; Dawson D.M. & Abdallah, C.T. (2003). *Robot Manipulator Control: Theory and Practice, Second Edition*, Marcel Dekker, ISBN: 0824740726, New York
- Liu, Z.L. & Svoboda, J. (2006). A new control scheme for nonlinear systems with disturbances. *IEEE Transactions on Control System Technology*. Vol. 14, No. 1, 2006, pp. 176-181, ISSN: 1063-6536
- Loria, A.; Lefeber, E. & Nijmeijer, H. (2000). Global asymptotic stability of robot manipulators with linear PID and PI2D control. *Stability and Control: Theory and Applications*. Vol. 3, No. 2, 2000, pp. 138-149, ISSN: 1563-3276
- Ortega, R. & Spong, M.W. (1988). Adaptive motion control of rigid robots: a tutorial. *Proceedings of IEEE Conference on Decision and Control*, pp. 1575-1584, Austin, TX, December 1988
- Qu, Z. (2000). Robust control of nonlinear systems by estimating time variant uncertainties. *Proceedings of IEEE Conference on Decision and Control*, pp. 3019-3024, Sydney, Australia, December 2000
- Qu, Z. & Dorsey, J. (1991). Robust PID control of robots. *International Journal of Robotics and Automation*. Vol. 6, No. 12, 1991, pp. 228-235, ISSN: 0826-8185
- Rocco, P. (1996). Stability of PID Control for Industrial Robot Arms. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, August 1996, pp. 606-614, ISSN: 1042-296X
- Slotine, J-J.E. (1985). The robust control of robot manipulators. *International Journal of Robotics Research*. Vol. 4, No. 2, 1985, pp. 49-64, ISSN: 0278-3649
- Slotine, J-J.E. & Li, W. (1987). On the adaptive control of robot manipulators. *International Journal of Robotics Research*. Vol. 6, No. 3, 1987, pp. 49-59, ISSN: 0278-3649

- 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, July 2009, pp. 892-899, ISSN: 1063-6536
- Umeno, T.; Kaneko, T. & Hori, Y. (1993). Robust servosystem design with two degrees of freedom and its application to novel motion control of robot manipulators. *IEEE Transactions on Industrial Electronics*. Vol. 40, No. 5, October 1993, pp. 473-485, ISSN: 0278-0046
- Young K-K.D. (1978). Controller design for a manipulator using theory of variable structure systems. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 8, No. 2, February 1978, pp. 210-218, ISSN: 0018-9472

Global Stiffness Optimization of Parallel Robots Using Kinetostatic Performance Indices

Dan Zhang
*University of Ontario Institute of Technology
Canada*

1. Introduction

Global Stiffness design and optimization of parallel mechanisms can be a difficult and time-consuming exercise in parallel robot design, especially when the variables are multifarious and the objective functions are too complex. To address this issue, optimization techniques based on kinetostatic model and genetic algorithms are investigated as the effective criteria. First, a 5-DOF parallel mechanism with a passive constraining leg and five identical legs with prismatic actuators for machine tool is proposed, and its corresponding inverse kinematics, Jacobian matrices and global velocity equation are derived. Second, with the kinetostatic model, the mean value and the standard deviation of the trace of the global compliance distribution are proposed as these two kinetostatic performance indices. Finally, the effectiveness of this optimization design methodology for global stiffness indices is validated with simulation.

Compared with traditional serial manipulators, a parallel robot manipulator offers different potential benefits, including high rigidity, high accuracy, and high loading capacities. The parallel robot manipulator is used for applications where the demand on workspace and manoeuvrability is relatively low, while the dynamic loading is severe, and high speed and precision motions are primarily required. These applications include parallel kinematic machines (PKMs), aircraft flight simulators, telescope positioning, position tracker, and medical devices (Zhang & Gosselin, 2000; Dunlop & Jones, 1999; Carretero & Podhorodeski, 2000; Staicu *et al.*, 2006; Zhang & Wang, 2000; Liu *et al.*, 2005). Past research and development efforts with parallel robot manipulators have shown the ever-increasing demand on the robot's rigidity which is directly related to the system stiffness. In order to increase the production, a parallel manipulator which is capable of high speed operations with optimal rigidity is necessary.

Recently, researchers have been trying to utilize these advantages to develop parallel robot based multi-axis machining tools and precision assembly tools. Since most machining operations only require a maximum of five axes, new configurations with less than six axes would be more appropriate (Bi *et al.*, 2005). A 5-DOF parallel mechanism with a passive constraining leg and five identical legs with prismatic actuators for machine tool is proposed in this work. Kinetostatic analysis is essential for PKMs. A great deal of work so far has been

done on kinetostatic analysis that has direct application to PKMs (Birglen & Gosselin, 2004; Chablat & Angeles, 2002; Zhang & Gosselin, 2002), the issue of how to optimize the global stiffness based on kinetostatic modelling has not been well addressed. Two global compliance indices (kinetostatic performance indices) are introduced in this study, namely, the mean value and the standard deviation of the trace of the generalized stiffness matrix. The mean value represents the average stiffness of the PKMs over the workspace, while the standard deviation indicates the stiffness fluctuation relative to the mean value.

Many scholars have studied on optimum design of robot manipulators (Bergamaschi *et al.*, 2006; Stock & Miller, 2003; Rout & Mittal, 2008; Ceccarelli & Lanni, 2004). Lum *et al.* (Mitchell *et al.*, 2006) presented kinematic optimization to confirm the smallest configuration that would satisfy the workspace requirements for a lightweight and compact surgical manipulator. Chablat and Angeles (Chablat & Angeles, 2002) investigated on optimum dimensioning of revolute-coupled planar manipulators based on the concept of distance of Jacobian matrix to a given isotropic matrix which was used as a reference model. Zhao *et al.* (Zhao *et al.*, 2007) exploited the least number method of variables to optimize the leg length of a spatial parallel manipulator. Boeij *et al.* (Boeij *et al.*, 2008) proposed numerical integration and sequential quadratic programming method for optimization of a contactless electromagnetic planar 6-DOF actuator with manipulator on top of the floating platform. However, the traditional optimization methods only handle a few geometric variables due to the lack of convergence of the optimization algorithm. Genetic algorithms have applied the powerful and broadly applicable stochastic search methods and optimization techniques, and they can escape from local optima (Holland, 1975).

The remainder of the chapter is organized as follows. In Section 2, the structure of the tripod parallel manipulator and its parametric description is introduced. In Section 3, the kinetostatic analysis and stiffness modelling process is derived. In Section 4, the application of the integration approach is conducted to optimize the performance indices. Finally, the conclusions are given in Section 5.

2. Structure of the parallel manipulator

In this work, a 5-DOF parallel mechanism and its joint distributions both on the base and on the platform are shown in Figures 1, 2. This mechanism consists of six kinematic chains, including five variable length legs with identical topology and one passive leg which connect the fixed base to the moving platform. In this 5-DOF parallel mechanism, the kinematic chains associated with the five identical legs consist, from base to platform, of a fixed Hooke joint, a moving link, an actuated prismatic joint, a second moving link and a spherical joint attached to the platform. The sixth chain (central leg) connecting the base centre to the platform is a passive constraining leg and has architecture different from the other chains. It consists of a revolute joint attached to the base, a moving link, a Hooke joint, a second moving link and another Hooke joint attached to the platform. This last leg is used to constrain the motion of the platform to only five degrees of freedom. This mechanism could be built using only five legs, i.e., by removing one of the five identical legs and actuating the first joint of the passive constraining leg. However, the uniformity of the actuation would be lost.

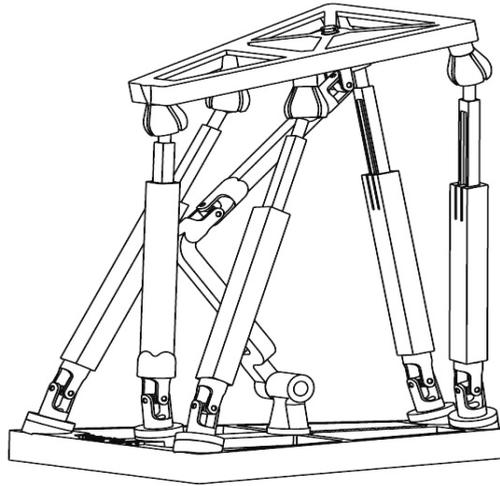


Fig. 1. CAD model of the 5-DOF parallel manipulator (by Gabriel Cote)

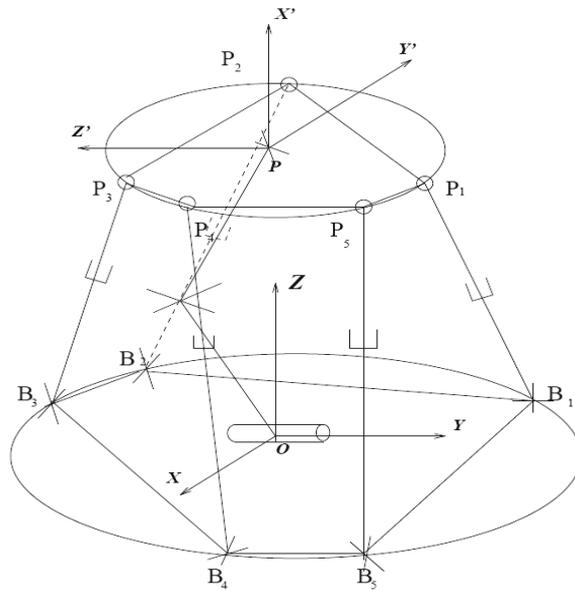


Fig. 2. Schematic representation of the 5-DOF parallel mechanism

Assume that the centres of the joints located on the base and on the platform are located on circles with radii R_b and R_p , respectively. A fixed reference frame $O-xyz$ is attached to the base of the mechanism and a moving coordinate frame $P-x'y'z'$ is attached to the platform. In Figure 2, the points of attachment of the actuated legs to the base are represented with B_i

and the points of attachment of all legs to the platform are represented by \mathbf{P}_i , with $i=1\dots 5$. Point \mathbf{P} is the reference point on the platform and its position coordinate is $\mathbf{P}(x,y,z)$.

The Cartesian coordinates of the platform are given by the position of point \mathbf{P} with respect to the fixed frame, and the orientation of the platform (orientation of frame $\mathbf{P} - x'y'z'$ with respect to the fixed frame), represented by rotation matrix \mathbf{Q} .

If the coordinates of the point \mathbf{P}_i in the moving reference frame are represented with (x'_i, y'_i, z'_i) and the coordinates of the point \mathbf{B}_i in the fixed frame are represented by vector \mathbf{b}_i , then for $i=1,\dots,5$, one has

$$\mathbf{p}_i = [x_i, y_i, z_i]^T \quad (1)$$

$$\mathbf{r}'_i = [x'_i, y'_i, z'_i]^T \quad (2)$$

$$\mathbf{p} = [x, y, z]^T \quad (3)$$

$$\mathbf{b}_i = [b_{ix}, b_{iy}, b_{iz}]^T \quad (4)$$

Where \mathbf{P}_i is the position vector of point \mathbf{P}_i expressed in the fixed coordinate frame whose coordinates are defined as (x_i, y_i, z_i) , \mathbf{r}'_i is the position vector of point \mathbf{P}_i expressed in the moving coordinate frame, and \mathbf{P} is the position vector of point \mathbf{P} expressed in the fixed frame as defined above.

One can then write

$$\mathbf{p}_i = \mathbf{p} + \mathbf{Q}\mathbf{r}'_i, \quad i = 1, \dots, 5 \quad (5)$$

Where, \mathbf{Q} is the rotation matrix from the fixed reference frame to the moving coordinate frame.

Subtracting vector \mathbf{b}_i from both sides of Eq. (5), one obtains

$$\mathbf{p}_i - \mathbf{b}_i = \mathbf{p} + \mathbf{Q}\mathbf{r}'_i - \mathbf{b}_i, \quad i = 1, \dots, 5 \quad (6)$$

Then, taking the Euclidean norm on both sides of Eq. (6), one has

$$\|\mathbf{p}_i - \mathbf{b}_i\| = \|\mathbf{p} + \mathbf{Q}\mathbf{r}'_i - \mathbf{b}_i\| = \rho_i, \quad i = 1, \dots, 5 \quad (7)$$

Where ρ_i is the length of the i th leg, i.e., the value of the i th joint coordinate. The solution of the inverse kinematic problem for the 5-DOF manipulator is therefore completed and can be written as

$$\rho_i^2 = (\mathbf{p}_i - \mathbf{b}_i)^T (\mathbf{p}_i - \mathbf{b}_i), \quad i = 1, \dots, 5 \quad (8)$$

Now considering the parallel component of the mechanism, the parallel Jacobian matrix can be obtained by differentiating Eq. (8) with respect to time, one obtains

$$\rho_i \dot{\rho}_i = (\mathbf{p}_i - \mathbf{b}_i)^T \dot{\mathbf{p}}_i, \quad i = 1, \dots, 5 \quad (9)$$

Since one has

$$\dot{\mathbf{Q}} = \mathbf{\Omega}\mathbf{Q} \quad (10)$$

with

$$\mathbf{\Omega} = \mathbf{1} \times \boldsymbol{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (11)$$

differentiating Eq. (5), one obtains

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}} + \dot{\mathbf{Q}}\mathbf{r}'_i \quad (12)$$

Then, Eq. (9) can be rewritten as

$$\boldsymbol{\rho}_i \dot{\mathbf{p}}_i = (\mathbf{p}_i - \mathbf{b}_i)^T \dot{\mathbf{p}} + [(\mathbf{Q}\mathbf{r}'_i) \times (\mathbf{p}_i - \mathbf{b}_i)]^T \boldsymbol{\omega}, \quad i = 1, \dots, 5 \quad (13)$$

Hence, one can write the velocity equation as

$$\mathbf{A}\mathbf{t} = \mathbf{B}\dot{\boldsymbol{\rho}} \quad (14)$$

where vector $\dot{\boldsymbol{\rho}}$ is defined as

$$\dot{\boldsymbol{\rho}} = [\dot{\rho}_1 \quad \dot{\rho}_2 \quad \dots \quad \dot{\rho}_5]^T \quad (15)$$

and

$$\mathbf{A} = [\mathbf{m}_1 \quad \mathbf{m}_2 \quad \dots \quad \mathbf{m}_5]^T \quad (16)$$

$$\mathbf{B} = \text{diag}[\rho_1 \quad \rho_2 \quad \dots \quad \rho_5] \quad (17)$$

where \mathbf{m}_i is a vector with 6 components, which can be expressed as

$$\mathbf{m}_i = \begin{bmatrix} (\mathbf{Q}\mathbf{r}'_i) \times (\mathbf{p}_i - \mathbf{b}_i) \\ (\mathbf{p}_i - \mathbf{b}_i) \end{bmatrix} \quad (18)$$

3. Kinetostatic analysis and stiffness modelling

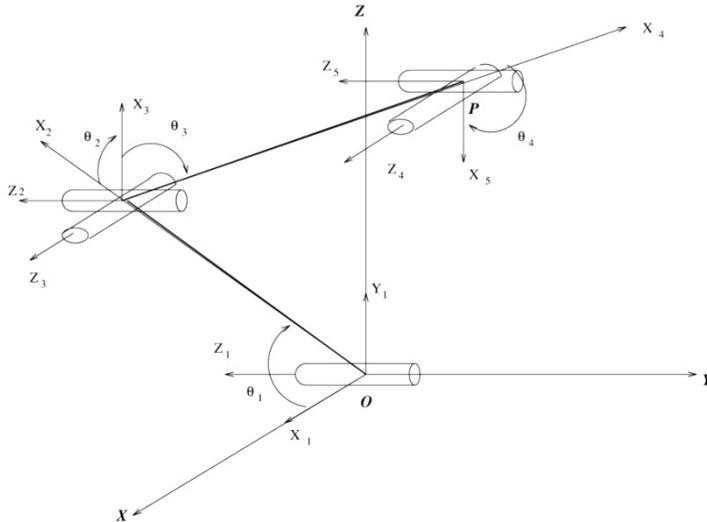


Fig. 3. The passive constraining leg with rigid links

i	a_i	b_i	α_i	θ_i
0	0	0	90°	0
1	$l_{passive,1}$	0	0	$\theta_{passive,1}$
2	0	0	90°	$\theta_{passive,2}$
3	$l_{passive,2}$	0	0	$\theta_{passive,3}$
4	0	0	90°	$\theta_{passive,4}$
5	0	0	0	$\theta_{passive,5}$

Table 1. The DH parameters for the passive constraining leg with rigid links

From Figure 3, one can obtain the Danavit-Hartenberg parameters of the passive leg as in Table 1. We take the Cartesian coordinate frame O , and define $\alpha_0=90^\circ$, $\theta_0=0^\circ$, then one obtains

$$\mathbf{Q}_{passive,0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (19)$$

The expressions for vectors $e_{passive,i}$ and $r_{passive,i}$ are then obtained following the procedure given above.

Stiffness is a very important factor in many applications including machine tool design, as it affects the precision of machining. Induced vibration is explicitly linked to machine tool stiffness. For a metal cutting machine tool, high stiffness allows higher machining speeds and feeds while providing the desired precision, thus reduce vibration (such as chatter). Therefore, to build and study a general stiffness model of parallel mechanisms is very important for machine tool design.

The parallel mechanisms studied here comprise two main components, namely, the constraining leg - which can be considered as a serial mechanism - and the actuated legs acting in parallel.

Considering the constraining leg, one can write

$$\mathbf{J}_{passive} \dot{\boldsymbol{\theta}}_{passive} = \mathbf{t} \quad (20)$$

where $\mathbf{t} = [\boldsymbol{\omega}^T \quad \dot{\mathbf{p}}^T]^T$ is the twist of the platform, with $\boldsymbol{\omega}$ the angular velocity of the platform and

$$\dot{\boldsymbol{\theta}}_{passive} = [\dot{\theta}_{passive,1} \quad \dots \quad \dot{\theta}_{passive,5}]^T \quad (21)$$

$$\mathbf{J}_{passive} = \begin{bmatrix} \mathbf{e}_{passive,1} & \mathbf{e}_{passive,2} & \mathbf{e}_{passive,3} & \mathbf{e}_{passive,4} & \mathbf{e}_{passive,5} \\ \mathbf{e}_{passive,1} \times \mathbf{r}_{passive,1} & \mathbf{e}_{passive,2} \times \mathbf{r}_{passive,2} & \mathbf{e}_{passive,3} \times \mathbf{r}_{passive,3} & \mathbf{e}_{passive,4} \times \mathbf{r}_{passive,4} & \mathbf{e}_{passive,5} \times \mathbf{r}_{passive,5} \end{bmatrix} \quad (22)$$

is the joint velocity vector associated with the constraining leg. Matrix $\mathbf{J}_{passive}$ is the Jacobian matrix of the constraining leg considered as a serial 5-dof mechanism.

According to the principle of virtual work, one has

$$\boldsymbol{\tau}^T \dot{\boldsymbol{\rho}} = \mathbf{w}^T \mathbf{t} \quad (23)$$

Where $\boldsymbol{\tau}$ is the vector of actuator forces applied at each actuated joint and \mathbf{w} is the wrench (torque and force) applied to the platform and where it is assumed that no gravitational forces act on any of the intermediate links. In practice, gravitational forces may often be neglected in machine tool applications.

One has $\mathbf{w} = [\mathbf{n}^T \quad \mathbf{f}^T]^T$ where \mathbf{n} and \mathbf{f} are respectively the external torque and force applied to the platform.

Rearranging Eq. (14) and substituting it into Eq. (23), one obtains

$$\boldsymbol{\tau}^T \mathbf{B}^{-1} \mathbf{A} \mathbf{t} = \mathbf{w}^T \mathbf{t} \quad (24)$$

Now, substituting Eq. (20) into Eq. (24), one has

$$\boldsymbol{\tau}^T \mathbf{B}^{-1} \mathbf{A} \mathbf{J}_{passive} \dot{\boldsymbol{\theta}}_{passive} = \mathbf{w}^T \mathbf{J}_{passive} \dot{\boldsymbol{\theta}}_{passive} \quad (25)$$

The latter equation must be satisfied for arbitrary values of $\dot{\theta}_{\text{passive}}$ and hence one can write

$$(\mathbf{A}\mathbf{J}_{\text{passive}})^T \mathbf{B}^{-T} \boldsymbol{\tau} = \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (26)$$

The latter equation relates the actuator forces to the Cartesian wrench, \mathbf{w} , applied at the end-effector in static mode. Since all links are assumed rigid, the compliance of the mechanism will be induced solely by the compliance of the actuators. An actuator compliance matrix \mathbf{C} is therefore defined as

$$\mathbf{C}\boldsymbol{\tau} = \Delta\boldsymbol{\rho} \quad (27)$$

Where $\boldsymbol{\tau}$ is the vector of actuated joint forces and $\Delta\boldsymbol{\rho}$ is the induced joint displacement. Matrix \mathbf{C} is a $(n \times n)$ diagonal matrix whose i th diagonal entry is the compliance of the i th actuator.

Now, Eq. (26) can be rewritten as

$$\boldsymbol{\tau} = \mathbf{B}^T (\mathbf{A}\mathbf{J}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (28)$$

The substitution of Eq. (28) into Eq. (27) then leads to

$$\Delta\boldsymbol{\rho} = \mathbf{C}\mathbf{B}^T (\mathbf{A}\mathbf{J}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (29)$$

Moreover, for a small displacement vector $\Delta\boldsymbol{\rho}$, Eq. (14) can be written as

$$\Delta\boldsymbol{\rho} \cong \mathbf{B}^{-1} \mathbf{A} \Delta\mathbf{c} \quad (30)$$

where $\Delta\mathbf{c}$ is a vector of small Cartesian displacement and rotation defined as

$$\Delta\mathbf{c} = \begin{bmatrix} \Delta\mathbf{p}^T & \Delta\boldsymbol{\alpha}^T \end{bmatrix}^T \quad (31)$$

in which $\Delta\boldsymbol{\alpha}$, the change of orientation, is defined from Eqs. (10) and (11) as

$$\Delta\boldsymbol{\alpha} = \text{vect}(\Delta\mathbf{Q}\mathbf{Q}^T) \quad (32)$$

where $\Delta\mathbf{Q}$ is the variation of the rotation matrix and $\text{vect}(\cdot)$ is the vector linear invariant of its matrix argument.

Similarly, Eq. (20) can also be written, for small displacements, as

$$\mathbf{J}_{\text{passive}}^T \Delta\boldsymbol{\theta}_{\text{passive}} \cong \Delta\mathbf{c} \quad (33)$$

where $\Delta\boldsymbol{\theta}_{\text{passive}}$ is a vector of small variations of the joint coordinates of the constraining leg. Substituting Eq. (29) into Eq. (30), one obtains

$$\mathbf{B}^{-1} \mathbf{A} \Delta\mathbf{c} = \mathbf{C}\mathbf{B}^T (\mathbf{A}\mathbf{J}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (34)$$

Premultiplying both sides of Eq. (34) by \mathbf{B} , and substituting Eq. (33) into Eq. (34), one obtains,

$$\mathbf{AJ}_{\text{passive}} \Delta \boldsymbol{\theta}_{\text{passive}} = \mathbf{BCB}^T (\mathbf{AJ}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (35)$$

Then, pre-multiplying both sides of Eq. (35) by $(\mathbf{AJ}_{\text{passive}})^{-1}$, one obtains,

$$\Delta \boldsymbol{\theta}_{\text{passive}} = (\mathbf{AJ}_{\text{passive}})^{-1} \mathbf{BCB}^T (\mathbf{AJ}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (36)$$

and finally, premultiplying both sides of Eq. (36) by $\mathbf{J}_{\text{passive}}$, one obtains,

$$\Delta \mathbf{c} = \mathbf{J}_{\text{passive}} (\mathbf{AJ}_{\text{passive}})^{-1} \mathbf{BCB}^T (\mathbf{AJ}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \mathbf{w} \quad (37)$$

Hence, one obtains the Cartesian compliance matrix as

$$\mathbf{C}_c = \mathbf{J}_{\text{passive}} (\mathbf{AJ}_{\text{passive}})^{-1} \mathbf{BCB}^T (\mathbf{AJ}_{\text{passive}})^{-T} \mathbf{J}_{\text{passive}}^T \quad (38)$$

with

$$\Delta \mathbf{c} = \mathbf{C}_c \mathbf{w} \quad (39)$$

where \mathbf{C}_c is a symmetric positive semi-definite (6×6) matrix, as expected.

It is noted that, in non-singular configurations, the rank of \mathbf{B} , \mathbf{C} and $\mathbf{J}_{\text{passive}}$ is five, and hence the rank of \mathbf{C}_c will be five, depending on the degree of freedom of the mechanism. Hence, the nullspace of matrix \mathbf{C}_c will not be empty and there will exist a set of vectors \mathbf{w} that will induce no Cartesian displacement $\Delta \mathbf{C}$. This corresponds to the wrenches that are supported by the constraining leg, which is considered infinitely rigid. These wrenches are orthogonal complements of the allowable twists at the platform. Hence, matrix \mathbf{C}_c cannot be inverted and this is why it was more convenient to use compliance matrices rather than stiffness matrices in the above derivation.

Furthermore, the diagonal elements of the stiffness matrix are used as the system stiffness value. These elements represent the pure stiffness in each direction, and they reflect the rigidity of machine tools more clearly and directly. The objective function for mean value and standard deviation of system stiffness can be written as:

$$\boldsymbol{\mu}\text{-compliance} = E(\text{tr}(\mathbf{C}_c)) \quad (40)$$

$$\boldsymbol{\sigma}\text{-compliance} = \text{STD}(\text{tr}(\mathbf{C}_c)) \quad (41)$$

Where $E(\cdot)$ and $\text{STD}(\cdot)$ represent the mean value and the standard deviation respectively, and tr is the trace of the stiffness matrix.

4. Design optimization

4.1 Principles

We propose the mean value and the standard deviation of the trace of the generalized compliance matrix as the design indices. The purpose of design optimization is to evolve the performance indices by adjusting the structure parameters. It is noted that the trace of the matrix is an invariant of the matrix, so the distribution of the system stiffness/compliance is the distribution of the trace. The mean value represents the average compliance of the parallel robot manipulator over the workspace, while the standard deviation indicates the compliance fluctuation relative to the mean value. In general the lower the mean value the

less the deformation. Similarly, the lower the standard deviation the more uniform the stiffness distribution over the workspace. The suitability of these design indices for the system stiffness can be examined by developing their relationship with the stiffness of links and joints. We will further study a design optimization based on the compliance indices.

Since only a few geometric parameters can be handled due to the lack of convergence, this arises from the fact that traditional optimization methods use a local search by a convergent stepwise procedure, e.g. gradient, Hessians, linearity, and continuity, which compares the values of the next points and moves to the relative optimal points (Gosselin & Guillot, 1991). Global optima can be found only if the problem possesses certain convexity properties which essentially guarantee that any local optima are a global optimum. In other words, conventional methods are based on a point-to-point rule; it has the danger of falling in local optima. The genetic algorithms are based on the population-to-population rule; it can escape from local optima.

Genetic algorithms have the advantages of robustness and good convergence properties:

- They require no knowledge or gradient information about the optimization problems; only the objective function and corresponding fitness levels influence the directions of search.
- Discontinuities present on the optimization problems have little effect on the overall optimization performance.
- They are generally more straightforward to introduce, since no restrictions for the definition of the objective function exist.
- They use probabilistic transition rules, not deterministic ones.
- They perform well for large-scale optimization problems.

4.2 Optimization

In order to obtain the maximum global stiffness, the global compliance (since there are infinite terms among the diagonal stiffness elements) is minimized. As Cartesian stiffness is a monotonically increasing function of the link and actuator stiffness, the optimum solution always corresponds to the maximum link or actuator stiffness and these parameters are not included in the optimization variables. Seven geometrical parameters are selected as the pending optimization variables in order to obtain the optimal system stiffness, i.e.

$$\mathbf{s} = [R_p, R_b, l_{61}, l_{62}, z, T_p, T_b] \quad (42)$$

where R_p is the radius of the platform, R_b is the radius of the base, l_{61} , l_{62} are the link length for the 1st and 2nd link of the passive leg, respectively, z is the height of the platform, T_p, T_b are the angles to determine the attachment points on the base and on the platform, and their bounds are

$$\begin{aligned} R_p &\in [0.10, 0.14] m, R_b \in [0.20, 0.26] m, \\ l_{61} &\in [0.52, 0.70] m, l_{62} \in [0.52, 0.70] m, \\ z &\in [0.66, 0.70] m, \\ T_p &\in [18, 26]^\circ, T_b \in [38, 48]^\circ \end{aligned}$$

Some genetic parameters and operators are set as:

Variable representation format: real value

Selection: roulette wheel approach

Crossover operator: multi-point crossover

Crossover rate: 0.9

Mutation operator: multiple-point bit mutation

Mutation rate: 0.005

Population size: 200

Maximum number of generations: 40

The input vectors are the random arrangement of discretization values from the seven structure variables. The objective function is defined as

$$ObjFun = E(tr(\mathbf{C}_c)) + STD(tr(\mathbf{C}_c)) = \mu + \sigma \quad (43)$$

The evolution of system stiffness/compliance value arises from the optimization of architecture and behaviour variables in the implementation process of genetic algorithm as shown in Figure 4. By simultaneously adjusting the seven parameters, optimization results are obtained after 40 generations as follows

$$\mathbf{s} = [R_p, R_b, l_{61}, l_{62}, z, T_p, T_b]$$

$$= [0.14m, 0.209m, 0.52m, 0.7m, 0.66m, 18^\circ, 48^\circ]$$

and the compliances in each direction are

$$\mathbf{K} = \begin{bmatrix} K_{\theta_x} & K_{\theta_y} & K_{\theta_z} & K_x & K_y & K_z \end{bmatrix}$$

$$= [0.0363, 0.0293, 0.0381, 0.0413, 0.0177, 0.0002]$$

Figure 5 describes the evolution of the best individual, and the sum of the compliances is convergent at 0.0691 m/N after 40 generations.

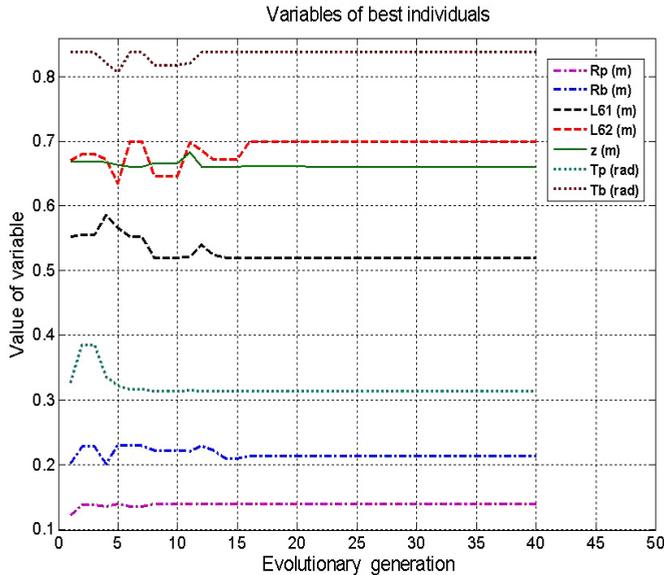


Fig. 4. The evolution of geometrical parameters for stiffness index

Before optimization, the parameter values of the mechanism were given as

$$\begin{aligned} \mathbf{s}' &= [R_p, R_b, l_{61}, l_{62}, z, T_p, T_b] \\ &= [0.11m, 0.25m, 0.68m, 0.52m, 0.68m, 22^\circ, 42^\circ] \end{aligned}$$

and the compliances in each direction were

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} K'_{\theta_x} & K'_{\theta_y} & K'_{\theta_z} & K'_x & K'_y & K'_z \end{bmatrix} \\ &= [0.1224, 0.1971, 0.7151, 0.0640, 0.0198, 0.0003] \end{aligned}$$

the compliance sum is 0.6416 m/N. After optimization, the compliance sum is improved 9.3 times.

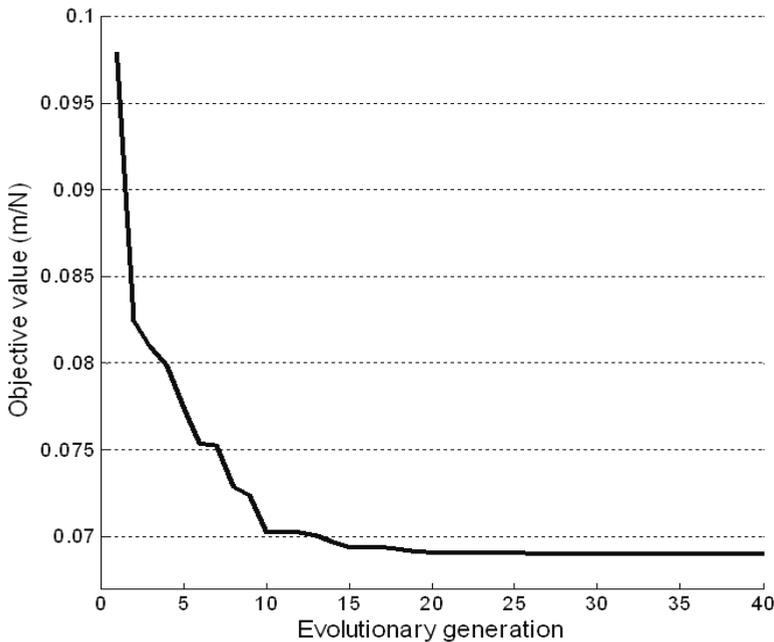


Fig. 5. The evolution of the performance

5. Conclusions

This chapter focused on the stiffness optimization of a spatial 5-DOF parallel manipulator. It is shown that the mean value and the standard deviation of the trace of the generalized compliance matrix can not only be used to characterize the kinetostatic behaviour of PKMs globally, but can be used for design optimization. This methodology paves the way for providing not only the effective guidance, but also a new approach of dimensional synthesis for the optimal design of general parallel mechanisms.

6. References

- Zhang D, Gosselin C M. Kinetostatic modeling of parallel mechanisms with a passive constraining leg and revolute actuators. *Mech Mach Theory*, 2000, 37: 599-617
- Dunlop G R, Jones T P. Position analysis of a two DOF parallel mechanism-Canterbury tracker. *Mech Mach Theory*, 1999, 34: 599-614
- Carretero J A., Podhorodeski R P, Nahon M N, *et al.* Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. *J Mech Design*, 2000, 122:17-24
- Staicu S, Zhang D, Rugescu R D. Dynamic modelling of a 3-DOF parallel manipulator using iterative matrix relations. *Robotica*, 2006, 24(1):125-130
- Zhang D, Wang L. PKM capabilities and applications exploration in a collaborative virtual environment. *Robot Cim-Int Manuf*, 2000, 22: 384-395
- Liu X J, Wang J S, Pritschow G. A new family of spatial 3-DoF fully-parallel manipulators with high rotational capability. *Mech Mach Theory*, 2005, 40: 475-494
- Bi Z M, Lang S Y, Zhang D, *et al.* An integrated design toolbox for tripod-based parallel kinematic machines. *J Mech Design*, 2007, 129: 799-807
- Birglen L, Gosselin C M. Kinetostatic analysis of underactuated fingers. *IEEE T Robot*, 2004, 20 (2):211 - 221
- Chablat D, Angeles J. On the kinetostatic optimization of revolute-coupled planar manipulators. *Mech Mach Theory*, 2002, 37: 351-374
- Zhang D and Gosselin C M. Parallel kinematic machine design with kinetostatic model. *Robotica*, 2002, 4(7): 429-438
- Bergamaschi P R, Nogueira A C, Saramago S F. Design and optimization of 3R manipulators using the workspace features. *Appl Math Comput*, 2006, 172: 439-463
- Stock M, Miller K. Optimal kinematic design of spatial parallel manipulators: Application to linear delta robot. *J Mech Design*, 2003, 125: 292-301
- Rout B K, Mittal R K. Parametric design optimization of 2-DOF R-R planar manipulator: A design of experiment approach. *Robot Cim-Int Manuf*, 2008, 24:239-248
- Ceccarelli M, Lanni C. A multi-objective optimum design of general 3R manipulators for prescribed workspace limits. *Mech Mach Theory*, 2004, 39:119-132
- Mitchell J. H. Lum, Jacob Rosen, Mika N. Sinanan, *et al.* Optimization of a spherical mechanism for a minimally invasive surgical robot: theoretical and experimental approaches, *IEEE T Bio-Med Eng*, 2006, 53(7):1440 - 445
- Zhao J S, Zhang S L, Dong J X. Optimizing the kinematic chains for a spatial parallel manipulator via searching the desired dexterous workspace. *Robot Cim-Int Manuf*, 2007, 23: 38-46
- Boeij J, Lomonova E A, André J A, *et al.* Optimization of contactless planar actuator with manipulator, *IEEE T Magn*, 2008, 44(6):1118 -1121
- Holland J. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, MI, 1975
- Gosselin CM, Guillot M. The synthesis of manipulators with prescribed workspace, *ASME J Mech Des* 1991; 113(1):451-5.

Measurement Analysis and Diagnosis for Robot Manipulators using Advanced Nonlinear Control Techniques

Amr Pertew, Ph.D, P.Eng.
*Computer and Systems Engineering Department, University of Alexandria
 Egypt*

Horacio Marquez, Ph. D, P. Eng and Qing Zhao, Ph. D, P. Eng
*Electrical and Computer Engineering Department, University of Alberta
 Canada*

1. Introduction

The *measurement analysis and diagnosis problem* is gaining increasing consideration worldwide in both theory and application. This is due to the growing demand for higher reliability in control systems, and hence the importance of having a monitoring system that detects any existing measurement errors, and indicates their location and significance in the control loop. The observer-based approach is one of the most popular techniques used for fault diagnosis in general and for the measurement error diagnosis problem in particular. Many standard observer-based techniques exist in the literature providing different solutions to both the theoretical and practical aspects of the problem for the *Linear Time-Invariant* (LTI) case (see (Frank, 1990; Willsky, 1976) for good surveys on this subject). The main idea behind the observer-based approach is to estimate the outputs of the system from the measurements by using either *static gain* observers in a deterministic framework (Zhong et al., 2003) or *Kalman* filters in a stochastic framework (Chen et al., 2003). The output estimation error is then used as the residual signal, which can be analyzed further to obtain an accurate estimation of the measurement errors which affect the control system. Unlike the LTI case, however, the nonlinear problem lacks a universal approach and is currently an active area of research (see (Adjallah et al., 1994; Garcia & Frank, 1997; Hammouri et al., 1999; Kabore & Wang, 2001; Vemuri, 2001; Wang et al., 1997; Yu & Shields, 1996) for important results in this direction). The main obstacle in the solution of the observer-based nonlinear fault detection problem is the lack of a universal approach for nonlinear observer synthesis. Robot manipulators, characterized by largely nonlinear dynamics, are no exception to this dilemma and therefore need a unified framework for measurement error detection and diagnosis.

The well known Euler-Lagrange model of a robot manipulator is as follows (Sciavicco & Siciliano, 1989):

$$u = M(\theta) \ddot{\theta} + V(\theta, \dot{\theta}) \quad (1)$$

where θ , $\dot{\theta}$ and $\ddot{\theta} \in \mathbb{R}^n$ are vectors representing the position, velocity and acceleration of the n -link robot respectively, and $u \in \mathbb{R}^n$ represents the actuator torques. By defining the state variables as $x_1 = \theta$, $x_2 = \dot{\theta}$, we get the state space model:

$$\dot{x} = f(x) + g(x)u \quad (2)$$

where $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $f(x) = \begin{bmatrix} x_2 \\ -M^{-1}(x_1)V(x_1, x_2) \end{bmatrix}$ and $g(x) = \begin{bmatrix} 0_n \\ M^{-1}(x_1) \end{bmatrix}$. This model clearly shows the nonlinearities resulting from the inertia matrix $M(\theta)$, and from the vector $V(\theta, \dot{\theta})$ which combines all the centrifugal, coriolis, gravity and friction terms that are an integral part of the manipulator dynamical equations.

The problem of measurement error diagnosis for robot manipulators described by the Euler-Lagrange model is an interesting one. Sensor faults could affect both the position and velocity measurements, and would have a detrimental effect on the control performance. The dynamic observer structure which was introduced in (Pertew et al., 2006; 2007) can serve as a general framework for solving this problem. Necessary and sufficient design conditions for the problem of error detection can be derived from the nonlinear dissipativity properties inherent in the robot structure. Another important advantage of this approach stems from combining the objectives of estimating the error magnitudes, as well as detecting and isolating the error sources. Using the extra degrees of freedom in the dynamic observer structure, the problem is solvable by minimizing the fault effects in a narrow frequency band on the observer's estimation error. This is an important advantage of the new framework over the classical constant gain structure. The robot manipulator measurement error diagnosis problem could then be generally modelled as a convex optimization problem and solved using a standard *Linear Matrix Inequality* (LMI) design procedure. Using standard weightings, different frequency patterns for the measurement errors can also be considered.

2. Nonlinear Observers and Terminology

An important class of nonlinear systems is:

$$\dot{x}(t) = Ax(t) + \Gamma(u, t) + \Phi(x, u, t) \quad (3)$$

$$y(t) = Cx(t) + f(t), \quad A \in \mathbb{R}^{n \times n}, \quad C \in \mathbb{R}^{p \times n} \quad (4)$$

where (A, C) detectable, $f(t)$ represent measurement errors, and $\Phi(x, u, t)$ satisfies:

$$\| \Phi(x_1, u, t) - \Phi(x_2, u, t) \| \leq \alpha \| x_1 - x_2 \| \quad (5)$$

$\forall u \in \mathbb{R}^m$ and $t \in \mathbb{R}$ and $\forall x_1$ and $x_2 \in D$, where D is a closed and bounded region containing the origin. These systems have been widely referred to as nonlinear Lipschitz systems, due to the Lipschitz continuity condition in (5) that affect them. Due to their importance and the variety of nonlinear systems that they cover, much effort has been done towards solving the observer design problem for this class of nonlinear systems. One important advantage is that robot manipulators fall in the class of nonlinear Lipschitz systems. The classical observer structure for Lipschitz systems of the form (3)-(5), with no measurement errors is the well-known Luenberger structure, represented by the following equations:

$$\dot{\hat{x}} = A\hat{x} + \Gamma(u, t) + \Phi(\hat{x}, u, t) + L(y - \hat{y}), \quad L \in \mathbb{R}^{n \times p} \quad (6)$$

$$\hat{y} = C\hat{x} \quad (7)$$

A more general framework introduced in (Pertew et al., 2006), makes use of dynamical observers of the form:

$$\dot{\hat{x}} = A\hat{x} + \Gamma(u, t) + \Phi(\hat{x}, u, t) + \eta \quad (8)$$

$$\dot{\xi} = A_L \xi + B_L(y - \hat{y}), \quad A_L \in \mathbb{R}^{k \times k}, \quad B_L \in \mathbb{R}^{k \times p} \quad (9)$$

$$\eta = C_L \xi + D_L(y - \hat{y}), \quad C_L \in \mathbb{R}^{n \times k}, \quad D_L \in \mathbb{R}^{n \times p} \quad (10)$$

$$\hat{y} = C\hat{x} \quad (11)$$

We will write $K = \left[\begin{array}{c|c} A_L & B_L \\ \hline C_L & D_L \end{array} \right]$ to represent the dynamic observer gain in (23)-(24). It can be shown that K , sufficient to achieve observer convergence, can be represented by a set of controllers. This design freedom can be used in the measurement error diagnosis problem discussed in this chapter, by analyzing the residual signal:

$$r(t) = y(t) - \hat{y}(t) \quad (12)$$

This will be discussed in details throughout this chapter. To this end, the following definitions and notation are widely accepted and used by the *Fault Detection and Diagnosis* (FDD) community:

Definition 1. *Fault detection: The residual in equation (12) achieves fault detection (strong fault detection) if the following condition is satisfied:*

$$r(t) = 0; \forall t \text{ if (if and only if) } f(t) = 0; \forall t$$

Definition 2. *Fault isolation: The residual in (12) achieves fault isolation if:*

$$(r_i(t) = 0; \forall t \iff f_i(t) = 0; \forall t); \text{ for } i = 1, \dots, p$$

Definition 3. *Fault identification: Fault identification is satisfied by (12) if:*

$$(r_i(t) = f_i(t); \forall t); \text{ for } i = 1, \dots, p$$

The previous definitions are borrowed from (Chen & Patton, 1999). Note that, in these definitions, the transient period of the residual signal is not considered, and that since the focus in this chapter is on sensor faults the term “measurement error” will be used instead of “fault” throughout the chapter.

The following definition and notation will also be used in this chapter:

Definition 4. \mathcal{L}_2 space: Space \mathcal{L}_2 consists of all Lebesgue measurable functions $u : \mathbb{R}^+ \rightarrow \mathbb{R}^q$, with finite $\|u\|_{\mathcal{L}_2}$, where $\|u\|_{\mathcal{L}_2} \triangleq \sqrt{\int_0^\infty \|u(t)\|^2 dt}$.

For a system $H : \mathcal{L}_2 \rightarrow \mathcal{L}_2$, we will represent by $\gamma(H)$ the \mathcal{L}_2 gain of H defined by $\gamma(H) = \sup_u \frac{\|Hu\|_{\mathcal{L}_2}}{\|u\|_{\mathcal{L}_2}}$. It is well known that, for a LTI system $H : \mathcal{L}_2 \rightarrow \mathcal{L}_2$ (with a transfer matrix $\hat{H}(s)$), $\gamma(H) \equiv \|\hat{H}(s)\|_\infty \triangleq \sup_{\omega \in \mathbb{R}} \sigma_{\max}(\hat{H}(j\omega))$. The matrices I_n , 0_n and 0_{nm} will represent the identity matrix of order n , the zero square matrix of order n and the zero n by m matrix respectively. $Diag_r(a)$ represents the diagonal square matrix of order r with $[a \ a \ \dots \ a]_{1 \times r}$ as its diagonal vector, while $diag(a_1, a_2, \dots, a_r)$ represents the one with $[a_1 \ a_2 \ \dots \ a_r]$ as its diagonal vector. \hat{T}_{yu} represents the transfer matrix

from input u to output y . RH_∞ denotes the space of all proper real rational stable transfer matrices. The partitioned matrix $H = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ (when used in $y = Hu$) represents $(\dot{\xi} = A\xi + Bu, y = C\xi + Du)$, and $\hat{H}(s) = C(sI - A)^{-1}B + D$. We will make use of (13) (Zhou & Doyle, 1998), if s is not an eigenvalue of A :

$$\text{rank} \begin{bmatrix} A - sI & B \\ C & D \end{bmatrix} = n + \text{rank}(\hat{H}(s)), \quad n \text{ being the dimension of } A \quad (13)$$

The setup in Fig. 1 will also be used throughout the chapter along with:

$$G = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \quad (14)$$

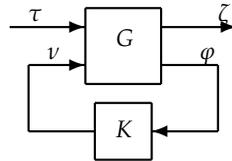


Fig. 1. Standard setup.

We will also make use of the following result from (Gahinet & Apkarian, 1994):

Theorem 1. Assume stabilizability and detectability of (A, B_2, C_2) and that $D_{22} = 0$, and let \mathcal{N}_{12} and \mathcal{N}_{21} denote orthonormal bases of the null spaces of (B_2^T, D_{12}^T) and (C_2, D_{21}) . There exists a controller K such that $\|\hat{T}_{z\tau}\|_\infty < \gamma$ if and only if there exist symmetric $R, S \in \mathbb{R}^{n \times n}$ satisfying the following system of LMIs:

$$\begin{bmatrix} \mathcal{N}_{12} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} AR + RA^T & RC_1^T & B_1 \\ C_1R & -\gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{bmatrix} \begin{bmatrix} \mathcal{N}_{12} & 0 \\ 0 & I \end{bmatrix} < 0 \quad (15)$$

$$\begin{bmatrix} \mathcal{N}_{21} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} A^T S + SA & SB_1 & C_1^T \\ B_1^T S & -\gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix} \begin{bmatrix} \mathcal{N}_{21} & 0 \\ 0 & I \end{bmatrix} < 0 \quad (16)$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \quad (17)$$

3. Robot Manipulator Measurement Error Detection

As mentioned earlier, robot manipulators fall in the category of nonlinear Lipschitz systems. This is clear by rewriting the robot manipulator model in (2) as:

$$\dot{x} = Ax + \Phi(x, u, t) \quad (18)$$

where $A = \begin{bmatrix} 0_n & I_n \\ 0_n & 0_n \end{bmatrix}$, $\Phi = \begin{bmatrix} 0_n \\ M^{-1}(x_1)u - M^{-1}(x_1)V(x_1, x_2) \end{bmatrix}$. It is important to note that the nonlinear terms in Φ are mainly trigonometric terms which are locally Lipschitz, and an upper bound on the Lipschitz constant can be found by computing $\|\partial\Phi(x, u, t)/\partial x\|$ over the operating range (Marquez, 2003). Another representation of (2) around an operating point x^* of interest is:

$$\dot{x} = Ax + Bu + \Phi(x, u, t) \quad (19)$$

where $A = \left(\frac{\partial f}{\partial x}\right)|_{x^*}$, $B = \left(\frac{\partial g}{\partial x}\right)|_{x^*}$, $\Phi = (f(x) - Ax + g(x)u - Bu)$.

It is also important to note that (18) and (19) are both exact models of (2). By neglecting the terms in Φ in (19), one gets the well known approximate linearized model around the operating point x^* , i.e:

$$\dot{x} = Ax + Bu \quad (20)$$

where $A = \left(\frac{\partial f}{\partial x}\right)|_{x^*}$, $B = \left(\frac{\partial g}{\partial x}\right)|_{x^*}$, which is an approximate model of (2). In the formulation used in this chapter, however, no approximation is needed and the exact Lipschitz model in (19) can be directly use for solving the measurement error diagnosis problem.

Starting by the measurement error detection problem represented by Definition 1, the robot measurements are assumed to be any linear combination of the position and velocity sensors which satisfies the detectability condition needed for observer design. Errors in measurements affect the system, and the purpose is to design an observer which satisfies the measurement error detectability condition as per Definition 1. By making use of dynamical observers of the form:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + \Gamma(u, t) + \Phi(\hat{x}, u, t) + \eta(t) \quad (21)$$

$$\hat{y}(t) = C\hat{x}(t) \quad (22)$$

where $\eta(t)$ is obtained by applying a dynamical compensator K of order k ("k" being arbitrary) on the output estimation error. In other words $\eta(t)$ is given from

$$\dot{\zeta} = A_L\zeta + B_L(y - \hat{y}), \quad A_L \in \mathbb{R}^{k \times k}, \quad B_L \in \mathbb{R}^{k \times p} \quad (23)$$

$$\eta = C_L\zeta + D_L(y - \hat{y}), \quad C_L \in \mathbb{R}^{n \times k}, \quad D_L \in \mathbb{R}^{n \times p} \quad (24)$$

We will also write

$$K = \left[\begin{array}{c|c} A_L & B_L \\ \hline C_L & D_L \end{array} \right] \quad (25)$$

to represent the compensator in (23)-(24). It is straightforward to see that this observer structure reduces to the usual observer in (6)-(7) when $K = \left[\begin{array}{c|c} 0_k & 0_{kp} \\ \hline 0_{nk} & L \end{array} \right]$. The additional dynamics brings additional degrees of freedom in the design, something that could be used to add the measurement error detection objective to the state estimation problem in the observer design. The observer error dynamics is now given by

$$\dot{e}(t) = A e + \Phi(x, u, t) - \Phi(\hat{x}, u, t) - \eta(t) \quad (26)$$

$$r(t) = Ce(t) + f(t) \quad (27)$$

which can also be represented by the setup in Fig. 1 where G has the state space representation in (14) with appropriate matrices and with the following variables:

$$\tau = \tilde{\phi} = \Phi(x, u, t) - \Phi(\hat{x}, u, t), \zeta = e = x - \hat{x}, v = \eta = K(y - \hat{y}), \text{ and } \varphi = y - \hat{y} \quad (28)$$

We denote by $\hat{T}_{\zeta\tau}$ the transfer function between τ and ζ for this setup. The following theorem provides a general solution to the dynamic observer condition needed to achieve measurement error detection:

Theorem 2. *Given the nonlinear system in (3)-(5), the residual signal in (8)-(12) achieves measurement error detection, $\forall \Phi$ satisfying the Lipschitz condition in (5) with a Lipschitz constant α , if the observer gain K is chosen such that: $\sup_{\omega \in \mathbb{R}} \sigma_{\max}[\hat{T}_{\zeta\tau}(j\omega)] < \frac{1}{\alpha}$.*

Proof: The proof is built on proving that, when the measurement error vector $f = 0$, the state \hat{x} of the observer (8)-(11) asymptotically converges to the system state x for all $\Phi(x, u, t)$ satisfying (5) with a Lipschitz constant α if the dynamic observer gain K is chosen s.t:

$$\sup_{\omega \in \mathbb{R}} \sigma_{\max}[\hat{T}_{\zeta\tau}(j\omega)] < \frac{1}{\alpha} \quad (29)$$

Using the variable definitions in (28) it can be seen that $\hat{T}_{\zeta\tau}$ can be represented as:

$$\hat{T}_{\zeta\tau} = \hat{T}_{e\tilde{\phi}} = \left[\begin{array}{cc|c} A - D_L C & -C_L & I_n \\ B_L C & A_L & 0_{kn} \\ \hline I_n & 0_{nk} & 0_n \end{array} \right] \quad (30)$$

and is such that $\gamma(\hat{T}_{e\tilde{\phi}}) = \|\hat{T}_{e\tilde{\phi}}\|_{\infty} < \frac{1}{\alpha}$ according to (29). The proof for sufficiency follows from noting that the estimation error e is given from the feedback interconnection of $\hat{T}_{e\tilde{\phi}}$ and Δ as shown in Fig. 2 where Δ is the static nonlinear time-varying operator defined as follows:

$$\begin{aligned} \Delta(t) : e \rightarrow \tilde{\phi} &= \Phi(x, u, t) - \Phi(\hat{x}, u, t) \\ &= \Phi(e + \hat{x}(t), u(t), t) - \Phi(\hat{x}(t), u(t), t) \end{aligned}$$

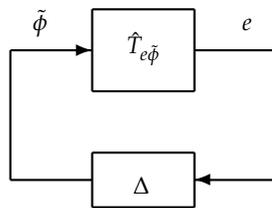


Fig. 2. Feedback interconnection.

In this loop, $\gamma(\hat{T}_{e\tilde{\phi}}) < \frac{1}{\alpha}$ as mentioned earlier. Although an exact expression for Δ is not available, we have $\gamma(\Delta) \leq \alpha$ as from the Lipschitz condition in (5) it follows that

$$\gamma(\Delta) \leq \frac{\sqrt{\int_0^{\infty} \alpha^2 \|x - \hat{x}\|^2 dt}}{\sqrt{\int_0^{\infty} \|x - \hat{x}\|^2 dt}} \leq \alpha$$

Using the bounds on the \mathcal{L}_2 gains of the operators $\hat{T}_{e\tilde{\phi}}$ and Δ , nonlinear dissipativity of the feedback interconnection is satisfied, by noting the following properties for the feedback loop in Fig. 2:

- (a) Δ is a static nonlinearity (no internal states) and $\hat{T}_{e\tilde{\phi}}$ is the dynamic LTI system in (30).
- (b) The mappings $\hat{T}_{e\tilde{\phi}} : \tilde{\phi} \rightarrow e$ and $\Delta : e \rightarrow \tilde{\phi}$ have finite \mathcal{L}_2 gains $\gamma(\hat{T}_{e\tilde{\phi}})$ and $\gamma(\Delta)$, and moreover they satisfy $\gamma(\hat{T}_{e\tilde{\phi}})\gamma(\Delta) < 1$.
- (c) $\hat{T}_{e\tilde{\phi}}$ and Δ are dissipative with the supply rates $\omega_1 = -e^T e + \gamma(\hat{T}_{e\tilde{\phi}})^2 \tilde{\phi}^T \tilde{\phi}$ and $\omega_2 = -\tilde{\phi}^T \tilde{\phi} + a^2 e^T e$ respectively. We will denote by S_1 and S_2 the corresponding storage functions.

It is a direct application of Corollary 1 in (Hill & Moylan, 1977) (see also (Marquez, 2003), Chapter 9, for a comprehensive review of the subject) that $S_1 + aS_2$, $a > 0$, is a Lyapunov function for this system, and that, since $\gamma(\hat{T}_{e\tilde{\phi}})\gamma(\Delta) < 1$, the system is asymptotically stable. This implies that $e \rightarrow 0$ as $t \rightarrow \infty$.

It also follows that under the same conditions of Theorem 2, if condition (5) holds locally, then local asymptotic convergence of the observer is guaranteed (and in this case the observer (8)-(11) is a local one, i.e. it is local in “ x ” and in the estimation error “ e ”).

Since the feedback interconnection is asymptotically stable when the measurement error vector $f(t)$ is equal to zero, the residual vector $r(t)$ consequently will converge to zero, and this completes the proof. \triangle

The importance of this theorem is twofold: First, it presents a generalized condition which guarantees observer stability, and hence measurement error detectability, for the new dynamic observer framework. Second, it paves the way for an analytical solution for the design problem, and a design procedure which could easily be implemented using available software packages as will be discussed next.

3.1 An analytical design procedure based on H_∞ regularization

As mentioned earlier, the stability condition in (29) can be represented by the H_∞ norm of the setup in Fig. 1 where G has the state space representation in (14) with appropriate matrices. However, this H_∞ problem does not satisfy all the regularity assumptions in the H_∞ framework (notice that $D_{12}^T D_{12}$ and $D_{21} D_{21}^T$ are both singular). Although the LMI approach in (Iwasaki & Skelton, 1994), or the techniques in (Scherer, 1992); (Stoorvogel, 1996) can be used to solve this singular problem, we here focus on the Riccati approach in (Doyle et al., 1989) by showing that the problem is actually equivalent to the so-called “Simplified H_∞ problem” defined in (Doyle et al., 1989); (Zhou & Doyle, 1998). This helps to directly relate the stability condition to two Riccati equations, instead of the one defined for the static observer framework in (Raghavan & Hedrick, 1994), and lays the ground to a systematic design procedure which is less restrictive than the existing design approaches. This also has the advantage of classifying the set of all possible observer gains by using the standard parameterization of H_∞ controllers in (Doyle et al., 1989); (Zhou & Doyle, 1998). Towards this objective, the following standard regularization procedure is adopted: By replacing the measurement error vector with a “weighted” disturbance term $\epsilon d(t)$ ($\epsilon > 0$) in the output equation (4), and using the same observer defined by (21)-(24), it can be seen that the standard H_∞ problem has now the form:

$$\dot{z} = [A] z + \begin{bmatrix} [I_n & 0_{np}] & -I_n \end{bmatrix} \begin{bmatrix} \tau \\ d \\ v \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} \zeta \\ \beta v \\ \varphi \end{bmatrix} = \begin{bmatrix} I_n \\ 0_n \\ C \end{bmatrix} z + \begin{bmatrix} 0_n & 0_{np} \\ 0_n & 0_{np} \\ 0_{pn} & \epsilon I_p \end{bmatrix} \begin{bmatrix} 0_n \\ \beta I_n \\ 0_{pn} \end{bmatrix} \begin{bmatrix} \tau \\ d \\ v \end{bmatrix} \quad (32)$$

which can still be represented by the setup in Fig. 1, by redefining the matrices in (14) and by replacing τ by $\bar{\tau}$ defined as: $\bar{\tau} \triangleq [\tau \quad d(t)]^T$ and ζ by $\bar{\zeta}$ defined as: $\bar{\zeta} = [\zeta \quad \beta v]^T$, ($\beta > 0$). It follows that the standard form in (31)-(32) satisfies the conditions of the so-called “Simplified H_∞ problem” (Doyle et al., 1989); (Zhou & Doyle, 1998) if and only if (A, C) is detectable, which does not impose any new design restrictions on the observer design. The equivalence between the original problem and this “Simplified H_∞ problem” can also be shown as follows: Assume T_1 as the setup in Fig. 1 associated with the original τ and ζ . And assume T_2 as the one associated with $\bar{\tau}$ and $\bar{\zeta}$, i.e the one described by equations (31)-(32). Assume both setups use the observer gain K in (25). And let $\hat{T}_1(s)$ and $\hat{T}_2(s)$ be their corresponding transfer matrices. The following lemma demonstrates certain equivalence relationships among these two setups (the proof of this Lemma is omitted and can be found in (Pertew et al., 2005)).

Lemma 1. *Given the same observer gain controller K for the setups T_1 and T_2 defined above, then $\|\hat{T}_1(s)\|_\infty < \gamma$ if and only if $\exists \epsilon > 0, \beta > 0$ such that $\|\hat{T}_2(s)\|_\infty < \gamma$.*

This now lays the ground to the main result of this section, in the form of a theorem showing that the observer gain K needed to stabilize the observer error dynamics and achieve measurement error detection according to Theorem 2 must solve a “Simplified H_∞ control problem” according to the definition used in (Zhou & Doyle, 1998). To this end, we define the “Nonlinear Lipschitz observer design problem” as follows:

Definition 5. (Nonlinear Lipschitz observer design problem) *Given $\epsilon > 0$ and $\beta > 0$, find S , the set of admissible observer gains K satisfying $\|\hat{T}_{\bar{\zeta}\bar{\tau}}\|_\infty < \frac{1}{\alpha}$ for the setup in Fig. 1 with G having the state space representation in (14) along with the matrices in (31)-(32).*

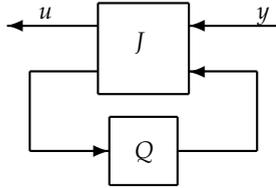
Defining the following two Hamiltonian matrices associated with this problem:

$$N_\infty = \begin{bmatrix} A & \alpha^2 I_n - \frac{1}{\beta^2} I_n \\ -I_n & -A^T \end{bmatrix}, \quad J_\infty = \begin{bmatrix} A^T & \alpha^2 I_n - \frac{1}{\epsilon^2} C^T C \\ -I_n & -A \end{bmatrix} \quad (33)$$

the main result is then summarized as follows:

Theorem 3. *There exists a dynamic observer gain K for the observer (8)-(11) (or a static gain L for the classical observer in (6)-(7)) that achieves measurement error detection according to Theorem 2 if and only if $\exists \epsilon, \beta > 0$ such that:*

- 1) $N_\infty \in \text{dom}(\text{Ric})$ and $X_\infty = \text{Ric}(N_\infty) > 0$.
- 2) $J_\infty \in \text{dom}(\text{Ric})$ and $Y_\infty = \text{Ric}(J_\infty) > 0$.
- 3) $\rho(X_\infty Y_\infty) < \frac{1}{\alpha^2}$ (where $\rho(\cdot)$ is the spectral radius).



$$\hat{J}(s) = \left[\begin{array}{c|cc} \hat{A}_\infty & \frac{1}{\epsilon^2}(I_n - \alpha^2 Y_\infty X_\infty)^{-1} Y_\infty C^T & -\frac{1}{\beta}(I_n - \alpha^2 Y_\infty X_\infty)^{-1} \\ \frac{1}{\beta^2} X_\infty & 0_{np} & \frac{1}{\beta} I_n \\ -\frac{1}{\epsilon} C & \frac{1}{\epsilon} I_p & 0_{pn} \end{array} \right]$$

Fig. 3. Parametrization of all observer gains.

Proof: A direct result of Theorem 2 and Lemma 1. △

Moreover, by using the result in (Doyle et al., 1989), the set of all observer gains K can be represented by the set of all transfer matrices from y to u in Fig. 3:

where $\hat{A}_\infty = A + (\alpha^2 - \frac{1}{\beta^2})X_\infty - \frac{1}{\epsilon^2}(I_n - \alpha^2 Y_\infty X_\infty)^{-1} Y_\infty C^T C$, and Q is such that $\|Q\|_\infty < \frac{1}{\alpha}$.

Based on the previous results, the following iterative “binary search” procedure is proposed to evaluate the observer gain:

Design Procedure:

Step 1 Set $\epsilon, \beta > 0$.

Step 2 Test solvability of the problem in Definition 5 . If the test fails then go to Step 3 ; otherwise solve the problem (using available software packages or using the analytical result of Theorem 3) and any $K \in \mathcal{S}$ (the set of admissible observer gains) is a candidate observer gain that stabilizes the error dynamics and achieves measurement error detection.

Step 3 Set $\epsilon \leftarrow \frac{\epsilon}{2}, \beta \leftarrow \frac{\beta}{2}$. If ϵ or $\beta < r$, a threshold value, then *stop* ; otherwise go to Step 2.

Remarks:

- This design procedure is less restrictive than the designs introduced in (Raghavan & Hedrick, 1994); (Rajamani, 1998); (Rajamani & Cho, 1998); (Aboky et al., 2002), since it is directly related to the stability condition through the result of Theorem 3.
- If the H_∞ problem can not be solved due to its infeasibility or due to the software limitations, one can decrease the Lipschitz constant α and this decreases the region of convergence if α is obtained through linearization but is still a possible way to solve the problem. The word *stop* in step 3 can then be replaced by: *decrease α and go to Step 1*. The algorithm is then guaranteed to work as $\alpha \rightarrow 0$. The choice of the threshold in step 3 is also important to avoid numerical instability of the used software.
- Design of the H_∞ observer can also be done by including appropriate weightings to emphasize the performance requirements of the observer over specific frequency ranges.
- If some states are not affected by nonlinearities (i.e, if some entries of the Lipschitz function Φ are zeros), the corresponding 1's of the Identity matrix in the matrix B_1 of the setup (31)-(32) can be replaced by zeros. As long as (A, B_1) is controllable, and the

regularity assumptions are satisfied, the observer design is still equivalent to a “Simplified H_∞ problem”.

4. The Measurement Error Identification Problem

After solving the measurement error detection problem in Section 3, it would be interesting if those results are extended to the identification problem, where the objective of estimating the error magnitudes and locating the error sources are also considered. The advantage of the dynamic formulation is clear at this point: the measurement error detection condition (29) (in Theorem 2) is satisfied by a family of observers, which helps to include the “identification” objective according to Definition 3 as an additional objective.

Since the residual “ r ” is given by equation (27), it is clear that the observer estimation error “ e ” constitutes a part of the residual response, and that by minimizing “ e ” the residual converges to “ f ” which guarantees measurement error identification in this case. This could be seen by noting that the estimation error “ e ” can be represented by the feedback interconnection in Fig. 4 where “ f ” is the measurement error vector that affects the system (compare this representation to the loop dynamics in Fig. 2 when the measurement error vector “ f ” is equal to zero).

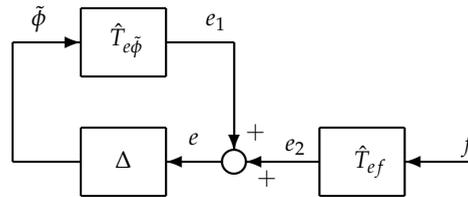


Fig. 4. Observer dynamics with measurement errors.

Therefore, minimizing “ e ” is equivalent to minimizing the effect of “ f ” on the feedback interconnection of Fig. 4. This minimization problem can be considered in \mathcal{L}_2 sense by assuming the measurement error to have finite energy, and applying the small gain theorem to Fig. 4. This assumption is with no loss of generality, since it is guaranteed over any finite time operation of the observer. In this section, the solution of this minimization problem is considered when “ f ” is in a narrow frequency band around a nominal frequency ω_0 . From the special cases of interest is the case of sensor bias and the case of measurement errors of known harmonics. We first show that the problem is not tractable for the classical structure in (6)-(7), and then present a solution using the dynamic observer. Towards that goal, we will first assume that the Fourier transform of the measurement error $F(j\omega)$ have a frequency pattern restricted to the narrow band $\omega_0 \pm \Delta\omega$ as described by equation (34).

$$|F(j\omega)| \leq \begin{cases} A; & |\omega - \omega_0| < \Delta\omega \\ \delta; & \text{otherwise} \end{cases} \quad (34)$$

where δ is a small neglected number for the frequency magnitudes outside the region of interest, and where A is a positive upper bound on these magnitudes inside the considered domain. We will then define an observer gain K as optimal if $\|e\|_{\mathcal{L}_2}$ can be made

arbitrarily small for all possible measurement errors satisfying (34). But by applying the small gain theorem to Fig. 4 when measurement error detection is satisfied (i.e, when K satisfies $\|\hat{T}_{e\tilde{\phi}}\|_{\infty} = \mu < \frac{1}{\alpha}$) we have: $\|e\|_{\mathcal{L}_2} \leq \frac{1}{1-\mu\alpha} \|e_2\|_{\mathcal{L}_2}$. And since (as $\Delta\omega \rightarrow 0$), $\hat{T}_{ef}(j\omega) \rightarrow \hat{T}_{ef}(j\omega_0)$ then we have $\|e_2\|_{\mathcal{L}_2} \leq \sigma_{max}(\hat{T}_{ef}(j\omega_0)) \|f\|_{\mathcal{L}_2}$, and therefore, it is easy to see that an optimal gain K is one that satisfies $\hat{T}_{ef}(j\omega_0) = 0$. By assuming that the measurement error detection objective is satisfied (as stated in Theorem 2, it follows that *measurement error identification* according to Definition 3 is satisfied if the following two conditions are satisfied: (i) $\|\hat{T}_{e\tilde{\phi}}\|_{\infty} < \frac{1}{\alpha}$, (ii) $\hat{T}_{ef}(j\omega_0) = 0$, where the first one is a sufficient condition in order to achieve *measurement error detection* according to Definition 1.

Moreover, to include the effect of measurement errors on the standard setup which was used in conjunction with the H_{∞} problem in Section 3, the vectors of the setup of Fig. 1 are redefined as: $\tau \triangleq \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \tilde{\phi} \\ f \end{bmatrix}$, $v = \eta$, $\zeta = e$, $\varphi = r$. The residual can be then represented by:

$$\dot{z} = [A]z + \begin{bmatrix} [I_n & 0_{np}] & -I_n \end{bmatrix} \begin{bmatrix} \tau \\ v \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} \zeta \\ \varphi \end{bmatrix} = \begin{bmatrix} I_n \\ C \end{bmatrix} z + \begin{bmatrix} \begin{bmatrix} 0_n & 0_{np} \end{bmatrix} & 0_n \\ \begin{bmatrix} 0_{pn} & I_p \end{bmatrix} & 0_{pn} \end{bmatrix} \begin{bmatrix} \tau \\ v \end{bmatrix} \quad (36)$$

which can also be represented by the standard form in Fig. 1 where G has the representation in (14) with the matrices in (35)-(36) and where K is the dynamic observer gain.

Based on the previous discussion, we define an optimal residual generator as:

Definition 6. (*Optimal residual generator for narrow frequency band*) An observer of the form (8)-(12) is said to be an optimal residual generator for the measurement error identification problem (with measurement errors in a narrow frequency band around ω_0) if the observer gain K satisfies $\|\hat{T}_{e\tilde{\phi}}\|_{\infty} < \frac{1}{\alpha}$ and $\hat{T}_{ef}(j\omega_0) = 0$, for the standard setup in Fig. 1 where the plant G has the state space representation in (14) with the matrices defined in (35)-(36).

The main result of this section is now presented in the form of a theorem showing that the classical observer structure in (6)-(7) can never be an optimal residual generator. This shows the importance of having a dynamic observer gain in this case.

Theorem 4. An observer of the form (6)-(7) with a static observer gain L can never be an optimal residual generator according to Definition 6.

Proof: First, using (35)-(36) and the dynamic observer gain K it can be shown that \hat{T}_{ef} is given from:

$$T_{ef} = T_{\zeta\tau_2} = \left[\begin{array}{cc|c} A - D_L C & -C_L & -D_L \\ B_L C & A_L & B_L \\ \hline I_n & 0_{nk} & 0_{np} \end{array} \right] \quad (37)$$

The proof follows by noting that (when the observer gain K is replaced by the static gain L) the transfer matrix from f to e in (37) is given by $T_{ef} = \left[\begin{array}{c|c} A - LC & -L \\ \hline I_n & 0_{np} \end{array} \right]$. Since the gain L is chosen to stabilize $(A - LC)$, then $(\forall \omega_0) j\omega_0$ is not an eigenvalue of $(A - LC)$. Therefore, by using (13), we have: $\text{rank}(\hat{T}_{ef}(j\omega_0)) = \text{rank} \left[\begin{array}{cc|c} A - LC - j\omega_0 I_n & -L \\ \hline I_n & 0_{np} \end{array} \right] - n$. But

$\text{rank} \begin{bmatrix} A - LC - j\omega_0 I_n & -L \\ I_n & 0_{np} \end{bmatrix} = \text{rank} \begin{bmatrix} 0_n & L \\ I_n & 0_{np} \end{bmatrix} = n + \text{rank}(L)$ (using the invariant zero property (13)).

Therefore, $\text{rank}(\hat{T}_{ef}(j\omega_0)) \neq 0$ unless $L = 0$. This implies that no gain L can satisfy $\hat{T}_{ef}(j\omega_0) = 0$, and therefore the static observer structure can never be an optimal residual generator according to Definition 6. \triangle

In the following section (section 4.1), a numerical approach based on LMIs is provided, by modelling the problem as a convex optimization problem using the dynamic observer structure in (8)-(12).

4.1 A LMI Design Procedure

The second objective, i.e $\hat{T}_{ef}(j\omega_0) = 0$, can also be modelled as a weighted H_∞ problem solvable using the dynamic observer formulation. To this end, we first note that for an observer gain K that satisfies the measurement error detection condition (as stated in Theorem 2, the following two statements are equivalent: (i) $\hat{T}_{ef}(j\omega_0) = 0$, (ii) $W(s)\hat{T}_{ef}(s) \in RH_\infty$, where $W(s) = \text{diag}_p(\frac{1}{s})$ if $\omega_0 = 0$ and $W(s) = \text{diag}_p(\frac{1}{s^2 + \omega_0^2})$ if $\omega_0 \neq 0$. The equivalence of these two statements can be seen by first noting that the condition in Theorem 2 implies that $\|\hat{T}_{e\tilde{\phi}}\|_\infty < \frac{1}{\alpha}$ and hence that $\hat{T}_{e\tilde{\phi}} \in RH_\infty$. It then follows that $\hat{T}_{ef}(s) \in RH_\infty$ since T_{ef} in (37) and $T_{e\tilde{\phi}}$ both have the same state transition matrix. Finally, since $\hat{T}_{ef}(j\omega_0) = 0$ corresponds to $j\omega_0$ being a system zero of $\hat{T}_{ef}(s)$ (which is equivalent to cancelling the poles of $W(s)$ on the imaginary axis), it follows that $\hat{T}_{ef}(j\omega_0) = 0$ is equivalent to having $W(s)\hat{T}_{ef}(s) \in RH_\infty$.

According to the previous discussion, it follows that the objective $\hat{T}_{ef}(j\omega_0) = 0$ can be restated as follows: $\exists \epsilon > 0$ such that $\{\epsilon \| W(s)\hat{T}_{ef}(s) \|_\infty < \frac{1}{\alpha}\}$, where the scalar “ ϵ ” is used for compatibility with the first objective (i.e, $\|\hat{T}_{e\tilde{\phi}}\|_\infty < \frac{1}{\alpha}$). The two objectives can then be combined in the unified framework in Fig. 5, where the plant G has the state space representation in (14) with the matrices defined in (35)-(36).

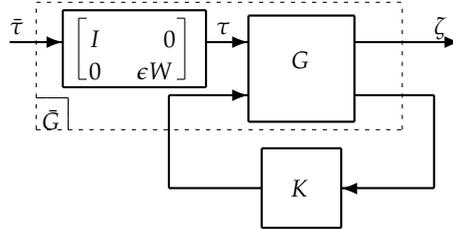


Fig. 5. Weighted standard setup.

It can be seen that the augmented plant \bar{G} in Fig. 5 is given by:

$$\bar{G} = \left[\begin{array}{c|cc} \bar{A} & \bar{B}_1 & \bar{B}_2 \\ \hline \bar{C}_1 & \bar{D}_{11} & \bar{D}_{12} \\ \bar{C}_2 & \bar{D}_{21} & \bar{D}_{22} \end{array} \right] = \left[\begin{array}{c|cc|c} \begin{bmatrix} A_\theta & 0_{\ell n} \\ 0_{n\ell} & A \end{bmatrix} & \begin{bmatrix} 0_{\ell n} & B_\theta \\ I_n & 0_{np} \end{bmatrix} & \begin{bmatrix} 0_{\ell n} \\ -I_n \end{bmatrix} \\ \hline \begin{bmatrix} 0_{n\ell} & I_n \\ \epsilon C_\theta & C \end{bmatrix} & \begin{bmatrix} 0_n & 0_{np} \\ 0_{pn} & 0_p \end{bmatrix} & \begin{bmatrix} 0_n \\ 0_{pn} \end{bmatrix} \end{array} \right] \quad (38)$$

where:

$$\left\{ \begin{array}{l} \ell = p, A_\theta = 0_p, B_\theta = I_p, C_\theta = I_p; \text{ if } \omega_o = 0 \\ \ell = 2p, A_\theta = \text{diag}_p \begin{bmatrix} 0 & 1 \\ -\omega_o^2 & 0 \end{bmatrix}, B_\theta = \text{diag}_p \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_\theta = \text{diag}_p [1 \ 0]; \text{ if } \omega_o \neq 0 \end{array} \right. \quad (39)$$

The following theorem gives necessary and sufficient conditions for solving the measurement error identification problem in Definition 6:

Theorem 5. *Given the system (3)-(4), there exists an optimal residual according to Definition 6, $\forall \Phi$ satisfying (5) with a Lipschitz constant α , if and only if $\exists \epsilon > 0$ and a dynamic observer gain K satisfying $\|\hat{T}_{\zeta\bar{\tau}}\|_\infty < \frac{1}{\alpha}$.*

Proof: A direct result of Definition 6 and the discussion in the beginning of Section 4.1 showing the equivalence between the two objectives. \triangle

However, standard H_∞ tools can not be directly applied for the H_∞ problem defined in Theorem 5, unlike the measurement error detection problem discussed in Section 3. For instance the Riccati approach in (Zhou & Doyle, 1998) can not be implemented since the augmented plant \bar{G} in (38) does not satisfy the needed regularity assumptions. Also, the LMIs in equation (15)-(17) are not feasible due to the poles that \bar{G} has on the imaginary axis, making the use of the LMI approach in (Gahinet & Apkarian, 1994) impossible. However, by replacing the weightings $W(s)$ by the modified weightings $\bar{W}(s)$ where $\bar{W}(s) = \text{diag}_p(\frac{1}{s+\lambda})$ if $\omega_o = 0$ and $\bar{W}(s) = \text{diag}_p(\frac{1}{s^2+2\lambda\omega_o s+\omega_o^2})$ if $\omega_o \neq 0$, with $\lambda \in \mathbb{R}^+$, the augmented plant \bar{G} in Fig. 5 is still given by equation (38), but with A_θ as:

$$A_\theta = \left\{ \begin{array}{l} \text{diag}_p(-\lambda) \ ; \ \omega_o = 0 \\ \text{diag}_p \begin{bmatrix} 0 & 1 \\ -\omega_o^2 & -2\lambda\omega_o \end{bmatrix}; \ \omega_o \neq 0 \end{array} \right. \quad (40)$$

which has no poles on the imaginary axis. Using the modified plant and the result in Theorem 1, the following convex optimization problem is proposed to solve the problem in Theorem 5:

$$\begin{array}{l} \min_{R,S} \lambda \\ \text{subject to " the 3 LMIs in (15)-(17) with } \gamma = \frac{1}{\alpha} \text{ "} \end{array}$$

with the matrices in (15)-(17) replaced by the corresponding ones in (38)-(40).

The set of all admissible observer gains K for a given λ can then be parameterized using R, S by using the result in (Gahinet & Apkarian, 1994). It can also be seen, that these LMIs are feasible for all $\lambda > 0$, and that minimizing λ in this case is equivalent to minimizing $\sigma_{\max}(\hat{T}_{ef}(j\omega_o))$. This guarantees that the proposed optimization problem converges to the existing solution as $\lambda \rightarrow 0$. It also guarantees that standard software packages can be used to solve this optimization problem.

The optimal residual generator guarantees measurement errors estimation and at the same time state estimation. An advantage of having *state* estimation in the presence of measurement errors is the possibility to use the observer in fault tolerant output feedback control (i.e, if a reconfiguration control action is involved). Also, from the special cases of interest is the case of sensor bias, where the previous approach can be used to get an *exact* estimation of all sensor biases at the same time. An important advantage over the adaptive approaches used to

diagnose sensor biases in nonlinear systems, such as (Vemuri, 2001); (Wang et al., 1997), is the ability to diagnose piecewise constant bias with the same observer. Moreover, the proposed approach is not limited to sensor biases and can be used to diagnose measurement errors of any harmonics.

5. Measurement Error Identification for Low and High Frequencies

We now consider measurement errors of low frequencies determined by a cutoff frequency ω_l . The SISO weighting $\hat{w}_l(s) = \frac{as+b}{s}$, (Zhou & Doyle, 1998), emphasizes this range with “ b ” selected as ω_l and “ a ” as an arbitrary small number for the magnitude of $\hat{w}_l(j\omega)$ as $\omega \rightarrow \infty$. With a diagonal transfer matrix $\hat{W}(s)$ that consists of these SISO weightings (and similar to the approach adopted in section 4.1), the detection and identification objectives can be combined in the unified framework represented by the weighted setup of Fig. 5. In this case, the augmented plant \bar{G} is given by:

$$\bar{G} = \left[\begin{array}{c|cc} \bar{A} & \bar{B}_1 & \bar{B}_2 \\ \hline \bar{C}_1 & \bar{D}_{11} & \bar{D}_{12} \\ \bar{C}_2 & \bar{D}_{21} & \bar{D}_{22} \end{array} \right] = \left[\begin{array}{c|cc|c} \left[\begin{array}{cc} A_\theta & 0_{pn} \\ 0_{np} & A \end{array} \right] & \left[\begin{array}{cc} 0_{pn} & B_\theta \\ I_n & 0_{np} \end{array} \right] & \left[\begin{array}{c} 0_{pn} \\ -I_n \end{array} \right] \\ \hline \left[\begin{array}{cc} 0_{np} & I_n \\ \epsilon C_\theta & C \end{array} \right] & \left[\begin{array}{cc} 0_n & 0_{np} \\ 0_{pn} & \epsilon D_\theta \end{array} \right] & \left[\begin{array}{c} 0_n \\ 0_{pn} \end{array} \right] \end{array} \right] \quad (41)$$

where $A_\theta=0_p$, $B_\theta=I_p$, $C_\theta=diag_p(b)$ and $D_\theta=diag_p(a)$. This form also violates the assumptions of Theorem 1 (note that (\bar{A}, \bar{B}_2) is not stabilizable). Similar to Section 4, we introduce the modified weighting $\hat{w}_{lmod}(s)=\frac{as+b}{s+\lambda}$; with arbitrary small positive “ λ ”. The augmented plant \bar{G} is then the same as (41) except for A_θ which is now given by the stable matrix $diag_p(-\lambda)$ and C_θ given by $diag_p(b - a\lambda)$. Similar to the narrow frequency band case, the assumptions of Theorem 1 are now satisfied and the LMI approach in (Gahinet & Apkarian, 1994) can be used to solve the H_∞ problem. To this end, we define the H_∞ problem associated with the low frequency range as follows:

Definition 7. (Low frequency H_∞) Given $\lambda > 0$, $\epsilon > 0$, find \mathcal{S} , the set of admissible controllers K satisfying $\| \hat{T}_{z\bar{r}} \|_\infty < \gamma$ for the setup in Fig. 5 where \bar{G} has the state space representation (41) with $A_\theta = diag_p(-\lambda)$, $B_\theta = I_p$, $C_\theta = diag_p(b - a\lambda)$ and $D_\theta = diag_p(a)$.

Based on all the above, we now present the main result of this section in the form of the following definition for an optimal residual generator in \mathcal{L}_2 sense:

Definition 8. (Optimal residual for low frequencies) An observer of the form (8)-(12) is an optimal residual generator for the measurement error identification problem (with low frequency measurement errors below the cutoff frequency ω_l) if the dynamic gain $K \in \mathcal{S}^*$ (the set of controllers solving the H_∞ problem in Definition 7 for $\gamma = 1/\alpha$ with the minimum possible λ).

Similar to the low frequency range, a proper weighting $\hat{w}_{hmod}(s) = \frac{s+(a \times b)}{\lambda s + b}$, (Zhou & Doyle, 1998), with an arbitrary small $\lambda > 0$, could be selected to emphasize the high frequency range $[w_h, \infty)$ with “ b ” selected as w_h and “ a ” as an arbitrary small number for $|\hat{w}_h(j\omega)|$ as $\omega \rightarrow 0$. With the help of $\hat{w}_{hmod}(s)$, a suitable weighting W that emphasizes the high frequency range can be designed. The augmented \bar{G} is also given from (41) (same as the low frequency case), but with A_θ , B_θ , C_θ and D_θ given as $diag_p(-\frac{b}{\lambda})$, I_p , $diag_p(\frac{a \times b}{\lambda} - \frac{b}{\lambda^2})$ and $diag_p(\frac{1}{\lambda})$ respectively. It is straightforward that \bar{G} satisfies all of the assumptions of Theorem 1 and therefore, similar to the low frequency range, an H_∞ problem related to the high frequency range can be defined. An optimal residual generator can be defined in the same way as Definition 8 for the generalized low frequency case.

6. Experimental Results

The experimental results presented in this section (Pertew, 2006) are intended to illustrate the applicability of the theoretical results presented in this chapter for robotic systems.

6.1 The ROTPEN: Models and Assumptions

The *Quanser* rotary inverted pendulum (ROTPEN) is shown schematically in Fig. 6, Lynch (2004). The angle that the perfectly rigid link of length l_1 and inertia J_1 makes with the x -axis of an inertial frame is denoted θ_1 (degrees). Also, the angle of the pendulum (of length l_2 and mass m_2) from the z -axis of the inertial frame is denoted θ_2 (degrees).

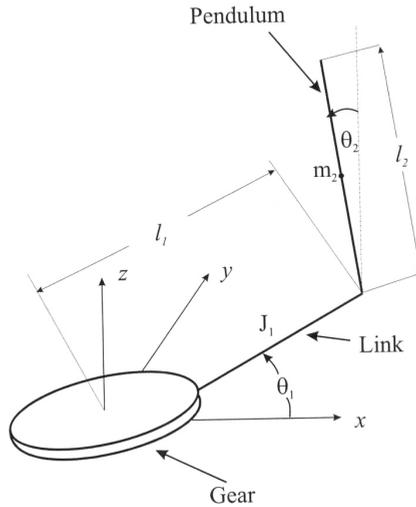


Fig. 6. The Rotary Inverted Pendulum (ROTPEN).

The system has one input which is the scalar servomotor voltage input (Volt). Therefore, the system is a special case of the robot manipulator model discussed in Section 1: a planar robot manipulator with two links ($n = 2$), with only one torque applied at the first joint, while the second joint is subject to the gravitational force. In fact, the ROTPEN has a state space model of the form $\dot{x} = f(x) + g(x)u$, where $x = [\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2]^T$ is the state vector, and u is the scalar servomotor voltage input (Volt). More details about this model and its parameters can be found in Appendix 9.1.

The system has an infinite number of equilibrium points, representing the following two equilibrium points:

- 1) Pendant position: $x_1 = 0$ (rad), $x_2 = \pi$ (rad), $x_3 = x_4 = 0$ (rad/sec).
- 2) Inverted position: $x_1 = x_2 = 0$ (rad), $x_3 = x_4 = 0$ (rad/sec).

By separating the nonlinear terms, the model can be put in the form $\dot{x} = Ax + \Phi(x, u)$, where:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -25.14 & -17.22 & 0.2210 \\ 0 & 68.13 & 16.57 & -0.599 \end{bmatrix}, \quad \Phi(x, u) = \begin{bmatrix} 0 \\ 0 \\ \phi_1(x, u) \\ \phi_2(x, u) \end{bmatrix}. \quad \text{The nonlinear terms in } \Phi \text{ are}$$

mainly trigonometric terms, and using the symbolic MATLAB toolbox, an upper bound on $\|\Phi(x, u)\|$ is found as 44.45, and hence the Lipschitz constant for the ROTPEN is $\alpha = 44.45$. This follows from the fact that if $\Phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$ is continuously differentiable on a domain D and the derivative of Φ with respect to the first argument satisfies $\|\frac{\partial \Phi}{\partial x}\| \leq \alpha$ on D , then Φ is Lipschitz continuous on D with constant α , i.e.:

$$\|\Phi(x, u) - \Phi(y, u)\| \leq \alpha \|x - y\|, \forall x, y \in D \quad (42)$$

There are two encoders to measure the angle of the servomotor output shaft (θ_1) and the angle of the pendulum (θ_2). An encoder is also available to measure the motor velocity $\dot{\theta}_1$, but no one is available to measure the pendulum velocity $\dot{\theta}_2$. In the experiments, linear as well as nonlinear control schemes are used to stabilize the pendulum at the inverted position ($\theta_2 = 0$), while tracking a step input of 30 degrees for the motor angle.

6.2 Case Study 1 - Lipschitz Observer Design

In this experiment, we focus on the nonlinear state estimation problem when no measurement errors are affecting the system. We consider situations in which the operating range of the pendulum is either close or far from the equilibrium point, comparing the Luenberger observer with the Lipschitz observer in these cases. For the purpose of applying the Lipschitz observer design, the nonlinear model discussed in section 6.1 is used. We also compare the dynamic Lipschitz observer of section 3 with the static design method in Reference (Raghavan & Hedrick, 1994). In this case study the full-order linear and Lipschitz models are used for observer design, where the output is assumed as $y = [x_1 \ x_2]^T$ (all the observer parameters that are used in this experiment can be found in Appendix 9.2).

First, a linear state feedback controller is used to stabilize the system in a small operating range around the inverted position, and three observers are compared:

- 1) Observer 1: A linear Luenberger observer where the observer gain is obtained by placing the poles of $(A - LC)$ at $\{-24, -3.8, -4.8, -12.8\}$ (see $L_{3-small}$ in Appendix 9.2).
- 2) Observer 2: A high gain Luenberger observer, which has the same form of Observer 1 but with the poles placed at $\{-200, -70, -20 + 15i, -20 - 15i\}$ (see $L_{3-large}$ in Appendix 9.2).
- 3) Observer 3: A Lipschitz observer of the form (8)-(11), based on the full-order Lipschitz model of the ROTPEN. The dynamic gain is computed using the design procedure in section 3.1, for $\alpha = 44.45$ (see K_3 in Appendix 9.2).

The three observers run successfully with stable estimation errors. Table 1 shows the maximum estimation errors in this case. It can be seen that both the Luenberger observer (large poles) and the Lipschitz observer achieve comparable performance, which is much better than the Luenberger observer with small poles. The three observers are also tested in observer-based control, and their tracking performance is compared in Table 2. We conclude that, due to the small operating range considered in this case study, a high-gain Luenberger observer achieves a good performance in terms of the state estimation errors and the tracking errors.

We then consider a large operating range by using a nonlinear control scheme that stabilizes the pendulum angle at the pendant position (see Appendix 9.2 for more details about the controller used in this case study). Using this controller, a large operating range is obtained as seen in Fig. 7. The same observers (Observers 2 and 3) are used in parallel with this control scheme, and the resulting estimation errors are compared in Fig. 8. The two observers are also

	Small-gain Luenberger	High-gain Luenberger	Lipschitz
$\max e_1 $	3.6485	0.4323	0.1716
$\max e_2 $	1.5681	0.0925	0.1865

Table 1. Case study 1 - Estimation errors “ e_1 ” and “ e_2 ” in degrees

	pure state feedback	High-gain Luenberger	Lipschitz
Percentage of overshoot	20.3613%	12.7440%	48.4863%
steady state error	2.5635	3.4424	3.7939

Table 2. Case study 1 - Tracking performance in degrees

compared in observer-based control, and the Luenberger observer fails in this case, causing total system instability. The Lipschitz observer, on the other hand, runs successfully and its performance (compared to the pure state feedback control) is shown in Fig. 9. This case study illustrates the importance of the Lipschitz observer in large operating regions, where the linear observer normally fails.

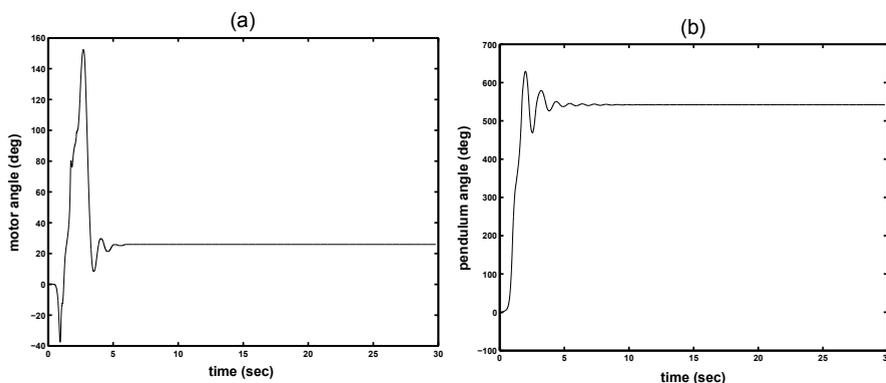


Fig. 7. Case Study 1 - (a) Motor Response, (b) Pendulum Response.

Finally, we conduct a comparison between static and dynamic Lipschitz observers, namely the observer (6)-(7) and the one in (8)-(11). The comparison is between the new design proposed in Section 3 and the one in Reference (Raghavan & Hedrick, 1994). First, the design algorithm in (Raghavan & Hedrick, 1994) is tested for different values of α and ϵ . It fails for all values of $\alpha > 1$, and the maximum attainable value is $\alpha = 1$ (see L_5 in Appendix 9.2), while the Lipschitz constant of the ROTPEN model is 44.45 as mentioned earlier. This observer is then compared to the dynamic Lipschitz observer having the dynamic gain K_3 , and the estimation errors are shown in Fig. 10. It is also important to note that the static Lipschitz observer fails in stabilizing the system, when used in observer-based control, for both the small and large operating range experiments. This shows the importance of the dynamic Lipschitz observer design in this case.

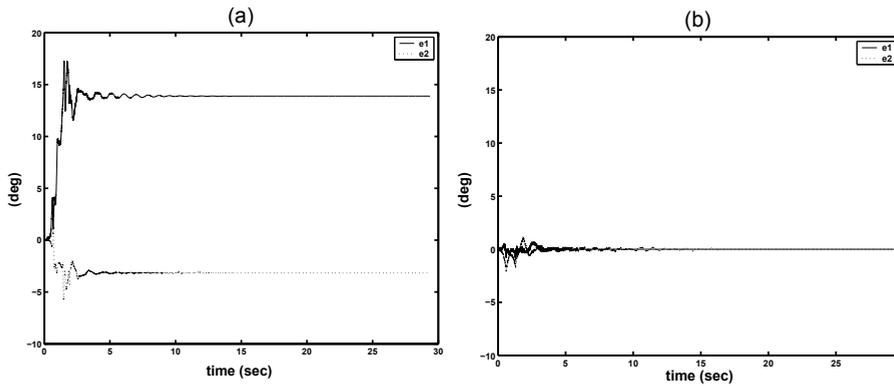


Fig. 8. Case Study 1 - (a) High-gain Luenberger Errors, (b) Dynamic Lipschitz Errors.

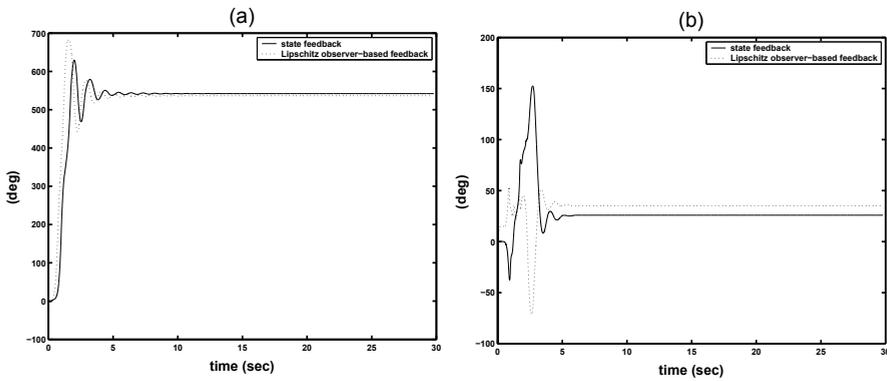


Fig. 9. Case Study 1 - (a) Pendulum Angle, (b) Motor Angle.

6.3 Case Study 2 - Lipschitz Measurement Error Diagnosis

In this experiment, the results of Sections 4 and 5 are assessed on the nonlinear Lipschitz model. A large operating range is considered by using a nonlinear, switching, LQR control scheme (with integrator) that stabilizes the pendulum at the inverted position (starting from the pendant position) while tracking a step input of 30 degrees for the motor angle as seen in Fig. 11 (the no-bias case). In the first part of this experiment, an important measurement error that affects the ROTPEN in real-time is considered. This is a sensor fault introduced by the pendulum encoder. The encoder returns the pendulum angle relative to the initial condition, assuming this initial condition to be $\theta_2 = 0$. This constitutes a source of bias, as shown in Fig. 11(b), when the pendulum initial condition is unknown or is deviated from the inverted position. The effect of this measurement error on the tracking performance is also illustrated in Fig. 11(a) for two different bias situations. The dynamic Lipschitz observer (discussed in section 4) is applied to diagnose and tolerate this fault. In addition to this bias fault, the observer is also applied for a 2 rad/sec fault introduced in real-time, as well as for the case of a low frequency fault in the range $[0, 1 \text{ rad/sec}]$.

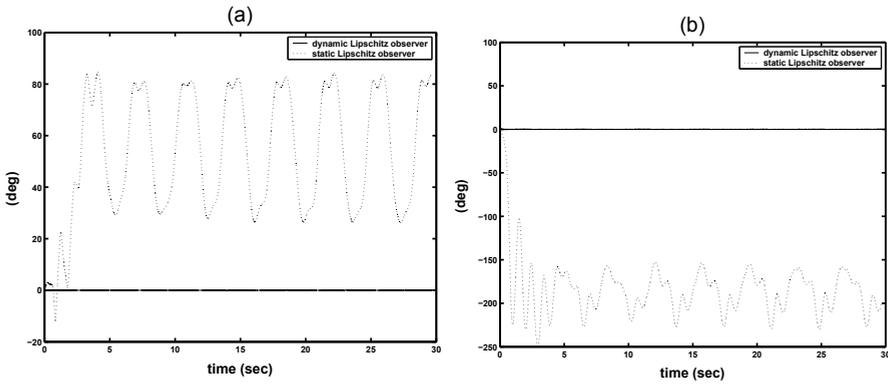


Fig. 10. Case Study 1 - (a) Estimation Error “ e_1 ”, (b) Estimation Error “ e_2 ”.

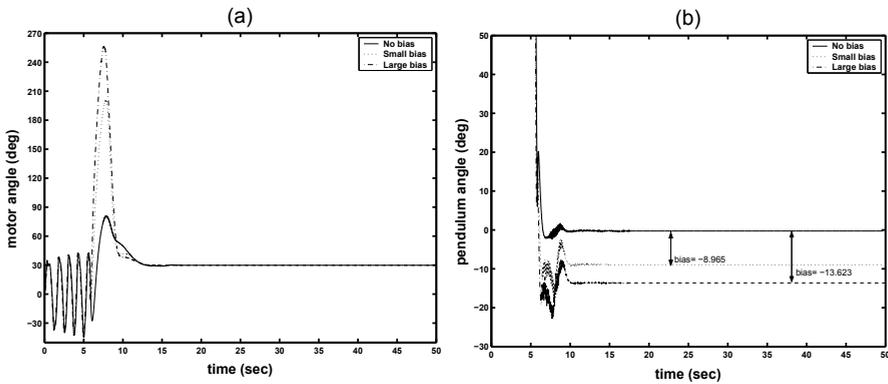


Fig. 11. Case Study 2 - (a) Tracking Performance, (b) Pendulum Angle.

First, the design procedure in section 4 is used to accurately estimate and tolerate the bias faults shown in Fig. 11(b). This is the special case where $\omega_o = 0$. Using the reduced-order Lipschitz model with $\alpha = 44.45$ (and using the LMI design procedure, the dynamic gain for the observer (8)-(12) that achieves measurement error identification is obtained as K_6 (see Appendix 9.3 for more details). Using this observer, the biases affecting the system in Fig. 11 are successfully estimated as shown in Fig. 12. Moreover, by using this observer in an observer-based control scheme, the tracking performance in the large bias case is illustrated in Fig. 13. The performance is much improved over the one with no fault tolerance as seen in Fig. 13(b). It also gives less overshoot than the no bias case, as seen in Fig. 13(a). Similar results are obtained for the small bias case.

The case of measurement error in the form of harmonics is now considered, with a sensor fault having a frequency of 2 rad/sec. The dynamic gain for the observer (8)-(12) is computed using the design approach discussed in section 5. This is the special case where $\omega_o = 2$. The gain is obtained at $\lambda = 10^{-12}$ as K_7 (see Appendix 9.3). Using this observer, Fig. 14 shows the correct estimation of a measurement error of amplitude 20 degrees and frequency 2 rad/sec.

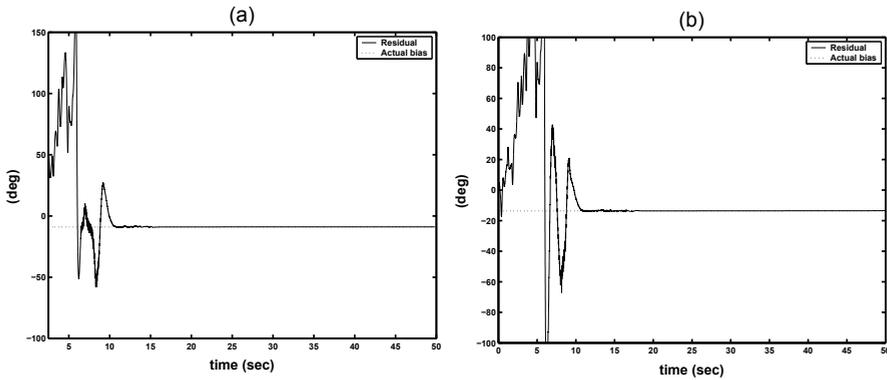


Fig. 12. Case Study 2 - (a) Estimation of the Small Bias, (b) Estimation of the Large Bias.

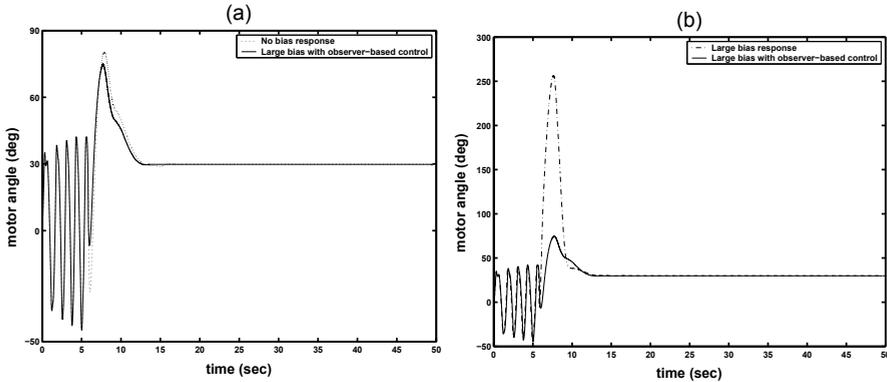


Fig. 13. Case Study 2 - (a) No-bias versus Observer-based, (b) Large Bias versus Observer-based.

We then consider the case of low frequency sensor faults (in the range $[0, 1 \text{ rad/sec}]$). Using the design introduced in section 5 (and with $a = 0.1$, $b = 1$ and $\epsilon = 0.1$), the optimal observer gain is obtained using the command *hinflmi* in MATLAB, with minimum λ as 10^{-12} (see K_8 in Appendix 9.3). Using this observer for measurement error diagnosis, a correct estimation of a low frequency sensor fault (generated using the MATLAB command *idinput*) is shown in Fig. 15.

7. Conclusion

The Lipschitz observer design approach provides an important framework for solving the measurement error diagnosis problem in robot manipulators. The classical observer structure is not directly applicable to the detection and identification problems. This is in part due to the restrictive observer structure, and also due to the idealized assumptions inherent in this structure that do not take into account uncertain model parameters and disturbances. The dynamic observer structure offers two important advantages in that regard: (i) The observer stability condition that ensures asymptotic convergence of the state estimates is satisfied by a family

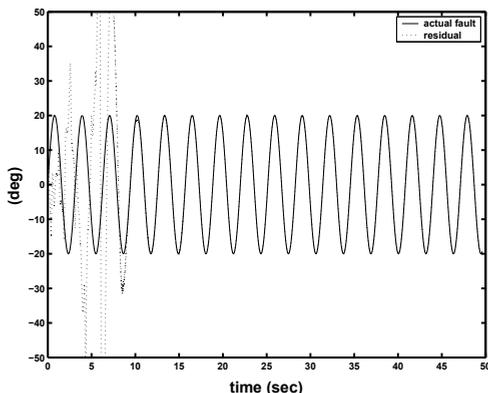


Fig. 14. Case Study 2 - Frequency Band Estimation.

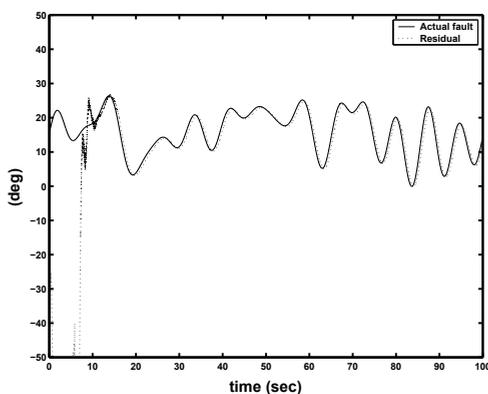


Fig. 15. Case Study 2 - Diagnosis of Low Frequency Sensor Fault.

of observers, adding extra degrees of freedom to the observer which lay the ground to the addition of the detection and identification objectives in the design, (ii) The observer design can be carried out using a systematic design procedure which is less restrictive than the existing design approaches and which is solvable using commercially available software. The design depends heavily on the nature of the objectives considered. While an analytical solution can be used for measurement error detection, the identification problem is more demanding and needs a more general design framework. This problem is shown to be equivalent to a standard convex optimization problem which is solvable using Linear Matrix Inequalities (LMIs). Using this generalized framework, different frequency patterns for the measurement errors that affect the robot manipulator could be considered, and systematic design procedures could be used to solve the problem. A practical example, namely the *Quanser* rotary inverted pendulum (ROTPEN) in the Control Systems Lab, Electrical and Computer Engineering department, University of Alberta, is used to illustrate these results. The ROTPEN model falls in the category of planar robot manipulators, and the experimental results illustrate the applicability of the proposed techniques in the robotics field by showing the following:

- i) How to model a robot manipulator as a standard Lipschitz system.
- ii) The importance of the dynamic Lipschitz observer in large operating regions where the linear observer normally fails.
- iii) The accurate velocity estimations obtained using the dynamic observer, alleviating the need to introduce velocity sensors in real-time.
- iv) How the static observer fails, compared to the dynamic observer, when applied to Robotic Systems due to the large Lipschitz constant that these systems normally have.
- v) The efficiency of the dynamic observer in diagnosing and tolerating measurement errors of different frequencies, including an important bias introduced by the error in the initial conditions of the pendulum encoder.

8. Acknowledgement

The author would like to thank the Advanced Control Systems Laboratory members at University of Alberta. Special thanks to Dr. Alan Lynch and to Dr. Thomas Grochmal for providing the ROTPEN equations and the switching swingup control scheme used in the experiments.

9. Appendix

9.1 The ROTPEN Model

The system parameters are: $l_1 = 0.215$ m, $l_2 = 0.335$ m, $m_2 = 0.1246$ Kg, $\beta = 0.135$ Nm/s, $\mu = 0.2065$ Nm/V, $b_2 = 0.0018$ Kg/s, $g = 9.81$ m/s², and $J_1 = 0.0064$ Kg.m². With the state defined as $x = [x_1 \ x_2 \ x_3 \ x_4]^T = [\theta_1(\text{rad}) \ \theta_2(\text{rad}) \ \dot{\theta}_1(\text{rad/s}) \ \dot{\theta}_2(\text{rad/s})]^T$, the state space model has the form $\dot{x} = f(x) + g(x)u$ as follows (This model was derived in Lynch (2004)):

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ h_3(x) - \frac{m_2 l_2^2 \beta x_3}{3\Delta} \\ h_4(x) + \frac{m_2 l_1 l_2 \beta c_2}{2\Delta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\mu m_2 l_2^2}{3\Delta} \\ -\frac{\mu m_2 l_1 l_2 c_2}{2\Delta} \end{bmatrix} u$$

where $s_k = \sin(x_k)$, $c_k = \cos(x_k)$ are used to simplify notation, and where:

$$h_3(x) = \frac{m_2^2 l_2^2 \left(-\frac{1}{2} g l_1 s_2 c_2 - \frac{1}{4} l_1 l_2 x_3^2 s_2 c_2^2 + b_2 l_1 x_4 c_2 / (m_2 l_2) + \frac{1}{3} l_1 l_2 x_4^2 s_2 - \frac{1}{3} l_2^2 x_3 x_4 s_2 c_2 \right)}{2\Delta}$$

$$h_4(x) = \frac{\frac{1}{2} m_2 g l_2 \left(m_2 l_1^2 + \frac{1}{4} m_2 l_2^2 s_2^2 + J_1 \right) s_2}{\Delta} - \frac{\left(m_2 l_1^2 + \frac{1}{4} m_2 l_2^2 s_2^2 + J_1 \right) b_2 x_4}{\Delta}$$

$$+ \frac{\frac{1}{4} m_2 l_2^2 [m_2 l_1^2 (x_3^2 - x_4^2) s_2 c_2 + \frac{1}{4} m_2 l_2^2 x_3^2 s_2^3 c_2 + J_1 x_3^2 s_2 c_2 + m_2 l_1 l_2 x_3 x_4 s_2 c_2^2]}{\Delta}$$

$$\Delta = m_2 l_2^2 \left(\frac{1}{3} m_2 l_1^2 + \frac{1}{12} m_2 l_2^2 s_2^2 + \frac{1}{3} J_1 - \frac{1}{4} m_2 l_1^2 c_2^2 \right).$$

9.2 Models and Parameters for Case Study 1

Luenberger observer with small gain :

$$L_{3-small} = \begin{bmatrix} 5.9207 & -7.4414 & -13.0209 & -9.9019 \\ -1.5356 & 21.6603 & -7.2493 & 108.1343 \end{bmatrix}^T.$$

High-gain Luenberger observer :

$$L_{3-large} = 10^3 \begin{bmatrix} 0.0716 & 0.0070 & 0.1432 & -0.5022 \\ 0.0203 & 0.2206 & 1.4312 & 4.4841 \end{bmatrix}^T.$$

Dynamic Lipschitz observer : (K_3 , obtained for $\alpha = 44.45$, $\epsilon = \beta = 0.00048828$)

$$A_{L3} = 10^4 \begin{bmatrix} -0.3428 & 0 & 0 & 0 \\ 0 & -0.3428 & 0 & 0 \\ -6.2073 & 0 & -0.2048 & 0 \\ 0 & -6.2073 & 0 & -0.2048 \end{bmatrix}, B_{L3} = 10^4 \begin{bmatrix} 0.138 & 0 \\ 0 & 0.138 \\ 6.2072 & 0 \\ 0 & 6.2072 \end{bmatrix},$$

$$C_{L3} = 10^3 \begin{bmatrix} 2.048 & 0 & 0.0005 & 0 \\ 0 & 2.048 & 0 & 0.0005 \\ 0.0005 & 0 & 2.0480 & 0 \\ 0 & 0.0005 & 0 & 2.0485 \end{bmatrix}, D_{L3} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Nonlinear "normal form" Controller :

By considering $y = x_2$, and using the nonlinear model of the ROTPEN in Appendix 9.1, the following coordinate transformation:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_3 \left(\frac{l_1}{2} c_2 \right) + x_4 \frac{l_2}{3} \end{bmatrix}$$

is used to put the system in the so-called *normal* or *tracking form* (Marino & Tomei, 1995), that is:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} \xi_2 \\ f_4(x) + g_4(x)u \\ x_3 \\ -\frac{l_1}{2} x_3 x_4 s_2 + \frac{l_1}{2} c_2 f_3(x) + \frac{l_2}{3} f_4(x) \end{bmatrix}$$

and using the control law:

$$u = \frac{1}{g_4(x)} [-9x_2 - 6x_4 - f_4(x)]$$

where $f_4(x)$ and $g_4(x)$ denote the 4th elements of $f(x)$ and $g(x)$ in Appendix 9.1 respectively. The subsystem (ξ_1, ξ_2) is then stabilized. It is important to note that the zero dynamics in this case, i.e the subsystem (η_1, η_2) is unstable, and therefore the motor angle is not guaranteed to converge to the reference input.

Static Lipschitz observer : (obtained for $\alpha = 1$, $\epsilon = 0.5$)

$$L_5 = \begin{bmatrix} 1.7108 & -2.1247 & 1.9837 & -5.4019 \\ 0.4338 & -0.2089 & 1.1030 & -2.8972 \end{bmatrix}^T.$$

9.3 Models and Parameters for Case Study 2

Lipschitz reduced-order model for observer design ($\bar{x} = [\theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2]^T$):

$$\dot{\bar{x}} = \begin{bmatrix} 0 & 0 & 1 \\ -25.14 & -17.22 & 0.2210 \\ 68.13 & 16.57 & -0.599 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ \phi_1(\bar{x}, u) \\ \phi_2(\bar{x}, u) \end{bmatrix}$$

$$\bar{y} = [1 \ 0 \ 0] \bar{x}$$

Lipschitz dynamic observer for sensor bias : (K_6 , obtained for $\lambda = 10^{-12}$, $\epsilon = 0.1$)

$$A_{L6} = \begin{bmatrix} -175.7353 & 3.8503 & 0.1710 & -30.6336 \\ 16.8182 & -171.9539 & 26.7652 & 32.1257 \\ 35.1361 & 16.5360 & -97.3465 & 114.1349 \\ -87.9041 & 25.7568 & 62.1442 & -87.8099 \end{bmatrix}, B_{L6} = \begin{bmatrix} 5.0462 \\ -44.8932 \\ -75.4539 \\ 106.5497 \end{bmatrix},$$

$$C_{L6} = \begin{bmatrix} 167.6750 & -5.0531 & -8.5208 & 42.0138 \\ -7.1899 & 155.5373 & -42.6804 & -11.1441 \\ 5.3053 & -18.7128 & -120.8293 & 171.1055 \end{bmatrix}, D_{L6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Lipschitz dynamic observer for fault of 2 rad/sec : (K_7 , obtained for $\lambda = 10^{-12}$, $\epsilon = 0.1$)

$$A_{L7} = \begin{bmatrix} -816.9997 & -12.5050 & -51.0842 & -64.0861 & 31.8003 \\ 23.8482 & -772.7024 & 149.1621 & 122.7602 & -75.3718 \\ -3.0714 & 139.9543 & -412.1421 & 361.2027 & -176.7926 \\ -193.3011 & 128.2831 & 346.2370 & -405.3024 & 201.2094 \\ 71.5547 & -47.7237 & -104.0209 & 129.8922 & -64.7247 \end{bmatrix}, B_{L7} = \begin{bmatrix} 9.2096 \\ -73.6540 \\ -80.3861 \\ 177.6628 \\ -67.4227 \end{bmatrix},$$

$$C_{L7} = \begin{bmatrix} 809.4037 & 11.3091 & 28.1928 & 88.3295 & -43.7581 \\ -13.1309 & 758.2718 & -276.6110 & 4.7255 & 12.0717 \\ -15.9908 & -176.8554 & -509.7118 & 587.8999 & -294.7496 \end{bmatrix}, D_{L7} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Lipschitz dynamic observer for low frequencies : (K_8 , obtained for $\lambda = 10^{-12}$, $\epsilon = 0.1$)

$$A_{L8} = \begin{bmatrix} -217.7814 & 1.8898 & -4.8573 & -38.2385 \\ -1.5288 & -185.0261 & 38.1186 & 36.8585 \\ 108.5437 & 28.4810 & -87.0920 & 135.1710 \\ -618.9648 & 28.9348 & 82.1016 & -164.6086 \end{bmatrix}, B_{L8} = \begin{bmatrix} -30.2950 \\ 26.3896 \\ 147.7784 \\ -637.5223 \end{bmatrix},$$

$$C_{L8} = \begin{bmatrix} -184.6168 & 3.4213 & 1.8716 & -51.2266 \\ 6.5728 & -171.5615 & 49.1851 & 16.3542 \\ -4.3022 & 15.0586 & 114.2413 & -224.5769 \end{bmatrix}, D_{L8} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

10. References

- Aboky, C., Sallet, G. & Vivalda, J. (2002). Observers for Lipschitz nonlinear systems, *Int. J. of Contr.*, vol. 75, No. 3, pp. 204-212.
- Adjallah, K., Maquin, D. & Ragot, J. (1994). Nonlinear observer based fault detection, *IEEE Trans. on Automat. Contr.*, pp. 1115-1120.
- Chen, R., Mingori, D. & Speyer, J. (2003). Optimal stochastic fault detection filter, *Automatica*, vol. 39, No. 3, pp. 377-390.

- Chen, J. & Patton, R. (1999). *Robust model-based fault diagnosis for dynamic systems*, Kluwer Academic Publishers.
- Doyle, J., Glover, K., Khargonekar P. & Francis, B. (1989). State space solutions to standard H_2 and H_∞ control problems, *IEEE Trans. Automat. Contr.*, Vol. 34, No. 8, pp. 831-847.
- Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results, *Automatica*, vol. 26, No. 3, pp. 459-474.
- Gahinet, P. & Apkarian, P. (1994). A linear matrix inequality approach to H_∞ control, *Int. J. of Robust and Nonlinear Contr.*, vol. 4, pp. 421-448.
- Garcia, E. & Frank, P. (1997). Deterministic nonlinear observer based approaches to fault diagnosis: A survey, *Contr. Eng. Practice*, vol. 5, No. 5, pp. 663-670.
- Hammouri, H., Kinnaert, M. & El Yaagoubi, E. (1999). Observer-based approach to fault detection and isolation for nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol. 44, No. 10.
- Hill, D. & Moylan, P. (1977). Stability Results for Nonlinear Feedback Systems, *Automatica*, Vol. 13, pp. 377-382.
- Iwasaki, T. & Skelton, R. (1994). All controllers for the general H_∞ control problem: LMI existence conditions and state space formulas, *Automatica*, Vol. 30, No. 8, pp. 1307-1317.
- Kabore, P. & Wang, H. (2001). Design of fault diagnosis filters and fault-tolerant control for a class of nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol. 46, No. 11.
- Lynch, A. (2004). *Control Systems II (Lab Manual)*, University of Alberta.
- Marino, R. & Tomei, P. (1995). *Nonlinear Control Design - Geometric, Adaptive and Robust*, Prentice Hall Europe, 1995.
- Marquez, H. (2003). *Nonlinear Control Systems: Analysis and Design*, Wiley, NY.
- Pertew, A. (2006). Nonlinear observer-based fault detection and diagnosis, *Ph.D Thesis*, Department of Electrical and Computer Engineering, University of Alberta.
- Pertew, A., Marquez, H. & Zhao, Q. (2005). H_∞ synthesis of unknown input observers for nonlinear Lipschitz systems, *International J. Contr.*, vol. 78, No. 15, pp. 1155-1165.
- Pertew, A., Marquez, H. & Zhao, Q. (2006). H_∞ observer design for Lipschitz nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol. 51, No. 7, pp. 1211-1216.
- Pertew, A., Marquez, H. & Zhao, Q. (2007). LMI-based sensor fault diagnosis for nonlinear Lipschitz systems, *IEEE Trans. on Automat. Contr.*, vol. 43, pp. 1464-1469.
- Raghavan, S. & Hedrick, J. (1994). Observer design for a class of nonlinear systems, *Int. J. of Contr.*, vol. 59, No. 2, pp. 5515-528.
- Rajamani, R. (1998). Observers for Lipschitz nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol. 43, No. 3, pp. 397-401.
- Rajamani, R. & Cho, Y. (1998). Existence and design of observers for nonlinear systems: relation to distance of unobservability, *Int. J. Contr.*, Vol. 69, pp. 717-731.
- Scherer, C. (1992). H_∞ optimization without assumptions on finite or infinite zeros, *Int. J. Contr. and Optim.*, Vol. 30, No. 1, pp. 143-166.
- Sciavicco, L. & Siciliano, B. (1989). *Modeling and Control of Robot Manipulators*, McGraw Hill.
- Stoorvogel, A. (1996). The H_∞ control problem with zeros on the boundary of the stability domain, *Int. J. Contr.*, Vol. 63, pp. 1029-1053.
- Vemuri, A. (2001). Sensor bias fault diagnosis in a class of nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol. 46, No. 6.
- Wang, H., Huang, Z. & Daley, S. (1997). On the Use of Adaptive Updating Rules for Actuator and Sensor Fault Diagnosis, *Automatica*, Vol. 33, No. 2, pp. 217-225.

- Willsky, A. (1976). A survey of design methods for failure detection in dynamic systems, *Automatica*, vol. 12, pp. 601-611.
- Yu, D. & Shields, D. (1996). A bilinear fault detection observer, *Automatica*, vol. 32, No. 11, pp. 1597-1602.
- Zhong, M., Ding, S., Lam, J. & Wang, H. (2003). An LMI approach to design robust fault detection filter for uncertain LTI systems, *Automatica*, vol. 39, No. 3, pp. 543-550.
- Zhou, K. & Doyle, J. (1998). *Essentials of robust control*, Prentice-Hall, NY.

Cartesian Control for Robot Manipulators

Pablo Sánchez-Sánchez and Fernando Reyes-Cortés
Benemérita Universidad Autónoma de Puebla (BUAP)
Facultad de Ciencias de la Electrónica
México

1. Introduction

A robot is a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions, all this for a best performance in a variety of tasks. A useful robot is the one which is able to control its movements and the forces it applies to its environment. Typically, robot manipulators are studied in consideration of their displacements on joint space, in other words, robot's displacements inside of its workspace usually are considered as joint displacements, for this reason the robot is analyzed in a joint space reference. These considerations generate an important and complex theory of control in which many physical characteristics appear, this kind of control is known as *joint control*.

The joint control theory expresses the relations of position, velocity and acceleration of the robot in its native language, in other words, describes its movements using the torque and angles necessary to complete the task; in majority of cases this language is difficult to understand by the end user who interprets space movements in cartesian space easily. The singularities in the boundary workspace are those which occur when the manipulator is completely stretched-out or folded back on itself such as the end-effector is near or at the boundary workspace. It's necessary to understand that singularity is a mathematical problem that undefined the system, that is, indicates the absence of velocity control which specifies that the end-effector never get the desired position at some specific point in the workspace, this doesn't mean the robot cannot reach the desired position structurally, whenever this position is defined inside the workspace. This problem was solved by S. Arimoto and M. Takegaki in 1981 when they proposed a new control scheme based on the Jacobian Transposed matrix; eliminating the possibility of singularities and giving origin to the *cartesian control*.

The joint control is used for determining the main characteristics of the cartesian control based on the Jacobian Transposed matrix. It is necessary to keep in mind that to consider the robot's workspace like a joint space, has some problems with interpretation because the user needs having a joint dimensional knowledge, thus, when the user wants to move the robot's end-effector through a desired position he needs to understand the joint displacements the robot needs to do, to get the desired position. This interpretation problem is solved by using the cartesian space, that is, to interpret the robot's movements by using cartesian coordinates on reference of cartesian space; the advantage is for the final user who has the cartesian dimensional knowledge for understanding the robot's movements. Due this reason, learning the mathematical tools for analysis by the robot's movements on cartesian space is necessary, this allows us to propose control structures, to use the dynamic model and to understand the

physical phenomena on robot manipulators on cartesian space. When we control the global motion or position of general manipulators, we are confronted with the nonlinear dynamics in a lot of degrees of freedom. In literature focused with the dynamic control of manipulators, the complexity of nonlinear dynamics is emphasized and some methods, compensating all nonlinear terms in dynamics in real time, are developed in order to reduce the complexity in system control. However, these methods require a large amount of complicated calculation so it is difficult to implement these methods with low level controllers such as microcomputers. In addition, the reliability of these methods may be lost when a small error in computation or a small change in system parameters occurs, occurs because they are not considered in the control. Most industrial robots, each joint of manipulator is independently controlled by a simple linear feedback. However, convergence for target position has not been enough investigated for general nonlinear mechanical systems.

This chapter is focused on the position control for robot manipulators by using control structures defined on the cartesian space because the robot move freely in its workspace, which is understood by the final user like cartesian space. Besides, the mathematical tools will be detailed for propose, analyze and evaluating control structures in cartesian space.

2. Preliminaries: forward kinematics and Jacobian matrix

A rigid multi-body system consists in a set of rigid objects, called links, joined together by joints. Simple kinds of joints include revolute (rotational) and prismatic (translational) joints. It is also possible to work with more general types of joints, and thereby simulate non-rigid objects. Well-known applications of rigid multi-bodies include robotic arms. A robot manipulator is modeled with a set of links connected by joints. There are a variety of possible joint types. Perhaps the most common type is a rotational joint with its configuration described by a single scalar angle value. The key point is: "the configuration of a joint is a continuous function of one or more real scalars; for rotational joints", the scalar is the angle of the joint. Complete configuration in robot manipulators is specified by vectors, for example the position is described as:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (1)$$

where $q \in \mathbb{R}^{n \times 1}$. We assume there are n joints and each q_n value is called a *joint position*. The robot manipulator will be controlled by specifying target positions by the end-effectors. The desired positions are also given by a vector

$$q_d = \begin{bmatrix} q_{d_1} \\ q_{d_2} \\ \vdots \\ q_{d_n} \end{bmatrix} \quad (2)$$

where q_{d_i} is the desired position for the i th end-effector. We let $\tilde{q}_i = q_{d_i} - q_i$, the desired change in position of the i th end effector, also this vector is well-known as an error position. The end-effector positions (x, y, z) are functions of the joint angles q ; this fact can be expressed as:

$$x_i = f_i(q) \quad \text{for } i = 1, 2, \dots, k \tag{3}$$

this equation is well-known as *forward kinematics*.

2.1 Case of study: Cartesian robot (forward kinematics)

In order to understand application of cartesian control in robot manipulators a case of study will be used, which all the concepts were evaluated. In this section we will obtain the forward kinematics of a three degrees of freedom cartesian robot, Figure 1; and we will use this information in the following sections.

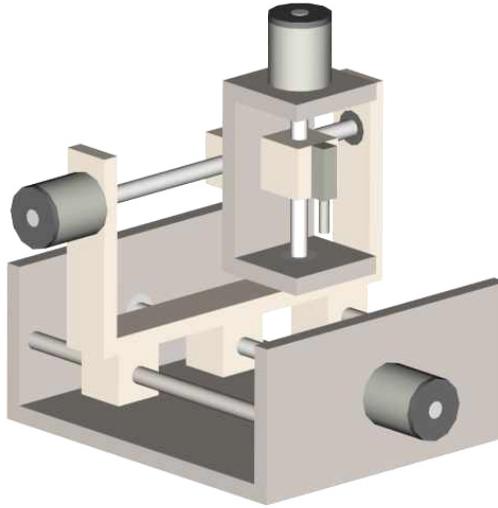


Fig. 1. Three degrees of freedom cartesian robot.

In order to obtain the forward kinematics of three degrees of freedom cartesian robot we need to draw a system diagram, Figura 2,

where q_1, q_2, q_3 are joint displacements; and m_1, m_2, m_3 represent the masses of each link. As it is observed, translation is the unique movement that realizes this kind of robots, then the forward kinematics are defined as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \\ 0 \end{bmatrix} ; \quad \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ 0 \end{bmatrix} ; \quad \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} . \tag{4}$$

We can observe, that in the first vector is contemplated only by the first displacement of value q_1 , in the second one considers the movement of translation in q_1 and q_2 respecting the axis x and y , and finally the complete displacement in third axis described in the last vector, being this representation the robot forward kinematics.

2.2 Jacobian matrix

The Jacobian matrix $J(q)$ is a multidimensional form of the derivative. This matrix is used to relate the joint velocity \dot{q} with the cartesian velocity \dot{x} , based on this reason we are able to think about Jacobian matrix as mapping velocities in q to those in x :

$$\dot{x} = J(q) \dot{q}. \quad (5)$$

where \dot{x} is the velocity on cartesian space; \dot{q} is the velocity in joint space; and $J(q)$ is the Jacobian matrix of the system.

In many cases, we use modeling and simulation as a tool for analysis about the behavior of a given system. Even though at this stage, we have not formed the equations of motion for a robotic manipulator, by inspecting the kinematic models, we are able to reveal many characteristics from the system. One of the most important quantities (for the purpose of analysis) in (5), is the Jacobian matrix $J(q)$. It reveals many properties of a system and can be used for the formulation of motion equations, analysis of special system configurations, static analysis, motion planning, etc. The robot manipulator's Jacobian matrix $J(q)$ is defined as follow:

$$J(q) = \frac{\partial f(q)}{\partial q} = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \frac{\partial f_1(q)}{\partial q_2} & \dots & \frac{\partial f_1(q)}{\partial q_n} \\ \frac{\partial f_2(q)}{\partial q_1} & \frac{\partial f_2(q)}{\partial q_2} & \dots & \frac{\partial f_2(q)}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(q)}{\partial q_1} & \frac{\partial f_m(q)}{\partial q_2} & \dots & \frac{\partial f_m(q)}{\partial q_n} \end{bmatrix} \quad (6)$$

where $f(q)$ is the relationship of forward kinematics, equation (3); n is the dimension of q ; and m is the dimension of x . We are interested about finding what joint velocities \dot{q} result in given (desired) v . Hence, we need to solve a system equations.

2.2.1 Case of study: Jacobian matrix of the cartesian robot

In order to obtain the Jacobian matrix of the three degrees of freedom cartesian robot it is necessary to use the forward kinematics which is defined as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (7)$$

Now, doing the partial derivation of x in reference to q_1, q_2, q_3 we have:

$$\begin{aligned}\frac{\partial x}{\partial q_1} &= \frac{\partial q_1}{\partial q_1} = \dot{q}_1 \\ \frac{\partial x}{\partial q_2} &= \frac{\partial q_1}{\partial q_2} = 0 \\ \frac{\partial x}{\partial q_3} &= \frac{\partial q_1}{\partial q_3} = 0\end{aligned}\tag{8}$$

The partial derivation of y in reference to q_1, q_2, q_3 are:

$$\begin{aligned}\frac{\partial y}{\partial q_1} &= \frac{\partial q_2}{\partial q_1} = 0 \\ \frac{\partial y}{\partial q_2} &= \frac{\partial q_2}{\partial q_2} = \dot{q}_2 \\ \frac{\partial y}{\partial q_3} &= \frac{\partial q_2}{\partial q_3} = 0\end{aligned}\tag{9}$$

The partial derivation of z in reference to q_1, q_2, q_3 , we have:

$$\begin{aligned}\frac{\partial z}{\partial q_1} &= \frac{\partial q_3}{\partial q_1} = 0 \\ \frac{\partial z}{\partial q_2} &= \frac{\partial q_3}{\partial q_2} = 0 \\ \frac{\partial z}{\partial q_3} &= \frac{\partial q_3}{\partial q_3} = \dot{q}_3\end{aligned}\tag{10}$$

The system $\dot{x} = J(q)\dot{q}$ is described by following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}\tag{11}$$

where the Jacobian matrix elements are defined using the equations (8), (9) and (10):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}\tag{12}$$

2.3 Jacobian transpose matrix

The transpose of a matrix $J(q)$ is another matrix $J(q)^T$ created by anyone of the following equivalent actions: write the $J(q)^T$ rows as the $J(q)^T$ columns; write the $J(q)^T$ columns as the $J(q)^T$ rows; and reflect $J(q)$ by its main diagonal (which starts from the top left) to obtain $J(q)^T$. Formally, the transpose of an $m \times n$ matrix $J(q)$ with elements $J(q)_{ij}$ is $n \times m$ matrix as follow

$$J_{ji}(q)^T = J_{ij}(q) \quad \text{for } 1 \leq i \leq n, 1 \leq j \leq m. \quad (13)$$

The transposing of a scalar is the same scalar.

2.3.1 Case of study: Jacobian transpose matrix of the cartesian robot

In order to obtain the Jacobian transpose matrix $J(q)^T$ we apply (13) leaving of the equation (12). In particular case of cartesian robot the Jacobian matrix $J(q)$ is equal to the identity matrix I , thus its transposed matrix $J(q)^T$ is the same, thus we have:

$$J(q)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

2.4 Singularities

Singularities correspond certain configurations in robot manipulators which have to be avoided because they lead to an abrupt loss of manipulator rigidity. In the vicinity of these configurations, manipulator can become uncontrollable and the joint forces could increase considerably and may there would be risk to even damage the manipulator mechanisms. The singularities in a workspace can be identified mathematically when the determinant in the Jacobian matrix is zero:

$$\det J(q) = 0. \quad (15)$$

Mathematically this means that matrix $J(q)$ is degenerated and there is, in the inverse geometrical model, an infinity of solutions in the vicinity of these points.

2.5 Singular configurations

Due to the tuning of derivative and proportional matrices from the control algorithms of which objective is to maintain in every moment the error position nearest to zero, it exists the possibility that in certain values of the determinant in Jacobian matrix the system is singular undefined. It's denominated *singular configurations of a robot* those distributions in which that determinant of the Jacobian matrix is zero, equation (15). Because of this circumstance, in the singular configurations the inverse Jacobian matrix doesn't exist. For a undefine Jacobian matrix, an infinitesimal increment in the cartesian coordinates would suppose an infinite increment at joint coordinates, which is translated as movements from the articulations to inaccessible velocities on some part of its links for reaching the desired position for a constant velocity in the practice. Therefore, in the vicinity of the singular configurations lost some degrees in the robot's freedom, being impossible their end-effector moves in a certain cartesian address.

Different singular configurations on robot can be classified as:

- *Singularities in the limits in the robot's workspace.* These singularities are presented when the robot's boundary is in some point of the limit of interior or external workspace. In this situation it is obvious the robot won't be able to move in the addresses that were taken away from this workspace.
- *Singularities inside the robot's workspace.* They take place generally inside the work area and for the alignment of two or more axes in the robot's articulations.

2.5.1 Case of study: determinant of the Jacobian matrix of the cartesian robot

In order to determine if there are singularities in the system, it is necessary to obtain the determinant on the system $\det J(q)$, considering a general structure of the Jacobian matrix, thus we have:

$$\det J(q) = j_{11} \begin{bmatrix} j_{22} & j_{23} \\ j_{32} & j_{33} \end{bmatrix} - j_{12} \begin{bmatrix} j_{21} & j_{23} \\ j_{31} & j_{33} \end{bmatrix} + j_{13} \begin{bmatrix} j_{21} & j_{22} \\ j_{31} & j_{32} \end{bmatrix}$$

$$\det J(q) = j_{11}(j_{22}j_{33} - j_{32}j_{23}) - j_{12}(j_{21}j_{33} - j_{31}j_{23}) + j_{13}(j_{21}j_{32} - j_{31}j_{22}) \quad (16)$$

$$\det J(q) = 1$$

As it is observed, the determinant in the Jacobian matrix is not undefined in any point which indicates the workspace for the cartesian robot is complete.

2.5.2 Workspace

The workspace is the area where the robot can move freely with no damage. This area is determined by the robot's physical and mechanical capacities. The workspace is defined without considering the robot's end-effector, in the Figure 3 the workspace of a robot of three degrees of freedom is described.

2.6 Inverse Jacobian matrix

In mathematics, and especially in linear algebra, a matrix squared A with an order $n \times n$ it is said is reversible, nonsingular, non-degenerate or regular if exists another squared matrix with order $n \times n$ called *inverse matrix* A^{-1} and represented matrix like

$$AA^{-1} = A^{-1}A = I \quad (17)$$

I is the identity matrix with order $n \times n$ and the used product is the usual product of matrices. The mathematical definition in the inverse matrix is defined as follow:

$$J(q)^{-1} = \frac{C^T}{\det J(q)} \quad (18)$$

where C is the co-factors matrix.

2.6.1 Case of study: co-factors matrix in the cartesian robot

In order to obtain the co-factor matrix it is necessary to apply the following procedure: Considering the matrix A defined like:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (19)$$

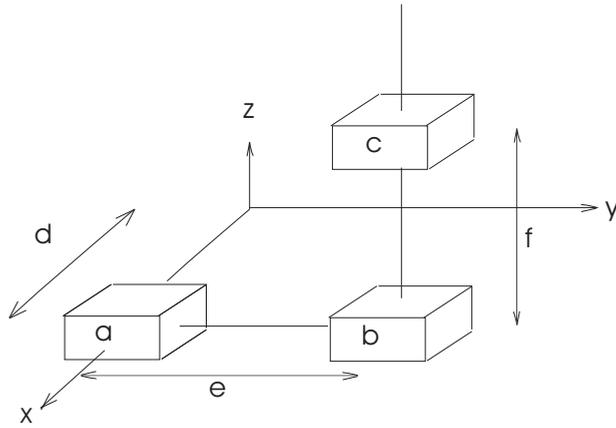


Fig. 2. Diagram of three degrees of freedom cartesian robot.

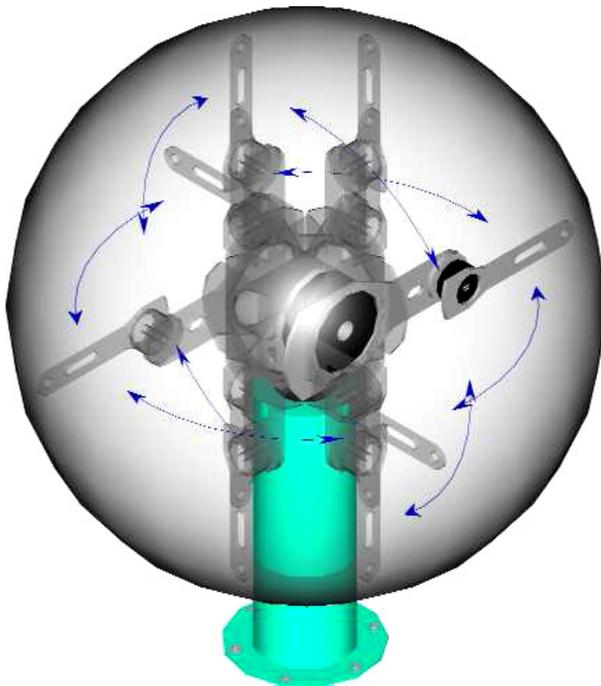


Fig. 3. Robot manipulator's workspace.

we obtain the following co-factors matrix:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (20)$$

where each component is defined as:

$$\begin{aligned} c_{11} &= + (ei - hf) \\ c_{12} &= - (di - gf) \\ c_{13} &= + (dh - ge) \\ c_{21} &= - (bi - hc) \\ c_{22} &= + (ai - gc) \\ c_{23} &= - (ah - gb) \\ c_{31} &= + (bf - ec) \\ c_{32} &= - (af - dc) \\ c_{33} &= + (ae - db) \end{aligned} \quad (21)$$

Considering the Jacobian matrix (12) we can obtain the following co-factors matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (22)$$

where the components of the matrix are defined as:

$$\begin{aligned} c_{11} &= + (ei - hf) &= + (1 - 0) &= 1 \\ c_{12} &= - (di - gf) &= - (0 - 0) &= 0 \\ c_{13} &= + (dh - ge) &= + (0 - 0) &= 0 \\ c_{21} &= - (bi - hc) &= - (0 - 0) &= 0 \\ c_{22} &= + (ai - gc) &= + (1 - 0) &= 1 \\ c_{23} &= - (ah - gb) &= - (0 - 0) &= 0 \\ c_{31} &= + (bf - ec) &= + (0 - 0) &= 0 \\ c_{32} &= - (af - dc) &= - (0 - 0) &= 0 \\ c_{33} &= + (ae - db) &= + (1 - 0) &= 1 \end{aligned} \quad (23)$$

2.6.2 Case of study: inverse Jacobian matrix of the cartesian robot

In order to obtain the inverse Jacobian matrix $J(q)^{-1}$ according the definition on (18), it is necessary the transposing co-factor matrix C^T ,

$$C^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

and the determinant of the Jacobian matrix (16), we obtain:

$$J(q)^{-1} = \frac{C^T}{\det J(q)} = \frac{1}{1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (25)$$

$$J(q)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As it is observed, for the specific case of the three degrees of freedom cartesian robot the inverse matrix does exist.

3. Dynamic model

The dynamic model is the mathematical representation of a system which describes its behavior in the internal and external stimulus presented in the system. For cartesian control design purposes, and for designing better controllers, it is necessary to reveal the dynamic behavior of the robot via a mathematical model obtained from some basic physical laws. We use Lagrangian dynamics to obtain the describing mathematical equations. We begin our development with the general Lagrange equation about motion. Considering Lagrange's equation for a conservative system as given by:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau - f(\tau, \dot{q}) \quad (26)$$

where $q, \dot{q} \in \mathbb{R}^{n \times 1}$ are position and velocity in a joint space, respectively; $\tau \in \mathbb{R}^{n \times 1}$ is a vector of an applied torque; $f(\tau, \dot{q}) \in \mathbb{R}^{n \times 1}$ is the friction vector; and the Lagrangian $\mathcal{L}(q, \dot{q})$ is the difference between kinetic $\mathcal{K}(q, \dot{q})$ and potential $\mathcal{U}(q)$ energies:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q). \quad (27)$$

The application of the Lagrange's equation results in the mathematical equation which describes the system behavior at any stimulus, *dynamic model equation*. Then it can be shown the robot dynamics are given by:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) + f(\tau, \dot{q}) = \tau \quad (28)$$

where q, \dot{q}, \ddot{q} are the position, velocity and acceleration in joint space, respectively; $M(q) \in \mathbb{R}^{n \times n}$ is symmetric, positive-definite inertial matrix; $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is a matrix containing the Coriolis and centripetal torques effects; $g(q) \in \mathbb{R}^{n \times 1}$ is a vector of gravity torque obtained as gradient result on the potential energy,

$$g(q) = \frac{\partial \mathcal{U}(q)}{\partial q}, \quad (29)$$

and $f(\tau, \dot{q}) \in \mathbb{R}^{n \times 1}$ are the vector of friction torques. The friction torque is decentralize in the sense that $f(\tau, \dot{q})$ depends only on τ and \dot{q}

$$f(\tau, \dot{q}) = \begin{bmatrix} f_1(\tau_1, \dot{q}_1) \\ f_2(\tau_2, \dot{q}_2) \\ \vdots \\ f_n(\tau_n, \dot{q}_n) \end{bmatrix}. \quad (30)$$

Friction is the tangential reaction force between two surfaces in contact. Physically these reaction forces are the result of many different mechanisms, which depend on geometry and topology contact, properties of bulk and surface materials on the bodies, displacement and relative velocity on the bodies and presence of lubrication.

It is well known that exist two friction models: the *static and dynamic*. The static models of friction consist on different components, each take care about certain friction force issues. The main idea is: friction opposes motion and its magnitude is independent on velocity and contact area. The friction torques are assumed to be a dissipated energy at all nonzero velocities, therefore, their entries are bounded within the first and third quadrants. The friction force is given by a static function possibly except for a zero velocity. Figure 4(a) shows Coulomb friction; Figure 4(b) Coulomb plus viscous friction; Stiction plus Coulomb and viscous friction is shown in Figure 4(c); and Figure 4(d) shows how the friction force may decrease continuously from the static friction level.

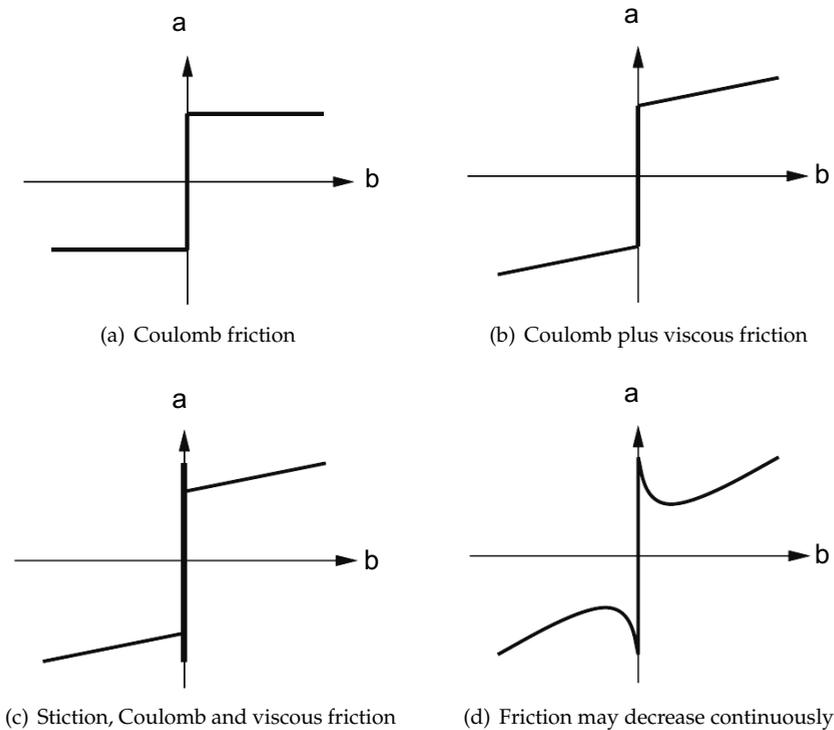


Fig. 4. Examples of static friction models.

This feature allows considering the common Coulomb and viscous friction models. At zero velocities, only static friction is satisfying presented

$$f_i(\tau_i, 0) = \tau_i - g_i(q) \tag{31}$$

for $-f_i \leq \tau_i - g_i(q) \leq f_i$ with f_i being the limit on the static friction torques for joint i .

Lately there has been a significant interest in dynamic friction models. This has been driven by intellectual curiosity, demands for precision servos and advances in hardware that make it possible for implementing friction compensators. The *Dahl model* was developed with the purpose of simulating control systems with friction. Dahl's starting point had several experiments on friction in servo systems with ball bearings. One of his findings was that bearing friction behave was very similar on solid friction. These experiments indicate that there are metal contacts between the surfaces. Dahl developed a simple comparatively model and was used extensively to simulate systems with ball bearing friction.

The starting point for Dahl's model is the stress-strain curve in classical solid mechanics, Figure 5. When the subject is under stress the friction force increases gradually until a rupture occurs. Dahl modeled the stress-strain curve by a differential equation. x will be the displacement, F the friction force, and F_c the Coulomb friction force. Then Dahl's model has this form:

$$\frac{dF}{dx} = \sigma \left(1 - \frac{F}{F_c} \operatorname{sgn}(v)\right)^\alpha \quad (32)$$

where σ is the stiffness coefficient; and α is a parameter which determines the shape of the stress-strain curve. The value $\alpha = 1$ is most commonly used. Higher values will give a stress strain curve with a sharper bend. The friction force $|F|$ will never be larger than F_c if its initial value is such that $|F(0)| < F_c$.

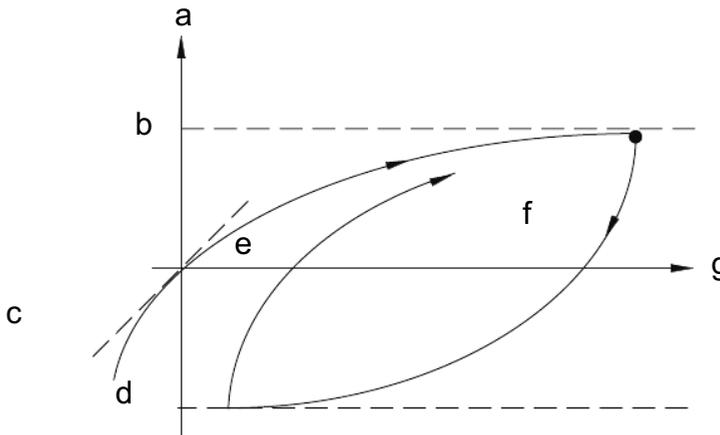


Fig. 5. Friction force as a function of displacement for Dahl's model.

With an absence of friction and other disturbances, the dynamic model (28) about n -links rigid robot manipulator can be written as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau. \quad (33)$$

It is assumed that the robot links are joined together with revolute joints. Although the equation (28) is complex, for this reason we use the equation (33) for the analysis to facilitate control system design. It is necessary to indicate that the Euler-Lagrange's methodology is not the

only procedure to obtain the robot's dynamic model since this issue has been object of many study and researching. Researchers have developed alternative formulations based on the Newtonian and Lagrangian mechanics with one objective: obtaining a more efficient model.

3.1 Properties

It is essential to analyze the properties on the model to be able to apply them in the obtaining in the model on cartesian space.

3.1.1 Inertial matrix properties

The inertia matrix $M(q)$ has an important characteristic like its intimate relation with kinetic energy $\mathcal{K}(q, \dot{q})$,

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}. \quad (34)$$

The inertial matrix $M(q)$ is a positive definite matrix $M(q) > 0$; so is a symmetric matrix, $M(q) > 0 \Rightarrow \exists M(q)^{-1} > 0$. Another vital property of $M(q)$ is that it could be bounded above and below. So,

$$\mu_1(q) I \leq M(q) \leq \mu_2(q) I \quad (35)$$

where I is the identity matrix, $\mu_1(q) \neq 0$ and $\mu_2(q)$ are scalar constants for a revolute arm and scalar function of q for an arm generally containing prismatic joints. It is easy to realize then that $M^{-1}(q)$ is also bounded since

$$0 \leq \frac{1}{\mu_2(q)} I \leq M^{-1}(q) \leq \frac{1}{\mu_1(q)} I. \quad (36)$$

In the case of robots provided solely of rotational joints, a constant $\beta > 0$ exists like:

$$\lambda_{\max} \{M(q)\} \leq \beta \quad \forall q \in \mathbb{R}^{n \times 1} \quad (37)$$

where β is calculated as:

$$\beta \geq n \left[\max_{i,j,q} |M_{ij}(q)| \right] \quad (38)$$

where $M_{ij}(q)$ are the ij th element of the matrix $M(q)$.

3.1.2 Coriolis and centripetal terms properties

The matrix of Coriolis and centripetal force $C(q, \dot{q})$ is $n \times n$ matrix, of which elements are functions of q and \dot{q} . Matrix $C(q, \dot{q})$ can't be unique, but the vector $C(q, \dot{q})\dot{q}$ can. If the matrix $C(q, \dot{q})$ is evaluated considering the joint velocity \dot{q} in zero, the matrix is zero for all vector q

$$C(q, 0) = 0 \quad \forall q. \quad (39)$$

For all vector $q, x, y, z \in \mathbb{R}^{n \times 1}$ and the scale α we have:

$$C(q, x)y = C(q, y)x \quad (40)$$

$$C(q, z + \alpha x)y = C(q, z)y + \alpha C(q, x)y$$

Vector $C(q, x)y$ can be expressed on the form:

$$C(q, x)y = \begin{bmatrix} x^T C_1(q)y \\ x^T C_2(q)y \\ \vdots \\ x^T C_n(q)y \end{bmatrix} \quad (41)$$

where $C_k(q)$ are symmetrical matrices of dimensions $n \times n$ for all $k = 1, 2, \dots, n$; In fact the ij th element $C_{kij}(q)$ of matrix $C_k(q)$ corresponds to the symbol of Christoffel:

$$C_{ijk}(q) = \frac{1}{2} \left[\frac{\partial M_{kj}(q)}{\partial q_i} + \frac{\partial M_{ki}(q)}{\partial q_j} - \frac{\partial M_{ij}(q)}{\partial q_k} \right]. \quad (42)$$

In the case of robots provided solely of rotational joints, a constant $k_{C_1} > 0$ exists like:

$$\|C(q, x)y\| \leq k_{C_1} \|x\| \|y\| \quad \text{for all } q, x, y \in \mathbb{R}^{n \times 1} \quad (43)$$

In the case of robots provided solely of rotational joints, a constants $k_{C_1} > 0$ and $k_{C_2} > 0$ exist like:

$$\|C(x, z)w - C(yu)w\| \leq k_{C_1} \|z - u\| \|w\| + k_{C_2} \|x - y\| \|w\| \|z\| \quad \forall u, x, y, w \in \mathbb{R}^{n \times 1} \quad (44)$$

Matrix $C(q, \dot{q})$ is related with the inertial matrix $M(q)$ by the expression:

$$x^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] x = 0 \quad \forall q, \dot{q}, x \in \mathbb{R}^{n \times 1} \quad (45)$$

In analogous form, the matrix $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric, and it also is certain that

$$\dot{M}(q) = C(q, \dot{q}) + C(q, \dot{q})^T. \quad (46)$$

3.1.3 Gravity terms properties

The gravity vector is present in robots which have not been designed mechanically with compensation of gravity, so, without counterbalances; or for robots assigned to move outside the horizontal plane. The gravity terms only depends on joint positions q ; and the gravity terms can be related with the joint velocity \dot{q} this means:

$$\int_0^T g(q)^T \dot{q} dt = \mathcal{U}(q(T)) - \mathcal{U}(q(0)) \quad \text{for all } T \in \mathbb{R}_+. \quad (47)$$

In the case of robots provided solely of rotational joints, a constant $k_{\mathcal{U}}$ exists like:

$$\int_0^T g(q)^T \dot{q} dt + \mathcal{U}(q(0)) \geq k_{\mathcal{U}} \quad \text{for all } T \in \mathbb{R}_+ \text{ and where } k_{\mathcal{U}} = \min_q \{\mathcal{U}(q)\} \quad (48)$$

In the case of robots provided solely of rotational joints the gravity vector $g(q)$ is Lipschitz, this means that a constant $k_g > 0$ exists like:

$$\|g(x) - g(y)\| \leq k_g \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^{n \times 1} \quad (49)$$

A simple form to calculate k_g is:

$$k_g \geq n \left[\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right] \quad (50)$$

In addition k_g satisfies:

$$k_g \geq \left\| \frac{\partial g(q)}{\partial q} \right\| \geq \lambda \max \left\{ \frac{\partial g(q)}{\partial q} \right\} \quad (51)$$

The gravity term $g(q)$ is bounded only if q is bounded:

$$\|g(q)\| \leq g_b \quad (52)$$

where g_b is a scalar constant for revolute arms and a scalar function of q for arms containing revolute joints.

3.2 Case of study: Dynamic model of cartesian robot

In this section we will obtain the dynamic model on three degrees of freedom cartesian robot, and we will use this information in the following sections. The case of study is represented in Figure 1. In order to obtain the dynamic model we need to consider its forward kinematics (4). However, as the movement of the cartesian robot only is about transferring, the rotation energy is zero, therefore, the equation in the kinetic energy is reduced to:

$$\mathcal{K}(q, \dot{q}) = \frac{mv^2}{2} = \frac{q^T M(q) q}{2}. \quad (53)$$

Considering velocity, it is defined as:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (54)$$

and in (53) is represented in the squared way, it is necessary its vectorial representation,

$$v^2 = \|v\|^2 = v^T v = [v_1 \quad v_2 \quad v_3] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \quad (55)$$

Deriving (4) and solving (55) we have:

$$\begin{aligned} v_1^2 &= \dot{q}_1^2 \\ v_2^2 &= \dot{q}_1^2 + \dot{q}_2^2 \\ v_3^2 &= \dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2. \end{aligned} \quad (56)$$

Replacing the values on v_1^2 , v_2^2 and v_3^2 in (53) we obtain the kinetic energy of each link,

$$\begin{aligned}
\mathcal{K}_1(q, \dot{q}) &= \frac{m_1 \dot{q}_1^2}{2} \\
\mathcal{K}_2(q, \dot{q}) &= \frac{m_2(\dot{q}_1^2 + \dot{q}_2^2)}{2} \\
\mathcal{K}_3(q, \dot{q}) &= \frac{m_3(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2)}{2}.
\end{aligned} \tag{57}$$

Adding the kinetic energy of each link,

$$\mathcal{K}(q, \dot{q}) = \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2(\dot{q}_1^2 + \dot{q}_2^2)}{2} + \frac{m_3(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2)}{2}, \tag{58}$$

we can expand the equation by the following form:

$$\mathcal{K}(q, \dot{q}) = \frac{m_1 \dot{q}_1^2}{2} + \frac{m_2 \dot{q}_1^2}{2} + \frac{m_2 \dot{q}_2^2}{2} + \frac{m_3 \dot{q}_1^2}{2} + \frac{m_3 \dot{q}_2^2}{2} + \frac{m_3 \dot{q}_3^2}{2}, \tag{59}$$

obtaining the total kinetic energy on the robot when grouping terms:

$$\mathcal{K}(q, \dot{q}) = \frac{(m_1 + m_2 + m_3)}{2} \dot{q}_1^2 + \frac{(m_2 + m_3)}{2} \dot{q}_2^2 + \frac{m_3}{2} \dot{q}_3^2. \tag{60}$$

The potential energy $\mathcal{U}(q)$ is obtained considering in this case $h = q_3$ and $m = (m_1 + m_2 + m_3)$:

$$\mathcal{U}(q) = (m_1 + m_2 + m_3)gq_3. \tag{61}$$

After calculating the potential and kinetic energy on the robot we calculated the Lagrangian using (27):

$$\mathcal{L}(q, \dot{q}) = \frac{(m)}{2} \dot{q}_1^2 + \frac{(m_2 + m_3)}{2} \dot{q}_2^2 + \frac{m_3 \dot{q}_3^2}{2} - (m_1 + m_2 + m_3)gq_3. \tag{62}$$

When we used the obtained representation of the Lagrangian (27), we solve part by part the Euler-Lagrange equation for a conservativo system (26), we begin to solve the partial derived one from the Lagrangian with respect to the joint velocity \dot{q} :

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & (m_2 + m_3) & 0 \\ 0 & 0 & (m_1 + m_2 + m_3) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}; \tag{63}$$

We continued with the derived in (63) with respect time:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & (m_2 + m_3) & 0 \\ 0 & 0 & (m_1 + m_2 + m_3) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix}, \tag{64}$$

and finally the independent term is solved:

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \begin{bmatrix} 0 \\ 0 \\ (m_1 + m_2 + m_3)g \end{bmatrix}. \tag{65}$$

Thus, the dynamic model of the cartesian robot is:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_1 + m_2 + m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 + m_2 + m_3 \end{bmatrix} g \quad (66)$$

being τ_1 , τ_2 and τ_3 the applied torques. As we can observe, the dynamic model represented in (66) it is not under a friction influence.

Considering the following physical parameters:

Description	Notation	Value	Units
Link mass 1	m_1	16.180	kg
Link mass 2	m_2	14.562	kg
Link mass 3	m_3	12.944	kg
Gravity acceleration	g	9.8100	$\frac{m}{s^2}$

Table 1. Physical parameters on the three degrees of freedom cartesian robot

we can describe the dynamic model of the robot of three degrees of freedom by following:

$$M(q) = \begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.742 & 0 \\ 0 & 0 & 43.686 \end{bmatrix} \quad (67)$$

$$g(q) = \begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix}$$

As it is observed in a cartesian robot the presence of the Coriolis and centripetal forces matrix $C(q, \dot{q})$ does not exist.

4. Hamilton's equations

Elegant and powerful methods have also been devised for solving dynamic problems with constraints. One of the best known is called Lagrange's equations, equation (26), where the Lagrangian $\mathcal{L}(q, \dot{q})$ is defined in (27). There is even a more powerful method called *Hamilton's equations*. It begins by defining a generalized momentum ρ , which is related to the Lagrangian $\mathcal{L}(q, \dot{q})$ and the generalized velocity \dot{q} by:

$$\rho = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \quad (68)$$

A new function, the Hamiltonian $\mathcal{H}(q, \rho)$, is defined by the addition of kinetic and potential energy:

$$\mathcal{H}(q, \rho) = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q). \quad (69)$$

From this point it is not difficult to derive

$$\dot{q} = \frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \quad (70)$$

and

$$\dot{\rho} = \tau - \frac{\partial \mathcal{H}(q, \rho)}{\partial q} \quad (71)$$

These are called *Hamilton's equations*. There are two of them for each generalized coordinates. They may be used in place of Lagrange's equations, with the advantage that only the first derivatives not the second ones are involved.

Proof. In order to verify the obtaining of Hamilton's equations, the procedure begins by solving the first element on the equation:

$$\underbrace{\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right]}_{\text{first element}} - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau. \quad (72)$$

In order to solve this part in the equation, we consider the Lagrangian $\mathcal{L}(q, \dot{q})$ as the difference between the kinetic $\mathcal{K}(q, \dot{q})$ and potential $\mathcal{U}(q)$ energy, equation (27); and we substitute it in the equation (72):

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \dot{q}} - \frac{\partial \mathcal{U}(q)}{\partial \dot{q}}. \quad (73)$$

It is observed when we solve the partial derivation, the term which contains the potential energy $\mathcal{U}(q)$ is eliminated:

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \dot{q}} - \cancel{\frac{\partial \mathcal{U}(q)}{\partial \dot{q}}} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \dot{q}} \quad (74)$$

and considering that kinetic energy $\mathcal{K}(q, \dot{q})$ is defined as:

$$\mathcal{K}(q, \dot{q}) = \frac{\dot{q}^T M(q) \dot{q}}{2} = \frac{\rho^T M^{-1}(q) \rho}{2} \quad (75)$$

we can represent and solve the equation as follows:

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}} \left(\frac{\dot{q}^T M(q) \dot{q}}{2} \right) = M(q) \dot{q} = \rho \quad (76)$$

where ρ is the momentum. Finally, deriving (76) we have:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] = M(q) \ddot{q} + \dot{M}(q) \dot{q} = \dot{\rho}. \quad (77)$$

Substituting the equation (77) in the equation (72) we have:

$$\dot{\rho} - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau. \quad (78)$$

Now, in order to solve the second part of the equation (72) we need to consider the possible relationships between $\dot{\rho}$ and the energies as follows:

$$\dot{\rho} = \tau + \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau - \left(\frac{\partial \mathcal{U}(q)}{\partial q} \right) = \tau - \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right) \quad (79)$$

this allows us to represent the equation (72) in the way:

$$\boxed{\dot{\rho} + \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right) = \tau} \quad (80)$$

where $\mathcal{H}(q, \rho)$ is the Hamiltonian which represents the total energy of the system and it is defined as the sum of the kinetic $\mathcal{K}(q, \dot{q})$ and potential $\mathcal{U}(q)$ energy, equation (69). It is assumed that the potential energy $\mathcal{U}(q)$ of the system is twice differentiable respect q and any entry of the Hessian of $\mathcal{U}(q)$, it is bounded for all q . This assumption is done for general manipulators. Now, if we evaluate the Hamiltonian $\mathcal{H}(q, \rho)$ using the partial derivation in function the momentum ρ , we can observe the potential energy is eliminated:

$$\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \rho} + \frac{\partial \mathcal{U}(q)}{\partial \rho} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \rho} + \cancel{\frac{\partial \mathcal{U}(q)}{\partial \rho}} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \rho} \quad (81)$$

and considering the form on the kinetic energy $\mathcal{K}(q, \dot{q})$, equation (75), we'll get:

$$\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} = \frac{\partial \mathcal{K}(q, \dot{q})}{\partial \rho} = \frac{\partial}{\partial \rho} \left(\frac{\rho^T M^{-1}(q) \rho}{2} \right) = M^{-1}(q) \rho = \dot{q} \quad (82)$$

Until now, we have obtained the equations (79) and (82), these equations are called as *Hamiltonian's equations of motion*:

$$\dot{\rho} = \tau - \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right) \quad (83)$$

$$\dot{q} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right) \quad (84)$$

□

4.1 Work and energy Principle

When forces act on a mechanism, *work* (in technical sense) is accomplish if the mechanism moves through a displacement. Work \mathcal{W} is defined as force acting through a distance and it is a scalar with units of energy.

In order to understand and apply the concept about work \mathcal{W} , we must start from the physics point of view. In physics, many forces are the function of an appear position. Two examples are gravitational and the electrical forces. The one-dimensional movement equation which describes a body under the action of a force that depends on position q is:

$$F(q) = m \frac{d\dot{q}}{dt} \quad (85)$$

Integral on a force F which depends on position q , the realized force which represents the work \mathcal{W} on a body while this one moves from starting point q_0 to the final q :

$$\mathcal{W} = \int_{q_0}^q F(q) dq. \quad (86)$$

The integral on the work \mathcal{W} can be always found, in an explicit or numerical way, in both cases we can define a function $\mathcal{U}(q)$:

$$\mathcal{U}(q) = - \int_{q_0}^q F(q) dq \quad (87)$$

like:

$$F(q) = - \frac{d\mathcal{U}(q)}{dq}. \quad (88)$$

This allows us to express the work \mathcal{W} meaning the difference between $\mathcal{U}(q)$ value in the external points:

$$\mathcal{W} = \int_{q_0}^q F(q) dq = -\mathcal{U}(q) \Big|_{q_0}^q = -\mathcal{U}(q) + \mathcal{U}(q_0) \quad (89)$$

Function $\mathcal{U}(q)$ is the potential energy which has the body when it is placed on point q . Up to this moment we have found that: work \mathcal{W} carried out by force $F(q)$ is equal to the difference between initial and final value on potential energy $\mathcal{U}(q)$ of the body.

Starting off in (86), we can observe that the value of the force $F(q)$ can be replace by us on equation (85), as follows:

$$\mathcal{W} = \int_{q_0}^q F(q) dq = \int_{q_0}^q m \frac{d\dot{q}}{dt} dq \quad (90)$$

Considering velocity definition \dot{q} :

$$\dot{q} = \frac{dq}{dt} \quad (91)$$

We can realize a change on variable based on dq

$$dq = \dot{q} dt, \quad (92)$$

thus we have:

$$\mathcal{W} = \int_{q_0}^q F(q) dq = \int_{q_0}^q m \frac{d\dot{q}}{dt} \dot{q} dt = \int_{t_0}^t m \dot{q} d\dot{q} = m \int_{t_0}^t \dot{q} d\dot{q} \quad (93)$$

Solving the integral, we obtain:

$$\mathcal{W} = m \int_{t_0}^t \dot{q} d\dot{q} = \frac{1}{2} m \dot{q}^2 \Big|_{t_0}^t = \frac{1}{2} m \dot{q}^2(t) - \frac{1}{2} m \dot{q}^2(t_0) \quad (94)$$

As it is observed in (94), when it is solved the integral, appears the kinetic energy of the body. It is well known the unit of kinetic energy $\mathcal{K}(q, \dot{q})$ in MKS system equal to work \mathcal{W} , that is the Joule. Until this moment we have found for any force $F(q)$

$$\mathcal{W} = \int_{q_0}^q F(q) dq = -\mathcal{U}(q) + \mathcal{U}(q_0) = \mathcal{K}(q, \dot{q}) - \mathcal{K}(q_0, \dot{q}_0) \quad (95)$$

and using (89) and (94) it is possible to express as:

$$\mathcal{W} = -\mathcal{U}(q) + \mathcal{U}(q_0) = \mathcal{K}(q, \dot{q}) - \mathcal{K}(q_0, \dot{q}_0), \quad (96)$$

and putting together all the equation elements we have:

$$\mathcal{W} = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q) = \mathcal{U}(q_0) + \mathcal{K}(q_0, \dot{q}_0) \quad (97)$$

(97) expresses *the total conservation of energy principle* $\mathcal{E}(q, \dot{q})$.

$$\mathcal{W} = \underbrace{\mathcal{K}(q, \dot{q}) + \mathcal{U}(q)}_{\mathcal{E}(q, \dot{q})} = \mathcal{U}(q_0) + \mathcal{K}(q_0, \dot{q}_0) \quad (98)$$

This principle is applicable for any one-dimensional problem where the force just represents function of position q . Equation (98) also receives the name of *the work and energy principle*, and establishes the work \mathcal{W} carried out by a force $F(q)$ on a body is equal to the change or variation of its kinetic energy $\mathcal{K}(q, \dot{q})$.

One must observe that the function on defined potential energy $\mathcal{U}(q)$ in (87) is indefinite by a constant value, constant integration. Nevertheless, this has no matter, since in any application it will only seem the difference of potential energies, equation (89). It is important to remember this, because it will allow us to choose arbitrarily the point where the body has potential energy zero $\mathcal{U}(q) = 0$. In addition, it will allow us, at any time, being able to do it, adding in all points the same constant amount to the potential energy $\mathcal{U}(q)$ of a body without affecting the results.

4.2 Principle of energy

The total energy of the system $\mathcal{E}(q, \dot{q})$ expressed in (98) can be considered like *the Hamiltonian* $\mathcal{H}(q, \rho)$,

$$\mathcal{W} = \mathcal{E}(q, \dot{q}) = \underbrace{\mathcal{K}(q, \dot{q}) + \mathcal{U}(q)}_{\mathcal{H}(q, \rho)} \quad (99)$$

and its derived can be considered like the *power* \mathcal{P} . Power \mathcal{P} can be interpreted like the work velocity or the work carried out by a time unit; and it is defined as follow:

$$\mathcal{P} = \frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}(\dot{q}, \rho)}{dt} \quad (100)$$

From a mechanical point of view, the power (mechanical power) is the transmitted force by means on the associated mechanical element or by means on the contact forces. The simplest case is that a variable force acts in a free particle. According to the classic dynamics, this power is used by some variation from its kinetic energy $\mathcal{K}(q, \dot{q})$ or carried out by a time unit. Whereas in mechanical systems more complex like rotating elements on a constant axis, and where the moment of inertia I remains constant, the mechanical power can be related to the engine torque or torque applied τ , and the joint velocity \dot{q} , being the variation power of the angular kinetic energy by time unit; in case of expressed vectorial systems, thus we have:

$$\mathcal{P} = \frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}(\dot{q}, \rho)}{dt} = \tau^T \dot{q}. \quad (101)$$

where $\tau \in \mathbb{R}^{n \times 1}$ represents the vector of forces and torques at the end-effector; and $\dot{q} \in \mathbb{R}^{n \times 1}$ is a joint velocity.

Proof. We begin differentiating (100); we can observed that the following balance energy immediately appears:

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right)^T \dot{q} + \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \dot{\rho} \quad (102)$$

Substituting the joint velocity \dot{q} from equation (84) in (102) we have:

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right)^T \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right) + \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \dot{\rho}. \quad (103)$$

Applying the matrix property $x^T y = y^T x$ we get:

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right) + \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \dot{\rho}. \quad (104)$$

The factorization of the equation (104):

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \left[\left(\frac{\partial \mathcal{H}(q, \rho)}{\partial q} \right) + \dot{\rho} \right] \quad (105)$$

this allow us to substitute the equation (80) in the equation (105)

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right)^T \tau. \quad (106)$$

Applying the matrix property $x^T y = y^T x$ we get:

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \tau^T \left(\frac{\partial \mathcal{H}(q, \rho)}{\partial \rho} \right) \quad (107)$$

and substituting the joint velocity \dot{q} from equation (84) in (107) we obtain:

$$\frac{d\mathcal{H}(q, \rho)}{dt} = \tau^T \dot{q} \quad (108)$$

□

5. Cartesian space

The joint space is analyzed because it offers mathematical bases for the cartesian space. Cartesian space gives advantages of interpretation to the end-user, and for him is easier to locate the cartesian coordinates (x, y, z) which joint displacements (q_1, q_2, \dots, q_n) ; that is, for the final user it is intuitive to understand the space location of a body expressed in cartesian coordinates; so it is important to describe the characteristics and properties of the cartesian space. The analysis of the cartesian space leaving of the joint space begins by considering the inverse kinematics, which are one of the basic functions for control systems robot manipulators. Inverse kinematics is the process which determines the joint parameters of a based object on the cartesian position which is described as a function f on the joint variable q :

$$x = f(q). \quad (109)$$

In order to solve the inverse problem in (109) it is necessary to determine q using a partial derivation as follow:

$$\dot{x} = J(q)\dot{q}, \quad (110)$$

where $J(q)$ is the Jacobian matrix, \dot{q} is the joint velocity; and \dot{x} is the cartesian velocity. The equation (110) allows us to obtain the joint velocity representation as follow:

$$\dot{q} = J(q)^{-1}\dot{x} \quad (111)$$

After some operations, we can relate the joint space with cartesian space using some equations, Table 2.

Joint space	Cartesian space
$\dot{q} = J(q)^{-1}\dot{x}$	$\dot{x} = J(q)\dot{q}$
$\ddot{q} = J(q)^{-1}\ddot{x} - J(q)^{-1}\dot{J}(q)J(q)^{-1}\dot{x}$	$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$

Table 2. Equations which relate both workspaces.

Partial derivation on the inverse kinematic model establishes a relationship between the joint and cartesian velocity. The inverse Jacobian matrix obtained will be used for study on singular positions of the robot manipulator, for evaluation of its maneuverability and also for optimization of its architecture.

The forward kinematics model, equation (109), provides the relationships to determine cartesian and joint position on the end-effector given by the joint position. As it is observed in the equations where they relate the workspaces, Table 2, the Jacobian matrix $J(q)$ appears.

5.1 Jacobian transpose controller

In order to define the cartesian space for control proposes is required the dynamic model in joint space, equation (33); the equations which relate the work spaces, Table 1; and a new scheme of control known *Jacobian transpose controller*. In 1981 Suguru Arimoto and Morikazu Takegaki members of the mechanical engineering department in the Osaka University in Japan, they published in the *Journal of Dynamic Systems, Measurement and Control* Vol. 103 a new control scheme based on the Jacobian Transpose matrix; the conservation energy idea; the principle of virtual works and generalized force; and the static equilibrium. Jacobian transpose method removes the problematic from the Jacobian inversion mentioned above. The annoying inversion is replaced by a simple transposition. This control scheme was used for stability proof in the PD controller in global way, this was the first stability proof in the PD controller. This proposal changed the point of view from the control theory because it avoided singularities, doing as possible the robot manipulator all desired positions inside its workspace. It is well known, the applied torque and cartesian force satisfies:

$$\tau = J(q)^T \mathcal{F} \quad (112)$$

where $\tau \in \mathbb{R}^{n \times 1}$ is the vector of applied torques, $J(q) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix and $\mathcal{F} \in \mathbb{R}^{n \times 1}$ is the vector from the applied force at the end-effector in cartesian space. The equation (112) is called *Jacobian transpose controller*. The external force \mathcal{F} is applied to the end-effector on the articulated structure and results in internal forces and torques in joints.

Proof. The *principle of energy* allows us to make certain statements about the static case by allowing the amount of this displacement to go to an infinitesimal. From the physical point of view, it is well known that the work has units of energy, this must be the same measured in any set of generalized coordinates, this allows us to describe the power \mathcal{P} as follow:

$$\frac{d\mathcal{W}}{dt} = \frac{d\mathcal{H}(\dot{q}, \rho)}{dt} = \tau^T \dot{q} \quad (113)$$

Specifically, we can equate the work done in cartesian terms with the work done in joint space terms.

In the multidimensional case, work \mathcal{W} is the dot product of a vector force or torque and the vector displacement. Thus we have:

$$\frac{d\mathcal{W}}{dt} = \mathcal{F}^T \dot{x}, \quad (114)$$

a necessary condition to satisfy the static equilibrium:

$$\mathcal{F}^T \dot{x} = \tau^T \dot{q} \quad (115)$$

where $\mathcal{F} \in \mathbb{R}^{n \times 1}$ represents the vector of forces and torques at the end-effector in cartesian coordinates; $\dot{x} \in \mathbb{R}^{n \times 1}$ is a cartesian velocity; $\tau \in \mathbb{R}^{n \times 1}$ is a vector of torque; and $\dot{q} \in \mathbb{R}^{n \times 1}$ is the joint velocity. Finally, let's \dot{q} represent the corresponding joint velocity. These velocity are related through the Jacobian matrix $J(q)$ according to equation (110):

$$\mathcal{F}^T J(q) \dot{q} = \tau^T \dot{q} \quad (116)$$

The *virtual work* of the system is defined as:

$$\mathcal{F}^T J(q) \dot{q} - \tau^T \dot{q} = 0, \quad (117)$$

this is equal to zero if the manipulator is in equilibrium. Factorizing the equation (117) we have:

$$\left(\mathcal{F}^T J(q) - \tau^T \right) \dot{q} = 0. \quad (118)$$

If we analyzed the equation (118) we can determine that the system is equal to zero, this assumption let us make the following equality:

$$\mathcal{F}^T J(q) - \tau^T = 0. \quad (119)$$

Applying the property $x^T y = y^T x$

$$J(q)^T \mathcal{F} - \left(\tau^T \right)^T = 0 \quad (120)$$

and the property $(x^T)^T = x$ in the equation (120) we have:

$$J(q)^T \mathcal{F} - \tau = 0 \quad (121)$$

and obtaining the applied torque τ we have:

$$\tau = J(q)^T \mathcal{F} \quad (122)$$

where $\tau \in \mathbb{R}^{n \times 1}$ is the applied torque; $\mathcal{F} \in \mathbb{R}^{n \times 1}$ represent the vector of forces and torques at the end-effector in cartesian coordinates; and $J(q) \in \mathbb{R}^{n \times n}$ is the Jacobian matrix on the system. In other words the end-effector forces are related to joint torques by the Jacobian transpose matrix according to (122).

□

5.2 Dynamic model based-on the Jacobian transpose controller

In 1981 Suguru Arimoto and Morikazu Takegaki members of the mechanical engineering department in the Osaka University in Japan, they published in the Journal of Dynamic Systems, Measurement and Control Vol. 103 a new scheme control based on the Jacobian Transpose matrix (122); the energy conservation idea; the principle of virtual works and generalized force; and the static equilibrium. Jacobian transpose method removes the problematic of the Jacobian inversion and the singularity problem. Suguru Arimoto and Morikazu Takegaki substituted the Jacobian transpose controller, equation (122), in the dynamic model, equation (33),

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = J(q)^T \mathcal{F} \quad (123)$$

and using the equations described in Table 1, we obtain:

$$M(x)\ddot{x} + C(x, \dot{x})\dot{x} + g(x) = \tau_x, \quad (124)$$

where:

$$M(x) = J(q)^{-T} M(q) J(q)^{-1} \quad (125)$$

$$C(x, \dot{x}) = J(q)^{-T} C(q, \dot{q}) J(q)^{-1} - J(q)^{-T} M(q) J(q)^{-1} \dot{J}(q) J(q)^{-1} \quad (126)$$

$$g(x) = J(q)^{-T} g(q) \quad (127)$$

$$\tau_x = \mathcal{F} \quad (128)$$

we obtained a dynamic model representation on Jacobian transpose terms.

5.2.1 Properties

Although the equation of motion (124) is complex, it has several fundamental properties which can be exploited to facilitate a control system design. We use the following important properties. In order to establish the properties on the described dynamic model in cartesian space it is necessary to do the following assumptions:

Assumption 1. Jacobian matrix does exist, $J(q) \exists$; is continuously differentiable respecting each entry of q , $J(q) \in \mathcal{C}^k$; and it is considered that is of a full rank:

$$\text{rank} \{J(q)\} = n.$$

This assumption is required for technical reason in stability analysis inside the workspace in cartesian space.

Assumption 2. According to the assumption 1, Jacobian matrix has a full rank, this consideration indicates that its inverse representation does exist,

$$\text{If } \text{rank} \{J(q)\} = n \text{ then } J(q)^{-1} \exists.$$

This assumption indicates the existence of the Jacobian matrix and its inverse within the workspace Ω .

Assumption 3. According to the assumption 1, Jacobian matrix is continuously differentiable respecting each entry of q , this consideration indicates that its derivative representation does exist,

$$\text{If rank } \{J(q)\} = n \text{ then } \dot{J}(q) \exists.$$

This assumption indicates the existence on the derivative representation of the Jacobian matrix within the workspace Ω .

Assumption 4. If the Jacobiana matrix does exist, its transpose does exist,

$$\text{If } J(q) \exists \text{ then } J(q)^T \exists.$$

This assumption indicates the existence of the transpose representation of the Jacobian matrix within the workspace Ω .

Assumption 5. According to the assumption 1, 2 and 4, the Jacobiana transpose matrix does exist, and its inverse does exist,

$$\text{If } J(q)^T \exists \text{ then } J(q)^{-T} \exists.$$

This assumption indicates the existence of the inverse transpose representation of the Jacobian matrix within the workspace Ω .

Assumption 6. According to assumptions 1, 2 and 5; and considering the definition of an inertial matrix $M(x)$, equation (125), we can say inertial matrix $M(x)$ does exist,

$$\text{If } J(q)^{-1} \exists, J(q)^{-T} \exists \text{ and } M(q) \exists \text{ then } M(x) \exists.$$

This assumption indicates the existence on the inertial matrix $M(x)$ within the workspace Ω . Obviously, matrix $M(q)$ must exists.

Assumption 7. According to assumption 6 the inertial matrix $M(x)$ does exist, then according to assumption 1 its inverse does exist,

$$M(x) \exists \quad \text{and} \quad M(x)^{-1} \exists.$$

Assumption 8. Matrix $M(x)$ does exist, and it is symmetric,

$$M(x) \exists \quad \text{and} \quad M(x) = M(x)^T \exists.$$

Proof. In order to verify this assumption it is necessary to consider the definition of matrix $M(x)$, equation (125), and transposing the matrix,

$$M(x)^T = \left(J(q)^{-T} M(q) J(q)^{-1} \right)^T \quad (129)$$

Applying the formula $(xyz)^T = z^T y^T x^T$ we have:

$$M(x)^T = \left(J(q)^{-1} \right)^T (M(q))^T \left(J(q)^{-T} \right)^T \quad (130)$$

$$M(x)^T = J(q)^{-T} M(q)^T J(q)^{-1}$$

Matrix $M(q)$ is symmetrical¹, this allows us to represent (130) the following form:

$$M(x)^T = J(q)^{-T} M(q) J(q)^{-1} \quad (131)$$

We can conclude that the following equality is fulfilled:

$$M(x) = M(x)^T \quad (132)$$

□

Assumption 9. Considering that the matrix $J(q)$ does exist, assumption 1, and its inverse does exist $J(q)^{-1}$, assumption 2, when multiplying $J(q)J(q)^{-1}$ or $J(q)^{-1}J(q)$ we obtain the identity matrix I .

$$\text{If } J(q) \exists \text{ and } J(q)^{-1} \exists \text{ then } J(q)J(q)^{-1} = I_{JJ^{-1}}$$

$$\text{If } J(q)^{-1} \exists \text{ and } J(q) \exists \text{ then } J(q)^{-1}J(q) = I_{J^{-1}J}$$

We observed that obtained matrices I are equal,

$$I_{JJ^{-1}} = I_{J^{-1}J} = I$$

Assumption 10. Considering that the matrix $J(q)^T$ does exist, assumption 4, and its inverse does exist $J(q)^{-T}$, assumption 5, when multiplying $J(q)^TJ(q)^{-T}$ or $J(q)^{-T}J(q)^T$ we obtain the identity matrix I .

$$\text{If } J(q)^T \exists \text{ and } J(q)^{-T} \exists \text{ then } J(q)^TJ(q)^{-T} = I_{J^TJ^{-T}}$$

$$\text{If } J(q)^T \exists \text{ and } J(q)^{-T} \exists \text{ then } J(q)^{-T}J(q)^T = I_{J^{-T}J^T}$$

We observed that obtained matrices I are equal,

$$I_{J^TJ^{-T}} = I_{J^{-T}J^T} = I$$

5.2.1.1 Inertial matrix $M(x)$ properties

In accordance with assumption 6 the inertial matrix $M(x)$ exists, according to assumption 7 the inverse inertial matrix exists, and in reference about assumption 8 the inertial matrix $M(x)$ is symmetric. Another vital property of $M(x)$ is that it is bounded above and below. So,

$$\mu_1(x)I \leq M(x) \leq \mu_2(x)I \quad (133)$$

where I is the identity matrix, $\mu_1(x) \neq 0$ and $\mu_1(x)$ are constant scalars for a revolte arm and generally the function scalar of x for an arm containing prismatic joints.

¹ For more information, consult section 3.1.1.

5.2.1.2 Coriolis and centripetal terms $C(x, \dot{x})$ properties

The matrix $\dot{x}^T [\dot{M}(x) - 2C(x, \dot{x})] \dot{x} \equiv 0$ is skew-symmetric, so,

$$\dot{M}(x) = C(x, \dot{x}) + C(x, \dot{x})^T. \quad (134)$$

We need to keep in mind that the equality described in (134) can be written in the following form:

$$\dot{M}(x) - [C(x, \dot{x}) + C(x, \dot{x})^T] = 0 \quad (135)$$

Proof. Considering the definition on the inertia matrix $M(x)$, equation (125), and the Coriolis and centripetal terms $C(x, \dot{x})$, equation (126), both in cartesian space; we will verify the equation (134) is fulfilled. Therefore we initiated transposing the Coriolis matrix, thus we have:

$$C(x, \dot{x})^T = J(q)^{-T} C(q, \dot{q})^T J(q)^{-1} - J(q)^{-T} \dot{j}(q)^T J(q)^{-T} M(q) J(q)^{-1} \quad (136)$$

what it allows us to solve operation $C(x, \dot{x}) + C(x, \dot{x})^T$:

$$\begin{aligned} C(x, \dot{x}) + C(x, \dot{x})^T &= J(q)^{-T} C(q, \dot{q}) J(q)^{-1} - J(q)^{-T} M(q) J(q)^{-1} \dot{j}(q) J(q)^{-1} \\ &\quad + J(q)^{-T} C(q, \dot{q})^T J(q)^{-1} - J(q)^{-T} \dot{j}(q)^T J(q)^{-T} M(q) J(q)^{-1} \end{aligned} \quad (137)$$

As is observed, we can put together the following terms:

$$\begin{aligned} C(x, \dot{x}) + C(x, \dot{x})^T &= J(q)^{-T} C(q, \dot{q}) J(q)^{-1} - J(q)^{-T} M(q) J(q)^{-1} \dot{j}(q) J(q)^{-1} \\ &\quad + J(q)^{-T} C(q, \dot{q})^T J(q)^{-1} - J(q)^{-T} \dot{j}(q)^T J(q)^{-T} M(q) J(q)^{-1} \end{aligned} \quad (138)$$

Thus we have:

$$\begin{aligned} C(x, \dot{x}) + C(x, \dot{x})^T &= J(q)^{-T} [C(q, \dot{q}) + C(q, \dot{q})^T] J(q)^{-1} - J(q)^{-T} M(q) J(q)^{-1} \dot{j}(q) J(q)^{-1} \\ &\quad - J(q)^{-T} \dot{j}(q)^T J(q)^{-T} M(q) J(q)^{-1} \end{aligned} \quad (139)$$

Applying (46) we have:

$$\begin{aligned} C(x, \dot{x}) + C(x, \dot{x})^T &= J(q)^{-T} \dot{M}(q) J(q)^{-1} - J(q)^{-T} M(q) J(q)^{-1} \dot{j}(q) J(q)^{-1} \\ &\quad - J(q)^{-T} \dot{j}(q)^T J(q)^{-T} M(q) J(q)^{-1} \end{aligned} \quad (140)$$

Now, replacing (125) in (140),

$$\begin{aligned}
C(x, \dot{x}) + C(x, \dot{x})^T &= J(q)^{-T} \dot{M}(q) J(q)^{-1} - \underbrace{J(q)^{-T} M(q) J(q)^{-1}}_{M(x)} \dot{J}(q) J(q)^{-1} \\
&\quad - J(q)^{-T} \dot{J}(q)^T \underbrace{J(q)^{-T} M(q) J(q)^{-1}}_{M(x)}
\end{aligned} \tag{141}$$

thus we have:

$$\boxed{C(x, \dot{x}) + C(x, \dot{x})^T = J(q)^{-T} \dot{M}(q) J(q)^{-1} - M(x) \dot{J}(q) J(q)^{-1} - J(q)^{-T} \dot{J}(q)^T M(x)} \tag{142}$$

Equation (142) represents the first part on the proof.

The second step consists on deriving matrix $M(x)$ defined in (125), thus we have:

$$\dot{M}(x) = \dot{J}(q)^{-T} M(q) J(q)^{-1} + J(q)^{-T} \dot{M}(q) J(q)^{-1} + J(q)^{-T} M(q) \dot{J}(q)^{-1} \tag{143}$$

Using the equation (125), we can find $M(q)$ as follows:

$$\begin{aligned}
M(x) &= J(q)^{-T} M(q) J(q)^{-1} \\
J(q)^T M(x) &= M(q) J(q)^{-1} \\
J(q)^T M(x) J(q) &= M(q)
\end{aligned} \tag{144}$$

This allows us to replace $M(q)$ expressed in (144) in (143), as follows:

$$\begin{aligned}
\dot{M}(x) &= \dot{J}(q)^{-T} J(q)^T M(x) J(q) J(q)^{-1} + J(q)^{-T} \dot{M}(q) J(q)^{-1} \\
&\quad + J(q)^{-T} J(q)^T M(x) J(q) \dot{J}(q)^{-1}
\end{aligned} \tag{145}$$

Some terms can be eliminated applying the identity matrix property:

$$\begin{aligned}
\dot{M}(x) &= \dot{J}(q)^{-T} J(q)^T M(x) \underbrace{J(q) J(q)^{-1}}_I + J(q)^{-T} \dot{M}(q) J(q)^{-1} \\
&\quad + \underbrace{J(q)^{-T} J(q)^T}_I M(x) J(q) \dot{J}(q)^{-1}
\end{aligned} \tag{146}$$

thus we have:

$$\boxed{\dot{M}(x) = \dot{J}(q)^{-T} J(q)^T M(x) + J(q)^{-T} \dot{M}(q) J(q)^{-1} + M(x) J(q) \dot{J}(q)^{-1}} \tag{147}$$

Equation (147) represents the second part on the proof.

For the following step on the proof, we must consider next equation:

$$\dot{I}_{J^{-T}J^T}M(x) + M(x)\dot{I}_{JJ^{-1}} = 0 \quad (148)$$

where $\dot{I}_{J^{-T}J^T}$ and $\dot{I}_{JJ^{-1}}$ are derivative forms on following equations:

$$I_{J^{-T}J^T} = J(q)^T J(q)^{-T} \quad (149)$$

$$I_{JJ^{-1}} = J(q)J(q)^{-1}$$

thus we have:

$$\dot{I}_{J^{-T}J^T} = \frac{d\left(J(q)^{-T}J(q)^T\right)}{dt} = \dot{J}(q)^{-T}J(q)^T + J(q)^{-T}\dot{J}(q)^T = 0 \quad (150)$$

$$\dot{I}_{JJ^{-1}} = \frac{d\left(J(q)J(q)^{-1}\right)}{dt} = \dot{J}(q)J(q)^{-1} + J(q)\dot{J}(q)^{-1} = 0 \quad (151)$$

In (150) and (151) we are applying assumption 9 and 10. It is well known that derivation of identity matrix is equal to zero, $\dot{I} = 0$. Now, replacing (150) and (151) in 148 we get:

$$\underbrace{\left[\dot{J}(q)^{-T}J(q)^T + J(q)^{-T}\dot{J}(q)^T\right]}_{\frac{d\left(J(q)^{-T}J(q)^T\right)}{dt}}M(x) + M(x)\underbrace{\left[\dot{J}(q)J(q)^{-1} + J(q)\dot{J}(q)^{-1}\right]}_{\frac{d\left(J(q)J(q)^{-1}\right)}{dt}} = 0 \quad (152)$$

Solving internal operations we have:

$$\dot{J}(q)^{-T}J(q)^T M(x) + J(q)^{-T}\dot{J}(q)^T M(x) + M(x)J(q)\dot{J}(q)^{-1} + M(x)\dot{J}(q)J(q)^{-1} = 0 \quad (153)$$

Adding a zero on form:

$$J(q)^{-T}\dot{M}(q)J(q)^{-1} - J(q)^{-T}\dot{M}(q)J(q)^{-1} = 0 \quad (154)$$

thus we have:

$$\begin{aligned} &\dot{J}(q)^{-T}J(q)^T M(x) + J(q)^{-T}\dot{J}(q)^T M(x) + J(q)^{-T}\dot{M}(q)J(q)^{-1} \\ &+ M(x)J(q)\dot{J}(q)^{-1} + M(x)\dot{J}(q)J(q)^{-1} - J(q)^{-T}\dot{M}(q)J(q)^{-1} = 0 \end{aligned} \quad (155)$$

As it is observed, the equality is conserved. Ordering equation (155) we get:

$$\begin{aligned} &\dot{J}(q)^{-T}J(q)^T M(x) + J(q)^{-T}\dot{M}(q)J(q)^{-1} + M(x)J(q)\dot{J}(q)^{-1} \\ &- J(q)^{-T}\dot{M}(q)J(q)^{-1} + J(q)^{-T}\dot{J}(q)^T M(x) + M(x)\dot{J}(q)J(q)^{-1} = 0 \end{aligned} \quad (156)$$

Replacing (142) and (147) in (156),

$$\underbrace{\dot{j}(q)^{-T} J(q)^T M(x) + J(q)^{-T} \dot{M}(q) J(q)^{-1} + M(x) J(q) \dot{j}(q)^{-1}}_{\dot{M}(x)} - \underbrace{\left[J(q)^{-T} \dot{M}(q) J(q)^{-1} - J(q)^{-T} \dot{j}(q)^T M(x) - M(x) \dot{j}(q) J(q)^{-1} \right]}_{C(x, \dot{x}) + C(x, \dot{x})^T} = 0 \quad (157)$$

thus we have:

$$\dot{M}(x) - [C(x, \dot{x}) + C(x, \dot{x})^T] = 0. \quad (158)$$

Ordering (158) we have:

$$\dot{M}(x) = C(x, \dot{x}) + C(x, \dot{x})^T. \quad (159)$$

□

5.2.1.3 Gravity terms properties

The generalized gravitational forces vector

$$g(x) = \frac{\partial \mathcal{U}(x)}{\partial x} \quad (160)$$

satisfies:

$$\left\| \frac{\partial g(x)}{\partial x} \right\| \leq k_g \quad (161)$$

for some $k_g \in \mathbb{R}_+$, where $\mathcal{U}(x)$ is the potential energy expressed in the cartesian space and is supposed to be bounded from below.

5.2.2 Case of study: Dynamic model based-on the $J(q)^T$ on cartesian robot

Along the chapter, we have evaluated a three degrees of freedom cartesian robot and we have obtained several equations which are required to obtain the dynamic model based on the Jacobian transpose controller, equation (124), these matrices are:

- The Jacobian matrix $J(q)$ defined in (12), this matrix fulfills assumption 1:

$$J(q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The transpose representation on the Jacobian matrix defined in (14), this matrix fulfills assumption 1 and 4:

$$J(q)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The inverse representation on the Jacobian matrix is defined in (25), this matrix fulfills assumption 1 and 2:

$$J(q)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The inverse representation of the Jacobian transpose matrix, this matrix fulfills assumption 1 and 5:

$$J(q)^{-T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The derivative representation on the Jacobian matrix, this matrix fulfills assumptions 1 and 3:

$$\dot{J}(q) = \frac{d}{dt} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

To obtain the defined dynamic model in (124) last set of matrices is needed, thus we have, for the inertial matrix $M(x)$ defined in (125):

$$M(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \quad (162)$$

Solving (162) we obtain:

$$M(x) = \begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}. \quad (163)$$

For the Coriolis and centripetal matrix $C(x, \dot{x})$ defined in (126) thus we have:

$$C(x, \dot{x}) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{C(q, \dot{q})} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \\ - \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\dot{J}(q)} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-1}} \quad (164)$$

Solving (164) we obtain:

$$C(x, \dot{x}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (165)$$

As it is observed by obtaining from matrix $C(x, \dot{x})$, in a cartesian robot rotation behavior does not exist, thus the matrix of Coriolis does not exist either.

For the gravity term $g(x)$ defined in (127) thus we have:

$$g(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(q)^{-T}} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix}}_{g(q)} \quad (166)$$

Solving (166) we obtain:

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix} \quad (167)$$

Now we have the dynamic model based on Jacobian transpose controller is defined as:

$$\underbrace{\begin{bmatrix} 16.180 & 0 & 0 \\ 0 & 30.472 & 0 \\ 0 & 0 & 43.686 \end{bmatrix}}_{M(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 43.686 \end{bmatrix}}_{g(x)} = \underbrace{\begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \mathcal{F}_3 \end{bmatrix}}_{\tau_x} \quad (168)$$

5.3 Cartesian controllers

In this section we present our main results concerning about stability analysis on the cartesian controllers. Now we are in position to formulate a cartesian control problem. Typically we propose controllers using the *energy shaping* on joint coordinates, now we use this methodology on cartesian space.

5.3.1 Energy shaping on cartesian space

The *energy shaping* is a technique which allows to design the control algorithms using kinetic and artificial potential energy which is shaping via a gradient for stabilization at the equilibrium point and damping injection to make this equilibrium attractive. The designed control algorithm is composed by the gradient on the artificial potential energy plus a velocity feedback. We use the following cartesian control scheme:

$$\tau_x = \nabla \mathcal{U}(k_p, \tilde{x}) - f_v(k_v, \dot{x}) + g(x) \quad (169)$$

where $\mathcal{U}(k_p, \tilde{x})$ is the *artificial potential energy* described by:

$$\mathcal{U}(k_p, \tilde{x}) = \frac{f(\tilde{x})^T k_p f(\tilde{x})}{2}, \quad (170)$$

and the term $f_v(k_v, \dot{x})$ is the derivative action. We use the following Lyapunov scheme:

$$V(\dot{x}, \bar{x}) = \frac{\dot{x}^T M(x) \dot{x}}{2} + \mathcal{U}(k_x, \bar{x}). \quad (171)$$

where $M(x)$ is a local definite function. The energy shaping methodology consists about finding a $\mathcal{U}(k_x, \bar{x})$ function to fulfill the next Lyapunov's conditions:

$$\begin{aligned} V(0,0) &= 0 & \forall \dot{x}, \bar{x} &= 0 \\ V(\dot{x}, \bar{x}) &> 0 & \forall \dot{x}, \bar{x} &\neq 0 \end{aligned} \quad (172)$$

and doing the derivation of the Lyapunov equation we get,

$$\dot{V}(\dot{x}, \bar{x}) = \dot{x}^T M(x) \ddot{x} + \frac{\dot{x}^T \dot{M}(x) \dot{x}}{2} - \frac{\partial \mathcal{U}(k_p, \bar{x})^T}{\partial \bar{x}} \dot{\bar{x}}, \quad (173)$$

fulfill the condition:

$$\dot{V}(\dot{x}, \bar{x}) \leq 0, \quad (174)$$

verify asymptotical stability with LaSalle theorem:

$$\dot{V}(\dot{x}, \bar{x}) < 0. \quad (175)$$

In general terms, when we consider the dynamic model on cartesian space, equation (124), together with control law (169), then the closed-loop system is locally stable and the positioning aim:

$$\lim_{t \rightarrow \infty} x(t) = x_d \quad \wedge \quad \lim_{t \rightarrow \infty} \dot{x}(t) = 0 \quad (176)$$

is achieved.

Proof. The closed-loop system equation obtained by combining the robot dynamic model on cartesian space, equation (124), and the control scheme (169), can be written as:

$$\frac{d}{dt} \begin{bmatrix} \bar{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\dot{x} \\ M(x)^{-1} [\tau_x - C(\dot{x}, x) \dot{x}] \end{bmatrix} \quad (177)$$

which is an autonomous differential equation, and the origin of state space is its unique equilibrium point, we need to keep in mind that the inverse representation of the inertial matrix $M(x)$ exists only if only the Jacobian matrix fulfills assumption 1. Considering the autonomous system:

$$\dot{x} = f(x), \quad (178)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is locally Lipschitz map in \mathbb{R}^n . Let x_e be an equilibrium point for $f(x_e) = 0$. Let $V: \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable, positive definite function such as $\dot{V}(x) \leq 0 \forall x \in \mathbb{R}^n$. Let $\Omega = \{\dot{x} \in \mathbb{R}^n | \dot{V}(x) = 0\}$ and suppose that no solution could stay identically in Ω , other than the trivial solution, then the origin is locally stable. In our case $f(x)$ is given by the closed-loop system equation (178), where $x = [\bar{x}, \dot{x}]^T \in \mathbb{R}^{2n}$. The origin of the space state is its unique equilibrium point for (178). Let $V: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable, positive definite function such as $\dot{V}(\bar{x}, \dot{x}) \leq 0 \forall \dot{x}, \bar{x} \in \mathbb{R}^n$. Let the region:

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^{2n} : \dot{V}(\tilde{x}, \dot{x}) = 0 \right\} \tag{179}$$

$$\Omega = \{ \tilde{x} \in \mathbb{R}^n, \dot{x} = 0 \in \mathbb{R}^n : \dot{V}(\tilde{x}, \dot{x}) = 0 \},$$

since $\dot{V}(\tilde{x}, \dot{x}) \leq 0 \in \Omega$, $V(\tilde{x}(t), \dot{x}(t))$ is a decreasing function of t . $V(\tilde{x}, \dot{x})$ is continuous on the compact set Ω , so it is bounded from below Ω .

For example, it satisfies $0 \leq V(\tilde{x}(t), \dot{x}(t)) \leq V(\tilde{x}(0), \dot{x}(0))$. Therefore, $V(\tilde{x}(t), \dot{x}(t))$ has limit α as $t \rightarrow \infty$. Hence $\dot{V}(\tilde{x}(t), \dot{x}(t)) = 0$ and the unique invariant is $\tilde{x} = 0$ and $\dot{x} = 0$. Since the trivial solution is the closed-loop system unique solution (178) restricted to Ω , then it is concluded that the origin of the state space is asymptotically stable in a local way.

□

The following block diagram describes the relationship between the robot manipulator on cartesian space dynamic model and the controller structure, specifying a position controller.

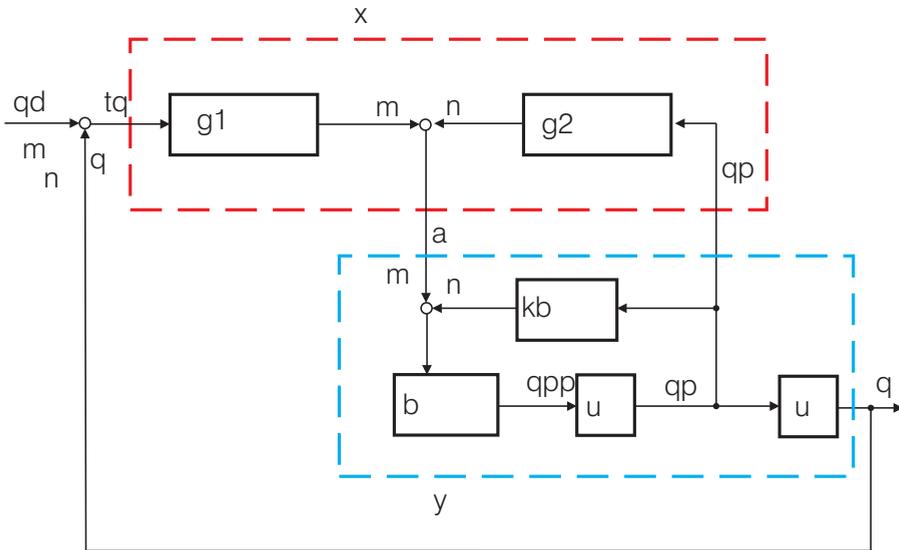


Fig. 6. Blocks of dynamic model and control scheme on cartesian space.

5.3.1.1 PD cartesian controller

In this section, we recall the stability proof of the simple PD cartesian controller which is given as:

$$\tau_x = K_p \tilde{x} - K_v \dot{x} + g(x) \tag{180}$$

where $\tilde{x} = x_d - x$ denotes the position error on cartesian coordinates, x_d is the desired position, and K_p and K_v are the proportional and derivative gains, respectively.

The control problem can be stated by selecting the design matrices K_p and K_v then the position error \tilde{x} vanishes asymptotically in a local way, i.e.

$$\lim_{t \rightarrow \infty} \tilde{x}(t) = 0 \in \mathbb{R}^n. \quad (181)$$

The closed-loop system equation obtained by combining the cartesian robot model, equation (124), and control scheme, equation (180), can be written as:

$$\frac{d}{dt} \begin{bmatrix} \tilde{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} -\dot{\tilde{x}} \\ M(x)^{-1} [K_p \tilde{x} - K_v \dot{\tilde{x}} - C(x, \dot{x}) \tilde{x}] \end{bmatrix} \quad (182)$$

with this is an autonomous differential equation and the origin of the state space is its unique equilibrium point. To accomplish the stability proof of equation (182), we proposed the following Lyapunov function candidate based on the energy shaping methodology oriented on cartesian space:

$$V(\dot{x}, \tilde{x}) = \frac{\dot{x}^T M(x) \dot{x}}{2} + \frac{\tilde{x}^T K_p \tilde{x}}{2}. \quad (183)$$

The first term of $V(\dot{x}, \tilde{x})$ is a positive definite function respecting to \dot{x} because $M(x)$ in the case of study is a positive definite matrix. The second one of Lyapunov function candidate (183) is a positive definite function respecting to position error \tilde{x} , because K_p is a positive definite matrix. Therefore $V(\dot{x}, \tilde{x})$ is a global positive definite and a radially unbounded function. The time derivative of Lyapunov function candidate (183) along the trajectories on the closed-loop (182),

$$\dot{V}(\dot{x}, \tilde{x}) = \dot{x}^T M(x) \ddot{x} + \frac{\dot{x}^T \dot{M}(x) \dot{x}}{2} + \tilde{x}^T K_p \dot{\tilde{x}} \quad (184)$$

and after some algebra and using the property of the Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{x}, \tilde{x}) = -\dot{x}^T K_v \dot{x} \leq 0 \quad (185)$$

which is a locally negative semi-definite function and therefore we conclude with stability on the equilibrium point.

In order to prove asymptotic stability in a local way, we exploit the autonomous nature of closed-loop (182) by applying the *La Salle's invariance principle*:

$$\dot{V}(\dot{x}, \tilde{x}) < 0. \quad (186)$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{\tilde{x}} \end{bmatrix} \in \mathbb{R}^n : V(\tilde{x}, \dot{\tilde{x}}) = 0 \right\} \quad (187)$$

the unique invariant is $[\tilde{x}^T \quad \dot{\tilde{x}}^T]^T = 0 \in \mathbb{R}^{2n}$.

5.3.1.2 A polynomial family of PD-type cartesian controller

This control structure is a control scheme in joint space generalization proposed in [Reyes & Rosado] and [Sánchez-Sánchez & Reyes-Cortés]. The family about proposed controllers with PD-type structure and its global asymptotic stability analysis. We intend to extend the results on the simple PD controller to a large class of polynomial PD-type controllers for robot manipulators on cartesian space. Considering the following control scheme with gravity compensation given by:

$$\tau_x = \sum_{j=1}^n \left(K_{p_{2j-1}} \tilde{x}^{2j-1} - K_{v_{2j-1}} \dot{\tilde{x}}^{2j-1} \right) + g(x) \quad (188)$$

where \tilde{x} denotes the position error on cartesian coordinates, x_d is the desired position, K_p and K_v are the propositional and derivative gains, respectively, and $2j - 1$ give the equation the polynomial characteristic. The closed-loop system equation obtained by combining the dynamic model on the robot manipulator on cartesian, equation (124), and the control scheme, equation (188), can be written as:

$$\frac{d}{dt} \begin{bmatrix} \tilde{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} -\dot{\tilde{x}} \\ M(x)^{-1} [\tau_x - C(x, \dot{x}) \dot{\tilde{x}}] \end{bmatrix} \quad (189)$$

where $\tau_x = \sum_{j=1}^n \left(K_{p_{2j-1}} \tilde{x}^{2j-1} - K_{v_{2j-1}} \dot{\tilde{x}}^{2j-1} \right)$, which is an autonomous differential equation and the origin of the state space is its unique equilibrium point. To analyze the existence of the equilibrium point we have evaluated \tilde{x} and $\dot{\tilde{x}}$ in the following way: For $I\dot{\tilde{x}} = 0 \Rightarrow \dot{\tilde{x}} = 0$, and $M(x)^{-1} K_{p_{2j-1}} \tilde{x}^{2j-1} = 0 \Rightarrow \tilde{x}^{2j-1} = 0 \Rightarrow \tilde{x} = 0$.

To make the proof of stability on the equation (189), we proposed the following Lyapunov function candidate based in the energy shaping methodology oriented on cartesian space:

$$V(\dot{\tilde{x}}, \tilde{x}) = \frac{\dot{\tilde{x}}^T M(x) \dot{\tilde{x}}}{2} + \frac{\sum_{j=1}^n K_{p_{2j-1}} \tilde{x}^{2j}}{2j}, \quad (190)$$

the first term of $V(\dot{\tilde{x}}, \tilde{x})$ is a positive define function respecting to $\dot{\tilde{x}}$ because $M(x)$ in the case of study is a positive definite matrix. The second one Lyapunov function candidate (190) is a positive definite function respecting to position error \tilde{x} , because K_p is a positive define matrix. Therefore $V(\dot{\tilde{x}}, \tilde{x})$ is a locally positive definite. The simple cartesian PD Controller is a particular case on the polynomial family of PD-type cartesian controller when $j = 1$. The time derivative of Lyapunov function candidate (190) along the trajectories of the closed-loop (189),

$$\dot{V}(\dot{\tilde{x}}, \tilde{x}) = \dot{\tilde{x}}^T M(x) \ddot{\tilde{x}} + \frac{\dot{\tilde{x}}^T \dot{M}(x) \dot{\tilde{x}}}{2} - \dot{\tilde{x}}^T \sum_{j=1}^n K_{p_{2j-1}} \tilde{x}^{2j-1} \quad (191)$$

after some algebra and using the property of the Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{\tilde{x}}, \tilde{x}) = -\dot{\tilde{x}}^T K_{v_{2j-1}} \dot{\tilde{x}}^{2j-1} \leq 0 \quad (192)$$

which is a locally negative semi-definite function and therefore we conclude stability on the equilibrium point. In order to prove asymptotic stability in a local way we exploit the autonomous nature of closed-loop (189) by applying the *La Salle's invariance principle*:

$$\dot{V}(\dot{x}, \bar{x}) < 0. \quad (193)$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \bar{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^n : V(\bar{x}, \dot{x}) = 0 \right\} \quad (194)$$

the unique invariant is $[\bar{x}^T \quad \dot{x}^T]^T = 0 \in \mathbb{R}^{2n}$. Since (192) is a locally negative semi-definite function in full state and the Lyapunov function (190) is a radially unbounded locally positive definite function, then it satisfies:

$$0 \leq V(\bar{x}(t), \dot{x}(t)) \leq V(\bar{x}(0), \dot{x}(0)) \quad (195)$$

the bounds for the position error are given by:

$$\begin{aligned} & \sum_{j=1}^n \lambda_{\min} \{K_{p_{2j-1}}\} \left\| \bar{x}^{2j-1}(t) \right\|^2 \\ & \leq \left\| \bar{x}(0) \right\|^2 \beta + \frac{1}{m} \sum_{j=1}^n \lambda_{\max} \{K_{p_{2j-1}}\} \left\| \bar{x}^{2j-1}(0) \right\|^2 \end{aligned} \quad (196)$$

$$\forall m \in \mathbb{Z}^+, t \geq 0$$

where $\lambda_{\min} \{K_{p_{2j-1}}\}$ and $\lambda_{\max} \{K_{p_{2j-1}}\}$ represent the smallest and largest eigenvalues on the diagonal matrix $K_{p_{2j-1}}$, respectively, for derivative gain bounds are:

$$\begin{aligned} & \sum_{j=1}^n \lambda_{\min} \{K_{v_{2j-1}}\} \left\| \dot{x}^{2j-1}(t) \right\|^2 \\ & \leq \left\| \dot{x}(0) \right\|^2 \beta + \frac{1}{m} \sum_{j=1}^n \lambda_{\max} \{K_{v_{2j-1}}\} \left\| \dot{x}^{2j-1}(0) \right\|^2 \end{aligned} \quad (197)$$

$$\forall m \in \mathbb{Z}^+, t \geq 0$$

where $\lambda_{\min} \{K_{v_{2j-1}}\}$ and $\lambda_{\max} \{K_{v_{2j-1}}\}$ represent the smallest and largest eigenvalues of the diagonal matrix $K_{v_{2j-1}}$, respectively, β is a positive constant, strictly speaking, boundlessness of the inertial matrix requires, generally, that all joints must be revolute:

$$\begin{aligned} & \beta \|\dot{x}\| \geq \|M(x)\dot{x}\| \quad \forall x, \dot{x} \in \mathbb{R}^n \\ & \beta \geq n \left(\max_{i,j,x} |M_{ij}(x)| \right) \end{aligned} \quad (198)$$

where M_{ij} are elements of $M(x)$.

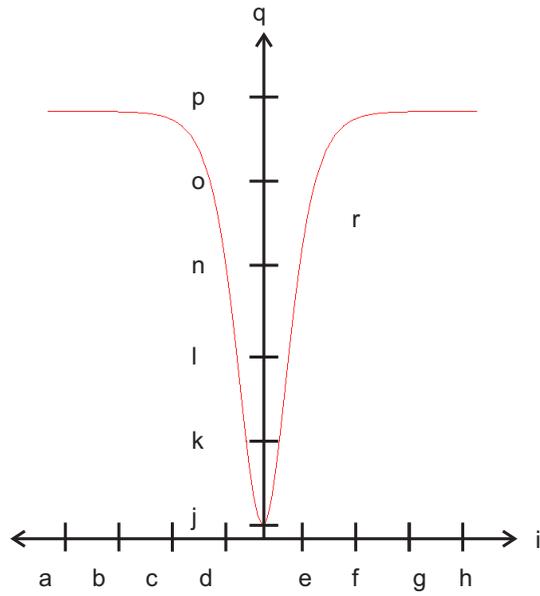


Fig. 7. terms within the radical.

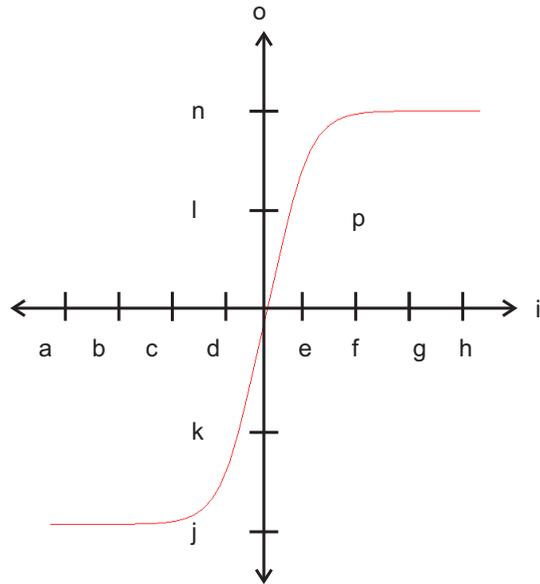


Fig. 8. Complete behavior.

$$V(\dot{x}, \tilde{x}) = \frac{\dot{x}^T M(x) \dot{x}}{2} + \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}^T K_p \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}, \quad (204)$$

the first term on $V(\dot{x}, \tilde{x})$ is a positive definite function respecting to \dot{x} because $M(x)$ in the case of study is a positive definite matrix. The second one of Lyapunov function candidate (204) is a positive definite function respecting to position error \tilde{x} , because K_p is a positive definite matrix. Therefore $V(\dot{x}, \tilde{x})$ is a locally positive definite. The time derivative of Lyapunov function candidate (204) along the trajectories of the closed-loop (200),

$$\dot{V}(\dot{x}, \tilde{x}) = \dot{x}^T M(x) \ddot{x} + \frac{\dot{x}^T \dot{M}(x) \dot{x}}{2} + \begin{bmatrix} \sqrt{\ln(\cosh(\tilde{x}_1))} \\ \sqrt{\ln(\cosh(\tilde{x}_2))} \\ \vdots \\ \sqrt{\ln(\cosh(\tilde{x}_n))} \end{bmatrix}^T K_p \begin{bmatrix} \tanh \tilde{x} \\ \sqrt{\ln(\cosh(\tilde{x}))} \end{bmatrix} \dot{\tilde{x}} \quad (205)$$

after some algebra and using the property of Coriolis and centripetal term described in section 5.2.1.2. it can be written as:

$$\dot{V}(\dot{x}, \tilde{x}) = -\dot{x}^T K_v \begin{bmatrix} \tanh(\dot{x}_1) \sqrt[2j]{1 + \tanh^{2j}(\dot{x}_1)} \\ \tanh(\dot{x}_2) \sqrt[2j]{1 + \tanh^{2j}(\dot{x}_2)} \\ \vdots \\ \tanh(\dot{x}_n) \sqrt[2j]{1 + \tanh^{2j}(\dot{x}_n)} \end{bmatrix} \leq 0. \quad (206)$$

which is a locally negative semi-definite function and therefore we conclude stability on the equilibrium point. In order to prove asymptotic stability in local way we exploit the autonomous nature of closed-loop (200) by applying the *LaSalle invariance principle*:

$$\dot{V}(\dot{x}, \tilde{x}) < 0. \quad (207)$$

In the region

$$\Omega = \left\{ \begin{bmatrix} \tilde{x} \\ \dot{x} \end{bmatrix} \in \mathbb{R}^n : V(\tilde{x}, \dot{x}) = 0 \right\} \quad (208)$$

the unique invariant is $[\tilde{x}^T \quad \dot{x}^T]^T = 0 \in \mathbb{R}^{2n}$.

5.4 Experimental Set-Up

We have designed and built an experimental system for researching on cartesian robot control algorithms and currently it is a turn key research system for developing and validation on cartesian control algorithms for robot manipulators. The experimental system is a servomotor robot manipulator with three degrees of freedom moving itself into a three dimensional space as it is shown in the Figure 5.

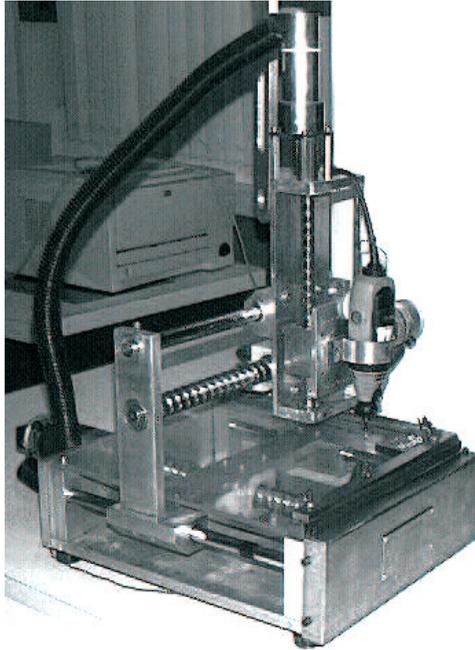


Fig. 9. Experimental Prototype. "DRILL-BOT"

The structure is made of stainless iron, direct-drive shaft with servomotors from Reliance Electronics[®]. Advantages in this kind of drive shaft includes a high torque. The servomotor has an Incremental Encoder from Hewlett Packard[®].

Motors used in the experimental system are **E450 model [450 oz-in.]**. Servos are operated in torque mode, so the motors act a reference if torque emits a signal information about position is obtained from incremental encoders located on the motors, which have a resolution of 1024000 *p/rev*.

5.4.1 Experimental Results

To support our theoretical developments, this section presents an experimental comparison between three position controllers on cartesian space by using an experimental system of three degrees of freedom. To investigate the performance among controllers, they have been classified as τ_{PD} for the simple PD controller; τ_{Poly} for the polynomial family of PD-type cartesian controller; and τ_{Pascal} represent the Pascal's cartesian controller, all the control structures on cartesian space. To analyze the controllers' behavior it is necessary to compare their performances. For this reason we have used the \mathcal{L}^2 norm; this norm is a scalar value. A \mathcal{L}^2 smaller represents minor position error and thus it is the better performance. A position control experiment has been designed to compare the performances of controllers on a cartesian robot. The experiment consists on moving the manipulator's end-effector from its initial position to a fixed desired target. To the present application the desired positions were chosen as:

$$\begin{bmatrix} x_{d_1} \\ y_{d_1} \\ z_{d_1} \end{bmatrix} = \begin{bmatrix} 0.785 \\ 0.615 \\ 0.349 \end{bmatrix} \quad (209)$$

where x_{d_1}, y_{d_1} and z_{d_1} are in meters and represent the x, y and z axes in the prototype. The initial positions and velocities were set to zero (for example a home position). The friction phenomena were not modeled for compensation purpose. That is, all the controllers did not show any type of friction compensation. We should keep in mind that the phenomenon of friction doesn't have a mathematical structure to be modeled. The evaluated controllers have been written in C language. The sampling rate was executed at 2.5 ms. For proposed controller family were used the gains showed in Table 3.

Parameter	Value
K_{p_1}	359.196
K_{v_1}	35.5960
K_{p_2}	4.85400
K_{v_2}	4.36860
K_{p_3}	22.6520
K_{v_3}	3.23600

Table 3. Gains used in the experiments

5.4.2 Performance index

Robot manipulator is a very complex mechanical system, due to the nonlinear and multivariable nature on the dynamic behavior. For this reason, in the robotics community there are not well-established criteria for a proper evaluation in controllers for robots. However, it is accepted in practice comparing performance of controllers by using the scalar-valued \mathcal{L}^2 norm as an objective numerical measure for an entire error curve. The performance index is used to measure \mathcal{L}^2 norm of the position error \tilde{x} . A small value in \mathcal{L}^2 represents a smaller error and therefore it indicates a better performance. A vectorial function $\mathbb{R}^n \rightarrow \mathbb{R}^n \in \mathcal{L}^2$, if when we evaluate:

$$\mathcal{L}^2 = \sqrt{\int_0^\infty \|f(x)\|^2 dx} < \infty \quad (210)$$

where $\|f(t)\|$ is the Euclidean norm of the function on the interval; it is a scalar number. This property in vectorial functions is a measure to determine the convergence while the time increases. As the simulation time is finite we must apply the concept of effective value to calculate the deviation in the function between the simulation intervals, thus we defined \mathcal{L}^2 norm on the form:

$$\mathcal{L}^2 = \sqrt{\frac{1}{T} \int_0^T \|\tilde{x}\|^2 dx}. \quad (211)$$

It is necessary to count on the discreet norm representation with the purpose of facilitate its implementation:

$$\int \|\tilde{x}\|^2 dx \rightarrow I_k = I_{k-1} + h \|\tilde{x}\|^2$$

$$\mathcal{L}^2 = \sqrt{\frac{1}{T} I_k} \quad (212)$$

where h is the period of sampling; and T is the evaluation interval. This is not the unique form to obtain the discrete integral representation, being applied the rule of the trapeze we can define the integral in an alternative form:

$$\int_0^T f(t) dt \rightarrow I_k = I_{k-1} \frac{T}{2} [f_k + f_{k-1}]. \quad (213)$$

In order to obtain the performance index of proposed controllers the following program in Matlab[®] receives data obtained in SIMNON[®] applying \mathcal{L}^2 norm.

```
% Platform : DRILL-BOT
% Program to evaluate controllers

% Load the files
load <archivo 1>.dat -ascii;
load <archivo 2>.dat -ascii;

% Time of the system
T=10;

% Reading of:
t =<archivo 1>(:,1);           %time
xt1=<archivo 1>(:,2);         %xtilde1
xt2=<archivo 2>(:,2);         %xtilde2

%.....integral.....
h=0.0025;
i=size(t);
ik(1)=0;

for j=2:i
    ik(j)=ik(j-1)+h*(xt1(j)*xt1(j)+xt2(j)*xt2(j));
end

%.....L2 norm .....
L=sqrt(ik(j)/T)
```

Results obtained by applying \mathcal{L}^2 norm are in Table 4.

The performance indices graph is observed in the Figure 10.

Overall results are summarized in Figure 10 which includes the performance indexes for analyzed controllers. To average stochastic influences, data presentation in this figure represents the meaning of root-mean-square position error vector norm of ten runs. For clarity, the data

Control structure	Performance index (rad)
PD cartesian control	0.2160
A polynomial family of PD-type	0.1804
Pascal's cartesian control	0.1618

Table 4. Performance index of the evaluated controllers

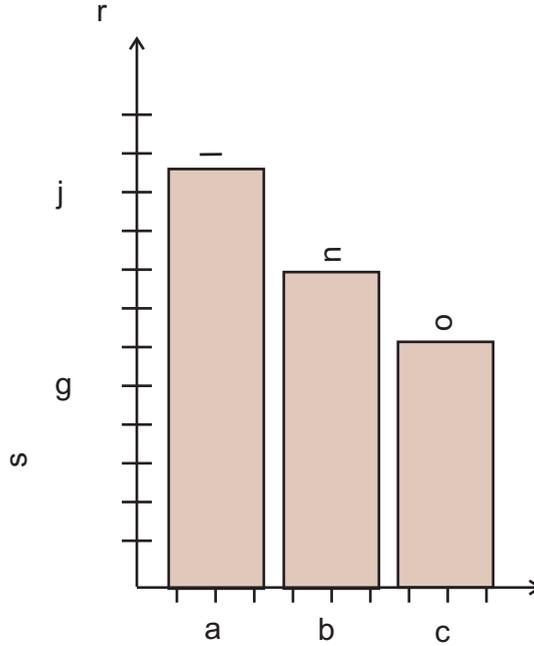


Fig. 10. Performance index of evaluated controllers

presented in Figure 10 are compared respecting to \mathcal{L}^2 norm of PD controller. The results from one run to another were observed to be less than 1% of their mean, which underscore the repeatability in the experiments. In general, the performance of the PD controller is improved by its counterpart.

5.4.2.1 Remarks

Through an analysis about obtained experimental data suggests the following results:

- Note that *A polynomial family of PD-type cartesian controller* and the *Pascal's cartesian controller* improves the performance obtained by the *PD cartesian controller*. The proposed controller families effectively exploits its exponential capability in order to enhance the position error, having a short transient phase and a small steady-state error. Fast convergence can be obtained (faster response). Consequently, the control performance is increased in comparison with the aforementioned controller.

- As it can be seen, the position error is bounded to increase the power those where the error signal is to be raised. However, for stability purposes, tuning procedure for the control schemes are sufficient to select a proportional and derivative gains as diagonal matrix, in order to ensure asymptotic stability in a local way.
- Nevertheless, in spite of the presence of friction, signals on position error are acceptably small for proposed families.

The problem about position control for robot manipulators could correspond to the configuration of a simple pick and place robot or a drilling robot. For example, when the robot reaches the desired point, it can return to the initial position. If this process is repetitive (robot plus controller), then it would be a simple pick and place robot used for manufacturing systems. Other applications could be: palletizing materials, press to press transferring, windshield glass handling, automotive components handling, cookie and bottle packing; and drilling. In those applications, the time spent on transferring a workpiece from one station to next or doing one or several perforations still high. In our prototype case, it becomes evident the use of position control due to the coordinates where a bore is desired. It is important to observe that after each perforation done by the robot it returns to their Initial position.

6. Conclusions

As a result about the assumptions and demonstrations realized in this chapter, is possible to conclude that the cartesian control is local. This characteristic restricts the system with its work area and it offers us a better understanding of the space in the location of the end-effector.

In this chapter we have described an experimental prototype for testing cartesian robot controllers with formal stability proof, which allows the programming a general class of cartesian robot controllers. The goal of the test system is to support the research as well as developing new cartesian control algorithms for robot manipulators. Our theoretical results are the propose on cartesian controllers. We have shown asymptotic stability in a local way by using Lyapunov's theory. Experiments on cartesian robot manipulator have been carried out to show the stability and performance for the cartesian controllers. For stability purposes, tuning procedure for the new scheme is enough to select a proportional and derivative gains as diagonal matrix in order to ensure asymptotic stability in a local way. However, the actual choice of gains can also produce torque saturation on the actuators, thus deteriorating the control system performance. To overcome these drawbacks, in this chapter it has been proposed a simple tuning rule. The scheme's performances were compared with the PD controller algorithm on cartesian coordinates by using a real time experiment on three degrees of freedom prototype. From experimental results the new scheme produced a brief transient and minimum steady-state position error.

In general, controllers showed better performance among the evaluated controllers and this statement can be proven by observing the performance index on the controllers. We can conclude that Pascal's cartesian controller is faster than PD cartesian controller and the polynomial family of PD-type cartesian controller, reason why the Pascal's cartesian controller offers some advantages in robot's control and in the time of operation.

7. Acknowledgement

The authors thanks the support received by Electronics Science Faculty on Autonomous University of Puebla, Mexico; and also by the revision on manuscript to Lic. Oscar R. Quirarte-Castellanos.

8. References

- Craig J. J. (1989) *Introduction to Robotics, Mechanics and Control*, Addison-Wesley Publishing Company, ISBN 0-201-09528-9, USA
- D'Souza A., Vijayakumar S. & Schaal S. (2001) *Learning inverse kinematics*, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 298-303.
- Gonin R. & H. Money A. (1989) *Nonlinear L_p -norm Estimation*, CRC, ISBN 0-8247-8125-2, USA.
- Hauser W. (1965) *Introduction to the principles of mechanics*, Addison-Wesley Publishing Company, Inc. Massachusetts, USA.
- Kelly R. & Santibáñez V. (2003) *Control de Movimiento de Robots Manipuladores*, Pearson Education SA, ISBN 84-205-3831-0, Madrid, España.
- Olsson H.; Åström K. J.; Canudas de Wit C.; Gäfvert M & Lischinsky P. (1998). Friction Models and Friction Compensation. *European Journal of Control*, Vol. 4, No. 3., (Dec. 1998) 176-195.
- Reyes F. & Rosado A.; (2005) Polynomial family of PD-type controllers for robot manipulators, *Journal on Control Engineering Practice*, Vol. 13, No. 4, (April 2005), 441-450, ISSN 0967-0661
- Sánchez-Sánchez P. & Reyes-Cortés F. (2005). *Pascal's cartesian controllers*, International Conference on Mechanics and Automation, Niagara Falls, Ontario, Canada, 94-100, ISBN 0-7803-9044-X
- Sánchez-Sánchez P. & Reyes-Cortés F. (2005) Position control through Pascal's cartesian controller, *Transactions on Systems*, Vol. 4, No. 12, 2417-2424, ISSN 1109-2777
- Sánchez-Sánchez P. & Reyes-Cortés F. (2006) *A new position controller: Pascal's cartesian controllers*, International Conference on Control and Applications, Montreal, Quebec, Canada, 126-132, ISBN 0-88986-596-5
- Sánchez-Sánchez P. & Reyes-Cortés F. (2008) A Polynomial Family of PD-Type Cartesian Controllers, *International Journal of Robotics and Automation*, Vol. 23, No. 2, 79-87, ISSN 0826-8185
- Santibáñez V.; Kelly R. & Reyes-Cortés F. (1998) A New Set-point controller with Bounded Torques for Robot Manipulators. *IEEE Transactions on Industrial Electronics*, Vol. 45, No. 1, 126-133
- Spong, M. W & Vidyasagar M. (1989) *Robot Dynamics and Control*, John Wiley & Sons, ISBN 0-471-61243-X, USA.
- Spong, M. W., Lewis F. L. & Adballah C. T. (1993) *Robot Control, Dynamics, Motion Planning and Analysis*, IEEE Press, ISBN 0-7803-0404-7, USA.
- Spong, M. W., Hutchinson S. & Vidyasagar M. (2006) *Robot modeling and Control*, John Wiley & Sons, Inc, ISBN 0-471-64990-8, USA.
- Synge L. J. (2008) *Principles of mechanics*, Milward Press, ISBN 1-443-72701-6, USA.
- Takegaki M. & Arimoto S. (1981) A New Feedback Method for Dynamic Control of Manipulators, *Journal of Dynamics System, Measurement and Control*, Vol. 103, No. 2, 119-125.
- Taylor R. J. (2005) *Classical Mechanics*, University Science Books, ISBN 1-8913-8922-X, USA.

Biomimetic Impedance Control of an EMG-Based Robotic Hand

Toshio Tsuji¹, Keisuke Shima¹, Nan Bu² and Osamu Fukuda²

1: Graduate School of Engineering, Hiroshima University,

2: Measurement Solution Research Center, National Institute of
Advanced Industrial Science and Technology
Japan

1. Introduction

The number of extremity amputations resulting from workplace mishaps, traffic accidents and other incidents has shown an increasing trend over time, although the importance of safety management and the prevention of such accidents is fully recognized. Since precise and complex motion may be very difficult in the daily activities of amputees, the development of prosthetic systems is necessary to support their lives and enable social integration. In particular, there is a mandatory requirement for the development of externally powered prosthetic hands with a natural feeling of control, since the role played by this part of the body is very important. However, the control of such hands is problematic, and they must be carefully designed in line with the amputee's remaining functions.

Many researchers have designed prosthetic limbs for amputees since the concept was proposed by N. Wiener in *Cybernetics* [1]. In previous research, electromyograms (EMGs) have been widely used as an interface tool for prosthetic hands because EMG signals contain information about the operator's intended motion [2] - [8]. For example, an EMG-prosthetic hand made in the USSR [2], the Waseda hand developed by Kato *et al.* [3], the Boston arm by MIT [4] and the Utha artificial arm by Jacobson *et al.* [5] were all pioneering steps in the field. Since EMG signals also include information on the force level and mechanical impedance properties of limb motion, Akazawa *et al.* [6] estimated the force of flexors and extensors from these signals and proposed a scheme to use them in controlling a prosthetic hand. Abul-haj and Hogan [8] also proposed prosthetic control based on an impedance model and analyzed its control characteristics.

Most previous research, however, dealt only with on/off control for prosthetic arms depending on the results of EMG pattern discrimination [2], [3], [7], or controlled only a particular joint depending on the torque estimated from EMG signals [4], [5], [6], [8]. Multi-joint control of prosthetic arms considering the variable viscoelasticity of flexors and extensors has not yet been realized.

This chapter introduces a biomimetic control for an externally powered multi-joint prosthetic hand that considers the muscular contraction levels of flexors and extensors using

neural networks. The method can express the difference between internal and external forces arising from the flexors and extensors, and all joints can be controlled as intended by the amputee. A natural feeling of control similar to that of the human arm can also be expected, since the viscoelasticity of each joint is regulated using EMG signals.

In this chapter, the biomimetic control for multi-joint motion is explained in Section 2. Section 3 describes the proposed impedance control system, and Sections 4 and 5 describe experiments conducted to assess the effectiveness of the method. Finally, Section 6 concludes the chapter and discusses the research work in further detail.

2. Biomimetic control of multi-joint motion

2.1 Control strategy

Various types of human motion are generally realized by multiple skeletal muscles related to each joint. Accordingly, multi-joint motion can be controlled if muscular contraction levels related to all joints can be estimated accurately from EMG signals. Even if only the human wrist joint is considered, however, there are at least three degrees of freedom and more than ten muscles involved with complex interrelations. Moreover, EMG signals measured on the skin surface contain only information on surface muscles in the human body, and such signals have non-linear and non-stationary characteristics. Estimation of all forces and torques caused by the muscles is therefore extremely difficult.

To overcome these difficulties, this chapter introduces a new control method involving two steps: In the first, the operator's intended joint motion is estimated using surface EMG signals measured from the operator's skin, and the joints to be driven in the prosthetic hand are selected. In the second step, the muscular contraction levels are estimated, and the joints selected in the first step are controlled using the impedance control.

Although many studies on discrimination of motion from EMG patterns [7], [10], [11] and impedance control of a single-joint prosthetic arm using EMG signals [6], [8], [9] have been reported so far, no previous method has realized multi-joint motion control based on EMG pattern discrimination and impedance control using internal and external forces arising from agonist and antagonist muscles. In this chapter, the method proposed by Tsuji *et al.* [17] is used for the first step.

2.2 Single-joint model considering flexors and extensors

Skilful motion of the human arm is realized by regulating its impedance properties, such as stiffness, viscosity and inertia [13]. A natural feeling of control similar to that of the human arm can be expected if prosthetic control is performed on the basis of impedance control with human-arm impedance properties.

As shown in Fig. 1 (a), the characteristics of joint motion can be represented by the tension balance between flexors and extensors. Here, each muscle tension f_i is modeled as

$$f_i = f_{0i}(\alpha_i) - k_i(\alpha_i)x_i - b_i(\alpha_i)\dot{x}_i, \quad (1)$$

where α_i ($0 \leq \alpha_i \leq 1$) is the muscular contraction level, $f_{0i}(\alpha_i)$ is the muscle tension under isometric contraction at a natural length, and x_i , $k_i(\alpha_i)$ and $b_i(\alpha_i)$ are the displacement, muscular stiffness and muscular viscosity, respectively. The subscript $i \in \{f, e\}$ indicates flexor or extensor.

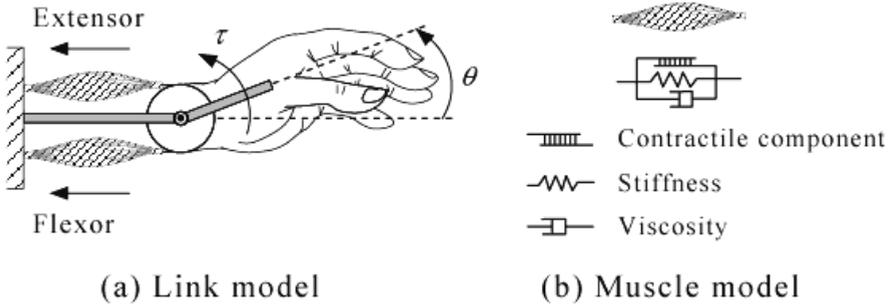


Fig. 1. Musculoskeletal model of human wrist joint

Accordingly, the joint torques τ_f and τ_e generated by the flexor and extensor are also described as

$$\tau_i = \tau_{0i}(\alpha_i) - K_i(\alpha_i)\theta - B_i(\alpha_i)\dot{\theta} , \tag{2}$$

where $\tau_{0i}(\alpha_i)$, $K_i(\alpha_i)$ and $B_i(\alpha_i)$ are the joint torque at the natural muscle length, joint stiffness and joint viscosity, respectively, and θ is the joint angle. Thus, the equation for wrist joint motion is represented as

$$\begin{aligned} I &= \tau_f + \tau_e \\ &= \tau_0(\alpha_f, \alpha_e) - K(\alpha_f, \alpha_e)\theta - B(\alpha_f, \alpha_e)\dot{\theta} \end{aligned} , \tag{3}$$

where I , $K(\alpha_f, \alpha_e)$ and $B(\alpha_f, \alpha_e)$ represent the impedance parameters of the wrist joint such as the moment of inertia, joint stiffness and viscosity, and $\tau_0(\alpha_f, \alpha_e)$ is the joint torque at the natural muscle length. In order to control the movement of the prosthetic hand based on (3), three problems must be solved: i) how to calculate muscle contraction levels α_f and α_e of the flexors and extensors during movement; ii) how to define the isometric joint torque $\tau_0(\alpha_f, \alpha_e)$; and iii) how to determine the impedance parameters of I , $K(\alpha_f, \alpha_e)$ and $B(\alpha_f, \alpha_e)$.

In this study, EMG signals and a neural network were utilized for the first and second problems. For the third one, the impedance parameters of wrist joints in non-amputee subjects were measured experimentally.

3. Biomimetic impedance control of a robotic hand

Figure 2 shows the proposed biomimetic impedance control system for a robotic hand. This system consists of four parts: force extraction; determination of the driven joint (raw EMG pattern discrimination); impedance control; and a robotic hand. The details of each part are described in following subsections.

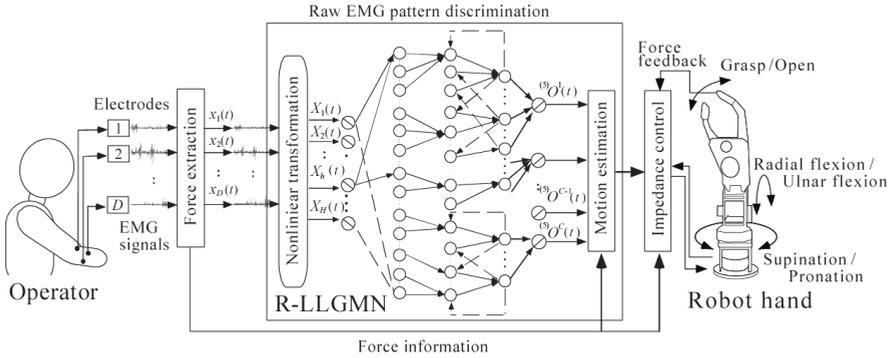


Fig. 2. Biomimetic control system for a robot hand

3.1 Force extraction

First, force information relating to the user was extracted from the EMG signals measured. The electrodes (NT-511G, NT-512G: Nihon Kohden Corp.) for EMG measurement were made of Ag/AgCl with a diameter of 0.012 m. The distance between the electrodes was set as 0.03 m. The measured EMG signals were amplified and filtered out with a low-pass filter (cut-off frequency: 100 Hz) in a multi-telemeter (Web5000: Nihon Kohden Corp.) and digitized using an A/D converter (sampling frequency: 200 Hz; quantization: 12 bits) after amplification (70 dB). The D channel EMG signals were denoted using $E_d(n)$ ($d = 1, 2, \dots, D$; $n = 1, 2, \dots, N$), where N is the number of all data.

The integral EMG (IEMG) $\beta(n)$ were obtained by calculating the moving average within the teacher vector length T of the neural network after rectification of $E_d(n)$:

$$\beta(n) = \frac{1}{D} \sum_{d=1}^D \frac{\overline{EMG}_d(n)}{\overline{EMG}_d^{\max}}, \quad (4)$$

$$\overline{EMG}_d(n) = \frac{1}{T} \sum_{t=0}^{T-1} |E_d(n-t)|, \quad (5)$$

where \overline{EMG}_d^{\max} is the pre-measured IEMG of each channel under the maximum voluntary contraction (MVC). It should be noted that $E_d(n-t) = 0$ when $n-t < 0$. In this system, $\beta(n)$ is used for recognition of the beginning and end of motion (i.e., when $\beta(n)$ is above the motion appearance threshold β_d , motion is judged to have occurred).

The input vector $\mathbf{x}_d(t) = [x_1(t), x_2(t), \dots, x_d(t)]^T$ ($t = 1, \dots, T$) for the neural network is defined as the normalized $E_d(n)$ with $\beta(n)$:

$$x_d(t) = \beta^{-1}(T) E_d(t). \quad (6)$$

This normalization enables motion estimation from the pattern of all channels and the amplitude of the raw EMG signals.

Furthermore, to allow impedance control of a robotic hand, the following values computed after $E_d(n)$ were rectified and filtered using a digital Butterworth filter (cut-off frequency: f_c Hz):

$$E_\mu(n) = \frac{1}{D} \sum_{d=1}^D (E_d(n) - E_d^{st}), \quad (7)$$

$$\delta_k(n) = \frac{E_\mu(n) - E^{th}}{E_k^{\max} - E^{th}}, \quad (8)$$

where E_d^{st} is the average of EMG signals $E_d(n)$ in a relaxed state and E_k^{\max} is the pre-measured $E_\mu(n)$ of each motion under the maximum voluntary contraction (MVC). $\delta_k(n)$ therefore describes the muscular contraction level for each motion ($0 \leq \delta_k(n) < 1$).

3.2 Determination of the driven joint

For the estimation of operator's intended motion, this subsection focuses on the pattern discrimination of the EMG signals using a probabilistic neural network (PNN). Since the PNN integrates statistical models into the neural network architecture as prior knowledge, outstanding performance has been reported [14]. For EMG pattern recognition using PNNs, the feature characteristics usually include: (1) amplitude, (2) frequency and (3) spatial information from multiple channels of EMG signals. However, significant temporal characteristics exist in the transient and non-stationary EMG signals, which cannot be considered by the traditional PNNs based on static stochastic models, and, in some cases, temporal characteristics could be only clues for reliable recognition. To overcome this problem, a Recurrent Log-Linearized Gaussian Mixture Network (R-LLGMN)[15] is utilized as the PNN for EMG pattern discrimination in the proposed system.

Since this network is composed of a feedforward NN including a Gaussian mixture model and feedback connections from output to input, the filtering process and the pattern discrimination are unified together and realized in a single network. The R-LLGMN includes a hidden Markov Model (HMM) [16] in its structure and can regulate the weight coefficients based on the learning scheme of the back-propagation through time (BPTT) algorithm [17]. The R-LLGMN ensures the filtering process and the pattern discrimination to be achieved at the same time and can attain high discrimination ability. The network therefore can classify time series of raw EMG signals [17].

The structure of R-LLGMN is shown in Fig. 2. This network is a five-layer recurrent NN with feedback connections between the 3rd layer and the 4th layer. First of all, the input vector $\mathbf{x}_d(t) = [x_1(t), x_2(t), \dots, x_d(t)]^T \in \mathfrak{R}^d$ is pre-processed with a non-linear computation and converted into the modified vector $\mathbf{X} \in \mathfrak{R}^H$:

$$\mathbf{X}(t) = \left[1, \mathbf{x}(t)^T, x_1(t)^2, x_1(t)x_2(t), \dots, x_1(t)x_d(t), x_2(t)^2, x_2(t)x_3(t), \dots, x_2(t)x_d(t), \dots, x_d(t)^2 \right]^T. \quad (9)$$

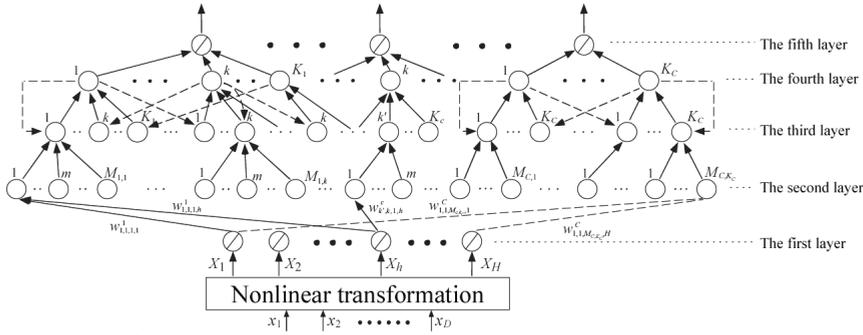


Fig. 3. Structure of the R-LLGMN

The first layer consists of H units corresponding to the dimension of X (the dimension H is determined as $H = 1 + d(d + 3)/2$) and the identity function is used for activation of each unit. Each unit in the first layer is defined as

$${}^{(1)}I_h(t) = X_h(t), \quad {}^{(1)}O_h(t) = {}^{(1)}I_h(t), \quad (10)$$

where ${}^{(1)}I_h(t)$ and ${}^{(1)}O_h(t)$ denote the input and the output, respectively, of the h th unit in the first layer.

Unit $\{c, k, k', m\}$ ($c = 1, \dots, C; k, k' = 1, \dots, K_c; m = 1, \dots, M_{c,k}$) in the second layer receives the output of the first layer weighted by the coefficient $w_{k',k,m,h}^c$. The relationship between the input and the output in the second layer is defined as

$${}^{(2)}J_{k',k,m}^c(t) = \sum_{h=1}^H {}^{(1)}O_h(t) w_{k',k,m,h}^c, \quad (11)$$

$${}^{(2)}O_{k',k,m}^c(t) = \exp\left({}^{(2)}J_{k',k,m}^c(t)\right), \quad (12)$$

where C is the number of classes, K_c is the number of states, $M_{c,k}$ is the number of the components of the Gaussian mixture distribution corresponding to the class c and the state k [18].

The input into a unit $\{c, k, k'\}$ in the third layer integrates the outputs of units $\{c, k, k', m\}$ ($m = 1, \dots, M_{c,k}$) in the second layer. The output in the third layer is that input weighted by the previous output in the fourth layer. The input-output relationship of the unit in the third layer is defined as

$${}^{(3)}J_{k',k}^c(t) = \sum_{m=1}^{M_{c,k}} {}^{(2)}O_{k',k,m}^c(t), \quad (13)$$

$${}^{(3)}O_{k',k}^c(t) = {}^{(4)}O_{k'}^c(t-1) {}^{(3)}J_{k',k}^c(t), \quad (14)$$

where ${}^{(4)}O_k^c(0) = 1.0$ for the initial state.

The fourth layer receives the integrated outputs of units $\{c, k, k'\}$ in the third layer. The input-output relationship in the fourth layer is defined as

$${}^{(4)}I_k^c(t) = \sum_{k'=1}^{K_c} {}^{(3)}O_{k',k}^c(t), \quad (15)$$

$${}^{(4)}O_k^c(t) = \frac{{}^{(4)}I_k^c(t)}{\sum_{c'=1}^C \sum_{k'=1}^K {}^{(4)}I_{k'}^{c'}(t)}. \quad (16)$$

At last, a unit c in the fifth layer integrates the outputs of K_c units $\{c, k\}$ ($k = 1, \dots, K_c$) in the fourth layer. The relationship in the fifth layer is defined as

$${}^{(5)}I^c(t) = \sum_{k=1}^{K_c} {}^{(4)}O_k^c(t), \quad (17)$$

$${}^{(5)}O^c(t) = {}^{(5)}I^c(t). \quad (18)$$

The output of the network ${}^{(5)}O^c(t)$ corresponds to the *a posteriori* probability of the input vector $\mathbf{x}(t)$ for the class c , while only the weight coefficients $w_{k',k,m,h}^c$ between the first layer and the second layer are adjusted by learning.

The entropy of R-LLGMN's output is calculated to prevent the risk of misdiscrimination. The entropy is defined as

$$H(t) = -\sum_{c=1}^C {}^{(5)}O^c(t) \log_2 {}^{(5)}O^c(t). \quad (19)$$

If the entropy $H(t)$ is less than the discrimination threshold H_d , the specific motion with the largest probability is determined according to the Bayes decision rule. If not, the determination is suspended.

The details of the raw EMG pattern discrimination using the R-LLGMN are described in [17]. In the proposed system, the driven joint j is selected based on the output of the R-LLGMN.

3.3 Impedance control

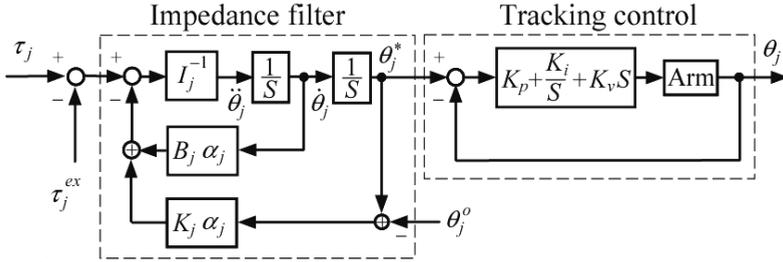


Fig. 4. Impedance control system

Human hand and wrist movement can be expressed based on a mechanical impedance model encompassing stiffness, viscosity and inertia [13]. The introduction of a human hand and wrist impedance model to control a robotic hand will make it possible to realize a natural feeling of control similar to that of human movement. This subsection explains a method to control a robot hand based on the mechanical impedance of the human hand's wrist movements.

First, the equation of motion around the manipulator's joint j is defined as

$$I_j \ddot{\theta}_j + B_j(\alpha_j) \dot{\theta}_j + K_j(\alpha_j)(\theta_j - \theta_j^o) = \tau_j - \tau_j^{ex}, \quad (20)$$

where I_j , $B_j(\alpha_j)$ and $K_j(\alpha_j)$ are the moment inertia, joint viscosity and stiffness, respectively. Here, it is assumed that the joint torques caused by muscular contraction of the flexors and extensors have almost the same properties, and the muscular contraction level of each joint is expressed as $\alpha_j = \alpha_{j,f} + \alpha_{j,e}$. θ_j and θ_j^o describe the joint angle and its equilibrium position, and τ_j and τ_j^{ex} are the joint torque and external torque, respectively. Muscular contraction level α_j and joint torque τ_j can be expressed as follows:

$$\alpha_j = \delta_k(n), \quad (21)$$

$$\tau_j(n) = \delta_k(n) \tau_k^{\max}, \quad (22)$$

where j is the joint driven during motion k , and τ_k^{\max} describes the maximum torque measured from the subject for each motion k in advance. It should be noted that the torque is assumed to remain the same (i.e., $\tau_j(n) = \tau_j(n-1)$) when the discrimination results are suspended.

Using the above equations, the desired joint angles can be calculated numerically using dynamic equation (20) considering changes in the EMG signals. Figure 4 shows the impedance control system used in this study. Here, K_p , K_i and K_v are the gain parameters for PID control. This method can be expected to provide a natural feeling of control similar to that of the original limb if the impedance parameters are set as values similar to those of the human arm. The hand can also react to external forces using a force sensor.

3.4 Robotic hand [12]

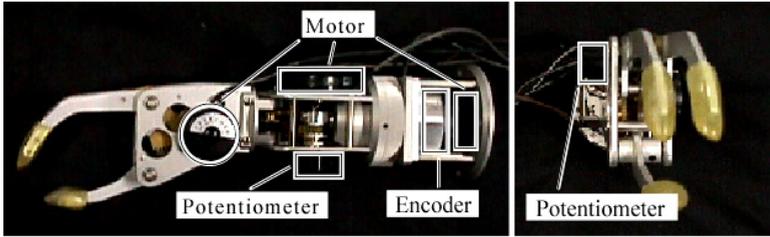


Fig. 5. Robotic hand

A photograph of the robotic hand utilized in this study is shown in Fig. 5 [12]. It is almost the same size as an adult hand, and weighs about 1.0 [kg]. The hand has three degrees of freedom (supination/pronation, radial flexion/ulnar flexion, grasp/open), and each joint is driven by an ultrasonic motor (Sinsei Corp.) [7]. An encoder or potentiometer is installed as an angular sensor for each joint as shown in Fig. 5. The unit can be attached to the amputation site and used as a prosthetic hand.

4. Determination of human wrist impedance parameters

Impedance control is an effective technique for achieving natural feeling similar to that of a human hand in prosthetic control. However, it is difficult to set the relevant parameters appropriately. Measurement experiments to ascertain human wrist joint impedance characteristics were therefore carried out.

4.1 Measurement of human wrist joint impedance

This section outlines Tsuji's method [13] for measurement of human wrist joint impedance. In a case where the subject performs single-joint motion of the wrist on a 2D plane, the dynamic properties of the hand can be approximated using a mechanical impedance model on a 2D plane as follows:

$$I_h \ddot{\theta}(t) + B_h \dot{\theta}(t) + K_h \theta(t) = \tau(t), \quad (23)$$

where $\theta(t)$ is the joint angle, $\tau(t)$ is the joint torque, and I_h , B_h and K_h are the moment of inertia, joint viscosity and stiffness, respectively.

In order to estimate the relevant parameters, the hand of the subject is displaced from equilibrium by means of a small short-duration disturbance (Fig. 6). A small disturbance is necessary to enable the assumption of an approximate constancy for I_h , B_h and K_h , which are known to depend on posture in smooth conditions. Here, at the onset time t_0 of the disturbance, we have

$$I_h \ddot{\theta}(t_0) + B_h \dot{\theta}(t_0) + K_h \theta(t_0) = \tau(t_0). \quad (24)$$

The dynamic hand properties at given time t are established by Eqs. (23) and (24):

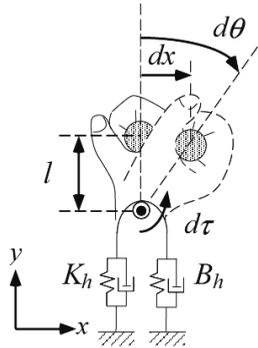


Fig. 6. Impedance model of the wrist joint

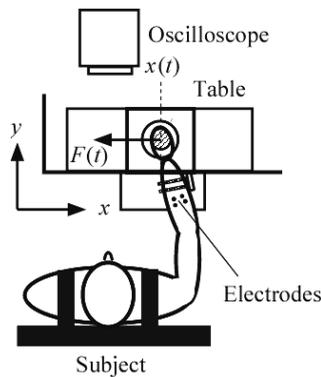


Fig. 7. Experimental apparatus

$$I_h d\ddot{\theta}(t) + B_h d\dot{\theta}(t) + K_h d\theta(t) = d\tau(t), \quad (25)$$

where $d\theta(t) = \theta(t) - \theta(t_0)$ and $d\tau(t) = \tau(t) - \tau(t_0)$. The parameters I_h , B_h and K_h of the subject can be estimated using the least square method from n equations for each sample ($t = t_1, \dots, t_n$). Figure 7 shows the experimental apparatus for measurement of impedance parameters, which uses a linear motor table (Nippon Thompson Co., Ltd.) with one degree of freedom adopting a magnetic drive to forcibly displace the subject's hand. In the experiments, the right hand of the subject sitting in front of the table was fixed to a handle through a cast made of glass fiber. In addition, the subject's forearm was fixed to an arm-supporting stand by the cuff. The surface EMG was then measured from the agonist and antagonist muscles of the wrist joint in the subject's forearm in order to clarify the activity of the muscles around the joint. A display for online monitoring of the muscle contraction level was set in front of the subject. The task for the subjects involved isometric flexion and extension movement of the wrist joint. During the experiments, the subjects could regulate the muscle contraction level as instructed by the experimenter since the contraction levels of each

muscle were monitored by the display. The force generated by the subject was measured using a six-axis force sensor (BL Autotec Co. Ltd; resolution ability: force x - and y -axes: 0.05 N; z -axis: 0.15 N) attached to the handle of the table. The output from the encoder and force sensor were sampled at 2 kHz.

From the slight hand displacement in the x axial direction $x(t)$ and the hand's resultant force along the x axis $F_x(t)$, we obtain the joint angle displacement $\theta(t)$ and joint torque $\tau(t)$ using the following equations:

$$\theta(t) = \sin^{-1}(x(t)/l), \tag{26}$$

$$\tau(t) = lF_x(t), \tag{27}$$

where l is the distance from the handle to the center of the wrist joint (see Fig. 6).

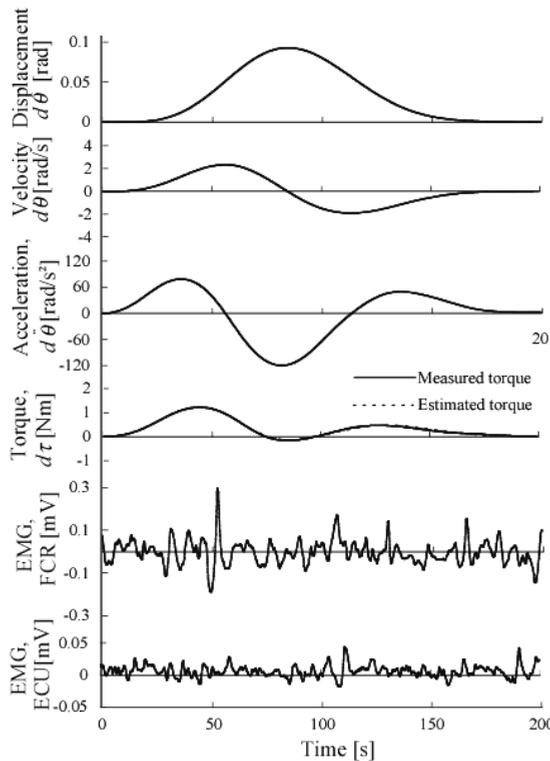


Fig. 8. A typical example of measured results for human wrist while maintaining posture

4.2 Results

Four healthy male subjects participated in the experiments. Figure 8 shows an example of the measurement results for wrist joint displacement in the extension direction. The figure shows the joint angle $d\theta(t)$, velocity $d\dot{\theta}(t)$, acceleration $d\ddot{\theta}(t)$, wrist joint torque $d\tau(t)$ and the EMG signals of the flexor carpi radialis (FCR) and extensor carpi ulnaris (ECU). From

the figure, it can be seen that the measured torque (shown by the solid line) corresponds with the estimated torque (the dotted line) using the measured $d\theta(t)$, $d\dot{\theta}(t)$ and $d\ddot{\theta}(t)$ and the impedance parameters I_h , B_h and K_h computed from them. The results indicate that an impedance model based on Eqs. (23), (24) and (25) can express the human wrist joint's characteristics.

Examples of the relationships between the muscular contraction levels and estimated parameters I_h , B_h and K_h are shown in Fig. 9. In this figure, α ($0 \leq \alpha \leq 1$) indicates the muscular contraction levels estimated from the EMG signals, and the plotted data show the measured values. From the figure, moment inertia I_h is maintained at a constant value, and viscosity B_h and stiffness K_h change according to the level of muscle contraction. In particular, B_h and K_h increase as muscle contraction levels become high. This study therefore approximated the relationships between muscular contraction levels and the impedance properties of stiffness and viscosity around the wrist joint as follows:

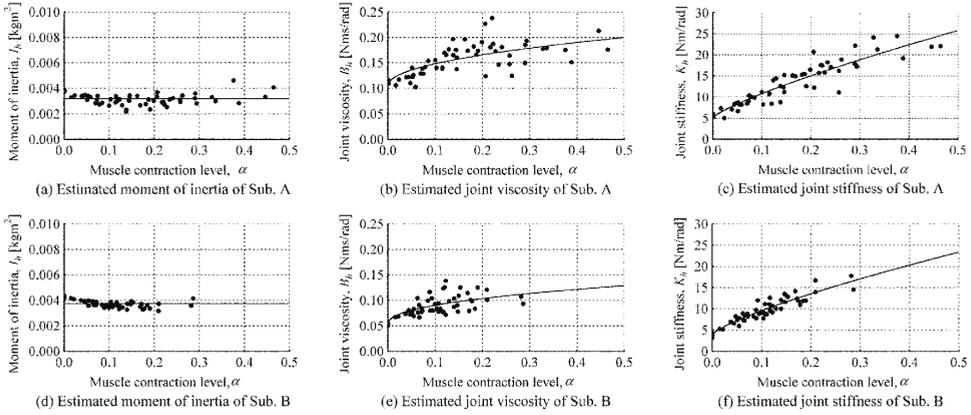


Fig. 9. Estimated impedance parameters of human wrist joint

$$\hat{B}_h(\alpha) = b_1\alpha^{b_2} + b_3, \quad (28)$$

$$\hat{K}_h(\alpha) = k_1\alpha^{k_2} + k_3, \quad (29)$$

where $\hat{B}_h(\alpha)$ and $\hat{K}_h(\alpha)$ are the estimated values of viscosity and stiffness around the wrist joint, k_i and b_i ($i = 1, 2, 3$) are constants, and k_3 and b_3 correspond to the level of viscoelasticity seen when the arm is in a relaxed state. The solid line in Fig. 9 shows the impedances estimated by Eqs. (28) and (29) using the least square method from the values obtained. From the figure, it is seen that the tendency of each parameter can be expressed using Eqs. (28) and (29).

5. Robotic hand control experiments

5.1 Experimental conditions

In the experiments, the cut-off frequency f_c and sampling frequency f_s were set as 3.0 Hz and 1,000 Hz, respectively. Based on the above-mentioned experiments, the impedance parameters for controlling the robotic hand were set as follows:

$$I_j = i_j, \quad (30)$$

$$B_j(\alpha_j) = b_{j,1}\alpha_j^{b_{j,2}} + b_{j,3}, \quad (31)$$

$$K_j(\alpha_j) = k_{j,1}\alpha_j^{k_{j,2}} + k_{j,3}, \quad (32)$$

where $j = 1, 2, 3$ is the joint number, and each parameter in Eqs. (30), (31) and (32) is shown in Table 1. Four healthy male subjects participated in the experiments. It should be noted that the impedance parameters estimated from the human impedance characteristics measured were for the wrist joint only (i.e., flexion and extension), and the parameters of other joints were defined by trial and error based on the wrist joint characteristics.

joint j	Motion (k)	$k_{j,1}$	$k_{j,2}$	$k_{j,3}$	$b_{j,1}$	$b_{j,2}$	$b_{j,3}$	i_j
1	Pronation (1) / spination (2)	32.0	0.6	3.2	0.14	0.2	0.110	0.002
2	Ulnar flexion (3) / radial flexion (4)	32.8	0.6	3.2	0.14	0.2	0.144	0.004
3	Open (5) / close (6)	0.90	0.6	0.3	0.08	0.2	0.090	0.001

Table 1. Impedance parameters used in the experiments

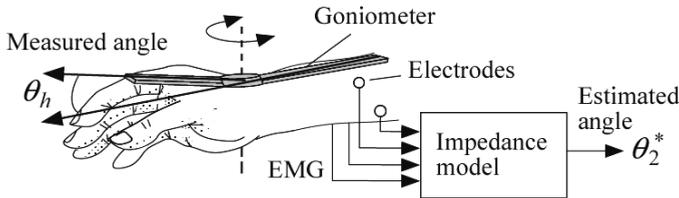


Fig. 10. Estimation of the joint angle from EMG signals based on biomimetic impedance model

5.2 Biomimetic impedance control

Here, in order to examine the validity of the impedance parameters, the motion of the subject's wrist joint and that of the manipulator were compared.

The subject executed wrist flexion and extension, and the wrist joint angle and EMG signals were measured as shown in Fig. 9. The muscle contraction level α_j was calculated from the EMG signals using the proposed method, and the desired joint angle θ_j^* (see Fig. 10) was calculated according to the muscle contraction level and the motion equation for the wrist joint with variable viscoelasticity (Eqs. (20), (21), and (22)).

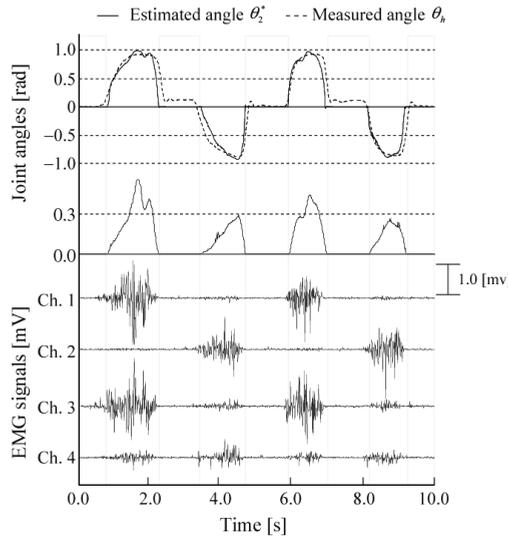


Fig. 11. Comparison between the estimated and measured angles of the wrist joint

In the experiment, four electrodes ($L = 4$: 1 ch. flexor carpi radialis, FCR; 2 ch. flexor carpi ulnaris, FCU; 3 ch. extensor carpi radialis, ECR; and 4 ch. brachioradialis, BR) were used, two motions ($k = 2$; flexion and extension) were discriminated using the neural network, and the motion appearance threshold was set as $E_d = 0.17$.

Figure 11 shows an example of the experimental results. The wrist joint angle of the manipulator almost corresponds with that of the subject. Moreover, the maintenance of posture under conditions of muscle co-contraction was realized. It can be seen that the joint angles of the prosthetic hand can be controlled using EMG signals.

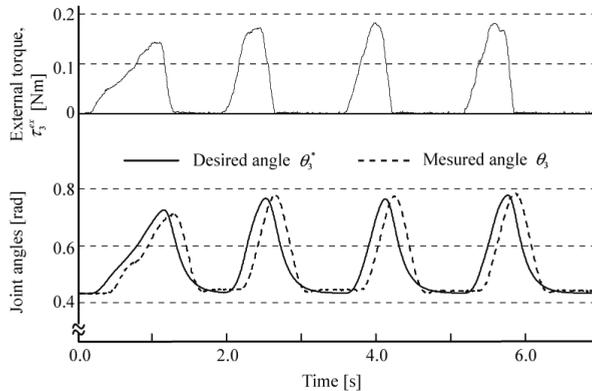


Fig. 12. Changes of joint angles of the finger part corresponding to external torque

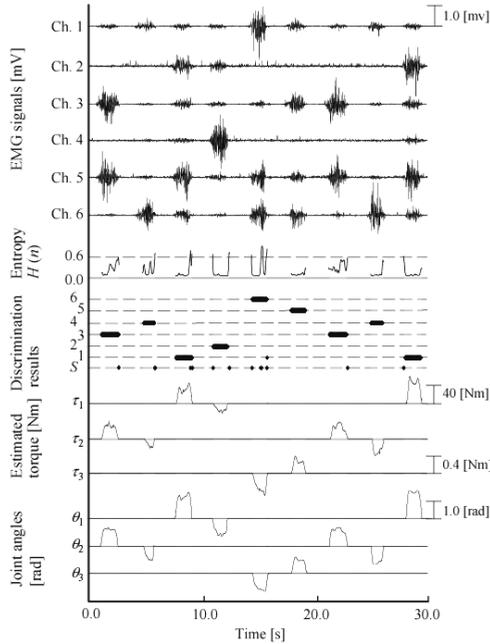


Fig. 13. An example of experimental control results (healthy subject)

To verify the response characteristics of the robotic hand for external torque, its movement was investigated by a human pressing the distal parts of its finger (attached to a pressure sensor) during in a relaxed state ($\alpha_j = 0$). The experimental results are shown in Fig. 12, in which the solid line shows the desired joint angle θ_3^* computed based on the impedance model, and the dotted line shows the joint angle θ_3 obtained (see Fig. 4). From the figure, it can be observed that the joint angle of the robotic hand changes according to the external force applied, but a time delay between the measured and estimated values is seen. This was mainly caused by the control gains of the PID controller being set low ($K_p = 0.16$, $K_i = 0.001$ and $K_v = 0.006$) because the robotic hand, which is operated with ultrasonic motors, cannot provide smooth control with large gains.

5.3 Robotic hand control

Control experiments of the robotic hand based on EMG signals were carried out to verify the proposed method using the estimated muscular contraction levels. The experimental conditions were almost the same as those of the previous experiments. Six electrodes ($L = 6$: 1 ch. flexor carpi radialis; 2 ch. triceps brachii; 3 ch. extensor carpi radialis; 4 ch. biceps brachii; 5 ch. brachioradialis; and 6 ch. flexor carpi ulnaris) were attached to the forearm of a healthy subject (male, 25 years old), and EMG signals were measured. Figure 13 shows an example of the experimental results, including EMG signals from the six channels, entropy, discrimination results, estimated torque and joint angle; the shaded areas indicate periods with no motion. From the figure, it can be seen that the measured EMG signals can be classified accurately using the proposed method, and that the joint angles of the robotic hand can be controlled using these EMG signals.

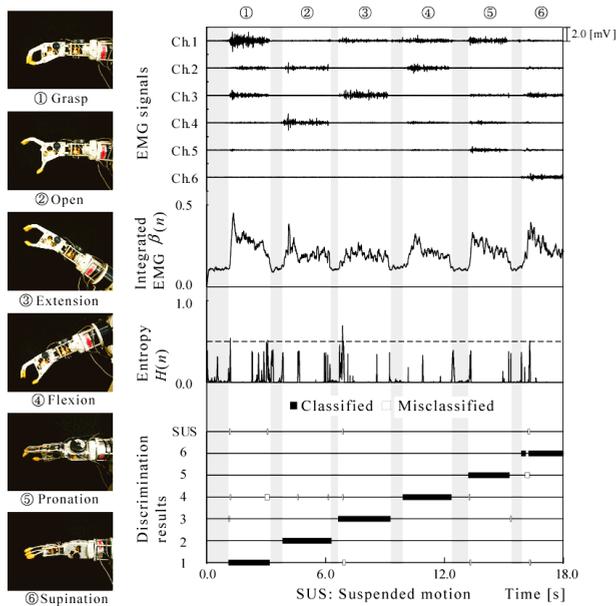


Fig. 14. Changes in the joint angles of the finger part in response to external torque

To confirm the suitability of the proposed method for people with physical disabilities, the EMG signals of an amputee subject (male, 44 years old) were measured and used to control the robotic hand for verification experiments. The subject's forearm was amputated at the age of 41. In this experiment, the R-LLGMN parameters were set as follows: the number of classes was $K = 6$; the number of states for each class was $K_{1,\dots,6} = 1$; the number of components was $M_{1,1,\dots,6,1} = 1$; the teacher vector length was $T = 20$ (about 100 ms), and eight data sets were used for learning. Figure 14 shows the experimental results, including photographs of the robotic hand during control, raw EMG signals, integrated EMG (IEMG), entropy and discrimination results, respectively. The shaded area shows periods when no motion was judged from the IEMG. From the figure, it can be seen that the subject could control the robotic hand smoothly using EMG patterns, and the discrimination rate in this operation was 95.5%. The results lead us to conclude that the proposed method can be used for robotic hand control based on human impedance characteristics.

5. Conclusion

This chapter explained a biomimetic impedance control method based on raw EMG classification for the development of a multi-joint robotic control system. In the experiments performed, a natural feeling of prosthetic control similar to that of the human hand was realized using the neural network and the biomimetic impedance control.

In the future, the authors plan to improve accuracy of estimation for the cooperation ratio of multiple muscles and construct a system that allows a more natural feeling of control.

Publications concerning this chapter are listed in the bibliography [17], [19].

6. References

- [1] Wiener, N. (1948); *CYBERNETICS or Control and Communication in the Animal and the Machine*, MIT Press
- [2] Sherman, E. D. (1964); A Russian Bioelectric-Controlled Prosthesis, *Canada, M. A. J.*, 91, pp. 1268-1270
- [3] Kato, I.; Okazaki, E.; Kikuchi H.; Iwanami K. (1967); Electro-Pneumatically Controlled Hand Prosthesis Using Pattern Recognition of Myo-Electric Signals, *Digest of 7th ICMBE*, p. 367
- [4] Jerard, R. B.; Williams, T. W.; Ohlenbusch, C. W. (1974); Practical Design of an EMG Controlled Above Elbow Prosthesis, *Proceedings of 1974 Conference on Engineering of Devices for Rehabilitation*, Tufts Univ. School of Med., p. 73, Boston, MA
- [5] Jacobson, S. C.; Knutti, D. F.; Johnson, R. T.; Sears, H. H. (1982); Development of the Utah Artificial Arm, *IEEE Transactions on Biomedical Engineering*, Vol. 29, No. 4, pp. 249-269, April
- [6] Akazawa, K.; Takizawa, H.; Hayashi, Y.; Fujii, K. (1988); Development of Control System and Myoelectric Signal Processor for Bio-Mimetic Prosthetic Hand, *Biomechanism 9*, pp. 43-53 (in Japanese)
- [7] Ito, K.; Tsuji, T.; Kato, A.; Ito, M. (1992); An EMG Controlled Prosthetic Forearm in Three Degrees of Freedom Using Ultrasonic Motors, *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 14, pp. 1487-1488
- [8] Abul-haj, C. J.; Hogan, N. (1990); Functional Assessment of Control Systems for Cybernetic Elbow Prostheses-Part I, Part II, *IEEE Transactions on Biomedical Engineering*, Vol. 37, No. 11, pp. 1025-1047, November
- [9] Ito, K.; Tsuji, T. (1985); Control Properties of Human-Prosthesis System with Bilinear Variable Structure, *Proceedings of the 2nd IFAC Conference on Man-Machine Systems*, pp. 353-358
- [10] Graupe, D. ; Magnussen, J.; Beex, A. A. M. (1978); A Microprocessor System for Multifunctional Control of Upper Limb Prostheses via Myoelectric Signal Identification, *IEEE Transactions on Automatic Control*, Vol. 23, No. 4, pp. 538-544, August
- [11] Hiraiwa, A.; Shimohara, K.; Tokunaga, Y. (1989); EMG Pattern Analysis and Classification by Neural Network, *Proceedings of IEEE International Conference on System, Man, and Cybernetics*, pp. 1113-1115
- [12] Fukuda, O.; Tsuji, T.; Kaneko, M.; Otsuka, A. (2003); A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm Motions, *IEEE Transactions on Robotics and Automation*, Vol.19, No.2, pp.210-222, April
- [13] Tsuji, T. (1997); Human arm impedance in multi-joint movements, Self-Organization, Computational Maps and Motor Control (P. Morasso and V. Sanguineti, Ed.), *Advances in Psychology*, 119, Elsevier, North-Holland, 1997, pp. 357-382
- [14] Zhang, G.D. (2000); Neural network for classification: A survey, *IEEE Transactions on Systems, Man and Cybernetics, Part C, Applications and Reviews*, Vol. 30, pp. 451-462
- [15] Tsuji, T.; Bu, N.; Fukuda, O.; Kaneko, M.; (2003); A Recurrent Log-linearized Gaussian Mixture Network, *IEEE Transactions on Neural Networks*, Vol. 14, No. 2, pp. 304-316, March

- [16] Rabiner, L.R. (1989); A tutorial on hidden Markov model and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77, pp. 257-286
- [17] Tsuji, T.; Bu, N.; Fukuda, O. (2006); A Recurrent Probabilistic Neural Network for EMG Pattern Recognition, *Neural Networks in Healthcare: Potential and Challenges* (R. Begg, J. Kamruzzaman, R. Sarker, Ed.), pp. 130-153 (Idea Group Inc.)
- [18] Tsuji, T.; Fukuda, O.; Ichinobe, H.; Kaneko, M. (1999); A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 29, No. 1, pp. 60-72, February
- [19] Tsuji, T.; Fukuda, O.; Shigeyoshi, H.; Kaneko, M. (2000); Bio-mimetic Impedance Control of an EMG-Controlled Prosthetic Hand, *CD-ROM Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju

Adaptive Robust Controller Designs Applied to Free-Floating Space Manipulators in Task Space

Tatiana Pazelli, Marco Terra
and Adriano Siqueira
*University of São Paulo
São Carlos, São Paulo
Brazil*

1. Introduction

Space robots are featured by a dynamic coupling which causes the rotation of the main body with the coordinated motions of the arm. A number of dynamic and control problems are unique to this area due to the distinctive and complex dynamics found in many applications. In-space operations such as assembly, inspection and maintenance of satellites or space stations have been receiving considerable research efforts. Considering the hostile environment where a space robot operates, which can deteriorate its structure and physical characteristics, and also considering the difficulty of taking the system back to reformulate its dynamic model due to these uncertainties, the proposal of intelligent adaptive robust controllers to this kind of system becomes very interesting.

One of the representative types of space robotic systems identified by [Dubowsky & Papadopoulos (1993)], free-floating space manipulators are systems that allows the spacecraft to move freely in response to the manipulator motions in order to conserve fuel and electrical power, [Papadopoulos & Dubowsky (1991)]. Trajectory planning algorithms have been developed in order to minimize the reaction motion of the free-floating base while executing the manipulator task, [Huang & Xu (2006); Liu et al. (2009); Papadopoulos et al. (2005); Torres & Dubowsky (1992); Tortopidis & Papadopoulos (2006)]. In case of redundant manipulators, coordinated spacecraft/manipulator motion control has been addressed in [Caccavale & Siciliano (2001); Dubowsky & Torres (1991)].

Solving control problems in joint space is an inconvenient task for a space robot with a free-floating base. When the base is free-floating, the kinematic mapping from task space to joint space, where the control is executed, becomes non-unique because of non-integrable angular momentum conservation. This may cause non-existence of the reference trajectory in joint space. Also, parametric uncertainties appear not only in the dynamic equation, but also in kinematic mapping from the joint space to the task space due to the absence of a fixed base. The model inaccuracies lead to the deviation of operational space trajectory provided by the kinematic mapping. [Parlaktuna & Ozkan (2004)] and [Abiko & Hirzinger (2009)] applied on-line adaptive techniques to deal with parametric uncertainties in controlling free-floating

manipulators at task space. [Fu et al. (2007)], on the other hand, established an off-line adaptive estimator to provide accurate identified parameters to a dynamic control law. In order to cope with the nonlinear parameterization problem of the dynamic model of the free-floating space robot system, [Gu & Xu (1993)], [Parlaktuna & Ozkan (2004)] and [Fu et al. (2007)] have modeled the system as an extended arm, and [Abiko & Hirzinger (2009)] used the inverted chain approach to explicitly describe the coupled dynamics between the end-effector and the robot arm.

This chapter deals with the problem of robust trajectory tracking control in task space for free-floating manipulator systems subject to plant uncertainties and external disturbances. To conduct a comparative study, adaptive techniques are developed considering nonlinear \mathcal{H}_∞ controllers based on game theory. A first approach is proposed considering a well defined structure for the plant, where the parameter uncertainties are represented as external disturbances. Artificial neural networks are applied in two other approaches. The first one applies the intelligent system to learn the dynamic behavior of the robotic system, which is considered totally unknown. The second intelligent strategy considers a well defined nominal model structure and the neural networks are applied to estimate only the behavior of parametric uncertainties and the spacecraft dynamics, considered here as non-modeled dynamics. The \mathcal{H}_∞ criterion is applied to the proposed techniques to attenuate the effect of estimation errors and external disturbances.

The dynamic model of the free-floating space manipulator (SM) is described in this chapter through the Dynamically Equivalent Manipulator (DEM) approach, [Liang et al. (1996)]. The DEM is a fixed-base manipulator with its first joint being a passive spherical one and, whose model is both kinematically and dynamically equivalent to the SM dynamics.

Trajectory tracking of the SM's end-effector in task space is considered for simulation. A fixed-base, three-link, planar manipulator whose first joint is configured as passive (UArmII - Underactuated Arm II) is taken as a dynamically equivalent reference to a free-floating space planar manipulator with two links. Parametric uncertainties and finite energy exogenous disturbances are included in the nominal model. In order to validate and compare the proposed strategies, graphical and numerical analysis are provided.

The chapter is organized as follows: Section 2 presents the model description through the DEM approach; the solutions for the nonlinear \mathcal{H}_∞ control problems based on model and neural networks are presented in Section 3; and, finally, simulation results for a two-link free-floating space manipulator are presented in Section 4.

2. Model Description and Problem Formulation

2.1 Free-Floating Space Manipulator Mapped by a Dynamically Equivalent Fixed-Base Manipulator

Consider an n -link serial-chain rigid manipulator mounted on a free-floating base and that no external forces and torques are applied on this system. Consider also the Dynamically Equivalent Manipulator (DEM) approach, [Liang et al. (1996)]. The DEM is an $(n + 1)$ -link fixed-base manipulator with its first joint being a passive spherical one and whose model is both kinematically and dynamically equivalent to the SM dynamics. Since it is a conventional manipulator, it can be physically built and experimentally used to study control algorithms for space manipulators.

Figure 1 shows the representation and the parameter notation for both SM and DEM manipulators. Let the SM parameters be identified by apostrophe ($\phi', \theta', \rho', J'$), the links of the manipulators are numbered from 2 to $n + 1$; the Z-Y-Z euler angles (ϕ, θ, ρ) represent the SM

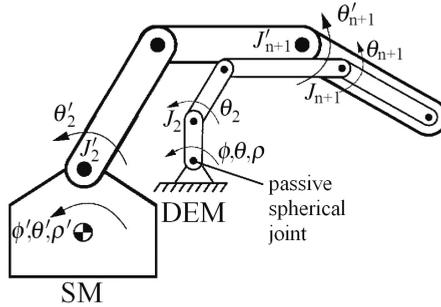


Fig. 1. The space manipulator and its corresponding DEM.

base attitude and the DEM first passive joint orientation; J_i is the joint connecting the $(i - 1)$ -th link and i -th link; θ_i is the rotation of the i -th link around joint J_i ; C_i is the center of mass of the i -th link; L_i is the vector connecting J'_i and C_i ; R_i is the vector connecting C_i and J_{i+1} ; l_{ci} is the vector connecting J_i and C_i ; and W_i is the vector connecting J_i and J_{i+1} . Considering that the DEM operates in the absence of gravity and that its base is located at the center of mass of the SM, the kinematic and dynamical parameters of the DEM can be found from the SM parameters as

$$\begin{aligned}
 m_i &= \frac{M_t^2 m'_i}{\sum_{k=1}^{i-1} m'_k \sum_{k=1}^i m'_k}, & i &= 2, \dots, n + 1, \\
 I_i &= I'_i, & i &= 1, \dots, n + 1, \\
 W_1 &= R_1 \frac{\sum_{k=1}^i m'_k}{M_t}, \\
 W_i &= R_i \frac{\sum_{k=1}^i m'_k}{M_t} + L_i \frac{\sum_{k=1}^{i-1} m'_k}{M_t}, & i &= 2, \dots, n + 1, \\
 l_{c1} &= 0, \\
 l_{ci} &= L_i \frac{\sum_{k=1}^{i-1} m'_k}{M_t}, & i &= 2, \dots, n + 1,
 \end{aligned}
 \tag{1}$$

where M_t is the total mass of the SM. Observe that the mass of the passive joint, m_1 , is not defined by the equivalence properties. Let the generalized coordinates $q = [\phi \ \theta \ \rho \ \theta_2 \ \dots \ \theta_{n+1}]^T$ be partitioned as $q = [q_b^T \ q_m^T]^T$, where the indexes b and m represent the passive spherical joint (base) and the active joints (manipulator), respectively. From Lagrange theory, dynamic equations of the DEM are given by

$$M(q_m)\ddot{q} + C(q_m, \dot{q})\dot{q} = \tau,
 \tag{2}$$

where $M(q_m) \in \mathbb{R}^{(n+3 \times n+3)}$ is the symmetric positive definite inertia matrix, $C(q_m, \dot{q}) \in \mathbb{R}^{(n+3 \times n+3)}$ is the matrix of Coriolis and centrifugal forces, and $\tau = [0 \ 0 \ 0 \ \tau_2 \ \dots \ \tau_{n+1}]^T$ is the torque vector acting upon the joints of the DEM. Parametric uncertainties can be introduced dividing the parameter matrices $M(q_m)$ and $C(q_m, \dot{q})$

into a nominal and a perturbed part

$$\begin{aligned} M(q_m) &= M_0(q_m) + \Delta M(q_m), \\ C(q_m, \dot{q}) &= C_0(q_m, \dot{q}) + \Delta C(q_m, \dot{q}), \end{aligned}$$

where $M_0(q_m)$ and $C_0(q_m, \dot{q})$ are nominal matrices and $\Delta M(q_m)$ and $\Delta C(q_m, \dot{q})$ are the parametric uncertainties.

2.2 Problem Formulation

As we are dealing with a free-floating space manipulator, it is considered that only the active joints of the DEM are controlled, with the passive spherical joint not locked. In this case, the passive joint dynamics intervenes with the control of the manipulator active joints.

The vector of orientation and inertial position of the end-effector,

$$p = [\phi_{ef} \quad \theta_{ef} \quad \psi_{ef} \quad x_{ef} \quad y_{ef} \quad z_{ef}],$$

is function of free-floating base position and attitude and of generalized coordinates of manipulator joints, q_m . Once the DEM modelling technique locates the inertial frame origin at the center of mass of the SM, the dependence of end-effector coordinates on base position is eliminated by integrating its equation of linear momentum, [Papadopoulos & Dubowsky (1991)]. However, the dependence on base attitude cannot be eliminated since the angular momentum of the system cannot be analytically integrated to provide the base attitude as a function of the variables of manipulator joints.

Let $J(q)$ be the Jacobian that relates the velocities of joints coordinates, \dot{q} , and the velocities of the end-effector, \dot{p} :

$$\dot{p} = J(q)\dot{q}. \quad (3)$$

Considering that $\det(J(q)) \neq 0$, applying (3) and its derivative, $\ddot{p} = \dot{J}(q)\dot{q} + J(q)\ddot{q}$, to (2), we have

$$\tau = M_{ef}(q)\ddot{p} + C_{ef}(q, \dot{q})\dot{p}, \quad (4)$$

where

$$\begin{aligned} M_{ef}(q) &= M(q_m)J^{-1}(q), \\ C_{ef}(q, \dot{q}) &= (C(q_m, \dot{q}) - M(q_m)J^{-1}(q)\dot{J}(q, \dot{q}))J^{-1}(q). \end{aligned}$$

It must be noted that the Jacobian, $J(q)$, introduces the values of spacecraft's attitude, q_b , in the dynamic equation matrices, M_{ef} and C_{ef} . This does not happen when the problem is formulated in joint space, [Taveira et al. (2006)]. Another remark is that, in this formulation, $M_{ef}(q)$ is not a symmetric positive definite matrix, neither $N_{ef}(q, \dot{q}) = \dot{M}_{ef}(q, \dot{q}) - 2C_{ef}(q, \dot{q})$ is skew-symmetric. In order to preserve the characteristics of dynamics formulated in joint space, a force transformation is applied to (4), [Lewis et al. (1993)]:

$$\tau = J^T(q)F, \quad (5)$$

where F is a vector of generalized forces of the end-effector in inertial space. Therefore,

$$F = \bar{M}_{ef}(q)\ddot{p} + \bar{C}_{ef}(q, \dot{q})\dot{p}, \quad (6)$$

with

$$\begin{aligned} \bar{M}_{ef}(q) &= J^{-T}(q)M_{ef}(q) = J^{-T}(q)M(q_m)J^{-1}(q), \\ \bar{C}_{ef}(q, \dot{q}) &= J^{-T}(q)C_{ef}(q, \dot{q}) = J^{-T}(q)(C(q_m, \dot{q}) - M(q_m)J^{-1}(q)\dot{J}(q, \dot{q}))J^{-1}(q). \end{aligned}$$

In this format the dynamic equation formulated in inertial space maintains the structure and properties found in joint space. So, $\bar{M}_{ef}(q)$ is symmetric positive definite and $\bar{N}_{ef}(q, \dot{q}) = \dot{\bar{M}}_{ef}(q, \dot{q}) - 2\bar{C}_{ef}(q, \dot{q})$ is skew-symmetric.

A characteristic inherited from underactuated manipulators, dealing with a system with n_a actuators leads to controlling only n_a degrees of freedom at a time, [Siqueira & Terra (2004)]. The DEM presents n active joints and, then, $n_a = n$. So, let's define $p = [p_u^T \ p_a^T]^T$ the vector of generalized coordinates of the system, with $p_u \in \mathbb{R}^{(6-n) \times 1}$ and $p_a \in \mathbb{R}^{n \times 1}$, where the indexes u and a represent the passive variables (which are let free during the control procedure) and the controlled variables, respectively. Partitioning equation (6), we have

$$\begin{bmatrix} F_u \\ F_a \end{bmatrix} = \begin{bmatrix} \bar{M}_{ef_{uu}}(q) & \bar{M}_{ef_{ua}}(q) \\ \bar{M}_{ef_{au}}(q) & \bar{M}_{ef_{aa}}(q) \end{bmatrix} \begin{bmatrix} \ddot{p}_u \\ \ddot{p}_a \end{bmatrix} + \begin{bmatrix} \bar{C}_{ef_{uu}}(q, \dot{q}) & \bar{C}_{ef_{ua}}(q, \dot{q}) \\ \bar{C}_{ef_{au}}(q, \dot{q}) & \bar{C}_{ef_{aa}}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{p}_u \\ \dot{p}_a \end{bmatrix}, \quad (7)$$

with $\bar{M}_{ef_{uu}} \in \mathbb{R}^{(6-n) \times (6-n)}$, $\bar{M}_{ef_{ua}} \in \mathbb{R}^{(6-n) \times n}$, $\bar{M}_{ef_{au}} \in \mathbb{R}^{n \times (6-n)}$, $\bar{M}_{ef_{aa}} \in \mathbb{R}^{n \times n}$, $\bar{C}_{ef_{uu}} \in \mathbb{R}^{(6-n) \times (6-n)}$, $\bar{C}_{ef_{ua}} \in \mathbb{R}^{(6-n) \times n}$, $\bar{C}_{ef_{au}} \in \mathbb{R}^{n \times (6-n)}$, $\bar{C}_{ef_{aa}} \in \mathbb{R}^{n \times n}$, $F_u \in \mathbb{R}^{(6-n) \times 1}$ e $F_a \in \mathbb{R}^{n \times 1}$. This decomposition should also preserve the properties of dynamic equation for the matrices $\bar{M}_{ef_{aa}}(q)$ and $\bar{C}_{ef_{aa}}(q, \dot{q})$:

- $\bar{M}_{ef_{aa}}(q) = \bar{M}_{ef_{aa}}^T(q) > 0$ and
- $\bar{N}_{ef_{aa}}(q, \dot{q}) = \dot{\bar{M}}_{ef_{aa}}(q, \dot{q}) - 2\bar{C}_{ef_{aa}}(q, \dot{q})$ is skew-symmetric.

Define $\delta = [\delta_u^T \ \delta_a^T]^T$ as a vector representing the sum of parametric uncertainties of the system and F_d as a finite energy external disturbance also introduced. Equation (7) can be rewritten as:

$$\begin{bmatrix} F_u \\ F_a \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_a \end{bmatrix} + F_d = \begin{bmatrix} \bar{M}_{ef_{uu}}(q) & \bar{M}_{ef_{ua}}(q) \\ \bar{M}_{ef_{au}}(q) & \bar{M}_{ef_{aa}}(q) \end{bmatrix} \begin{bmatrix} \ddot{p}_u \\ \ddot{p}_a \end{bmatrix} + \begin{bmatrix} \bar{C}_{ef_{uu}}(q, \dot{q}) & \bar{C}_{ef_{ua}}(q, \dot{q}) \\ \bar{C}_{ef_{au}}(q, \dot{q}) & \bar{C}_{ef_{aa}}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{p}_u \\ \dot{p}_a \end{bmatrix}, \quad (8)$$

where

$$\begin{bmatrix} \delta_u(q, \dot{q}, \ddot{p}, \ddot{p}, \tau_d) \\ \delta_a(q, \dot{q}, \ddot{p}, \ddot{p}, \tau_d) \end{bmatrix} = - \begin{bmatrix} \Delta \bar{M}_{ef_{uu}}(q) \ddot{p}_u + \Delta \bar{M}_{ef_{ua}}(q) \ddot{p}_a + \Delta \bar{C}_{ef_{uu}}(q, \dot{q}) \dot{p}_u + \Delta \bar{C}_{ef_{ua}}(q, \dot{q}) \dot{p}_a \\ \Delta \bar{M}_{ef_{au}}(q) \ddot{p}_u + \Delta \bar{M}_{ef_{aa}}(q) \ddot{p}_a + \Delta \bar{C}_{ef_{au}}(q, \dot{q}) \dot{p}_u + \Delta \bar{C}_{ef_{aa}}(q, \dot{q}) \dot{p}_a \end{bmatrix}$$

For simplicity of notation, the index 0 referring to the nominal system was suppressed.

Let $p_a^d \in \mathbb{R}^n$ and $\dot{p}_a^d \in \mathbb{R}^n$ be the desired reference trajectory and the corresponding velocity for the end-effector controlled variables, respectively. The state tracking error is defined as

$$\tilde{x}_{ef} = \begin{bmatrix} \dot{p}_a - \dot{p}_a^d \\ p_a - p_a^d \end{bmatrix} = \begin{bmatrix} \dot{\tilde{p}}_a \\ \tilde{p}_a \end{bmatrix}. \quad (9)$$

The variables p_a^d , \dot{p}_a^d , and \ddot{p}_a^d (desired acceleration) are assumed to be within the physical and kinematics limits of the control system and there exists no reference trajectory for the base. Also, assume that p_a^d , \dot{p}_a^d , and \ddot{p}_a^d , belong entirely to the path independent workspace (PIW), [Torres & Dubowsky (1992)], and therefore, they will not conduce to any dynamic singularity, i.e., $\det(J) \neq 0$ throughout the path.

Consider the following state transformation, [Johansson (1990)],

$$\tilde{z} = T_0 \tilde{x}_{ef} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\tilde{p}}_a \\ \tilde{p}_a \end{bmatrix}, \quad (10)$$

where $T_{11}, T_{12} \in \mathbb{R}^{n \times n}$ are constant matrices to be determined. From (8), (9) and (10), state space representation of the DEM is given by

$$\begin{aligned} \dot{\tilde{x}}_{ef} &= \bar{A}_{T_{ef}}(q, \dot{q}) \tilde{x}_{ef} + \bar{B}_{T_{ef}}(q) T_{11} (-\bar{F}(x_{e_{ef}}) - \bar{E}(x_{e_b}) + F_a + \delta_a + F_d) \\ &= \bar{A}_{T_{ef}}(q, \dot{q}) \tilde{x}_{ef} + \bar{B}_{T_{ef}}(q) u + \bar{B}_{T_{ef}}(q) \omega, \end{aligned} \quad (11)$$

where

$$\bar{A}_{T_{ef}}(q, \dot{q}) = T_0^{-1} \begin{bmatrix} -\bar{M}_{ef_{aa}}^{-1}(q) \bar{C}_{ef_{aa}}(q, \dot{q}) & 0 \\ I & 0 \end{bmatrix} T_0,$$

$$\bar{B}_{T_{ef}}(q) = T_0^{-1} \begin{bmatrix} \bar{M}_{ef_{aa}}^{-1}(q) \\ 0 \end{bmatrix},$$

$$\bar{F}(x_{e_{ef}}) = \bar{M}_{ef_{aa}}(q) (\dot{p}_a^d - T_{11}^{-1} T_{12} \dot{\tilde{p}}_a) + \bar{C}_{ef_{aa}}(q, \dot{q}) (\dot{p}_a^d - T_{11}^{-1} T_{12} \dot{\tilde{p}}_a),$$

$$\bar{E}(x_{e_b}) = \bar{M}_{ef_{au}}(q) \dot{p}_u + \bar{C}_{ef_{au}}(q, \dot{q}) \dot{p}_u,$$

$$u = T_{11} (-\bar{F}(x_{e_{ef}}) - \bar{E}(x_{e_b}) + F_a), \quad \text{and}$$

$$\omega = T_{11} (\delta_a + F_d),$$

with $x_{e_{ef}} = [q^T \ \dot{q}^T \ p_a^T \ \dot{p}_a^T \ (p_a^d)^T \ (\dot{p}_a^d)^T \ (\dot{p}_a^d)^T]^T$ and $x_{e_b} = [q^T \ \dot{q}^T \ \dot{p}_u^T \ (\dot{p}_u)^T]^T$.

The vector of generalized forces related to active variables, F_a , comes from the second line of equation (7):

$$F_a = \bar{M}_{ef_{au}}(q) \dot{p}_u + \bar{M}_{ef_{aa}}(q) \dot{p}_a + \bar{C}_{ef_{au}}(q, \dot{q}) \dot{p}_u + \bar{C}_{ef_{aa}}(q, \dot{q}) \dot{p}_a. \quad (12)$$

Decomposing (4) as in (7), the 2nd order non-holonomic constraint imposed by the free-floating base (passive joint at MDE) to the system is described by

$$M_{ef_{uu}}(q) \dot{p}_u + M_{ef_{ua}}(q) \dot{p}_a + C_{ef_{uu}}(q) \dot{p}_u + C_{ef_{ua}}(q) \dot{p}_a = 0. \quad (13)$$

From (12) and (13), we have

$$F_a = M_{F_a}(q) \dot{p}_a + C_{F_a}(q, \dot{q}) \dot{p}_a, \quad (14)$$

where

$$\begin{aligned} M_{F_a}(q) &= \bar{M}_{ef_{aa}}(q) - \bar{M}_{ef_{au}}(q) M_{ef_{uu}}^{-1}(q) M_{ef_{ua}}(q), \\ C_{F_a}(q, \dot{q}) &= \left[\bar{C}_{ef_{au}}(q, \dot{q}) - \bar{M}_{ef_{au}}(q) M_{ef_{uu}}^{-1}(q) C_{ef_{uu}}(q, \dot{q}) \quad \bar{C}_{ef_{aa}}(q, \dot{q}) - \bar{M}_{ef_{au}}(q) M_{ef_{uu}}^{-1}(q) C_{ef_{ua}}(q, \dot{q}) \right]. \end{aligned}$$

The controller applied to the state dynamic equation, (11), provides the necessary value of \dot{p}_a to set F_a that leads to the desired trajectory:

$$\dot{p}_a = \dot{p}_a^d - T_{11}^{-1} T_{12} \dot{\tilde{p}}_a - T_{11}^{-1} \bar{M}_{ef_{aa}}^{-1}(q) (\bar{C}_{ef_{aa}}(q, \dot{q}) B^T T_0 \tilde{x}_{ef} - \bar{u}). \quad (15)$$

Applying back the force transformation, (5), one obtains the value of joint torques as:

$$\begin{bmatrix} 0 \\ \tau_a \end{bmatrix} = \begin{bmatrix} J_{uu}(q) & J_{ua}(q) \\ J_{au}(q) & J_{aa}(q) \end{bmatrix}^T \begin{bmatrix} F_u \\ F_a \end{bmatrix}. \quad (16)$$

Since $J_{uu}(q)$ is admitted invertible, F_u is taken from the first line of (16) as $F_u = -J_{uu}^{-T}(q)J_{au}^T(q)F_a$, and replaced at the second line of (16), taking to

$$\tau_a = (-J_{ua}^T(q)J_{uu}^{-T}(q)J_{au}^T(q) + J_{aa}^T(q))F_a. \quad (17)$$

3. Robust Controller Design

3.1 Nonlinear \mathcal{H}_∞ Control

The state feedback \mathcal{H}_∞ control strategy proposed in [Chen et al. (1994)], seeks the disturbance attenuation in the system by a control law of the form $u = K(x)x$ in order to satisfy

$$\min_{u(\cdot) \in \mathcal{L}_2} \max_{0 \neq \omega(\cdot) \in \mathcal{L}_2} \frac{\int_0^\infty \left(\frac{1}{2} \tilde{x}_{ef}^T(t) Q \tilde{x}_{ef}(t) + \frac{1}{2} u^T(t) R u(t) \right) dt}{\int_0^\infty \left(\frac{1}{2} \omega^T(t) \omega(t) \right) dt} \leq \gamma^2, \quad (18)$$

where Q and R are symmetric positive definite weighting matrices defined by the designer, $\gamma > 0$ is the desired disturbance attenuation level, ω is referred to the disturbance term in (11) and $\tilde{x}_{ef}(0) = 0$. Following the game theory, the known solution of this problem is given, in a simplified form, in terms of the algebraic equation

$$\begin{bmatrix} 0 & K \\ K & 0 \end{bmatrix} - T_0^T B \left(R^{-1} - \frac{1}{\gamma^2} I \right) B^T T_0 + Q = 0, \quad (19)$$

with $B = [I \quad 0]^T$. Therefore, to solve the \mathcal{H}_∞ problem, it is only necessary to find matrices K and T_0 which solve (19). Considering the matrix R_1 the result of the Cholesky factorization

$$R_1^T R_1 = \left(R^{-1} - \frac{1}{\gamma^2} I \right)^{-1},$$

and Q factorized as

$$Q = \begin{bmatrix} Q_1^T Q_1 & Q_{12} \\ Q_{12}^T & Q_2^T Q_2 \end{bmatrix},$$

the solution of (19) is given by

$$T_0 = \begin{bmatrix} R_1^T Q_1 & R_1^T Q_2 \\ 0 & I \end{bmatrix} \quad \text{and} \quad K = \frac{1}{2} (Q_1^T Q_2 - Q_2^T Q_1) - \frac{1}{2} (Q_{21}^T + Q_{12}),$$

with the conditions: $K > 0$ and $R < \gamma^2 I$. The optimal control input is established for the proposed application in the following.

Given a desired disturbance attenuation level $\gamma > 0$, the \mathcal{H}_∞ control problem (18) subject to (11) has an optimal solution

$$\bar{u}^* = -R^{-1} B^T T_0 \tilde{x}_{ef}, \quad (20)$$

if $\gamma^2 I > R$ and if there exist matrices $K > 0$ and a non-singular T_0 solutions of (19). The forces related to active variables can be computed applying (20) at (15), and then, using this result at (14). Joint torques are computed by (17).

Remark 1. This nonlinear \mathcal{H}_∞ control strategy assumes that the model structure is completely known and represents parameter uncertainties as internal disturbances, treating them the same way as external disturbances.

Remark 2. This model-based nonlinear \mathcal{H}_∞ control strategy does not demand measured values of acceleration from the free-floating base neither from the arm.

3.2 Adaptive Neural Network Nonlinear \mathcal{H}_∞ Control

Define a set of n neural networks $E_k(x_e, \Theta_k)$, $k = 1, \dots, n$, where x_e is the input vector and Θ_k are the adjustable weights in the output layers. The single-output neural networks are of the form

$$E_k(x_e, \Theta_k) = \sum_{i=1}^{p_k} \theta_{ki} G \left(\sum_{j=1}^{q_k} w_{ij}^k x_{ej} + b_i^k \right) = \zeta_k^T \Theta_k, \quad (21)$$

where q_k is the size of vector x_e and p_k is the number of neurons in the hidden layer. The weights w_{ij}^k and the biases b_i^k for $1 \leq i \leq p_k$, $1 \leq j \leq q_k$ and $1 \leq k \leq n$ are assumed to be constant and specified by the designer. Thus, the adjustment of neural networks is performed only by updating the vectors Θ_k . The activation function for the neurons in the hidden layer is chosen to be $G(\cdot) = \tanh(\cdot)$. The complete neural network is denoted by

$$E(x_e, \Theta) = \begin{bmatrix} E_1(x_e, \Theta_1) \\ \vdots \\ E_n(x_e, \Theta_n) \end{bmatrix} = \begin{bmatrix} \zeta_1^T & 0 & \dots & 0 \\ 0 & \zeta_2^T & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \zeta_n^T \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_n \end{bmatrix} = \Xi \Theta. \quad (22)$$

Consider a first approach where the term

$$E^1(x_{e_{ef}}, x_{e_b}) = \bar{F}(x_{e_{ef}}) + \bar{E}(x_{e_b}) - \delta_a$$

in (11) is completely unknown regarding its structure and parameter values. The neural network defined in (22) is applied to learn the dynamic behavior of the robotic system:

$$E^1(x_{e_{ef}}, x_{e_b}) \approx \hat{E}(x_e, \Theta) = \Xi \Theta, \quad (23)$$

where the input vector x_e should be defined as

$$x_e = [q^T \quad \dot{q}^T \quad p_a^T \quad \dot{p}^T \quad \ddot{p}_u^T \quad (p_a^d)^T \quad (\dot{p}_a^d)^T \quad (\ddot{p}_a^d)^T]^T.$$

However, the values of q_b , \dot{q}_b , \dot{p}_u and \ddot{p}_u would be necessary, but they are not easy to obtain in practice. Considering that a neural network based approach is usually used when it is not possible to supply all the variables values to the system model, we have defined the vector x_e as

$$x_e = [q_m^T \quad \dot{q}_m^T \quad (p_a^d)^T \quad (\dot{p}_a^d)^T \quad (\ddot{p}_a^d)^T]^T, \quad (24)$$

avoiding the necessity of any data from the free-floating base or related to passive variables. Simulation results will show the feasibility of this assumption. Defining the following optimization problem

$$\Theta^* = \arg \min_{\Theta \in \Omega_\Theta} \max_{\tilde{x}_{ef} \in \Omega_{\tilde{x}_{ef}}} \left\| \hat{E}(x_e, \Theta) - E^1(x_{e_{ef}}, x_{e_b}) \right\|_2,$$

the modified error equation (11) may be rewritten as

$$\begin{aligned}
 \dot{\tilde{x}}_{ef} &= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)T_{11}(F_a - E^1(x_{ef}, x_{eb}) + F_d + \hat{E}(x_e, \Theta^*) - \hat{E}(x_e, \Theta)) \\
 &= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)T_{11}(F_a - \hat{E}(x_e, \Theta^*)) + \bar{B}_{T_{ef}}(q)T_{11}(\hat{E}(x_e, \Theta^*) - E^1(x_{ef}, x_{eb}) + F_d) \\
 &= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)u + \bar{B}_{T_{ef}}(q)\omega,
 \end{aligned} \tag{25}$$

with

$$u = T_{11}(F_a - \hat{E}(x_e, \Theta^*)), \tag{26}$$

$$\omega = T_{11}(\hat{E}(x_e, \Theta^*) - E^1(x_{ef}, x_{eb}) + F_d), \tag{27}$$

where ω refers to the estimation error from the neural network system and external disturbances. Considering $u = \bar{u}$ the control law provided by the nonlinear \mathcal{H}_∞ controller in (20), F_a can be computed by

$$F_a = \hat{E}(x_e, \Theta^*) + T_{11}^{-1}\bar{u}. \tag{28}$$

Thus, considering the stability analysis developed by [Chang & Chen (1997)], the adaptive neural network nonlinear \mathcal{H}_∞ control is stated for the proposed application as follows.

Let $E(x_e, \Theta)$ be a set of n neural networks defined by (22) with x_e being a vector of available data defined by (24) and Θ being a vector of adjustable parameters. Given a desired disturbance attenuation level $\gamma > 0$ and matrices $Z = Z^T > 0, Q = Q^T > 0, P_0 = P_0^T > 0, Z_0 = Z_0^T > 0$, and $R = R^T < \gamma^2 I$, the following performance criterion

$$\int_0^T (\tilde{x}_{ef}^T Q \tilde{x}_{ef} + \bar{u}^T R \bar{u}) dt \leq \tilde{x}_{ef}^T(0)P_0\tilde{x}_{ef}(0) + \tilde{\Theta}^T(0)Z_0\tilde{\Theta}(0) + \gamma^2 \int_0^T (\omega^T \omega) dt, \tag{29}$$

where $\tilde{\Theta} = \Theta - \Theta^*$ denotes the neural parameter estimation error, is satisfied, for any initial condition, if there exists a dynamic state feedback controller

$$\dot{\tilde{\Theta}} = \beta(t, \tilde{x}_{ef}) = -Z^{-T}\Xi^T T_{11}B^T T_0 \tilde{x}_{ef}, \tag{30}$$

$$F_a = F_a(t, \tilde{\Theta}, \tilde{x}_{ef}) = \Xi\tilde{\Theta} - T_{11}^{-1}R^{-1}B^T T_0 \tilde{x}_{ef}, \tag{31}$$

solution of the adaptive neural network nonlinear \mathcal{H}_∞ control problem subject to (25). The torques applied upon the joints are given by (17).

On the other hand, a second approach may be proposed. Consider that model structure and nominal values for the term $\bar{F}(x_{ef})$ are well defined and available for the controller. In this case, the neural network is applied to estimate only the behavior of parametric uncertainties and spacecraft dynamics (considered as a non-modeled dynamic):

$$E^2(x_{ef}, x_{eb}) \approx \hat{E}(x_e, \Theta) = \Xi\Theta, \tag{32}$$

where $E^2(x_{ef}, x_{eb}) = \bar{E}(x_{eb}) - \delta_a$.

Similarly, x_e is defined by (24), the optimal approximation parameters vector is given by

$$\Theta^* = \arg \min_{\Theta \in \Omega_\Theta} \max_{\tilde{x}_{ef} \in \Omega_{\tilde{x}_{ef}}} \left\| \hat{E}(x_e, \Theta^*) - E^2(x_{ef}, x_{eb}) \right\|_2,$$

and the modified error equation (11) may be rewritten as

$$\begin{aligned}
\dot{\tilde{x}}_{ef} &= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)T_{11}(F_a - \bar{F}(x_{ef}) - E^2(x_{ef}, x_{eb}) + F_d + \hat{E}(x_e, \Theta^*) - \hat{E}(x_e, \Theta^*)) \\
&= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)T_{11}(F_a - \bar{F}(x_{ef}) - \hat{E}(x_e, \Theta^*)) + \bar{B}_{T_{ef}}(q)T_{11}(\hat{E}(x_e, \Theta^*) + \\
&\quad - E^2(x_{ef}, x_{eb}) + F_d) \\
&= \bar{A}_{T_{ef}}(q, \dot{q})\tilde{x}_{ef} + \bar{B}_{T_{ef}}(q)u + \bar{B}_{T_{ef}}(q)\omega,
\end{aligned} \tag{33}$$

with

$$u = T_{11}(F_a - \bar{F}(x_{ef}) - \hat{E}(x_e, \Theta^*)), \tag{34}$$

$$\omega = T_{11}(\hat{E}(x_e, \Theta^*) - E^2(x_{ef}, x_{eb}) + F_d), \tag{35}$$

where ω refers to the estimation error from the neural network system and external disturbances. Considering $u = \bar{u}$ the control law provided by the nonlinear \mathcal{H}_∞ controller in (20), F_a can be computed by

$$F_a = \bar{F}(x_{ef}) + \hat{E}(x_e, \Theta^*) + T_{11}^{-1}\bar{u}. \tag{36}$$

For this approach, the adaptive neural network nonlinear \mathcal{H}_∞ control can be enunciated as follows.

Let $E(x_e, \Theta)$ be a set of n neural networks defined by (22) with x_e being a vector of available data defined by (24) and Θ being a vector of adjustable parameters. Given a desired disturbance attenuation level $\gamma > 0$ and matrices $Z = Z^T > 0, Q = Q^T > 0, P_0 = P_0^T > 0, Z_0 = Z_0^T > 0$, and $R = R^T < \gamma^2 I$, the following performance criterion

$$\int_0^T (\tilde{x}_{ef}^T Q \tilde{x}_{ef} + \bar{u}^T R \bar{u}) dt \leq \tilde{x}_{ef}^T(0)P_0\tilde{x}_{ef}(0) + \tilde{\Theta}^T(0)Z_0\tilde{\Theta}(0) + \gamma^2 \int_0^T (\omega^T \omega) dt, \tag{37}$$

where $\tilde{\Theta} = \Theta - \Theta^*$ denotes the neural parameter estimation error, is satisfied, for any initial condition, if there exists a dynamic state feedback controller

$$\dot{\tilde{\Theta}} = \beta(t, \tilde{x}_{ef}) = -Z^{-T}\Xi^T T_{11} B^T T_0 \tilde{x}_{ef}, \tag{38}$$

$$F_a = F_a(t, \tilde{\Theta}, \tilde{x}_{ef}) = \bar{F}(x_{ef}) + \Xi\tilde{\Theta} - T_{11}^{-1}R^{-1}B^T T_0 \tilde{x}_{ef}, \tag{39}$$

solution of the adaptive neural network nonlinear \mathcal{H}_∞ control problem subject to (33). The stability analysis developed in [Chang & Chen (1997)] is also valid for this case, [Petronilho et al. (2005)]. The torques applied upon the joints are given by (17).

Remark 3. *The adaptive neural network nonlinear \mathcal{H}_∞ strategies do not demand measured values of acceleration from the free-floating base neither from the arm.*

Remark 4. *The adaptive designs proposed apply an intelligent learning strategy to estimate uncertain parameters and also the behavior of non-modeled dynamics. The \mathcal{H}_∞ control law is applied to attenuate the effects of estimation errors and external disturbances.*

4. Results

For validation and comparison purposes, the proposed adaptive \mathcal{H}_∞ control solutions are applied to a free-floating, planar, two-link space manipulator system, whose nominal parameters are given in Table 1. The corresponding DEM is a fixed-base, three-link, planar manipulator whose first joint is configured as passive, that is, $q_m = [q_2 \ q_3]^T$ are the joints to be controlled. Its structure is based on the fixed-base manipulator UArmII (Underactuated Arm II), whose nominal parameters are given in Table 2. Nominal matrices $M(q)$ and $C(q, \dot{q})$ for the DEM can be found in [Liang et al. (1996)] and the Jacobian is given in the appendix.

Body	m_i' (kg)	I_i' (kgm ²)	R_i (m)	L_i (m)
Base	4.816	0.008251	0.253	0
Link 2	0.618	0.0075	0.118	0.12
Link 3	0.566	0.006	0.126	0.085

Table 1. SM Parameters

Body	m_i (kg)	I_i (kgm ²)	W_i (m)	l_{c_i} (m)
Link 1	1.932	0.008251	0.203	0
Link 2	0.850	0.0075	0.203	0.096
Link 3	0.625	0.006	0.203	0.077

Table 2. DEM Parameters

A trajectory tracking task is defined for the space manipulator end-effector. The Cartesian positions $p_a = [x_{ef} \ y_{ef}]^T$ of the end-effector are chosen to be the controlled variables, while its orientation ϕ_{ef} is let free. The reference trajectory is defined as a semi-circle starting at the end-effector initial position (set by $q(0) = [0^\circ \ 20^\circ \ -40^\circ]^T$) and characterized by radius = 5 cm. The angles that determine the semi-circle reference trajectory follows a fifth degree polynomial with $t_f = 3s$ (time defined for the task execution). During the simulation, a limited disturbance, initializing at $t = 1s$, was introduced in the following form

$$\tau_d = \begin{bmatrix} 0.025e^{-2t} \sin(2\pi t) \\ 0.015e^{-2t} \sin(2\pi t) \end{bmatrix}.$$

Compared to the torque applied in case that none disturbance is inserted, the disturbance τ_d presents peaks of approximately 75% of that torque peak value. Multiplicative uncertainties were also applied to the values of mass, moment of inertia, length and center of mass position as $\delta = [0.7 * m \ 1.2 * I \ 1.7 * W \ 0.5 * l_c]$.

In order to clearly identify the controllers actuation, Figures 2 to 5 illustrate the results obtained without adding disturbances and uncertainties to the model (nominal case) while Figures 6 to 9 show the results for the disturbed situation (disturbed case). To establish a basis for comparison, define a computed torque control by

$$\ddot{p}_a = K_p \tilde{p}_a + K_d \dot{\tilde{p}}_a + K_i \int \tilde{p}_a + \ddot{p}_a^d, \quad (40)$$

with $K_p = 50I_2$, $K_d = \begin{bmatrix} 50\sqrt{|K_p|} & 0 \\ 0 & 1.5\sqrt{|K_p|} \end{bmatrix}$ and $K_i = I_2$. Applying (40) to (14) provides

the forces related to active variables and joint torques are computed by (17). Simulation results are presented by Figures 2 and 6.

The level of disturbance attenuation defined for the proposed nonlinear \mathcal{H}_∞ controllers is $\gamma = 2$. The selected weighting matrices are shown in Table 3. Figures 3 and 7 are the results obtained by the nonlinear \mathcal{H}_∞ control described in Section 3.1. For the nonlinear \mathcal{H}_∞ controllers via neural network proposed, let $n = 2$ be the size of p_a determined by the number

of joints of the space manipulator (active joints in DEM), which define the size of $x_e, q_k = 10$. Define $E(x_e, \Theta) := [E_1(x_e, \Theta_1) \ E_2(x_e, \Theta_2)]^T$ with $p_k = 7$ neurons in the hidden layer, the bias vector $b_k = [-3 \ -2 \ -1 \ 0 \ 1 \ 2 \ 3]$ and the weighting matrix for the first layer

$$\Omega_i^k = [\omega_{ij}^k] = [1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1].$$

The uncertain vector Θ is defined as $\Theta = [\Theta_1 \ \Theta_2]^T$, with $\Theta_1^T = [\theta_{11} \ \dots \ \theta_{17}]$ and $\Theta_2^T = [\theta_{21} \ \dots \ \theta_{27}]$, and the matrix Ξ can be computed with $\xi_1^T = [\xi_{11} \ \dots \ \xi_{17}]$ and $\xi_2^T = [\xi_{21} \ \dots \ \xi_{27}]$. Simulation results for the adaptive neural network nonlinear \mathcal{H}_∞ controller are shown in Figures 4 and 8. For the second approach, neural network plus nominal model, simulation results are shown in Figures 5 and 9.

$\gamma = 2$	R	Q_1	Q_2	Z
Nonlinear \mathcal{H}_∞	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 49 & 0 \\ 0 & 9 \end{bmatrix}$	-
Adaptive Neural \mathcal{H}_∞ (1)	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 64 \end{bmatrix}$	$\begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$
Adaptive Neural \mathcal{H}_∞ (2)	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$	$\begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$

Table 3. Selected Weighting Matrices

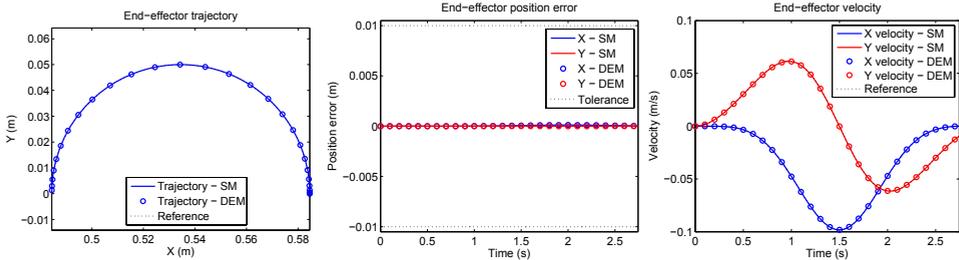


Fig. 2. Nominal case: End-effector trajectory, position errors and velocities - Computed torque

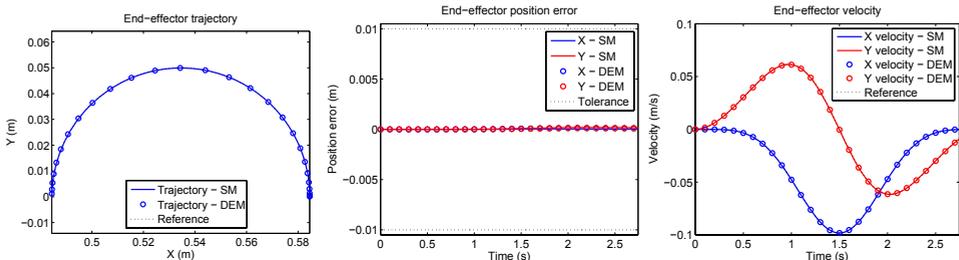


Fig. 3. Nominal case: End-effector trajectory, position errors and velocities - Nonlinear \mathcal{H}_∞

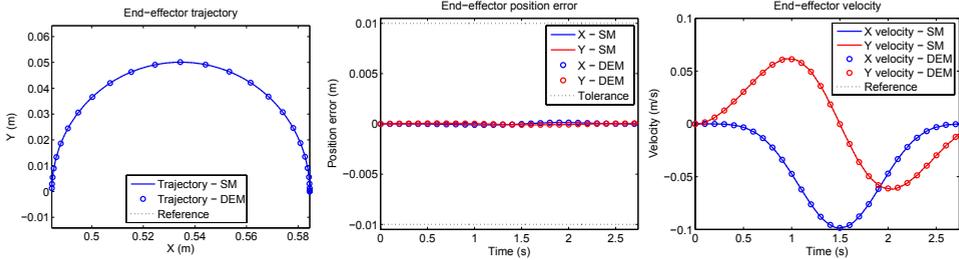


Fig. 4. Nominal case: End-effector trajectory, position errors and velocities - Adaptive Neural Network Nonlinear $\mathcal{H}_\infty (1)$

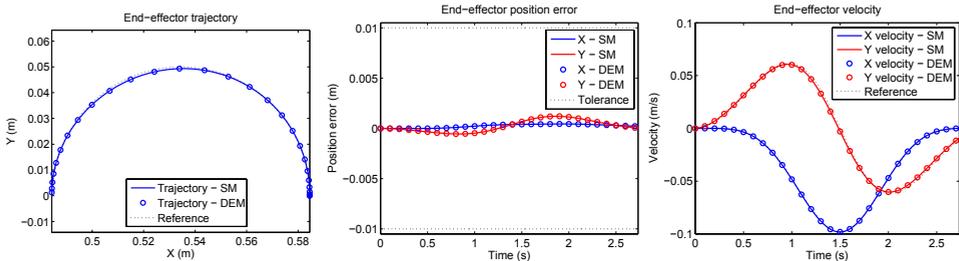


Fig. 5. Nominal case: End-effector trajectory, position errors and velocities - Adaptive Neural Network Nonlinear $\mathcal{H}_\infty (2)$

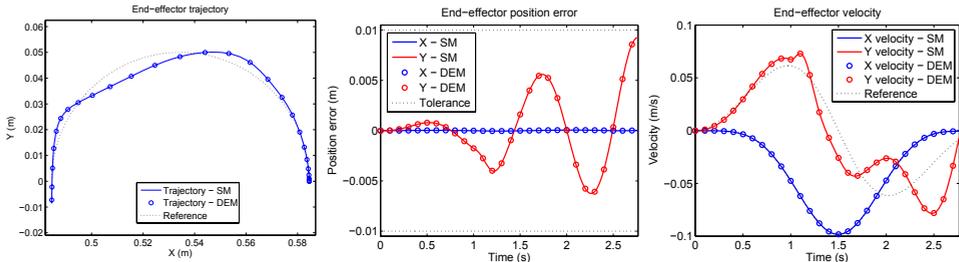


Fig. 6. Disturbed case: End-effector trajectory, position errors and velocities - Computed torque

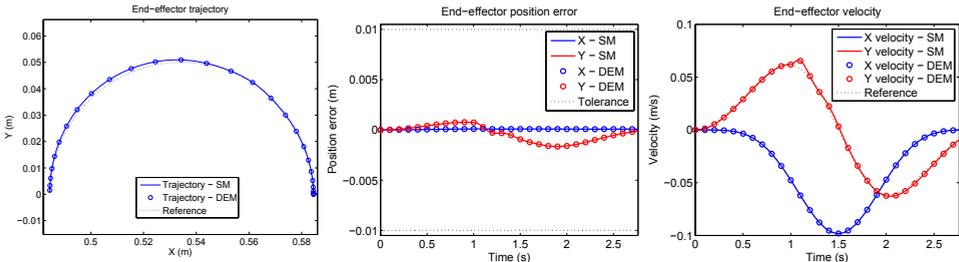


Fig. 7. Disturbed case: End-effector trajectory, position errors and velocities - Nonlinear \mathcal{H}_∞

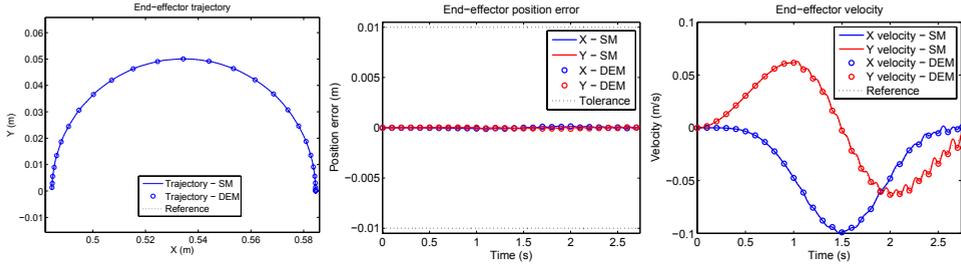


Fig. 8. Disturbed case: End-effector trajectory, position errors and velocities - Adaptive Neural Network Nonlinear \mathcal{H}_∞ (1)

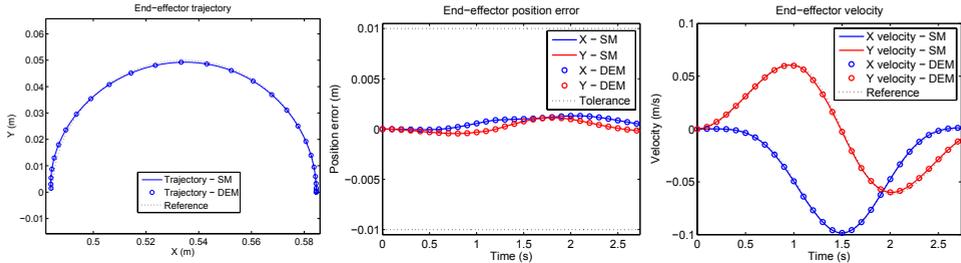


Fig. 9. Disturbed case: End-effector trajectory, position errors and velocities - Adaptive Neural Network Nonlinear \mathcal{H}_∞ (2)

A comparison among the graphical results illustrate that the proposed controllers reject disturbance efficiently and attenuate its effect in the trajectory tracking task. The computed torque control, however, presents an excellent result in Figure 2, where the plant model is completely defined and no disturbances and uncertainties are applied, but it is not able to reject the disturbances and uncertainties effects when they occur, Figure 6. These results clearly demonstrate the robustness of \mathcal{H}_∞ criterion.

Considering that the same value of γ was applied to all the proposed controllers, two performance indexes are used to numerically compare the controllers applied: the \mathcal{L}_2 norm of the state vector

$$\mathcal{L}_2[\tilde{x}_{ef}] = \left(\frac{1}{(t_r - t_0)} \int_{t_0}^{t_r} \|\tilde{x}_{ef}(t)\|_2^2 dt \right)^{1/2},$$

where $\|\cdot\|_2$ is the Euclidean norm, and the sum of the applied torques

$$E[\tau_a] = \sum_{i=1}^2 \left(\int_{t_0}^{t_r} |\tau_{a_i}(t)| dt \right).$$

The results are presented in Table 4 as follows. A quantitative analysis based on Table 4 endorses the results verified by the graphics. The robustness provided by the nonlinear \mathcal{H}_∞ control law is shown by the $\mathcal{L}_2[\tilde{x}_{ef}]$ results when compared to the results presented by the computed torque method. The results presented by the adaptive neural networks approaches exhibits their efficiency in estimating the effect of uncertainties, and mainly, the non-modeled dynamics of the spacecraft. The energy spent by the first approach of adaptive neural network strategy is greater than the one spent by strategy (2), however, their error avoidance capacity

	Nominal case		Disturbed case	
	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$
Computed Torque	1.24×10^{-4}	0.0449	0.0210	0.0592
Nonlinear \mathcal{H}_∞	1.39×10^{-4}	0.0449	0.0024	0.0458
Adaptive Neural \mathcal{H}_∞ (1)	3.63×10^{-4}	0.0452	0.0022	0.0559
Adaptive Neural \mathcal{H}_∞ (2)	16×10^{-4}	0.0442	0.0020	0.0444

Table 4. Performance Indexes

are very similar. Thus, the designer should ponder between energy demand and availability of the plant model. It must be emphasized that none of the proposed \mathcal{H}_∞ approaches demands measured acceleration values from the free-floating base or the arm.

5. Conclusion

This chapter presents an investigation on the motion control of a free-floating space manipulator subject to parametric uncertainties and external disturbances performed by different methods of nonlinear \mathcal{H}_∞ controllers.

The free-floating space manipulator model is developed based on the Dynamically Equivalent Manipulator concept. Trajectory tracking of the SM's end-effector in task (Cartesian) space is considered including the existence of parametric uncertainties in the model, the application of external disturbances and considering unknown the spacecraft dynamics. Nonlinear \mathcal{H}_∞ control techniques are designed according to the knowledge and availability of the parameter matrices for the controllers. Simulations in a free-floating space planar manipulator with two links were evaluated to demonstrate the effectiveness of the presented approaches. A qualitative analysis of trajectory tracking graphics identifies the action of the proposed control laws. A quantitative comparison among the strategies is provided by performance indexes regarding energy consumption and error avoidance.

The nonlinear \mathcal{H}_∞ controller procedure demands a precise knowledge of the model structure, but its efficiency in attenuating the effects of parametric uncertainties and external disturbances was verified before the results of a computed torque controller. The first approach for adaptive neural network \mathcal{H}_∞ controller applies the intelligent system to learn the dynamic behavior of the robotic system, which is considered totally unknown. This strategy proved to be very effective in estimating the non-modeled plant behavior. The proposal of combining the model-based technique (once the model is largely known) and the intelligent adaptive tool, joined the best characteristics of both strategies, robustness and flexibility, and exhibited the best numerical results.

The space applications area of research is wide and full of unique issues. Future research on the topic presented in this chapter should include application of the proposed strategies in a real robot arm, supported by the modeling concept used in this chapter, and also the proposal of other intelligent techniques, such as fuzzy systems.

6. References

- Abiko, S. & Hirzinger, G. (2009). *Frontiers in Adaptive Control*, IN-TECH, chapter An Adaptive Control for a Free-Floating Space Robot by Using Inverted Chain Approach, pp. 1–16.

- Caccavale, F. & Siciliano, B. (2001). Quaternion-based kinematic control of redundant spacecraft/manipulator systems, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 1, Seoul, Korea, pp. 435–440. doi:10.1109/ROBOT.2001.932589.
- Chang, Y. C. & Chen, B. S. (1997). A nonlinear adaptive \mathcal{H}_∞ tracking control design in robotic systems via neural networks, *IEEE Transactions on Control Systems Technology* 5(1): 13–29. doi:10.1109/87.553662.
- Chen, B. S., Lee, T. S. & Feng, J. H. (1994). A nonlinear \mathcal{H}_∞ control design in robotic systems under parameter perturbation and external disturbance, *International Journal of Control* 59(2): 439–461. doi:10.1080/00207179408923085.
- Dubowsky, S. & Papadopoulos, E. (1993). The kinematics, dynamics, and control of free-flying and free-floating space robotic systems, *IEEE Transactions on Robotics and Automation* 9(5): 531–543. doi:10.1109/70.258046.
- Dubowsky, S. & Torres, M. A. (1991). Path planning for space manipulators to minimize spacecraft attitude disturbances, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 3, Sacramento, USA, pp. 2522–2528. doi:10.1109/ROBOT.1991.132005.
- Fu, Y., Zhang, F., Wang, S. & Chi, Z. (2007). A dynamic control method for free-floating space manipulator in task space, *IEEE International Conference on Robotics and Biomimetics*, Sanya, China, pp. 1230–1235. doi:.
- Gu, Y. L. & Xu, Y. (1993). A normal form augmentation approach to adaptive control of space robot systems, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 2, Atlanta, USA, pp. 731–737. doi:10.1109/ROBOT.1993.291872.
- Huang, P. & Xu, Y. (2006). Pso-based time-optimal trajectory planning for space robot with dynamic constraints, *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Kunming, China, pp. 1402–1407. doi:10.1109/ROBIO.2006.340134.
- Johansson, R. (1990). Quadratic optimization of motion coordination and control, *IEEE Transactions on Automatic Control* 35(11): 1197–1208. doi:10.1109/9.59805.
- Lewis, F. L., Abdallah, C. T. & Dawson, D. M. (1993). *Control of Robot Manipulators*, Macmillan Publishing Company, New York.
- Liang, B., Xu, Y. & Bergerman, M. (1996). Mapping a space manipulator to a dynamically equivalent manipulator, *Technical Report CMU-RI-TR-96-33*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. doi:10.1.1.67.6435.
- Liu, Z., Huang, P., Yan, J. & Liu, G. (2009). Multi-objective genetic algorithms for trajectory optimization of space manipulator, *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 4th, Xi'an, China, pp. 2810–2815. doi:10.1109/ICIEA.2009.5138722.
- Papadopoulos, E. & Dubowsky, S. (1991). On the nature of control algorithms for free-floating space manipulators, *IEEE Transactions on Robotics and Automation* 7(6): 750–758. doi:10.1109/70.105384.
- Papadopoulos, E., Tortopidis, I. & Nanos, K. (2005). Smooth planning for free-floating space robots using polynomials, *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, pp. 4272–4277.
- Parlaktuna, O. & Ozkan, M. (2004). Adaptive control of free-floating space robots in cartesian coordinates, *Advanced Robotics* 18(9): 943–959. doi:10.1163/1568553042225732.
- Petronilho, A., Siqueira, A. A. G. & Terra, M. H. (2005). Adaptive \mathcal{H}_∞ tracking control design via neural networks of a constrained robot system, *IEEE Conference on Decision and Control*, 44th, Seville, Spain, pp. 5528–5533.

- Siqueira, A. A. G. & Terra, M. H. (2004). A fault tolerant robot manipulator based on \mathcal{H}_2 , \mathcal{H}_∞ and mixed $\mathcal{H}_2/\mathcal{H}_\infty$ markovian controls, *IEEE Conference on Control Applications (CCA)*, Taipei, Taiwan.
- Taveira, T. F. P. A., Siqueira, A. A. G. & Terra, M. H. (2006). Adaptive nonlinear \mathcal{H}_∞ controllers applied to a free-floating space manipulator, *IEEE International Conference on Control Applications (CCA)*, Munich, Germany, pp. 1476–1481. doi:10.1109/CCA.2006.286094.
- Torres, M. A. & Dubowsky, S. (1992). Minimizing spacecraft attitude disturbances in space manipulators systems, *AIAA Journal of Guidance, Control, and Dynamics* **15**(4): 1010–1017. doi:10.2514/3.20936.
- Tortopidis, I. & Papadopoulos, E. (2006). Point-to-point planning: Methodologies for underactuated space robots, *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, USA, pp. 3861–3866. doi:10.1109/ROBOT.2006.1642293.

A. Appendix

Let the auxiliary variables be:

$$\begin{aligned} r_1 &= \frac{R_1 m_1}{Mt}, \\ r_2 &= \frac{R_2(m_1 + m_2)}{Mt}, \\ r_3 &= \frac{R_3(m_1 + m_2 + m_3)}{Mt}, \\ l_2 &= \frac{L_2 m_1}{Mt}, \\ l_3 &= \frac{L_3(m_1 + m_2)}{Mt}. \end{aligned}$$

The Jacobian is given by

$$J(q) = \begin{bmatrix} J_{11}(q) & J_{12}(q) & J_{13}(q) \\ J_{21}(q) & J_{22}(q) & J_{23}(q) \\ J_{31}(q) & J_{32}(q) & J_{33}(q) \end{bmatrix}$$

$$\begin{aligned} J_{11}(q) &= J_{12}(q) = J_{13}(q) = 1, \\ J_{21}(q) &= -r_1 \sin(q_1) - (r_2 + l_2) \sin(q_1 + q_2) - (r_3 + l_3) \sin(q_1 + q_2 + q_3), \\ J_{22}(q) &= -(r_2 + l_2) \sin(q_1 + q_2) - (r_3 + l_3) \sin(q_1 + q_2 + q_3), \\ J_{23}(q) &= -(r_3 + l_3) \sin(q_1 + q_2 + q_3), \\ J_{31}(q) &= r_1 \cos(q_1) + (r_2 + l_2) \cos(q_1 + q_2) + (r_3 + l_3) \cos(q_1 + q_2 + q_3), \\ J_{32}(q) &= (r_2 + l_2) \cos(q_1 + q_2) + (r_3 + l_3) \cos(q_1 + q_2 + q_3), \\ J_{33}(q) &= (r_3 + l_3) \cos(q_1 + q_2 + q_3). \end{aligned}$$

Neural and Adaptive Control Strategies for a Rigid Link Manipulator

Dorin Popescu, Dan Selişteanu, Cosmin Ionete,
Monica Roman and Livia Popescu
*Department of Automation and Mechatronics,
University of Craiova Romania*

1. Introduction

The control of robotic manipulators has become important due to the development of the flexible automation. Requirements such as the high speed and high precision trajectory tracking make the modern control indispensable for versatile applications of manipulators (Middleton & Goodwin, 1998; Ortega & Spong, 1999; Popescu *et al.*, 2008). Rigid robot systems are subjects of the research in both robotic and control fields. The reported research leads to a variety of control methods for such rigid robot systems (Ortega & Spong, 1999; Raimondi *et al.*, 2004; Bobaşu & Popescu, 2006; Dinh *et al.*, 2008).

Conventional controllers for robotic structures are based on independent control schemes in which each joint is controlled separately by a simple servo loop. This classical control scheme (for example a PD control) is inadequate for precise trajectory tracking. The imposed performance for industrial applications requires the consideration of the complete dynamics of the manipulator. Moreover, in real-time applications, the ignoring parts of the robot dynamics or errors in the parameters of the robotic manipulator may cause the inefficiency of this classical control. An alternative solution to PD control is the computed torque technique. This classical method is in fact a nonlinear technique that takes account of the dynamic coupling between the robot links. The main disadvantage of this structure is the assumption of an exactly known dynamic model. However, the basic idea of this method remains important and it is the base of the neural and adaptive control structures (Gupta & Rao, 1994; Pham & Oh, 1994; Dumbravă & Olah, 1997; Ortega & Spong, 1999; Aoughellanet *et al.*, 2005; Popescu *et al.* 2008).

Industrial robotic manipulators are exposed to structured and unstructured uncertainties. Structured uncertainties are characterized by having a correct model but with parameter uncertainty (unknown loads and friction coefficients, imprecision of the manipulator link properties, etc.). Unstructured uncertainties are characterized by unmodelled dynamics. Generally speaking, two classes of strategies have been developed to maintain performance in the presence of the parameter uncertainties: robust control and adaptive control. The adaptive controllers can provide good performances in face of very large load variation. Therefore the adaptive approach is intuitively superior to robust approach in this type of application. When the dynamic model of the system is not known a priori (or is not

available), a control law is designed based on an estimated model. This is the basic idea behind adaptive control strategies (Ortega & Spong, 1999).

Over the last few years several authors (Zalzala & Morris, 1996; Miyamoto *et al.*, 1998; Popescu *et al.* 2001; Raimondi *et al.*, 2004; Popescu *et al.*, 2008) have considered the use of artificial neural networks (ANNs) within a control system for robotic arms. The differences in control schemes are in the role that ANN is playing, and the way it is trained for achieving desired trajectory tracking performance.

In this chapter, which is an extended work of the research achieved in some papers of the authors (Popescu, 1998; Popescu *et al.*, 2001, Selişteanu *et al.*, 2001; Popescu *et al.*, 2008), classical, adaptive and neural strategies for a robotic manipulator with two revolute joints are presented. The first section analyses the computed-torque method (based on the so-called inverse dynamics of the robotic manipulator), which is a starting point for the design of the adaptive and neural control techniques. In the next section, an overview of adaptive strategies is presented, and two adaptive controllers for rigid manipulators are designed. First, a direct adaptive control with adaptation law of gradient type is analyzed. Second, an indirect adaptive controller is designed; this controller uses the prediction errors of the filtered joint torques to generate parameter estimates. In the following section, various non-model and model-based neural control schemes have been designed. The ANN is used in order to generate auxiliary joint control torque to compensate for the uncertainties in the computed torque based primary robotic manipulator. Three neural control strategies are studied: feedforward neural control, feedback neural control, and feedback error based neural control. Also, numerical simulations are performed, in order to analyse the behaviour and the performance of the control strategies, and to make some useful comparisons. The final section deals with concluding remarks and further research directions.

2. The computed-torque control strategy

The robotic manipulator is modeled as a set of n rigid bodies connected in series with one end fixed to the ground and the other end free. The bodies are connected via either revolute or prismatic joints and a torque actuator acts at each joint.

The dynamic equation of an n -link robotic manipulator is given by (Ivănescu, 2003; Popescu, 1998):

$$T = J(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + F(\dot{q}), \quad (1)$$

where: T is an $(n \times 1)$ vector of joint torques; $J(q)$ is the $(n \times n)$ manipulator inertia matrix; $V(q, \dot{q})$ is an $(n \times n)$ matrix representing centrifugal and Coriolis effects; $G(q)$ is an $(n \times 1)$ vector representing gravity; $F(\dot{q})$ is an $(n \times 1)$ vector representing friction forces; q, \dot{q}, \ddot{q} are the $(n \times 1)$ vectors of joint positions, speeds and accelerations, respectively.

The equations (1) form a set of coupled nonlinear ordinary differential equations which are quite complex, even for simple robotic arms. For simplicity, we denote

$$V(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = H(q, \dot{q}), \quad (2)$$

so that (1) can be rewritten as:

$$T = J(q)\ddot{q} + H(q, \dot{q}). \tag{3}$$

The computed-torque method is a conventional control technique, which takes account of the dynamic coupling between the manipulator links. This method, also called the inverse model control technique (Zalzala & Morris, 1996; Ortega & Spong, 1999) leads to a completely decoupled error dynamics equation. The structure of this control strategy is illustrated in Fig. 1.

One of most used computed-torque control scheme is based on the exactly linearization of the nonlinear dynamics of the robotic manipulator. If the dynamic model is exact, the dynamic perturbations are exactly cancelled. The total torque driving the robotic manipulator is given by (Dumbravă & Olah, 1997):

$$T = \hat{J}(q)\Gamma' + \hat{V}(q, \dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) = \hat{J}(q)\Gamma' + \hat{H}(q, \dot{q}), \tag{4}$$

where: \hat{J} , \hat{V} , \hat{G} , \hat{F} , \hat{H} are estimates of J , V , G , F , H , respectively, and T' is defined as:

$$T' = \ddot{q}_d + K_v\dot{e} + K_p e. \tag{5}$$

The closed loop equation is found to be:

$$\ddot{e} + K_v\dot{e} + K_p e = \hat{J}^{-1}(q)[\tilde{J}(q)\ddot{q} + \tilde{V}(q, \dot{q})\dot{q} + \tilde{G}(q) + \tilde{F}(\dot{q})] = \hat{J}^{-1}(q)[\tilde{J}(q)\ddot{q} + \tilde{H}(q, \dot{q})], \tag{6}$$

where $\tilde{J} = J - \hat{J}$; $\tilde{V} = V - \hat{V}$; $\tilde{G} = G - \hat{G}$; $\tilde{F} = F - \hat{F}$; $\tilde{H} = H - \hat{H}$ are the modelling errors and the tracking error is $e = q_d - q$.

If the robotic manipulator's parameters are perfectly known, the closed loop equation (6) takes a linear, decoupled form:

$$\ddot{e} + K_v\dot{e} + K_p e = 0. \tag{7}$$

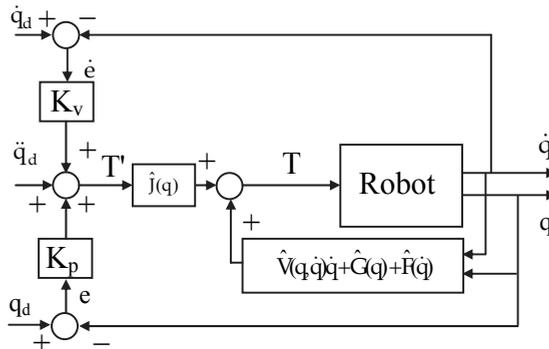


Fig. 1. The computed-torque control scheme

The computed-torque control method has performance problems because of its reliance on a fixed dynamic model. The robotic arm structures have to face uncertainty in the dynamics parameters. Two classes of approach have been studied to maintain performances in the presence of parametric uncertainties - the robust control and the adaptive control. The next section deals with the adaptive control strategy for the robotic manipulator.

3. The adaptive control method

Adaptive controllers can be a good alternative when it is neither possible nor economical to make a thorough investigation of the causes of the process variations. In other situations, some of the dynamics may be well understood, but other parts are unknown. This is the case of robots, for which the geometry, motors and gearboxes do not change, but the load does change. An adaptive controller can be defined as a controller with adjustable parameters and a mechanism for adjusting the parameters (Astrom & Wittenmark, 1995). The modern adaptive control approach consists in the explicit introduction of the linear parameterization of the robot dynamics. The adaptive controllers can be classified into three major categories (Zalzala & Morris, 1996): direct, indirect and composite.

3.1 Direct adaptive controller

The direct adaptive controllers use tracking errors of the joint motion to drive parameter adaptation. The main goal of the control strategy is to reduce the tracking errors. Such a direct technique is an adaptive control method based on computed torque control. This method has been pioneered by (Craig *et al.*, 1987), and the properties of stability and convergence are established in (Slotine & Li, 1987; Ortega & Spong, 1999). The controller is in fact composed of a modified computed-torque control and an adaptation law.

Next, this direct adaptive strategy is used for the robot arm structure (1). Let's consider θ the vector of the uncertain (unknown) parameters, which are the viscous friction coefficients, the Coulomb friction coefficients and the load mass. Then, the dynamics of the robot arm can be written as:

$$T = J(q, \theta)\ddot{q} + V(q, \dot{q}, \theta)\dot{q} + G(q, \theta) + F(\dot{q}, \theta). \quad (8)$$

A linear parameterization of (8) is:

$$J(q, \theta)\ddot{q} + V(q, \dot{q}, \theta)\dot{q} + G(q, \theta) + F(\dot{q}, \theta) = J_c(q)\ddot{q} + V_c(q, \dot{q})\dot{q} + G_c(q) + F_c(\dot{q}) + R(q, \dot{q}, \ddot{q})\theta, \quad (9)$$

where $J_c(\cdot)$, $V_c(\cdot)$, $G_c(\cdot)$, $F_c(\cdot)$ represent the known (certain) part of the dynamics and $R(q, \dot{q}, \ddot{q})$ is the regressor matrix.

The design of the control law is reached by using in (8) the vector of the estimated parameters. The linearization (9) allows us to obtain the torque:

$$T = J(q, \hat{\theta})\ddot{q} + V(q, \dot{q}, \hat{\theta})\dot{q} + G(q, \hat{\theta}) + F(\dot{q}, \hat{\theta}) = J_c(q)\ddot{q} + V_c(q, \dot{q})\dot{q} + G_c(q) + F_c(\dot{q}) + R(q, \dot{q}, \ddot{q})\hat{\theta}, \quad (10)$$

where $\hat{\theta}$ is the vector of estimated parameters.

From the equations (5), (10) the closed loop dynamics is obtained:

$$J(q, \hat{\theta})(\ddot{e} + K_v \dot{e} + K_p e) = R(q, \dot{q}, \ddot{q}) \tilde{\theta}, \quad (11)$$

with $\tilde{\theta} = \hat{\theta} - \theta$ the estimation parameter error vector.

If the inertia matrix is nonsingular, we can write:

$$\ddot{e} + K_v \dot{e} + K_p e = J^{-1}(q, \hat{\theta}) R(q, \dot{q}, \ddot{q}) \tilde{\theta}. \quad (12)$$

The state representation of (12) can be obtained if the state $x = [e \quad \dot{e}]^T$ is used:

$$\dot{x} = A_m x + B_m J^{-1}(q, \hat{\theta}) R(q, \dot{q}, \ddot{q}) \tilde{\theta}, \quad (13)$$

where $A_m = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}$, $B_m = \begin{bmatrix} 0 \\ I \end{bmatrix}$.

We can choose a gradient type adaptation law for the on-line estimation of the parameters:

$$\frac{d\tilde{\theta}(t)}{dt} = -\frac{d\hat{\theta}(t)}{dt} = -\Omega \cdot R^T(q, \dot{q}, \ddot{q}) J^{-1}(q, \hat{\theta}) \cdot B_m^T P x, \quad (14)$$

with $\Omega = \Omega^T > 0$ the amplification matrix and $P = P^T > 0$ a quadratic $n \times n$ matrix, solution of the Lyapunov equation:

$$A_m^T P + P A_m = -Q, \quad (15)$$

where $Q = Q^T > 0$.

Remark 1. The Lyapunov function $V = x^T P x + \tilde{\theta}^T \tilde{\theta}$ can be used to show that the tracking errors go to zero. \square

The final adaptive control law consists of the computed-torque Eq. (4) and the estimates provided by the adaptation law (14).

The global convergence of the direct adaptive controller based on computed-torque method is demonstrated in (Slotine & Li, 1987). The disadvantages of this adaptive method are the use of the acceleration measurements and the necessity of inversion of the estimated inertia matrix. The advantages are the simplicity of the method (comparatively to a least squares indirect method for example) and the rejection of the parametric disturbances, inherent for an adaptive method.

3.2 Indirect adaptive controller

The indirect adaptive control method for manipulators has been pioneered by (Middleton & Goodwin, 1998), who used prediction errors on the filtered joint torques to generate parameter estimates to be used in the control law.

Such indirect adaptive controller can be composed of a modified computed-torque control and a modified least-squares estimator.

The design of this indirect control law for the manipulator (1) is based on the estimate of the torque:

$$\hat{T} = J_C(q)\ddot{q} + V_C(q, \dot{q})\dot{q} + G_C(q) + F_C(\dot{q}) + R(q, \dot{q}, \ddot{q})\hat{\theta}, \quad (16)$$

where $\hat{\theta}$ is the vector of estimated parameters.

Now we can calculate the prediction error for the torque from (9), (16)

$$\varepsilon = \hat{T} - T = R(q, \dot{q}, \ddot{q}) \cdot (\hat{\theta} - \theta) = R(q, \dot{q}, \ddot{q})\tilde{\theta}, \quad (17)$$

with $\tilde{\theta} = \hat{\theta} - \theta$ the estimation parameter error vector.

The prediction error is filtered to eliminate the measurements of the accelerations in the control law. First, the torque T is filtered through a first-degree filter with the transfer function $H(s) = \frac{\omega_f}{s + \omega_f}$, where ω_f is the crossover frequency of the filter. The filtered torque is the convolution

$$T_f = h(t) * T(t), \quad (18)$$

where $h(t)$ is the impulse response of $H(s)$.

The estimated torque is also filtered. We define

$$T_C = J_C(q)\ddot{q} + V_C(q, \dot{q})\dot{q} + G_C(q) + F_C(\dot{q}), \quad (19)$$

and from (16), (19) the estimated torque can be written as

$$\hat{T} = T_C + R(q, \dot{q}, \ddot{q})\hat{\theta}. \quad (20)$$

We have

$$T_{Cf}(t) = h(t) * T_C(t), \quad (21)$$

$$\Phi(t) = h(t) * R(t). \quad (22)$$

In the relations (21), (22), T_{Cf} and the filtered regressor matrix Φ depend only of the state $q(t)$ and of the time derivative $\dot{q}(t)$, and not of the accelerations (Ivănescu, 2003):

$$T_{Cf}(t) = T_{Cf}(q(t), \dot{q}(t)); \quad \Phi(t) = \Phi(q(t), \dot{q}(t)). \quad (23)$$

We obtain the filtered estimated torque from (20), (21), and (22):

$$\hat{T}_f = T_{Cf} + \Phi \cdot \hat{\theta} . \quad (24)$$

Now we can obtain the filtered prediction error, which will be used in the adaptation law. From (17), (18), (24) the filtered prediction error is

$$\varepsilon_f = \hat{T}_f - T_f = T_{Cf}(t) + \Phi(t) \cdot \hat{\theta} - h(t) * T(t) . \quad (25)$$

The torque T can be written as

$$T = T_C + R \cdot \theta , \quad (26)$$

therefore the filtered prediction error becomes

$$\varepsilon_f = h(t) * T_C(t) + \Phi(t) \cdot \hat{\theta} - h(t) * T_C(t) - \Phi(t) \cdot \theta = \Phi(q(t), \dot{q}(t)) \cdot \tilde{\theta} . \quad (27)$$

The adaptation parameter law is based on a least-squares estimator (Dumbravă & Olah, 1997) that it has as input the filtered prediction error (27). The equations of the adaptation law are

$$\frac{d\tilde{\theta}(t)}{dt} = \frac{d\hat{\theta}(t)}{dt} = -\Gamma(t)\Phi^T(q, \dot{q})\varepsilon_f(t) , \quad (28)$$

$$\frac{d\Gamma(t)}{dt} = -\Gamma(t)\Phi^T(q, \dot{q})\Phi(q, \dot{q})\Gamma^T(t) , \quad (29)$$

with $\Gamma(0) = \Gamma^T(0) > 0$. The matrix $\Gamma(t) = \Gamma^T(t) > 0$ is the amplification matrix.

The final indirect adaptive control law consists of the computed-torque equation (4) and the estimates provided by the adaptation law (28), (29):

$$T = \hat{J}(q)\Gamma' + \hat{V}(q, \dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) = J(q, \hat{\theta})T' + V(q, \dot{q}, \hat{\theta})\dot{q} + G(q, \hat{\theta}) + F(\dot{q}, \hat{\theta}) , \quad (30)$$

with T' given by (5).

The indirect adaptive control structure is presented in Fig. 2.

The least-squares estimator (28), (29) has good convergence and stability properties (Ivănescu, 2003). A disadvantage can be the complexity of the algorithm and the correlation between the prediction error and the estimation parameter error (Ivănescu, 2003; Dumbravă & Olah, 1997). This disadvantage can be canceled by addition of a stabilizing signal to the control law (Dumbravă & Olah, 1997).

Remark 2. Indirect controllers allow the various parameter-estimation algorithms to be used to select time variations of the adaptation gains. \square

Composite adaptive controllers for manipulators have been developed by (Slotine & Li, 1989). These adaptive control strategies use both tracking errors in the joint motions and prediction errors on the filtered torque to drive the parameter adaptation.

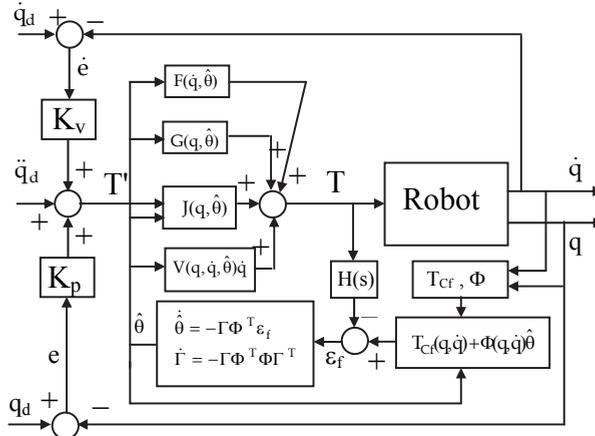


Fig. 2. The indirect adaptive control scheme for the rigid link manipulator

4. Neural control strategies

Various neural control schemes have been studied, proposed and compared. The differences in these schemes are in the role that artificial neural network (ANN) is playing in the control system and the way it is trained for achieving desired trajectory tracking performance structures (Psaltis *et al.*, 1988; Gupta & Rao, 1994; Zalzal & Morris, 1996; Raimondi *et al.*, 2004; Aoughellanet *et al.*, 2005; Dinh *et al.*, 2008). Two classes of approaches have been studied: non-model based neural control and model based neural control. Non-model based neural control consists of PD feedback controller and an ANN. The inverse dynamics is learned by measuring the input and output signals in the manipulator and then adjusting the connection weights vector by using a learning algorithm. After the learning was finished, the actual trajectory of the manipulator followed well the desired trajectory. But, when the desired trajectory was changed to one not used in the training of ANN, the error between the actual and desired trajectory became large. This means that the ANN had fitted a relationship between the input/output data but had no succeeded in learning the inverse-dynamics model (Zalzal & Morris, 1996). We want that training doesn't depend on desired trajectory. Hence, we proposed to train the ANN with $(q, \dot{q}, \ddot{q}_d, e, \dot{e})$ (see Fig. 3).

For training of ANN, there are two possibilities: off-line or on-line. From the viewpoint of real time control it's better to train ANN on-line. But, from the viewpoint of initial weights and biases, rate of convergence and stability of learning it's better to train ANN off-line. The tracking performance was better if ANN was trained off-line and then ANN was used to improve the performance of PD feedback controller (Popescu *et al.*, 2001).

In this section, model based neural control structures for a robotic manipulator are implemented. Various neural control schemes have been studied, proposed and compared. The differences in these schemes are in the role that ANN is playing in the control system

and the way it is trained for achieving desired trajectory tracking performance. The most popular control scheme is one which uses ANN to generate auxiliary joint control torque to compensate for the uncertainties in the computed torque based primary robotic manipulator controller that is designed based on a nominal robotic manipulator dynamic model.

This is accomplished by implementing the neural controller in either a feedforward or a feedback configuration, and the ANN is trained on-line. Based on the computed torque method, a training signal is derived for neural controller. Comparison studies based on a robotic planar manipulator have been made for the neural controller implemented in both feedforward and feedback configurations. Also, a feedback error based neural controller is proposed. In this approach, a feedback error function is minimized and the advantage over Jacobian based approach is that Jacobian estimation is not required.

4.1 Feedforward neural control strategy

The feedforward neural controller (Fig. 3) is designed to achieve perturbation rejection for a computed torque control system of a robotic manipulator. The ANN output cancels out the uncertainties caused by inaccurate robotic manipulator’s model in the computed torque controller. The robot joint torques are:

$$T = \hat{J}(q_d)(T' + \phi_f) + \hat{H}(q_d, \dot{q}_d). \tag{31}$$

The closed loop error system is

$$\ddot{e} + K_V \dot{e} + K_P e = \hat{J}^{-1}(\tilde{J}\ddot{q} + \tilde{H}) - \phi_f. \tag{32}$$

Since the control objective is to generate ϕ_f to reduce u to zero, we therefore propose to use:

$$u = \ddot{e} + K_V \dot{e} + K_P e, \tag{33}$$

as the error signal for training the ANN. Then, the ideal value of ϕ_f at $u = 0$ is:

$$\phi_f = \hat{J}^{-1}(\tilde{J}\ddot{q} + \tilde{H}). \tag{34}$$

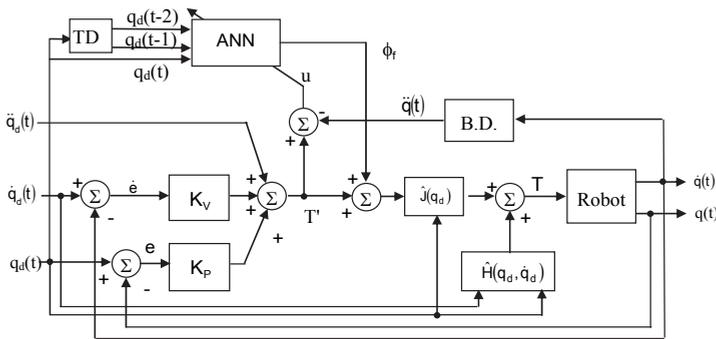


Fig. 3. The structure of the feedforward neural controller

4.2 Feedback neural control strategy

The main difference between feedforward and feedback neural controller schemes is that the joint variables used in the ANN inputs and the computed torque controller are either the desired values $q_d(t)$ or the actual values $q(t)$. The ANN inputs can be either $q_d(t)$, $\dot{q}_d(t)$, $\ddot{q}_d(t)$, or $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$, or the time-delayed values $q_d(t)$, $q_d(t-1)$, $q_d(t-2)$, or $q(t)$, $q(t-1)$, $q(t-2)$. Delay time is chosen as the sampling period of the controller. In simulations the ANN performs better when time-delayed joint values are used instead of the velocity and acceleration values calculated from finite difference approximations based on samples of $q(t)$.

For feedback neural controller (Fig. 4) the robotic manipulator joint torques are:

$$T = \hat{J}(q)(T' + \phi_b) + \hat{H}(q, \dot{q}). \tag{35}$$

The three-layer feedforward neural network is used as the compensator. It is composed of an input layer (6 neurons), a nonlinear hidden layer, and a linear output layer (2 neurons). The weight updating law minimizes the objective function J which is a quadratic function of the training signal u :

$$J = \frac{1}{2} (u^T u). \tag{36}$$

For simplicity, we use ϕ for ϕ_f or ϕ_b . Differentiating equation (36) and making use of (32) yields the gradient of J as follows:

$$\frac{\partial J}{\partial w} = \frac{\partial u^T}{\partial w} u = -\frac{\partial \phi^T}{\partial w} u. \tag{37}$$

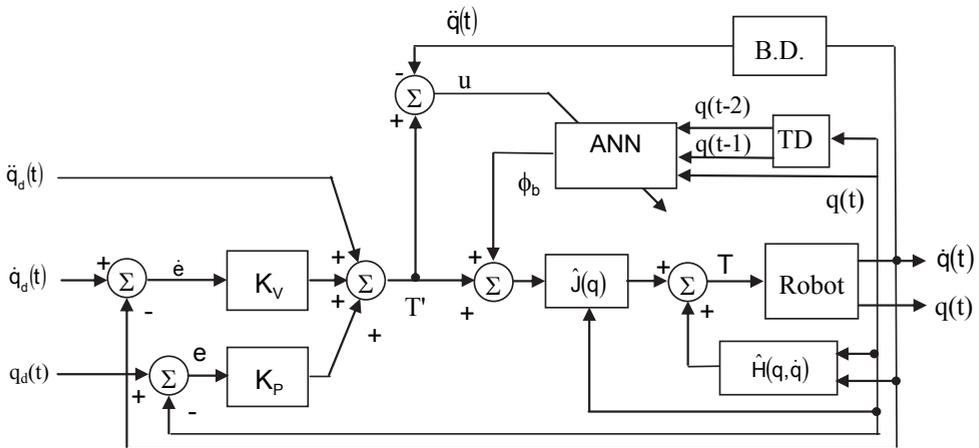


Fig. 4. The structure of the feedback neural controller

The backpropagation update rule for the weights with a momentum term is:

$$\Delta w(t) = -\eta \frac{\partial J}{\partial w} + \alpha \Delta w(t-1) = \eta \frac{\partial \phi^T}{\partial w} u + \alpha \Delta w(t-1), \tag{38}$$

where η is the learning rate and α is the momentum coefficient.

4.3 Feedback error based neural control strategy

In this approach, a feedback error function is minimized and the advantage over Jacobian based approach is that Jacobian estimation is not required. The inputs to the neural controller (Fig. 5) are the required trajectories $q_d(t)$, $\dot{q}_d(t)$, $\ddot{q}_d(t)$. The compensating signals from ANN, ϕ_p , ϕ_v , ϕ_a , are added to the desired trajectories.

The control law is:

$$T = \hat{J}(\ddot{q}_d + \phi_a + K_v(\dot{e} + \phi_v) + K_p(e + \phi_p)) + \hat{H}. \tag{39}$$

Combining (39) with dynamic equation of robotic manipulator yields:

$$u = \ddot{e} + K_v \dot{e} + K_p e = \hat{J}^{-1}(\tilde{J}\ddot{q} + \tilde{H}) - \Phi, \tag{40}$$

where $\Phi = \phi_a + K_v \phi_v + K_p \phi_p$. Ideally, at $u = 0$, the ideal value of Φ is:

$$\Phi = \hat{J}^{-1}(\tilde{J}\ddot{q} + \tilde{H}). \tag{41}$$

The error function u is minimized and objective function is the same – see expression (36).

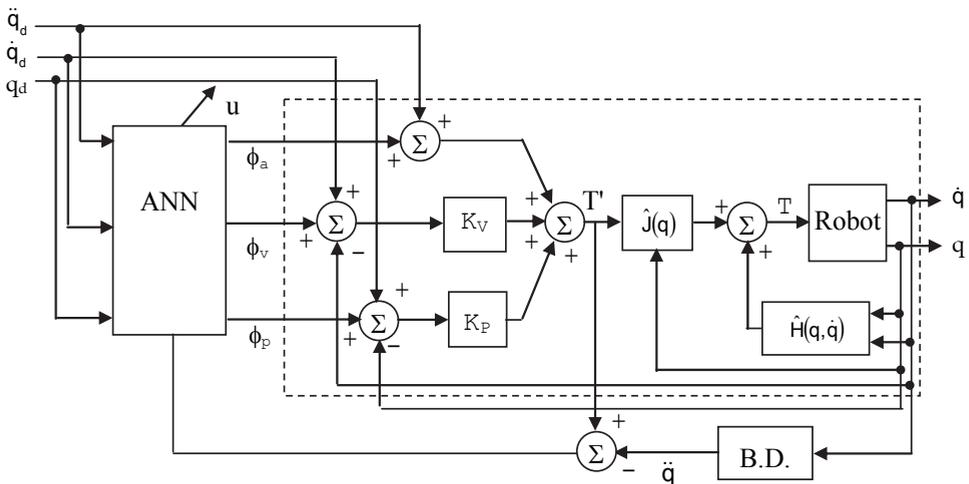


Fig. 5. The structure of the feedback error based neural controller

The gradient of J is:

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial \mathbf{u}^T}{\partial \mathbf{w}} \mathbf{u} = -\frac{\partial \Phi^T}{\partial \mathbf{w}} \mathbf{u}. \quad (42)$$

The backpropagation updating rule for the weights with momentum term is:

$$\Delta \mathbf{w}(t) = -\eta \frac{\partial J}{\partial \mathbf{w}} + \alpha \Delta \mathbf{w}(t-1) = \eta \frac{\partial \Phi^T}{\partial \mathbf{w}} \mathbf{u} + \alpha \Delta \mathbf{w}(t-1). \quad (43)$$

5. Simulation results and comparisons

In order to test the proposed adaptive and neural control strategies, the control of the simple planar robotic manipulator with two revolute joints shown in Fig. 6 was considered.

The elements of the dynamic equation (1) for this robotic manipulator with electrical motor dynamics are (Selișteanu *et al.*, 2001; Popescu *et al.*, 2008):

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} I_1^2(m_1 + m_2) + m_2 l_2^2 + 2m_2 l_1 l_2 c_2 + J_1 n_1^2 & m_2 l_2^2 + m_2 l_1 l_2 c_2 \\ m_2 l_2^2 + m_2 l_1 l_2 c_2 & m_2 l_2^2 + J_2 n_2^2 \end{bmatrix}, \quad (44)$$

$$\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) = m_2 l_1 l_2 s_2 \begin{bmatrix} 0 & -(2\dot{q}_1 + \dot{q}_2) \\ \dot{q}_1 & 0 \end{bmatrix}, \quad (45)$$

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} (m_1 + m_2) g l_1 s_1 + m_2 g l_2 s_{12} \\ m_2 g l_2 s_{12} \end{bmatrix}, \quad (46)$$

$$\mathbf{F}(\dot{\mathbf{q}}) = \begin{bmatrix} v_1 \dot{q}_1 + C_1 \text{sign}(\dot{q}_1) \\ v_2 \dot{q}_2 + C_2 \text{sign}(\dot{q}_2) \end{bmatrix}, \quad (47)$$

with m_1 - mass of link 1, $m_2 = m_{20} + m_p$, m_{20} - mass of link 2, m_p - mass of payload, l_1 - length of link 1, l_2 - length of link 2, $c_i = \cos(q_i)$, $i=1,2$, $s_i = \sin(q_i)$, $i=1,2$, $c_{12} = \cos(q_1 + q_2)$, $s_{12} = \sin(q_1 + q_2)$, J_i , $i=1,2$ - moments of inertia for electrical motor i , n_i , $i=1,2$ - factor of reduction gear i , v_i , $i=1,2$ - viscous friction for joint i , C_i , $i=1,2$ - Coulomb friction for joint i .

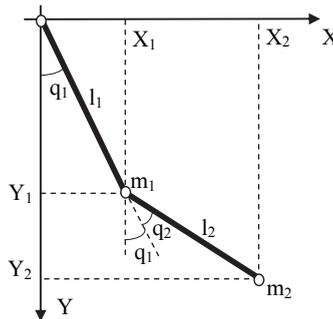


Fig. 6. The planar robotic manipulator with two revolute joints

For simulation and comparisons, the planar robot manipulator with two revolute joints (1), (44)-(47) is used. The simulation model parameters are (SI units): $m_1 = 10$, m_2 consists of mass of link 2, $m_{20} = 2.5$, and mass of payload m_p , $l_1 = 1, l_2 = 0.5$, link lengths.

The robot manipulator starts at position $(q_1 = 0, q_2 = 0)$, and the control objective is to track the desired trajectory given by:

$$q_{1d} = 0.4 \cdot \sin(0,4\pi t), \quad q_{2d} = -0.5 \cdot \sin(0,5\pi t). \tag{48}$$

In order to test the performance and to analyze the behaviour of the control strategies, several numerical simulations were performed. We considered three basic simulation cases:

1) When the model of robot manipulator is known, the use of the computed-torque method is recommended. The equations (4), (5) are used and a simulation has been done for the tuning parameters $K_{p1} = 50, K_{p2} = 50, K_{v1} = 6, K_{v2} = 15$ (matrices K_p, K_v of diagonal form).

The time evolution of tracking errors $e = [e_1 \ e_2]^T = [q_{1d} - q_1 \ q_{2d} - q_2]^T$ is presented in Fig. 7.

2) The computed-torque method provides good results when the model is exactly known. If parametric uncertainties occur, an adaptive control method can be used. Let's consider that the uncertain parameters are the viscous friction coefficients, the Coulomb friction coefficients and the load mass. Therefore we have: $\theta = [m_p \ v_1 \ C_1 \ v_2 \ C_2]^T$. The direct adaptive control law (9), (10), (14) is implemented with the design parameters $K_{p1} = 50, K_{p2} = 50, K_{v1} = 6, K_{v2} = 15$, and the diagonal matrix $\Omega = [\omega_{ii}]_{i=1,5}, \omega_{ii} = 15$. The results are presented in Fig. 8.

We can see that even if the estimated parameters $\hat{\theta} = [\hat{m}_p \ \hat{v}_1 \ \hat{C}_1 \ \hat{v}_2 \ \hat{C}_2]^T$ are used, the evolution of tracking errors remains good.

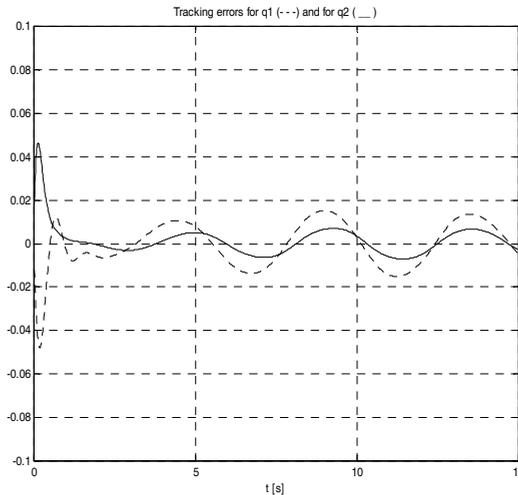


Fig. 7. Tracking errors – the computed-torque case

The imposed trajectories are preserved in the presence of the parametric uncertainties. Another simulation is done for the indirect adaptive control law (28)-(30), which is implemented with the parameters $K_{p1} = 50$, $K_{p2} = 50$, $K_{v1} = 6$, $K_{v2} = 15$, $\omega_f = 5$ and the diagonal matrix $\Gamma(0) = [\gamma_{ii}]_{i=1,5}$, $\gamma_{ii} = 15$. The estimated parameters have a fast convergence to their actual ("true") values. The evolution of tracking errors is illustrated in Fig. 9.

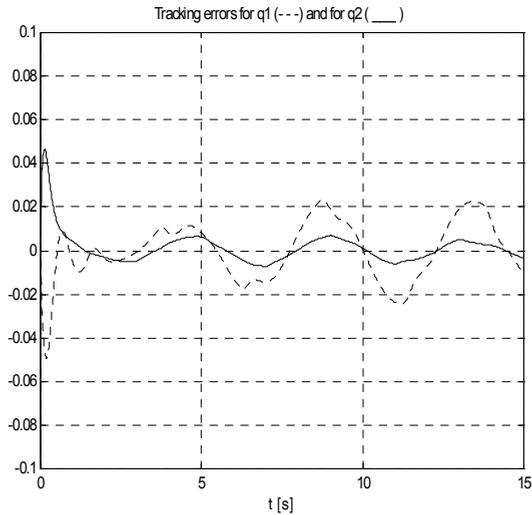


Fig. 8. Simulation results – direct adaptive control law

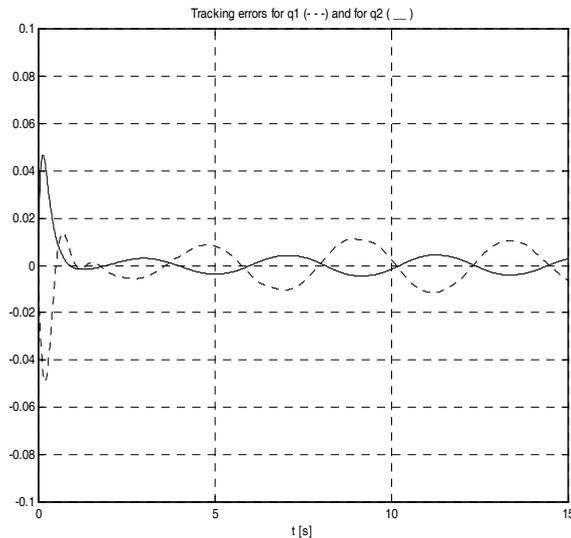


Fig. 9. Tracking errors – the indirect adaptive controller

3) The neural control strategies were implemented. For feedforward neural control law and for feedback neural controller (with the neural network structure of the form $6 \times 10 \times 2$), with update backpropagation rule (38), the time evolution of the tracking errors is presented in Fig. 10 and Fig. 11, respectively.

Also, for the feedback error based neural control law (with the neural network structure of the form $6 \times 9 \times 2$), and with the update backpropagation rule (43), the time profiles of the tracking errors are depicted in Fig. 12.

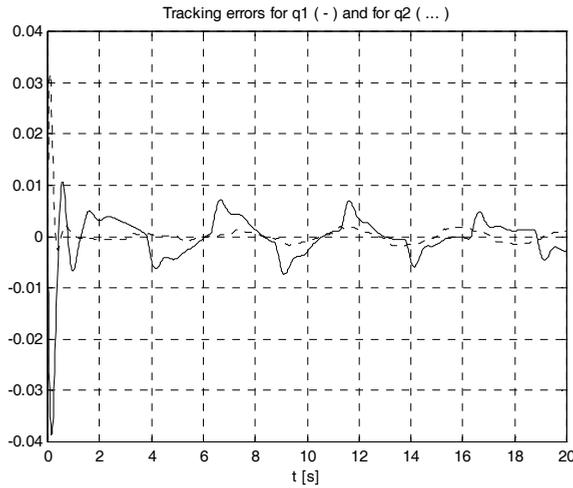


Fig. 10. Tracking errors – the feedforward neural control scheme

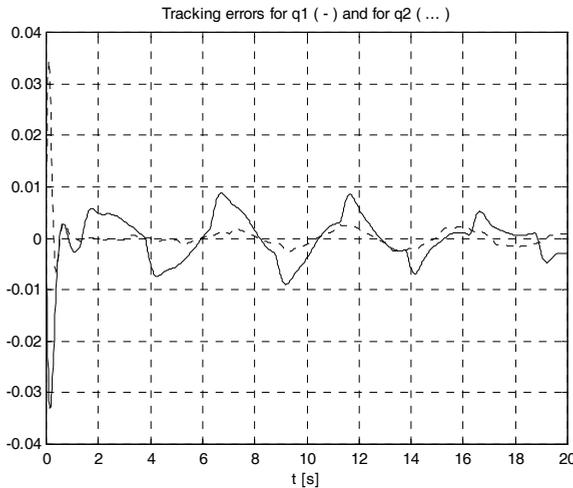


Fig. 11. Simulation results – the feedback neural controller

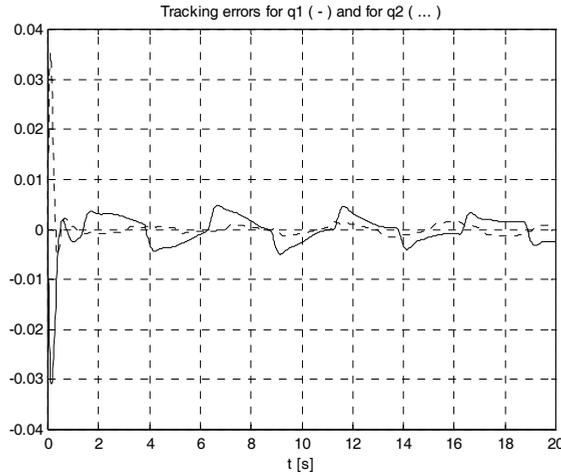


Fig. 12. Tracking errors – the feedback error based neural control scheme

The comparisons between the control strategies can be done by visualization of tracking errors, but accurate comparisons can be done by considering a criterion based on averaged square tracking errors – see (Whitcomb *et al.*, 1991; Popescu *et al.*, 2008):

$$I_1 = \frac{1}{T_s} \int_0^{T_s} e_1^2(t)dt, \quad I_2 = \frac{1}{T_s} \int_0^{T_s} e_2^2(t)dt, \tag{49}$$

where T_s is the total simulation time.

The values of I_1 and I_2 , computed for the studied control strategies (manipulator with two revolute joints), and for numerous simulations (including PD control law and non-model based neural controllers) are presented in Table 1.

Control Strategies	Performances	
	I_1	I_2
Classical PD controller (exactly known model and manipulator parameters)	$4.2 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$
Computed-torque method (exactly known model and manipulator parameters)	$1.2 \cdot 10^{-4}$	$0.8 \cdot 10^{-4}$
Direct adaptive controller (gradient adaptation law)	$1.7 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$
Indirect adaptive controller (least-squares estimator as adaptation law)	$1.1 \cdot 10^{-4}$	$0.7 \cdot 10^{-4}$
Non-model based neural controller	$2 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$
Model based neural controller	$0.9 \cdot 10^{-4}$	$0.5 \cdot 10^{-4}$

Table 1. Performance criterion results

6. Conclusion

In this chapter, some classical, adaptive and neural control strategies for a simple planar robotic manipulator with two revolute joints were designed and implemented.

First, the conventional computed-torque method was discussed. This control method solves the precision tracking problem, by using an exactly linearization of the nonlinearities of the manipulator model. The main disadvantage is the assumption of an exactly known dynamic model. If the model is imprecise known, it is necessary to design adaptive and/or neural control strategies.

Direct and indirect adaptive controllers have been studied and implemented in order to preserve the tracking performances when parameter uncertainties occur. From the simulation point of view, it can be noticed that the evolution of tracking errors remains good, even if the estimated parameters are used in the control law.

Also, three neural based control strategies were developed: a feedforward neural controller, a feedback neural control scheme, and a feedback error based neural controller. The simulations showed that the proposed neural controllers obtain results comparable to those achieved using adaptive control strategies. If a classical (PD or computed-torque) controller already controls a manipulator the advantage of proposed neural structures is that extension to a neural controller for performances improvement is easy.

7. Acknowledgment

This work was supported by the National Authority for Scientific Research, Romania, under the research project SICOTIR, 71-084/2007 (PNCDI II).

8. References

- Aoughellanet, S.; Mohammadi, T. & Bouterfa, Y. (2005). Neural network path planning applied to PUMA560 robot arm. *WSEAS Transactions on Systems*, Issue 4, Vol. 4, April 2005, pp. 446-450, ISSN: 1109-2777
- Astrom, K.J. & Wittenmark, B. (1995). *Adaptive Control*, Addison-Wesley, ISBN: 0-0201-55866-1, USA
- Bobașu, E. & Popescu, D. (2006). On modelling and multivariable adaptive control of robotic manipulators. *WSEAS Transactions on Systems*, Issue 7, Vol. 5, July 2006, pp. 1579-1586, ISSN: 1109-2777
- Craig, J.J.; Hsu, P. & Sastry, S. (1987). Adaptive control of mechanical manipulators. *International Journal of Robotics Research*, Vol. 2, pp. 10-20
- Dinh, B.H.; Dunnigan, M.W. & Reay, D.S. (2008). A practical approach for position control of a robotic manipulator using a radial basis function network and a simple vision system. *WSEAS Transactions on Systems and Control*, Issue 4, Vol. 3, April 2008, pp. 289-298, ISSN: 1991-8763
- Dumbravă, S. & Olah, I. (1997). Robustness analysis of computed torque based robot controllers, *Proceedings of 5-th Symposium on Automatic Control and Computer Science*, pp. 228-233, Iași, Romania, 1997
- Gupta, M.M. & Rao, D.H. (1994). *Neuro-Control Systems: Theory and Applications*, IEEE Press Piscataway, NJ, USA

- Ivănescu, M. (2003). *Advanced Systems for Robotic Control*, (in Romanian), Ed. Scrisul Romanesc, Craiova, ISBN: 973-38-0389-8
- Middleton, R. & Goodwin, G. (1998). Adaptive computed torque control for rigid link manipulators. *Systems and Control Letters*, Vol. 10, pp. 9-16, ISSN: 0167-6911
- Miyamoto, H.; Kawato, M.; Setoyama, T. & Suzuki, R. (1998). Feedback error learning neural networks for trajectory control of a robotic manipulator. *Neural Networks*, Vol. 1, pp. 251-265, ISSN: 1045-9227
- Ortega, R. & Spong, M.W. (1999). Adaptive motion control of rigid robots: a tutorial. *Automatica*, Vol. 25, pp. 877-888, ISSN: 0005-1098
- Pham, D.T. & Oh, S.J. (1994). Adaptive control of a robot using neural networks. *Robotica*, pp. 553-561, ISSN: 0263-5747
- Popescu, D. (1998). Neural control of manipulators using a supervisory algorithm, *Proceedings of A&Q'98 International Conference on Automation and Quality Control*, pp. A576-A581, Cluj-Napoca, Romania, 1998, Ed. Mediamira, ISBN: 973-9358-15-2
- Popescu, D.; Selișteanu, D. & Ionete, C. (2001). Non-model based neural robot control, *Proceedings of the 10th International Workshop on Robotics in Alpe-Adria-Danube Region*, RD-038, Vienna, Austria, 2001
- Popescu, D.; Selișteanu, D. & Popescu, L. (2008). Neural and adaptive control of a rigid link manipulator. *WSEAS Transactions on Systems*, Issue 6, Vol. 7, pp. 632-641, ISSN: 1109-2777
- Psaltis, D.; Sideris, A. & Yamamura, A. (1988). A multilayered neural network controller. *IEEE Control Systems Magazine*, Vol. 8, pp. 17-21, ISSN: 0272-1708
- Raimondi, F.M.; Melluso, M. & Bonafede, V. (2004). A neuro fuzzy controller for planar robot manipulators. *WSEAS Transactions on Systems*, Issue 10, Vol. 3, December 2004, pp. 2991-2996, ISSN: 1109-2777
- Selișteanu, D.; Popescu, D.; Bizdoacă, N. & Ionete, C. (2001). Adaptive and neural control of a rigid link manipulator, *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics SCI 2001*, Vol. IX, Part I, pp. 499-504, ISBN 980-07-7549-8, Orlando, SUA, July 2001
- Slotine, J.J.E. & Li, W. (1987). On the adaptive control of robot manipulators. *International Journal of Robotics Research*, Vol. 6, Issue 3, pp. 49-59, ISSN: 0278-3649
- Slotine, J.J.E. & Li, W. (1989). Composite adaptive manipulator control. *Automatica*, Vol. 25, No. 4, pp. 509-519, ISSN: 0005-1098
- Whitcomb, L.L.; Rizzi, A.A. & Kodishek, D.E. (1991). Comparative experiments with a new adaptive controller for robot arms, *Proceedings of IEEE Conference on Robotics & Automation*, pp. 2-7, Sacramento, USA, 1991
- Zalzala, A. & Morris, A. (1996). *Neural Networks for Robotic Control*, Prentice Hall, ISBN: 978-0131198920

Control of Flexible Manipulators. Theory and Practice

Pereira, E.; Becedas, J.; Payo, I.; Ramos, F. and Feliu, V.
*Universidad de Castilla-La Mancha,
ETS Ingenieros Industriales, Ciudad Real
Spain*

1. Introduction

Novel robotic applications have demanded lighter robots that can be driven using small amounts of energy, for example robotic booms in the aerospace industry, where lightweight manipulators with high performance requirements (high speed operation, better accuracy, high payload/weight ratio) are required (Wang & Gao, 2003). Unfortunately, the flexibility of these robots leads to oscillatory behaviour at the tip of the link, making precise pointing or tip positioning a daunting task that requires complex closed-loop control. In order to address control objectives, such as tip position accuracy and suppression of residual vibration, many control techniques have been applied to flexible robots (see, for instance, the survey (Benosman & Vey, 2004)). There are two main problems that complicate the control design for flexible manipulators viz: (i) the high order of the system, (ii) the no minimum phase dynamics that exists between the tip position and the input (torque applied at the joint). In addition, recently, geometric nonlinearities have been considered in the flexible elements. This chapter gives an overview to the modelling and control of flexible manipulators and focuses in the implementation of the main control techniques for single link flexible manipulators, which is the most studied case in the literature.

2. State of the art

Recently, some reviews in flexible robotics have been published. They divide the previous work attending to some short of classification: control schemes (Benosman & Vey, 2004), modelling (Dwivedy & Eberhard, 2006), overview of main researches (Feliu, 2006), etc. They are usually comprehensive enumerations of the different approaches and/or techniques used in the diverse fields involving flexible manipulators. However, this section intends to give a chronological overview of how flexible manipulators have evolved since visionaries such as Prof. Mark J. Balas or Prof. Wayne J. Book sowed the seeds of this challenging field of robotics. Moreover, some attention is given to main contributions attending to the impact of the work and the goodness of the results.

In the early 70's the necessity of building lighter manipulators able to perform mechanical tasks arises as a part of the USA Space Research. The abusive transportation costs of a gram

of material into orbit and the reduced room and energy available inside an spacecraft cause the imperative need for reducing weight and size as far as possible in any device aboard. Unfortunately, as the manipulator reduces weight, it reduces also accuracy in its manoeuvres due to the appearance of structural flexibility (and hence, vibrations) of the device.

The interest of NASA in creating these manipulators for use in spatial applications motivated the investment for the research of flexible robots and its associated new control problems. In 1974, Prof. Wayne J. Book provided the first known work dealing with this topic explicitly in his Ph. D. Thesis (Book, 1974) entitled as "*Modeling, design and control of flexible manipulators arms*" and supervised by Prof. Daniel E. Whitney, who was a professor at MIT Mechanical Engineering Department. In the same department than Prof. Book, the very same year Dr. Maizza-Neto also studied the control of flexible manipulator arms but from a modal analysis approach (Maizza-Neto, 1974). Fruits of their joint labour, the first work published in a journal in the field of flexible robotics appeared in 1975, dealing with the feedback control of a two-link-two-joints flexible robot (Book et al., 1975). After this milestone, Dr. Maizza-Neto quitted from study of elastic arms but Prof. Book continued with its theoretical analysis of flexible manipulators, e.g. taking frequency domain and space-state approaches (Book & Majette, 1983), until he finally came up with a recursive, lagrangian, assumed modes formulation for modelling a flexible arm (Book, 1984) that incorporates the approach taken by Denavit and Hartenberg (Denavit & Hartenberg, 1955), to describe in a efficient, complete and straightforward way the kinematics and dynamics of elastic manipulators. Due to the generality and simplicity of the technique applied, this work has become one of the most cited and well-known studies in flexible robotics. This structural flexibility was also intensively studied in satellites and other large spacecraft structures (again spatial purposes and NASA behind the scenes) which generally exhibit low structural damping in the materials used and lack of other forms of damping. A special mention deserves Prof. Mark J. Balas, whose generic studies on the control of flexible structures, mainly between 1978 and 1982, e.g. (Balas, 1978) and (Balas, 1982), established some key concepts such as the influence of high nonmodelled dynamics in the system controllability and performance, which is known as "spillover". In addition, the numerical/analytical examples included in his work dealt with controlling and modelling the elasticity of a pinned or cantilevered Euler-Bernoulli beam with a single actuator and a sensor, which is the typical configuration for a one degree of freedom flexible robot as we will discuss in later sections.

After these promising origins, the theoretical challenge of controlling a flexible arm (while still very open) turned into the technological challenge of building a real platform in which testing those control techniques. And there it was, the first known robot exhibiting notorious flexibility to be controlled was built by Dr. Eric Schmitz (Cannon & Schmitz, 1984) under the supervision of Prof. Robert H. Cannon Jr., founder of the Aerospace Robotics Lab and Professor Emeritus at Stanford University. A single-link flexible manipulator was precisely positioned by sensing its tip position while it was actuated on the other end of the link. In this work appeared another essential concept in flexible robots: a flexible robot it is a noncolocated system and thus of nonminimum phase nature. This work is the most referenced ever in the field of flexible robotics and it is considered unanimously as the breakthrough in this topic.

Point-to-point motion of elastic manipulators had been studied with remarkable success taking a number of different approaches, but it was not until 1989 that the tracking control problem of the end-point of a flexible robot was properly addressed. Prof. Siciliano collaborated with Prof. Alessandro De Luca to tackle the problem from a mixed open-closed loop control approach (De Luca & Siciliano, 1989) in the line proposed two years before by Prof. Bayo (Bayo, 1987). Also in 1989, another very important concept called passivity was used for the first time in this field. Prof. David Wang finished his Ph.D Thesis (Wang, 1989) under the advisement of Prof. Mathukumalli Vidyasagar, studying this passivity property of flexible links when an appropriate output of the system was chosen (Wang & Vidyasagar, 1991).

In (Book, 1993), a review on the elastic behaviour of manipulators was meticulously performed. In his conclusions, Prof. Book remarks the exponential growth in the number of publications and also the possibility of corroborating simulation results with experiments, what turns a flexible arm into "one test case for the evaluation of control and dynamics algorithms". And so it was. It is shown in (Benosman & Vey 2004) a summary of the main control theory contributions to flexible manipulators, such as PD-PID, feedforward, adaptive, intelligent, robust, strain feedback, energy-based, wave-based and among others.

3. Modelling of flexible manipulators

One of the most studied problems in flexible robotics is its dynamic modelling (Dwivedy & Eberhard, 2006). Differently to conventional rigid robots, the elastic behaviour of flexible robots makes the mathematical deduction of the models, which govern the real physical behaviour, quite difficult. One of the most important characteristic of the flexible manipulator models is that the low vibration modes have more influence in the system dynamics than the high ones, which allows us to use more simple controllers, with less computational costs and control efforts. Nevertheless, this high order dynamics, which is not considered directly in the controller designed, may give rise to the appearance of bad system behaviours, and sometimes, under specific conditions, instabilities. This problem is usually denoted in the literature as *spillover* (Balas, 1978).

The flexibility in robotics can appear in the joints (manipulators with flexible joints) or in the links (widely known as flexible link manipulators or simply flexible manipulators). The joint flexibility is due to the twisting of the elements that connect the joint and the link. This twisting appears, for instance, in reduction gears when very fast manoeuvres are involved, and produces changes in the joint angles. The link flexibility is due to its deflection when fast manoeuvres or heavy payloads are involved. From a control point of view, the flexibility link problem is quite more challenging than the joint flexibility.

3.1 Single-link flexible manipulators

Single-link flexible manipulators consist of a rigid part, also denominated as actuator, which produces the spatial movement of the structure; and by a flexible part, which presents distributed elasticity along the whole structure. Fig. 1 shows the parametric representation of a single-link flexible manipulator, which is composed of the following: (a) a motor and a reduction gear of $1:n_r$ reduction ratio at the base, with total inertia (rotor and hub) J_0 , dynamic friction coefficient ν and Coulomb friction torque Γ_f ; (b) a flexible link with uniform linear mass density ρ , uniform bending stiffness EI and length L ; and (c) a payload

of mass M_p and rotational inertia J_p . Furthermore, the applied torque is Γ_m , Γ_{coup} denotes the coupling torque between the motor and the link, θ_m is the joint angle and θ_t represents the tip angle.

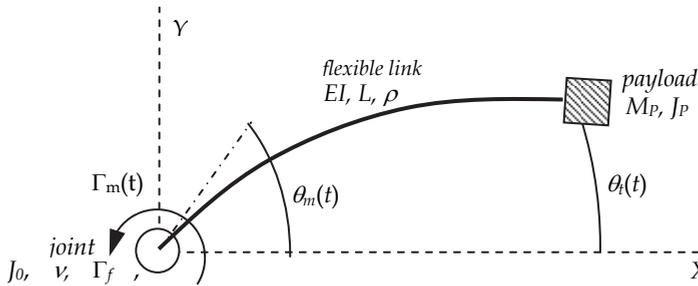


Fig. 1. Parametric representation of a single link flexible manipulator with a rotational joint.

The dynamic behaviour of the system is governed by a differential partial equation which presents infinite vibration modes. The objective is to obtain a simplified model (finite number of vibration modes) of the differential equation that characterizes the dynamics of the link. A number of models can be found in the literatures obtained from methods such as the truncation of the infinite dimensional model (Cannon & Schmitz, 1984); the discretization of the link based on finite elements (Bayo, 1987); or directly from concentrated mass models (Feliu et al., 1992).

The hypothesis of negligible gravity effect and horizontal motion are considered in the deduction of the model equations. In addition, the magnitudes seen from the motor side of the gear will be written with an upper hat, while the magnitudes seen from the link side will be denoted by standard letters. With this notation and these hypotheses, the momentum balance at the output side of the gear is given by the following expression

$$\hat{\Gamma}_m(t) = K_m u(t) = J_0 \hat{\theta}_m(t) + \nu \hat{\theta}_m(t) + \hat{\Gamma}_f(t) + \hat{\Gamma}_{coup}(t), \tag{1}$$

where K_m is the motor constant that models the electric part of the motor (using a current servoamplifier) and u is the motor input voltage. This equation can be represented in a block diagram as shown in

Fig. 2, where $G_c(s)$ and $G_t(s)$ are the transfer functions from θ_m to Γ_{coup} and θ_t respectively.

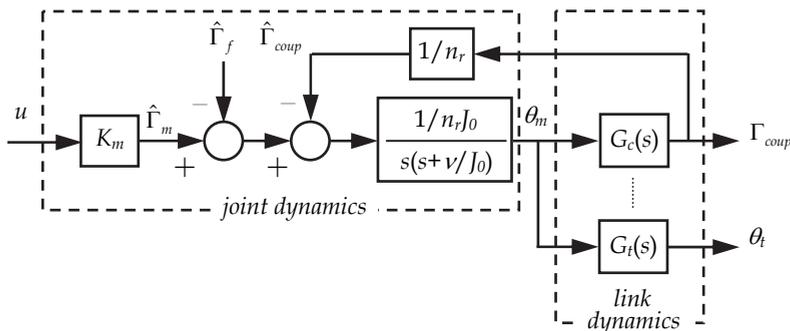


Fig. 2. Block diagram of the single-link flexible manipulator system.

The link model is deduced by considering small deformations, which allows us to use a linear beam model to obtain the dynamic equations. Based on this hypothesis, in this chapter we use models derived from the truncation of infinite dimensional model obtained from concentrated mass model and assumed mode method.

3.1.1 Concentrated mass models

In the concentrated mass models, the link mass is concentrated in several points along the whole structure (see Fig. 3), where the inertia produced by the point mass rotations is rejected. An example of this technique can be found in (Feliu et al., 1992). Fig. 3 shows the scheme of the concentrated mass model. The lumped masses are represented by m_i , with $1 \leq i \leq n$; the distance between two consecutive masses $i-1$ and i is l_i , l_1 is the distance between the motor shaft and the first mass; finally, the distance between the mass m_i and the motor shaft is L_i . F_n represents the applied external force at the tip of the link. Γ_n is the torque applied in the same location. Assuming small deflections and considering that the stiffness EI is constant through each interval of the beam the deflection is given by a third order polynomial:

$$y_i(x) = u_{i,0} + u_{i,1}(x - L_{i-1}) + u_{i,2}(x - L_{i-1})^2 + u_{i,3}(x - L_{i-1})^3, \quad (2)$$

where u_{ij} are the different coefficients for each interval, and $L_0=0$.

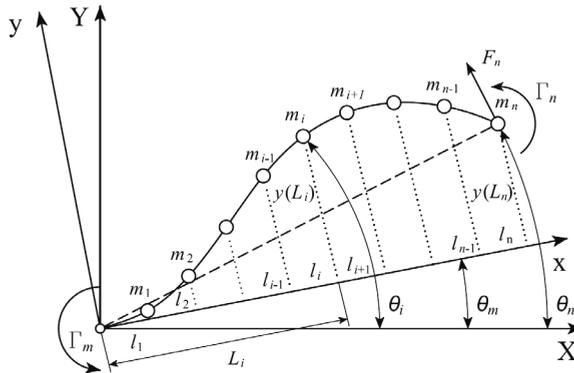


Fig. 3. Concentrated masses model of a single-link flexible manipulator.

The dynamic model of the flexible link is obtained from some geometric and dynamic equations as follows (see (Feliu et al., 1992) for more details):

$$\underline{M} \frac{d^2 \underline{\Theta}}{dt^2} = EI [\underline{A} \underline{\Theta} + \underline{B} \underline{\theta}_m] + \underline{P} \Gamma_n + \underline{Q} F_n, \quad (3)$$

where $\underline{M} = \text{diag}(m_1, m_2, \dots, m_n)$ represents the masses matrix of the system and $\underline{\Theta} = [\theta_1, \theta_2, \dots, \theta_n]^T$. On the other hand, $\underline{A} \in \mathfrak{R}^{n \times n}$ is a constant matrix, $\underline{B} = -\underline{A}[1, 1, \dots, 1]^T$, $\underline{P} \in \mathfrak{R}^{n \times 1}$ and $\underline{Q} \in \mathfrak{R}^{n \times 1}$ are constant column vectors, which only depend on the link geometry.

Finally, the coupling torque affecting the motor dynamics (see Equation (1)) is defined as $\Gamma_{coup} = -2EIu_{1,2}$. Notice that the coupling torque has the same magnitude and different sign to the joint torque $2EIu_{1,2}$. This torque can be expressed as a linear function:

$$\Gamma_{coup} = \underline{C}\underline{\Theta} - c_{n+1}\theta_m - c_{n+2}\Gamma_n, \quad (4)$$

where $\underline{C} = (c_1, c_2, \dots, c_n)$, c_i , $1 \leq i \leq n+2$, are parameters which do not depend on the concentrated masses along the structure and $c_{n+1} = -\underline{C}[1, 1, \dots, 1]^T$.

For example, the transfer functions $G_c(s)$ and $G_r(s)$ for only one point mass located in the tip (m_1) are as follows:

$$G_c(s) = (3EI/L)/(s^2 + \omega_1^2) \text{ and } G_r(s) = (\omega_1^2)/(s^2 + \omega_1^2), \quad (5)$$

in which $\omega_1 = \sqrt{3EI/L^3 m_1}$. This model can be used for flexible robots with a high payload/weight ratio.

3.1.2 Assumed mode method

The dynamic behaviour of an Euler-Bernoulli beam is governed by the following PDE (see, for example, (Meirovitch, 1996))

$$EIw^{IV}(x,t) + \rho\ddot{w}(x,t) = f(x,t), \quad (6)$$

where $f(x,t)$ is a distributed external force, w is the elastic deflection measured from the undeformed link. Then, from modal analysis of Equation (6), which considers $w(x,t)$ as

$$w(x,t) = \sum_{i=1}^{\infty} \phi_i(x) \eta_i(t), \quad (7)$$

in which $\phi_i(x)$ are the eigenfunctions and $\eta_i(t)$ are the generalized coordinates, the system model can be obtained (see (Belleza et al., 1990) for more details).

3.2 Multi-link flexible manipulators

For these types of manipulators truncated models are also used. Some examples are: (De Luca & Siciliano, 1991) for planar manipulators, (Pedersen & Pedersen, 1998) for 3 degree of freedom manipulators and (Schwertassek et al., 1999), in which the election of shape functions is discussed.

The deflections are calculated from the following expression:

$$w_i(x,t) = \underline{\Phi}_i^T(x) \cdot \underline{\Delta}_i(t), \quad 1 \leq i \leq n_L, \quad (8)$$

(see for example (Benosman & Vey 2004)), in which i means the number of the link, n_L the number of links, $\underline{\Phi}_i(x)$ is a column vector with the shape functions of the link (for each considered mode), $\underline{\Delta}_i(t) = (\eta_{1i}, \dots, \eta_{Ni})^T$ is a column vector that represents the dynamics of each mode, in which N is the number of modes considered.

The dynamics equations of the overall system from the Lagrange method are described as follows:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} + \frac{\partial D_R}{\partial \dot{q}_k} = u_k, \quad (9)$$

where L is the lagrangian defined as $L=E-P$, being E the total kinetic energy of the manipulator and P its potential energy. This expression is similar to the used in rigid robots, but in this case the potential energy is the sum of the gravity and the elastic deformation terms. The term D_R is the dissipation function of Rayleigh, which allows us to include dissipative terms like frictions, and u_k is the generalized force applied in q_k . From Equation (9) the robot dynamics can be deduced (see for example Chapter 1 of (Wang & Gao, 2003))

$$\underline{I}(\underline{Q}) \cdot \ddot{\underline{Q}} + \underline{b}(\underline{Q}, \dot{\underline{Q}}) + \underline{K}(\underline{Q}) \cdot \underline{Q} + \underline{D} \cdot \dot{\underline{Q}} + \underline{g}(\underline{Q}) = \underline{F} \cdot \underline{\Gamma}, \quad (10)$$

were $\underline{Q}=(\theta_1, \dots, \theta_{nL} | \underline{\Delta}_1, \dots, \underline{\Delta}_{nL})^T$ is the vector of generalized coordinates that includes the first block of joint angles θ_i (rigid part of the model) and the elastic deflections of the links $\underline{\Delta}_i$; $\underline{\Gamma}$ is the vector of motor torques of the joints, \underline{I} is the inertias matrix of the links and the payload of the robot, which is positive definite symmetric, \underline{b} is the vector that represents the spin and Coriolis forces ($\underline{b} = \beta(\underline{Q}, \dot{\underline{Q}}) \cdot \dot{\underline{Q}}$), \underline{K} is stiffness matrix, \underline{D} is the damping matrix, \underline{g} is the gravity vector and \underline{F} is the connection matrix between the joints and the mechanism. Equation (10) presents a similar structure to the dynamics of a rigid robot with the differences of: (i) the elasticity term ($\underline{K}(\underline{Q}) \cdot \underline{Q}$) and (ii) the vector of generalized coordinates is extended by vectors that include the link flexibility.

3.3 Flexible joints

In this sort of systems, differently to the flexible link robots, in which the flexibility was found in the whole structure from the hub with the actuator to the tip position, the flexibility appears as a consequence of a twist in those elements which connect the actuators with the links, and this effect has always rotational nature. Therefore, the reduction gears used to connect the actuators with the links can experiment this effect when they are subject to very fast movements. Such a joint flexibility can be modelled as a linear spring (Spong, 1987) or as a torsion spring (Yuan & Lin, 1990). Surveys devoted to this kind of robots are (Bridges et al., 1995) and (Ozgoli & Taghirad, 2006), in which a comparison between the most used methods in controlling this kind of systems is carried out. Nevertheless, this problem in flexible joints sometimes appears combined with flexible link manipulators. Examples of this problem are studied in (Yang & Donath, 1988) and (Yuan & Lin, 1990).

4. Control techniques

This section summarizes the main control techniques for flexible manipulators, which are classified into position and force control.

4.1 Position Control

The benefits and interests jointly with advantages and disadvantages of the most relevant contributions referent to open and closed control schemes for position control of flexible manipulators have been included in the following subsections:

4.1.1 Command generation

A great number of research works have proposed command generation techniques, which can be primarily classified into pre-computed and real-time. An example of pre-computed is (Aspinwall, 1980), where a Fourier expansion was proposed to generate a trajectory that reduces the peaks of the frequency spectrum at discrete points. Another pre-computed alternative uses multi-switch bang-bang functions that produce a time-optimal motion. However, this alternative requires the accurate selection of switching times which depends on the dynamic model of the system (Onsay & Akay, 1991). The main problem of pre-computed command profiles is that the vibration reduction is not guaranteed if a change in the trajectory is produced.

The most used reference command generation is based on filtering the desired trajectory in real time by using an input shaper (IS). An IS is a particular case of a finite impulse response filter that obtains the command reference by convolving the desired trajectory with a sequence of impulses (filter coefficients) ((Smith, 1958) and (Singer & Seering, 1990)). This control is widely extended in the industry and there are many different applications of IS such as spacecraft field (Tuttle & Seering, 1997), cranes and structures like cranes (see applications and performance comparisons in (Huey et al., 2008)) or nanopositioners (Jordan, 2002). One of the main problems of IS design is to deal with system uncertainties. The approaches to solve this main problem can be classified into robust (see the survey of (Vaughan et al., 2008)), learning ((Park & Chang, 2001) and (Park et al., 2006)) or adaptive input shaping (Bodson, 1998).

IS technique has also been combined with joint position control ((Feliu & Rattan 1999) and (Mohamed et al., 2005)), which guarantees trajectory tracking of the joint angle reference and makes the controlled system robust to joint frictions. The main advantages of this control scheme are the simplicity of the control design, since an accurate knowledge of the system is not necessary, and the robustness to unmodelled dynamics (spillover) and changes in the systems parameters (by using the aforementioned robust, adaptive and learning approaches). However, these control schemes are not robust to external disturbance, which has motivated closed loop controllers to be used in active vibration damping.

4.1.2 Classic control techniques

In this chapter, the term “classic control techniques” for flexible manipulators refers to control laws derived from the classic control theory, such as proportional, derivative and/or integral action, or phase-lag controllers. Thus, classic control techniques, like Proportional-Derivative (PD) control (De Luca & Siciliano, 1993) or Lead-Lag control (Feliu et al., 1993) among others, have been proposed in order to control the joint and tip position (angle) of a lightweight flexible manipulator. The main advantage of these techniques is the simplicity of its design, which makes this control very attractive from an industrial point of view. However, in situations of changes in the system, its performance is worse (slow time

response, worse accuracy in the control task...) than other control techniques such as robust, adaptive or learning approaches among others. Nevertheless, they can be used in combination with more modern and robust techniques (e.g. passive and robust control theories) to obtain a controller more adequate and versatile to do a determined control task, as a consequence of its easy implementation. Classic control techniques are more convenient when minimum phase systems are used (see discussions of (Wang et al., 1989)), which can be obtained by choosing an appropriate output ((Gerverter, 1970), (Luo, 1993) and (Pereira et al., 2007)) or by redefining it ((Wang & Vidyasagar 1992) and (Liu & Yuan, 2003)).

4.1.3 Robust, Optimal and Sliding Mode Control

It is widely recognized that many systems have inherently uncertainties, which can be parameters variations or simple lack of knowledge of their physical parameters, external disturbances, unmodelled dynamics or errors in the models because of simplicities or nonlinearities. These uncertainties may lead to inaccurate position control or even sometimes make the closed-loop system unstable. The robust control deals with these uncertainties (Korolov & Chen, 1989), taking them into account in the design of the control law or by using some analysis techniques to make the system robust to any or several of these uncertainties. The output/input linearization added to Linear Quadratic Regulator (LQR) was applied in (Singh & Schy, 1985). Nevertheless, LQR regulators are avoided to be applied in practical setups because of the well-known spillover problems. The Linear Quadratic Gaussian (LQG) was investigated in (Cannon & Schmitz, 1984) and (Balas, 1982). However, these LQG regulators do not guarantee general stability margins (Banavar & Dominic, 1995). Nonlinear robust control method has been proposed by using singular perturbation approach (Morita et al., 1997). To design robust controllers, Lyapunov's second method is widely used (Gutman, 1999). Nevertheless the design is not that simple, because the main difficulty is the non trivial finding of a Lyapunov function for control design. Some examples in using this technique to control the end-effector of a flexible manipulator are (Theodore & Ghosal, 2003) and (Jiang, 2004).

Another robust control technique which has been used by many researchers is the optimal H_∞ control, which is derived from the L2-gain analysis (Yim et al., 2006). Applications of this technique to control of flexible manipulators can be found in (Moser, 1993), (Landau et al., 1996), (Wang et al., 2002) and (Lizarraga & Etxebarria, 2003) among others.

Major research effort has been devoted to the development of the robust control based on Sliding Mode Control. This control is based on a nonlinear control law, which alters the dynamics of the system to be controlled by applying a high frequency switching control. One of the relevant characteristics of this sort of controllers is the augmented state feedback, which is not a continuous function of time. The goal of these controllers is to catch up with the designed sliding surface, which insures asymptotic stability. Some relevant publications in flexible robots are the following: (Choi et al., 1995), (Moallem et al., 1998), (Chen & Hsu, 2001) and (Thomas & Mija, 2008).

4.1.4 Adaptive control

Adaptive control arises as a solution for systems in which some of their parameters are unknown or change in time (Åström & Wittenmark, 1995). The answer to such a problem consists in developing a control system capable of monitoring his behaviour and adjusting

the controller parameters in order to increase the working accuracy. Thus, adaptive control is a combination of both control theory, which solves the problem of obtaining a desired system response to a given system input, and system identification theory, which deals with the problem of unknown parameters.

For obvious reasons, robotics has been a platinum client of adaptive control since first robot was foreseen. Manipulators are general purpose mechanisms designed to perform arbitrary tasks with arbitrary movements. That broad definition leaves the door open for changes in the system, some of which noticeably modify the dynamics of the system, e.g. payload changes (Bai et al., 1998).

Let us use a simple classification for adaptive control techniques, which groups them in (Åström & Wittenmark, 1995):

- **Direct Adaptive Control**, also called **Control with Implicit Identification (CII)**: the system parameters are not identified. Instead, the controller parameters are adjusted directly depending on the behaviour of the system. CII reduces the computational complexity and has a good performance in experimental applications. This reduction is mainly due to the controller parameters are adjusted only when an accurate estimation of the uncertainties is obtained, which requires, in addition to aforementioned accuracy, a fast estimation.

- **Indirect Adaptive Control**, also called **Control with Explicit Identification (CEI)**: the system parameters estimations are obtained on line and the controller parameters are adjusted or updated depending on such estimations. CEI presents good performance but they are not extendedly implemented in practical applications due to their complexity, high computational costs and insufficient control performance at start-up of the controllers.

First works on adaptive control applied to flexible robots were carried out in second half of 80's (Siciliano et al., 1986), (Rovner & Cannon, 1987) and (Koivo & Lee, 1989), but its study has been constant along the time up to date, with application to real projects such as the Canadian SRMS (Damaren, 1996). Works based on the direct adaptive control approach can be found: (Siciliano et al., 1986), (Christoforou & Damaren 2000) and (Damaren, 1996); and on the indirect adaptive control idea: (Rovner & Cannon, 1987) and (Feliu et al., 1990). In this last paper a camera was used as a sensorial system to close the control loop and track the tip position of the flexible robot. In other later work (Feliu et al., 1999), an accelerometer was used to carry out with the same objective, but presented some inaccuracies due to the inclusion of the actuator and its strong nonlinearities (Coulomb friction) in the estimation process. Recently, new indirect approaches have appeared due to improvements in sensorial system (Ramos & Feliu, 2008) or in estimation methods (Becedas et al., 2009), which reduce substantially the estimation time without reducing its accuracy. In both last works strain gauges located in the coupling between the flexible link and the actuator were used to estimate the tip position of the flexible robot.

4.1.5 Intelligent control

Ideally, an autonomous system must have the ability of learning what to do when there are changes in the plant or in the environment, ability that conventional control systems totally lack of. Intelligent control provides some techniques to obtain this learning and to apply it appropriately to achieve a good system performance. Learning control (as known in its

beginnings) started to be studied in the 60's (some surveys of this period are (Tsytkin, 1968) and (Fu, 1970)), and its popularity and applications have increased continuously since, being applied in almost all spheres of science and technology. Within these techniques, we can highlight *machine learning*, *fuzzy logic* and *neural networks*.

Due to the property of adaptability, inherent to any learning process, all of these schemes have been widely applied to control of robotic manipulator (see e.g. (Ge et al., 1998)), which are systems subjected to substantial and habitual changes in its dynamics (as commented before). In flexible robots, because of the undesired vibration in the structure due to elasticity, this ability becomes even more interesting. For instance, neural networks can be trained for attaining good responses without having an accurate model or any model at all. The drawbacks are: the need for being trained might take a considerable amount of time at the preparation stage; and their inherent nonlinear nature makes this systems quite demanding computationally. On the other hand, fuzzy logic is an empirical rules method that uses human experience in the control law. Again, model is not important to fuzzy logic as much as these rules implemented in the controller, which rely mainly on the experience of the designer when dealing with a particular system. This means that the controller can take into account not only numbers but also human knowledge. However, the performance of the controller depends strongly on the rules introduced, hence needing to take special care in the design-preparation stage, and the oversight of a certain conduct might lead to an unexpected behaviour. Some examples of these approaches are described in (Su & Khorasani, 2001), (Tian et al., 2004) and (Talebi et al., 2009) using neural networks; (Moudgal et al., 1995), (Green, & Sasiadek, 2002) and (Renno, 2007) using fuzzy logic; or (Caswar & Unbehauen, 2002) and (Subudhi & Morris, 2009) presenting hybrid neuro-fuzzy proposals.

4.2 Force control

Manipulator robots are designed to help to humans in their daily work, carrying out repetitive, precise or dangerous tasks. These tasks can be grouped into two categories: *unconstrained tasks*, in which the manipulator moves freely, and *constrained task*, in which the manipulator interacts with the environment, e.g. cutting, assembly, gripping, polishing or drilling.

Typically, the control techniques used for *unconstrained tasks* are focused to the motion control of the manipulator, in particular, so that the end-effector of the manipulator follows a planned trajectory. On the other hand, the control techniques used for *constrained tasks* can be grouped into two categories: *indirect force control* and *direct force control* (Siciliano & Villani, 1999). In the first case, the contact force control is achieved via motion control, without feeding back the contact force. In the second case, the contact force control is achieved thanks to a force feedback control scheme. In the *indirect force control* the position error is related to the contact force through a mechanical stiffness or impedance of adjustable parameters. Two control strategies which belong to this category are: *compliance (or stiffness) control* and *impedance control*. The *direct force control* can be used when a force sensor is available and therefore, the force measurements are considered in a closed loop control law. A control strategy belonging to this category is the *hybrid position/force control*, which performs a position control along the unconstrained task directions and a force control along the constrained task directions. Other strategy used in the *direct force control* is the *inner/outer motion /force control*, in which an outer closed loop force control works on an inner closed loop motion control.

There are also other advanced force controls that can work in combination with the previous techniques mentioned, e.g. adaptive, robust or intelligent control. A wide overview of the all above force control strategies can be found in the following works: (Whitney, 1987), (Zeng & Hemami, 1997) and (Siciliano & Villani, 1999). All these force control strategies are commonly used in rigid industrial manipulators but this kind of robots has some problems in interaction tasks because their high weight and inertia and their lack of touch senses in the structure. This becomes complicated any interaction task with any kind of surface because rigid robots do not absorb a great amount of energy in the impact, being any interaction between rigid robots and objects or humans quite dangerous.

The force control in flexible robots arises to solve these problems in interaction tasks in which the rigid robots are not appropriated. A comparative study between rigid and flexible robots performing constrained tasks in contact with a deformable environment is carried out in (Latornell et al., 1998). In these cases, a carefully analysis of the contact forces between the manipulator and the environment must be done. A literature survey of contact dynamics modelling is shown in (Gilardi & Sharf, 2002).

Some robotic applications demand manipulators with elastic links, like robotic arms mounted on other vehicles such a wheelchairs for handicapped people; minimally invasive surgery carried out with thin flexible instruments, and manipulation of fragile objects with elastic robotic fingers among others. The use of deformable flexible robotic fingers improves the limited capabilities of robotic rigid fingers, as is shown in survey (Shimoga, 1996). A review of robotic grasping and contact, for rigid and flexible fingers, can be also found in (Bicchi & Kumar, 2000).

Flexible robots are able to absorb a great amount of energy in the impact with any kind of surface, principally, those quite rigid, which can damage the robot, and those tender, like human parts, which can be damaged easily in an impact with any rigid object. Nevertheless, despite these favourable characteristics, an important aspect must be considered when a flexible robot is used: the appearance of vibrations because of the high structural flexibility. Thus, a greater control effort is required to deal with structural vibrations, which also requires more complex designs, because of the more complex dynamics models, to achieve a good control of these robots. Some of the published works on force control for flexible robots subject, by using different techniques, are, as e.g., (Chiou & Shahinpoor, 1988), (Yoshikawa et al., 1996), (Yamano et al., 2004) and (Palejiya & Tanner, 2006), where a hybrid position/force control was performed; in (Chapnik, et al., 1993) an open-loop control system using 2 frequency-domain techniques was designed; in (Matsuno & Kasai, 1998) and (Morita et al., 2001) an optimal control was used in experiments; in (Becedas et al., 2008) a force control based on a flatness technique was proposed; in (Tian et al., 2004) and (Shi & Trabia, 2005) neural networks and fuzzy logic techniques were respectively used; in (Siciliano & Villani, 2000) and (Vossoughi & Karimzadeh, 2006), the singular perturbation method was used to control, in both, a two degree-of-freedom planar flexible link manipulator; and finally in (Garcia et al., 2003) a force control is carried out for a robot of three degree-of-freedom.

Unlike the works before mentioned control, which only analyze the constrained motion of the robot, there are models and control laws designed to properly work on the force control, for free and constrained manipulator motions. The pre-impact (free motion) and post-impact (constrained motion) were analyzed in (Payo et al., 2009), where a modified PID controller was proposed to work properly for unconstrained and constrained tasks. The

authors only used measurements of the bending moment at the root of the arm in a closed loop control law. This same force control technique for flexible robots was also used in (Becedas et al., 2008) to design a flexible finger gripper, but in this case the implemented controller was a GPI controller that presents the characteristics described in Section 0

5. Design and implementation of the main control techniques for single-link flexible manipulators

Control of single link flexible manipulators is the most studied case in the literature (85% of the published works related to this field (Feliu, 2006)), but even nowadays, new control approaches are still being applied to this problem. Therefore, the examples presented in this section implement some recent control approaches of this kind of flexible manipulators.

5.1 Experimental platforms

5.1.1 Single link flexible manipulator with one significant vibration mode

In this case, the flexible arm is driven by a Harmonic Drive mini servo DC motor RH-8D-6006-E050A-SP(N), supported by a three-legged metallic structure, which has a gear with a reduction ratio of 1:50. The arm is made of a very lightweight carbon fibre rod and supports a load (several times the weight of the arm) at the tip. This load slides over an air table, which provides a friction-free tip planar motion. The load is a disc mass that can freely spin (thanks to a bearing) without producing a torque at the tip. The sensor system is integrated by an encoder embedded in the motor and a couple of strain gauges placed on to both sides of the root of the arm to measure the torque. The physical characteristics of the platform are specified in Table 1. Equation (5) is used for modelling the link of this flexible manipulator, in which the value of m_l is equal to M_p . For a better understanding of the setup, the following references can be consulted (Payo et al., 2009) and (Becedas et al., 2009). Fig. 4a shows a picture of the experimental platform.

5.1.2 Single link flexible manipulator with three significant vibration modes

The setup consists of a DC motor with a reduction gear 1:50 (HFUC-32-50-20H); a slender arm made of aluminium flexible beam with rectangular section, which is attached to the motor hub in such way that it rotates only in the horizontal plane, so that the effect of gravity can be ignored; and a mass at the end of the arm. In addition, two sensors are used: an encoder is mounted at the joint of the manipulator to measure the motor angle, and a strain-gauge bridge, placed at the base of the beam to measure the coupling torque. The physical characteristics of the system are shown in Table 1. The flexible arm is approximated by a truncated model of Equation (7) with the first three vibration modes to carry out the simulations (Bellezza et al., 1990). The natural frequencies of the one end clamped link model obtained from this approximate model, almost exactly reproduce the real frequencies of the system, which were determined experimentally. More information about this experimental setup can be found in (Feliu et al., 2006). Fig. 4b shows a picture of the experimental platform.

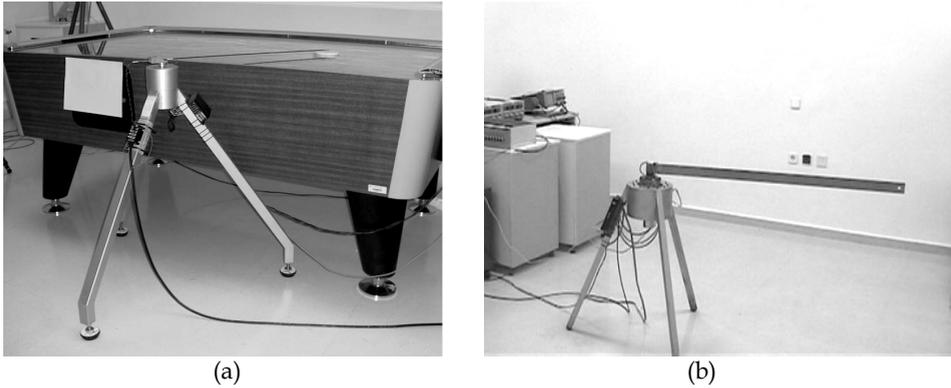


Fig. 4. Experimental platforms: (a) Single link flexible arm with one significant vibration mode; (b) Single link flexible arm with three significant vibration modes.

PARAMETER	DESCRIPTION	PLATFORM 1 VALUE	PLATFORM 2 VALUE
Data of the flexible link			
EI	Stiffness	0.37 Nm ²	2.40 Nm ²
l	Length	0.7 m	1.26 m
d	Diameter	$2.80 \cdot 10^{-3}$ m	-
h	Width	-	$5 \cdot 10^{-2}$ m
b	Thickness	-	$2 \cdot 10^{-3}$ m
M_p	Mass in the tip	0.03 kg	0-0.30 kg
J_p	Inertia in the tip	-	$0.588 \cdot 10^{-4}$ kgm ²
Data of the motor-gear set			
J_o	Inertia	$6.87 \cdot 10^{-5}$ kgm ²	$3.16 \cdot 10^{-4}$ kgm ²
ν	Viscous friction	$1.04 \cdot 10^{-3}$ kgm ² s	$1.39 \cdot 10^{-3}$ kgm ² s
n_r	Reduction ratio of the motor gear	50	50
K_m	Motor constant	$2.10 \cdot 10^{-1}$ Nm/V	$4.74 \cdot 10^{-1}$ Nm/V
u_{sat}	Saturation voltage of the servo amplifier	± 10 V	± 3.3 V

Table 1. Physical characteristics of the utilized experimental platforms.

5.2 Actuator position control.

Control scheme shown in Fig. 5 is used to position the joint angle. This controller makes the system less sensible to unknown bounded disturbances (Γ_{coup} in Equation (1)) and minimizes the effects of joint frictions (see, for instance (Feliu et al., 1993)). Thus, the joint angle can be controlled without considering the link dynamics by using a PD, PID or a Generalized Proportional Integral (GPI) controller, generically denoted as $C_a(s)$. In addition, this controller, as we will show bellow, can be combined with other control techniques, such as command generation, passivity based control, adaptive control or force control.

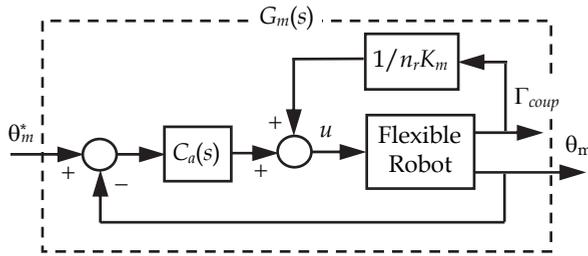


Fig. 5. Schematic of the inner control loop formed by a position control of θ_m plus the decoupling term $\Gamma_{coup}/n_r K_m$.

5.3 Command generation

The implementation of the IS technique as an example of command generation is described herein. It is usually accompanied by the feedback controller like the one shows in Fig. 5. Thus, the general control scheme showed in Fig. 6 is used, which has previously utilized with success for example in (Feliu & Rattan, 1999) or (Mohamed et al., 2005). The actuator controller is decided to be a PD with the following control law:

$$u(t) = \Gamma_{coup}(t)/n_r K_m + K_p (\theta_m^*(t) - \theta_m(t)) - K_v \dot{\theta}_m(t), \tag{11}$$

where $\Gamma_{coup}/n_r K_m$ (decoupling term) makes the design of the PD constants (K_p , K_v) independent of the link dynamics. Thus, if the tuning of the parameters of the PD controller (K_p , K_v) is carried out to achieve a critically damped second-order system, the dynamics of the inner control loop ($G_m(s)$) can be approximated by

$$\theta_m(s) \cong G_m(s) \theta_m^*(s) = \theta_m^*(s) / (1 + \alpha s)^2, \tag{12}$$

where α is the constant time of $G_m(s)$. From Equations (11) and (12) the values of K_p and K_v are obtained as

$$K_p = J_0 n_r / K_m \alpha^2, \quad K_v = n_r (2J_0 - v\alpha) / K_m \alpha. \tag{13}$$

As it was commented in Section 0, the IS ($C(s)$) can be a robust, learning or adaptive input shaper. In this section, a robust input shaper (RIS) for each vibration mode obtained by the so-called derivative method (Vaughan et al., 2008) is implemented. This multi-mode RIS is obtained as follows:

$$C(s) = \prod_{i=1}^N C_i(s) = \prod_{i=1}^N \left((1 + z_i e^{-sd_i}) / (1 + z_i) \right)^{p_i}, \tag{14}$$

in which

$$z_i = e^{\xi_i / \sqrt{1 - \xi_i^2}}, \quad d_i = \pi / \left(\omega_i \sqrt{1 - \xi_i^2} \right), \tag{15}$$

p_i is a positive integer used to increase the robustness of each $C_i(s)$ and ω_i and ξ_i denote the natural frequencies and damping ratio of each considered vibration mode.

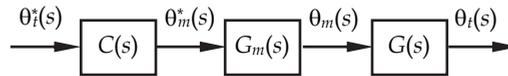
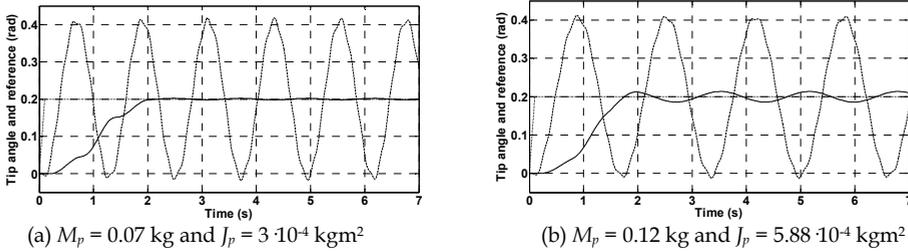


Fig. 6. General control scheme of the RIS implementation.

This example illustrates the design for the experimental platform of Fig. 4b of the multi-mode RIS of Equation (14) for a payload range $M_p \in [0.02, 0.12]$ kg and $J_p \in [0.0, 5.88 \cdot 10^{-4}]$ kgm². Each of one $C_i(s)$ is designed for the centre of three first frequency intervals, which has the next values: $\omega_1=5.16$ $\omega_2=35.34$ and $\omega_3=100.59$ rad/s. If the damping is neglected (ξ_1 , ξ_2 and ξ_3 equal to zero), the parameters of $C(s)$ are $z_1=z_2=z_3=1$, $d_1=0.61$, $d_2=0.089$ and $d_3=0.031$ s. In addition, if the maximum residual vibration is kept under 5% for all vibration modes, the value of each p_i is: $p_1=3$, $p_2=2$ and $p_3=2$. The dynamics of $G_m(s)$ is designed for $\alpha=0.01$. Then from Table 1 and Equations (12) and (13), the values of K_p and K_v were 350.9 and 6.9. This value of α makes the transfer function $G_m(s)$ robust to Coulomb friction and does not saturate the DC motor if the motor angle reference is ramp a reference with slope and final value equal to 2 and 0.2rad, respectively. Fig. 7 shows the experimental results for the multi-mode RIS design above. The residual vibration for the nominal payload ($M_p=0.07$ kg and $J_p=3 \cdot 10^{-4}$ kgm²) is approximately zero whereas one of the payload limits ($M_p = 0.12$ kg and $J_p = 5.88 \cdot 10^{-4}$ kgm²) has a residual vibration less than 5%.



(a) $M_p = 0.07$ kg and $J_p = 3 \cdot 10^{-4}$ kgm²

(b) $M_p = 0.12$ kg and $J_p = 5.88 \cdot 10^{-4}$ kgm²

Fig. 7. Experimental results for the multi-mode RIS. (...) References, (---) without RIS and (—) with RIS.

5.4 Classic control techniques

This subsection implements the new passivity methodology expounded in (Pereira et al., 2007) in the experimental platform of Fig. 4b, whose general control scheme is shown in Fig. 8. This control uses two control loops. The first one consists of the actuator control shown in Section 5.2, which allows us to employ an integral action or a high proportional gain. Thus, the system is robust to joint frictions. The outer controller is based on the passivity property of $\Gamma_{coup}(s)/s\theta_m(s)$, which is independent of the link and payload parameters. Thus, if $sC(s)G_m(s)$ is passive, the controller system is stable. The used outer controller is as following:

$$C(s) = K_c(\lambda s + 1)/s, \quad (16)$$

in which the parameter K_c imparts damping to the controlled system and λ must be chosen together with $G_m(s)$ to guarantee the stability. For example, if $G_m(s)$ is equal to Equation (12),

the necessary and sufficient stability condition is $0 < \alpha/2 < \lambda$ (see (Pereira et al., 2007) for more details).

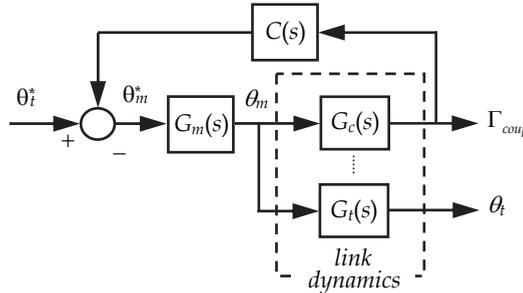


Fig. 8. General control scheme proposed in (Pereira, et al., 2007).

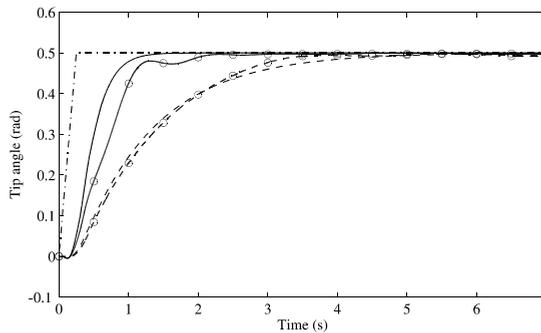


Fig. 9. Tip angle θ_t : (—). Simulation with $M_p = 0$; (\ominus) Experiment with $M_p = 0$; (---) Simulation with $M_p = 0.3$; (\oplus) Experiment with $M_p = 0.3$; (---) the reference.

Taking into account the maximum motor torque (i.e., u_{sat} in Table 1), the constant time of the inner loop is set to be $\alpha = 0.02$. Then, the parameters of the PD controller are obtained: $K_p = 83.72$ and $K_v = 3.35$. Next, the nominal condition is taken for $M_p = 0$ and $C'(s)$ is designed ($\lambda = 0.05$ and $K_c = 1.8$) in such a way that the poles corresponding to the first vibration mode are placed at -3.8 . Notice that λ fulfils the condition $0 < \alpha/2 < \lambda$ and is independent of the payload. Once the parameters of the control scheme are set, we carry out simulations and experiments for $M_p = 0$ and $M_p = 0.3$ kg (approximately the weight of the beam) and $J_p \cong 0$ kgm²). Figure 9 shows the tip angle, in which can be seen that the response for the two mass values without changing the control parameters is acceptable for both simulations and experiments. Notice that the experimental tip position response is estimated by a fully observer since it is not measured directly, which is not used for control purpose. Finally, a steady state error in the vicinity of 1% compared with the reference command arises for in the tip and motor angle for experimental results. This error is due to Coulomb friction and can be minimized using a PD with higher gains in the actuator control.

5.5 Adaptive control

Adaptive controller described in this section is based on the flatness characteristic of a flexible robotic system (see (Becedas, et al., 2009)). The control system is based on two

nested loops with two controllers designed for both motor and flexible link dynamics. The controller is called Generalized Proportional Integral (GPI). This presents robustness with respect to constant perturbations and does not require computation of derivatives of the system output signals. Therefore, the output signals are directly feedbacked in the control loops, then the usual delays produced by the computation of derivatives and the high computational costs that require the use of observers do not appear. In addition, due to the fact that one of the most changeable parameter in robotics is the payload, a fast algebraic continuous time estimator (see (Fließ & Sira-Ramírez, 2003)) is designed to on-line estimate the natural frequency of vibration in real time. The estimator calculates the real value of the natural frequency when the payload changes and updates the gains of the controllers. Therefore, this control scheme is an Indirect Adaptive Control. A scheme of the adaptive control system is depicted in Fig. 10, where $\omega_{n,e}$ represents the estimation of the vibration natural frequency of the flexible arm, used to update the system controller parameters.

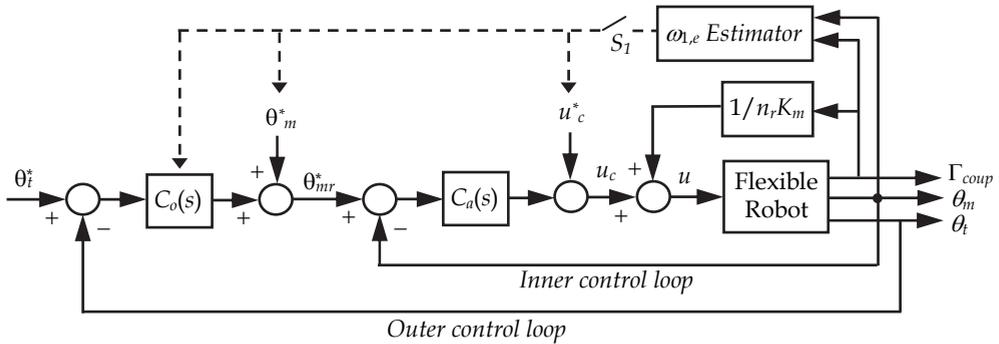


Fig. 10. Two-stage adaptive GPI control implemented in (Becedas, et al., 2009).

The system dynamics is obtained by the simplification to one vibration mode of the concentrated mass model (see Section 0). Adding the decoupling term defined in Section 5.2 to the voltage control signal u_c allows us to decouple both motor and link dynamics. Thus, the design of the controllers, one for each dynamics, is widely simplified. By using the flatness characteristic of the system, the two nested GPI controllers are designed as follows: Outer control law ($C_o(s)$):

$$(\theta_m - \theta_m^*) = [(\gamma_1 s + \gamma_0)/(s + \gamma_2)](\theta_t^* - \theta_t), \quad (17)$$

where θ_m^* is now an auxiliary ideal open loop control for the outer loop, θ_t^* represents the reference trajectory for the payload, and γ_i , $i=0, 1, 2$, are the outer loop controller gains, which are updated each time that the estimator estimates the real values of the system natural frequency.

Inner control law ($C_a(s)$):

$$(u_c - u_c^*) = [(\alpha_2 s^2 + \alpha_1 s + \alpha_0)/(s(s + \alpha_3))](\theta_{mr}^* - \theta_m), \quad (18)$$

where u^*_c represents the ideal open loop control for the inner loop, θ^*_{mr} represents the reference trajectory for the motor angle, and $\alpha_i, i=0, 1, 2, 3$ are the inner loop controller gains. The algebraic estimator for the natural frequency is given by the following equation:

$$\omega_{1e}^2 \begin{cases} \text{arbitrary}, t \in [0, \Delta) \\ n_e(t)/d_e(t), t \in [\Delta, +\infty) \end{cases} \tag{19}$$

where

$$\begin{aligned} n_e(t) &= t^2\theta_i(t) + z_1 & d_e(t) &= z \\ \dot{z}_1 &= z_2 - 4t\theta_i(t) & \dot{z}_3 &= z_4 \\ \dot{z}_2 &= 2\theta_i(t) & \dot{z}_4 &= t^2(\theta_m(t) - \theta_i(t)) \end{aligned} \tag{20}$$

Then, this control technique is implemented in the experimental platform of Figure 4a. The value of the tip angle is approximated by $\theta_t = \theta_m - L / (3EI) \Gamma_{coup}$, where θ_m and Γ_{coup} are obtained from the encoder and strain gauges measurements respectively. The desired reference to be tracked by the flexible robotic system is a two seconds Bezier eighth order trajectory with 1rad of amplitude. The control system starts working with an arbitrary computation of the tip mass, which is represented by a natural frequency $\omega_{0i}=9\text{rad/s}$, very different from the real value $\omega_{1e}=15.2\text{rad/s}$. In a small time interval $\Delta=0.5\text{s}$ (dashed line), the algebraic fast estimator estimates the real value ω_{1e} , and updates the inner (u^*_c) and outer ($\theta^*_{mr}, \alpha_2, \alpha_1$ and α_0) loop controllers (see details in (Becedas et al., 2009)). After the updating the control system perfectly tracks the desired trajectory (see Fig. 11).

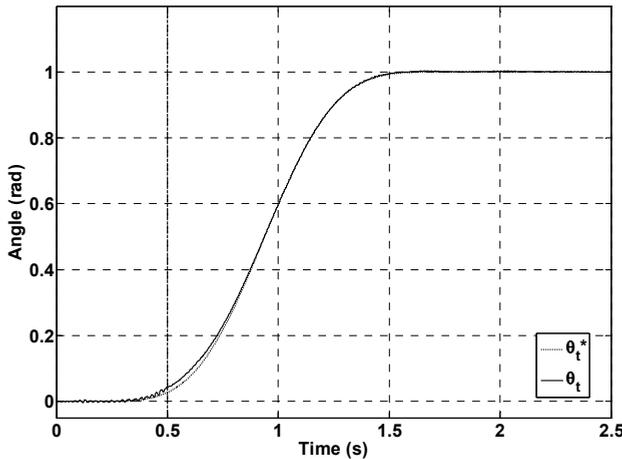


Fig. 11. Trajectory tracking of the reference trajectory with the GPI adaptive controller.

5.6 Force control

The special feature of the force control for flexible robots described here (Payo et al., 2009) is that the control law is designed to control the force exerted by a robot on the environment, which surrounds for both free and constrained motion tasks. The flexible robot of one

degree of freedom used is described in Section 0. The system dynamics of the arm is obtained by the simplification to one vibration mode of the concentrated mass model (see Section 0, specifically Equation (5)). The tracking of the desired force is obtained by using a feedback control loop of the torque at the root of the arm. This control law is based on a modified PID controller (I-PD controller (Ogata, 1998)), and it is demonstrated the effectiveness of the proposed controller for both free and constrained motion tasks. The sensor system used in this control law is constituted by a sole sensor very lightweight (two strain gauges placed at the root of the arm) to measure the torque, neither the contact force sensor nor the angular position sensor of the motor are used in the control method, unlike others methods described in Section 4.2. The controlled system presents robust stability conditions to changes in the tip mass, viscous friction and environment elasticity. It is also important to mention the good performance of the system response in spite of the nonlinear Coulomb friction term of the motor which was considered to be a perturbation. Fig. 12 shows the control scheme used to implement this force control technique, where the control law is given by the following equation:

$$u = a_0 \int_0^t (\Gamma_{coup}^d - \Gamma_{coup}) dt - a_1 \Gamma_{coup} - a_2 \frac{d\Gamma_{coup}}{dt}, \tag{21}$$

where a_0 , a_1 and a_2 are the design parameters of the I-PD and Γ_{coup}^d is the reference signal. The environment impedance is represented by the well known spring-dashpot model (Latornell et al., 1998) and (Erickson et al., 2003):

$$F_n = k_e x_e + b_e \dot{x}_e, \tag{22}$$

where k_e , b_e are the stiffness and damping characteristics of the environment and x_e is the local deformation of the environment. The plant dynamics for free and constrained motion tasks are given respectively by the following equations:

$$\frac{\Gamma_{coup}(s)}{U(s)} = \frac{sK_c / J_0 n_r}{s(s^2 + sv / J_0 + c / J_0 n_r^2) + \omega_0^2 (s + v / J_0)}, \tag{23}$$

$$\frac{\Gamma_{coup}(s)}{U(s)} = \frac{(K_c / J_0 n_r)(s^2 + sb_e / m + k_e / m)}{(s^2 + sv / J_0)(s^2 + sb_e / m + k_e / m + \omega_0^2) + c / J_0 n_r^2 (s^2 + sb_e / m + k_e / m)}. \tag{24}$$

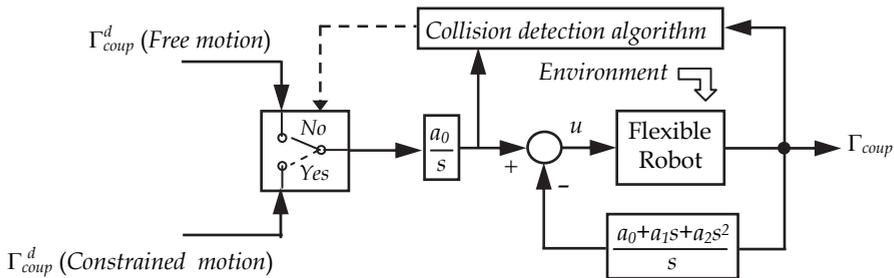


Fig. 12. Force control scheme.

The proposed strategy needs an online collision detection mechanism in order to switch between a command trajectory for free motion torque and a contact torque reference for the case of constrained motion. The collision was detected when the torque exceeded a threshold (γ) that depends on the amplitude of the reference signal, the Coulomb friction of the motor (Γ_c) and the noise in the measured signal (γ_3) according to the following equation (a detailed explication of this can be found in (Payo, et al., 2009)):

$$\gamma = \gamma_1 \Gamma_{coup} + \gamma_2 \Gamma_f + \gamma_3, \quad (25)$$

where γ_1 and γ_2 are normalized maximum deviations of the measured signal.

Fig. 13 and Fig. 14 show the results obtained in two experimental tests where the robot carried out both free and constrained motion tasks. The controlled torque is displayed before and after collision. A small value for the torque in free motion was used to prevent possible damages to the arm or to the object at the moment of collision. The chosen torque in these tests for free motion was equal to 0.07Nm. The constrained environment used in these tests was a rigid object with high impedance. Once the collision was detected, the Control law changed the reference value of the torque for constrained motion depending on the particular task carried out. For example, the first experiment matches a case in which the force exerted on the object was increased; and in the second experiment the force exerted on the object was decreased to avoid possible damages on the contact surfaces (case of fragile objects, for instance).

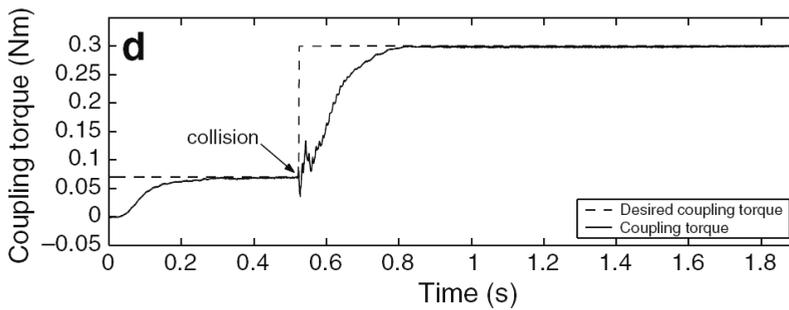


Fig. 13. System response for first experiment.

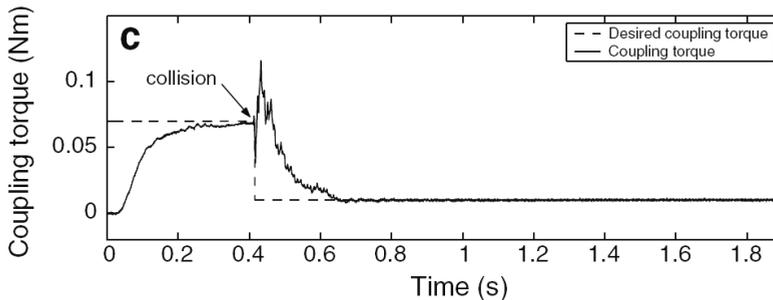


Fig. 14. System response for second experiment.

6. Future of flexible manipulators

After the huge amount of literature published on this topic during the last thirty years, flexible robotics is a deeply studied field of autonomous systems. Even complete books have been already devoted to the subject (Tokhi & Azad, 2008) and (Wang & Gao, 2003). Still, new control techniques can be studied due to simplicity of the physical platform, but, as discussed in (Benosman & Vey, 2004), most of the topics regarding modelling or controllability have been satisfactorily addressed in the previous literature.

However, some topics are still open and leave a considerable margin for improvement. Some manipulators with a small rigid arm attached to a large flexible base (called macro-micro manipulators, see (George & Book, 2003) for instance) have been developed for precision tasks, but the technological issue of building flexible robots with similar features to those of actual industrial robots has not been completely solved. While there exists a real prototype of a 3 dof flexible robot (Somolinos et al., 2002) achieving three dimensional positioning of the tip, a mechanical wrist still needs to be coupled for giving the manipulator the ability of reaching a particular position with a particular orientation.

On the control side, the search for the perfect controller is still open and, probably, never to be closed. All the robust, adaptive, intelligent techniques have their limitations and drawbacks. Many new controllers have been proposed but there is no standard measurement of the performance and, hence, no objective classification can be performed. The creation of a family of 'benchmark' problems would provide some objectivity to the results analysis.

One of the most potential aspects of flexible robots is their recently evolution in the position and force control. Such a combination provides of touch sensibility to the robotic system. Thus, the robot does not only have accuracy in the different positioning tasks, but also has the possibility of detecting whatever interaction with the environment that surrounds it. This characteristic allows the system to detect any collision with an object or surface, and to limit the actuating force in order not to damage the robotic arm nor the impact object or surface. Applications in this sense can be developed for robots involved in grasping, polishing, surface and shape recognition, and many other tasks (Becedas et al., 2008).

Nonlinear behaviour of flexible manipulators has been poorly accounted for in literature. A few works dealing with modelling of geometrical nonlinearities due to large displacements in the links have been published in (Payo et al., 2005) and (Lee, 2005) and a solution for achieving precise point-to-point motion of these systems has also been reported in (O'Connor et al., 2009). But these works are based on single link manipulators, and the multiple link case still has to be addressed. If we think of applications in which the robot is interacting with humans, these large displacements structures increase the safety of the subjects because the system is able to both absorb a great amount of energy in the impact and control effectively the contact force almost instantaneously (hybrid position/force controls). Thus, the development of human-machine interfaces becomes a potential application field for this kind of systems (Zinn, 2004).

Another interesting and not very studied approach to the flexibility of manipulators consists of taking advantage of it for specific purposes. Flexibility is considered as a potential benefit instead of a disadvantage, showing some examples with margin of improvement in assembling (Whitney, 1982), collision (García et al., 2003), sensors (Ueno et al., 1998) or mobile robots (Kitagawa et al., 2002).

7. References

- Aspinwall, D. M. (1980). Acceleration profiles for minimizing measurement machines. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102 (March of 1980), pp. 3-6.
- Åström, K. J. & Wittenmark, B. (1995). *Adaptive control*, Prentice Hall (2nd Edition), ISBN: 0201558661.
- Bai, M.; Zhou, D. & Fu, H. (1998). Adaptive augmented state feedback control. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6 pp. 940-950.
- Balas, M. J. (1978). Active control of flexible systems. *Journal of Optimisation Theory and Applications*, Vol. 25, No. 3, pp. 415-436.
- Balas, M. J. (1982). Trends in large space structures control theory: Fondest hopes, wildest dreams. *IEEE Transactions on Automatic Control*, Vol. 27, No. 3, pp. 522-535.
- Banavar, R. N. & Dominic, P. (1995). An LQG/ H^∞ Controller for a Flexible Manipulator. *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 4, pp. 409-416.
- Bayo, E. (1987). A finite-element approach to control the end-point motion of a single-link flexible robot. *Journal of Robotics Systems*, Vol. 4, No. 1, pp. 63-75.
- Becedas, J.; Payo, I.; Feliu, V. & Sira-Ramírez, H. (2008). Generalized Proportional Integral Control for a Robot with Flexible Finger Gripper, *Proceedings of the 17th IFAC World Congress*, pp. 6769-6775, Seoul (Korea).
- Becedas, J.; Trapero, J. R.; Feliu, V. & Sira-Ramírez, H. (2009). Adaptive controller for single-link flexible manipulators based on algebraic identification and generalized proportional integral control. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 39, No. 3, pp. 735-751.
- Belleza, F.; Lanari, L. & Ulivi, G. (1990). Exact modeling of the flexible slewing link, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 734-804.
- Benosman, M. & Vey, G. (2004). Control of flexible manipulators: A survey. *Robotica*, Vol. 22, pp. 533-545.
- Bicchi, A & Kumar, V. (2000). Robotic grasping and contact: a review, *Proceedings of the IEEE International Conference on Robotics and Automation*, No. 1, pp. 348-353.
- Bodson, M. (1998). An adaptive algorithm for the tuning of two input shaping methods. *Automatica*, Vol. 34, No. 6, pp. 771-776.
- Book, W. J. (1974). *Modeling, design and control of flexible manipulator arms*. Ph. D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge MA.
- Book, W. J.; Maizza-Neto, O. & Whitney, D.E. (1975). Feedback control of two beam, two joint systems with distributed flexibility. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 97G, No. 4, pp. 424-431.
- Book, W. J. & Majette, M. (1983). Controller design for flexible, distributed parameter mechanical arms via combined state space and frequency domain techniques. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 105, No. 4, pp. 245-254.
- Book, W. J. (1984). Recursive lagrangian dynamics of flexible manipulator arms. *International Journal of Robotics Research*, Vol. 3, No. 3, pp. 87-101.
- Book, W. J. (1993). Controlled motion in an elastic world. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 115, No. 2, pp. 252-261.

- Bridges, M. M.; Dawson, D. M. & Abdallah, C. T. (1995). Control of rigid-link flexible joint robots: a survey of backstepping approaches. *Journal of Robotic Systems*, Vol. 12, No. 3, pp. 199-216.
- Cannon, R. H. & Schmitz, E. (1984). Initial experiments on the end-point control of a flexible robot. *International Journal on Robotics Research*, Vol. 3, No. 3, pp. 62-75.
- Caswar, F. M. & Unbehauen, H. (2002). A neurofuzzy approach to the control of a flexible manipulator. *IEEE Transaction on Robotics and Automation*, Vol. 18, No. 6, pp. 932-944.
- Chapnik, B. V.; Heppler, G.R. & Aplevich, J.D. (1993). Controlling the impact response of a one-link flexible robotic arm. *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 3, pp. 346-351.
- Chen, Y. P. & Hsu, H. T. (2001). Regulation and vibration control of an FEM-based single-link flexible arm using sliding mode theory. *Journal Vibration and Control*, Vol. 7, No. 5, pp. 741-752.
- Chiou, B. C. & Shahinpoor, M. (1988). Dynamics stability analysis of a one-link force-controlled flexible manipulator. *Journal of Robotic Systems*, Vol. 5, No. 5, pp. 443-451.
- Choi S. B.; Cheong C. C. & Shin H. C. (1995). Sliding mode control of vibration in a single-link flexible arm with parameter variations, *Journal Vibration and Control*, Vol. 179, No. 5, pp. 737-748.
- Christoforou E. G. & Damaren C. J. (2000). The control of flexible-link robots manipulating large payloads: Theory and Experiments. *Journal of Robotic Systems*, Vol. 17, No. 5, pp. 255-271.
- Damaren C. J. (1996). Adaptive control of flexible manipulators carrying large uncertain payloads. *Journal of Robotic Systems*, Vol. 13, No. 4, pp. 219-228.
- De Luca, A. & Siciliano, B. (1989). Trajectory control of a non-linear one-link flexible arm. *International Journal of Control*, Vol. 50, No. 5, pp. 1699-1715.
- De Luca, A. & Siciliano, B. (1991). Closed form dynamic model of planar multilink lightweight robots. *IEEE Transactions Systems, Man and Cybernetics*, Vol. 21, No. 4, pp. 826-839.
- De Luca, A. & Siciliano, B. (1993). Regulation of flexible arms under gravity. *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, pp. 463-467.
- Denavit, J. & Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, (June), pp. 215-221.
- Dwivedy, S. K. & Eberhard, P (2006). Dynamic analysis of flexible manipulators, a literature review. *Mechanism and Machine Theory*, Vol. 41, No. 7, pp. 749-777.
- Erickson, D.; Weber, M. & Sharf, I. (2003). Contact stiffness and damping estimation for robotic systems. *International Journal of Robotic Research*, Vol. 22, No. 1, pp. 41-57.
- Feliu, V.; Rattan K. S. & Brown, H. B. (1990). Adaptive control of a single-link flexible manipulator. *IEEE Control Systems Magazine*, Vol. 10, No. 2, pp. 29-33.
- Feliu, V.; Rattan, K. & Brown, H. (1992). Modeling and control of single-link flexible arms with lumped masses. *Journal of Dynamic Systems, Measurement and Control*, Vol. 114, No. 7, pp. 59-69.
- Feliu V.; Rattan, K. & Brown, H. (1993). Control of flexible arms with friction in the joint. *IEEE Transactions on Robotics and Automation*. Vol. 9, No. 4, pp. 467-475.

- Feliu V. & Rattan K. S. (1999). Feedforward control of single-link flexible manipulators by discrete model inversion. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 121, pp. 713-721.
- Feliu, J. J.; Feliu, V. & Cerrada C. (1999). Load adaptive control of single-link flexible arms base don a new modeling technique. *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 5, pp. 793-804.
- Feliu, V. (2006). Robots flexibles: Hacia una generación de robots con nuevas prestaciones. *Revista Iberoamericana de Automática e Informática Industrial*, Vol. 3, No. 3, pp. 24-41.
- Feliu, V.; Pereira, E.; Díaz, I. M. & Roncero, P. (2006). Feedforward control of multimode single-link flexible manipulators based on an optimal mechanical design. *Robotics and Autonomous Systems*, Vol. 54, No. 8, pp. 651-666.
- Fliess, M.; Lévine, J.; Martin, P. & Rouchon, P. (1999). A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on Automatic Control*, Vol. 44, No. 5, pp. 922-937.
- Fliess, M. & Sira-Ramírez, H. (2003). An algebraic framework for linear identification. *ESAIM - Control, Optimisation and Calculus of Variations*, No. 9, pp. 151-168.
- Fu, K. S. (1970). Learning control systems-Review and Outlook. *IEEE Transactions on Automatic Control*, Vol. AC-15, pp. 210-221.
- García, A.; Feliu, V. & Somolinos, J. A. (2003). Experimental testing of a gauge based collision detection mechanism for a new three-degree-of-freedom flexible robot. *Journal of Robotic Systems*, Vol. 20, No. 6, pp. 271-284.
- Gervarter, W. (1970). Basic relations for controls of flexible vehicles. *AIAA Journal*, Vol. 8, No. 4, pp. 666-672.
- Ge, S. S.; Lee, T. H. & Harris, C. J. (1999). *Adaptive neural network control of robotic manipulators*. World Scientific, 981023452X.
- George, L. E. & Book, W. J. (2003). Inertial vibration damping control of a flexible base manipulator. *IEEE/ASME Transactions on Mechatronics*, Vol. 8, No. 2, pp. 268-271.
- Gilardi, G. & Sharf, I. (2002). Literature survey of contact dynamics modelling. *Mechanism and Machine Theory*, 2002, Vol. 37, No. 10, pp. 1213-1239.
- Green, A. & Sasiadek, J. Z. (2002). Inverse dynamics and fuzzy repetitive learning flexible robot control, *15th Triennial World Congress, IFAC*, Barcelona (Spain).
- Gutman, S. (1999). Uncertain Dynamical Systems-A Lyapunov Min-Max Approach. *IEEE Transactions on Automatic Control*, Vol. 24, No. 3, pp. 437-443.
- Huey, J. R.; Sorensen, K. L. & Singhose W. E. (2008). Useful applications of closed-loop signal shaping controllers. *Control Engineering Practice*, Vol. 16, No. 7, pp. 836-846.
- Jiang, Z. H. (2004). End-Effector Robust Trajectory Tracking Control for Flexible Robot Manipulators. *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 4394-4399.
- Jordan, S. (2002). Eliminating vibration in the nano-world. *Photonics Spectra*, Vol. 36, pp. 60-62.
- Kitagawa, H.; Beppu, T.; Kobayashi, T. & Terashima, K. (2002). Motion control of omnidirectional wheelchair considering patient comfort, *Proceedings of the 2002 IFAC World Congress*, Barcelona, (Spain).

- Koivo, A. J. & Lee K. S. (1989). Self-tuning control of planar two-link manipulator with non-rigid arm, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1030-1035.
- Korolov, V. V. & Chen, Y.H. (1989). Controller design robust to frequency variation in a one-link flexible robot arm. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 111, No. 1, pp. 9-14.
- Landau, I. D.; Daniel Rey, J. L. & Barnier J. (1996). Robust Control of a 360° Flexible Arm Using the Combined Pole Placemen/Sensitivity Function Shaping Method. *IEEE Transactions on Systems Technology*, Vol. 4, No. 4, pp. 369-383.
- Latornell, D. J.; Cherchas D. B. & Wong R. (1998). Dynamic characteristics of constrained manipulators for contact force control design. *International Journal of Robotics Research*, Vol. 17, No. 3, pp. 211-231.
- Lee, H. H. (2005). New Dynamic Modeling of Flexible-Link Robots. *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, Vol. 127, No. 2, pp. 307-309.
- Liu, L. Y. & Yuan, K. (2003). Noncollocated passivity-based PD control of a single-link flexible manipulator. *Robotica*, Vol. 21, No. 2, pp. 117-135.
- Lizarraga I. & Etxebarria V. (2003). Combined PD- H_∞ approach to control of flexible manipulators using only directly measurable variables. *Cybernetics and Systems*, Vol. 34, No. 1, pp. 19-32.
- Luo, Z. H. (1993). Direct strain feedback control of flexible robot arms: new theoretical and experimental results. *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, pp. 1610-1622.
- Maizza-Neto, O. (1974). *Modal analysis and control of flexible manipulator arms*. Ph. D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge MA.
- Matsuno, F. & Kasai, S. (1998). Modeling and robust force control of constrained one-link flexible arms. *Journal of Robotic Systems*, Vol. 15, No. 8, pp. 447-464.
- Meirovitch, L. (1996). *Principles and techniques of vibration*. Prentice Hall, 0023801417 Englewood Cliffs, New Jersey.
- Moallem, M.; Khorasani, K. & Patel, R.V. (1998). Inversion-based sliding control of a flexible-link manipulator. *International Journal of Control*, Vol. 71, No. 3, pp. 477-490.
- Mohamed, Z.; Martins, J. M.; Tokhi, M. O.; Sà da Costa, J. & Botto, M.A. (2005). Vibration control of a very flexible manipulator system. *Control Engineering Practice*, Vol. 13, No. 3, pp. 267-277.
- Morita, Y.; Ukai, H. & Kando, H. (1997). Robust Trajectory Tracking Control of Elastic Robot Manipulators. *Transactions on ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 119, No. 4 pp. 727-735.
- Morita, Y.; Kobayashi, Y.; Kando, H.; Matsuno, F.; Kanzawa, T. & Ukai, H. (2001). Robust force control of a flexible arm with a nonsymmetric rigid tip body. *Journal of Robotic Systems*, Vol. 18, No. 5 pp. 221-235.
- Moser A. N. (1993). Designing controllers for flexible structures with H_∞/μ -synthesis. *IEEE Control Systems Magazine*, Vol. 13, No. 2, pp. 79-89.
- Moudgal, V. G.; Kwong, W. A. & Passino, K. M. (1995). Fuzzy learning control for a flexible-link robot, *IEEE Transactions of fuzzy systems*, Vol. 3, No. 2, pp. 199-210.

- O'Connor, W. J. (2007). Wave-based analysis and control of lump-modeled flexible robots. *IEEE Transactions on Robotics*, Vol. 23, No. 2, pp. 342-352.
- O'Connor, W. J.; Ramos, F.; McKeown, D. & Feliu, V. (2007). Wave-based control of nonlinear flexible mechanical systems. *Nonlinear Dynamics*, Vol. 57, No. 1-2, pp. 113-123.
- Ogata, K. (2001). *Modern control engineering*. Prentice Hall, 0130609072.
- Onsay, T. & Akay, A. (1991). Vibration reduction of a flexible arm by time-optimal open-loop control. *Journal of Sound and Vibration*, Vol. 147, No. 2, pp. 283-300.
- Ozgoli, S. & Taghirad, H. D. (2006). A survey of the control of flexible joint robots. *Asian Journal of Control*, Vol. 8, No. 4, pp. 332-334.
- Palejiya, D. & Tanner, H. (2006). Hybrid velocity/force control for robot navigation in compliant unknown environments. *Robotica*. Vol. 24, No. 6, pp.745-758.
- Park, J. & Chang, P. H. (2001). Learning input shaping technique for non-LTI systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 123, No. 2, pp. 288-293.
- Park, J.; Chang, P. H.; Park, H.S. & Lee, E. (2006). Design of learning input shaping technique for residual vibration suppression in an industrial robot. *IEEE/ASME Transactions on Mechatronics*, Vol. 11, No. 1, pp. 55-65.
- Payo, I.; Ramos, F.; Feliu, V. & Cortázar, O. D. (2005). Experimental validation of nonlinear dynamic models for single-link very flexible arms, *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, December 2005, Seville, (Spain).
- Payo, I.; Feliu, V. & Cortazar, O. D. (2009). Force control of a very lightweight single-link flexible arm based on coupling torque feedback. *Mechatronics*, Vol. 19, No. 3, pp. 334-347.
- Pedersen, N. L. & Pedersen, M. L. (1998). A direct derivation of the equations of motion for 3-D flexible mechanical systems. *International Journal of Numerical Methods in Engineering*, Vol. 41, No. 4, pp. 697-719.
- Pereira, E.; Díaz, I. M.; Cela, J. J. & Feliu V. (2007). A new methodology for passivity based control of single-link flexible manipulator, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 978-1-4244-1263-1, Zurich (Switzerland).
- Ramos, R. & Feliu, V. (2008). New online payload identification for flexible robots. Application to adaptive control. *Journal of Sound and Vibration*, Vol. 315, No. 1-2, pp. 34-57.
- Renno, J. M. (2007). Inverse dynamics based tuning of a fuzzy logic controller for a single-link flexible manipulator. *Journal of Vibration and Control*, Vol. 13, No. 12, pp. 1741-1759.
- Rovner, D. M. & Cannon, R. H. (1987). Experiments toward on-line identification and control of a very flexible one-link manipulator. *The International Journal of Robotics Research*, Vol. 6, No. 4, pp. 3-19.
- Sanz, A. & Etxebarria V. (2005). Composite Robust Control of a Laboratory Flexible Manipulator. *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 3614-3619.
- Schwertassek, R.; Wallrapp O. & Shabana A. (1999). Flexible multibody simulation and choice of the shape functions. *Nonlinear dynamics*, Vol. 20, No. 4, pp. 361-380.

- Shimoga, K. B. (1996). Robot grasp synthesis algorithms: a survey. *International Journal of Robotic Research*, Vol. 15, No. 3, pp. 230-266.
- Shi, L. & Trabia, M. (2005). Comparison of distributed PD-like and importance based fuzzy logic controllers for a two-link rigid-flexible manipulator. *Journal of Vibration and Control*, Vol. 11, No. 6, pp. 723-747.
- Siciliano B.; Yuan B. S. & Book W. J. (1986). Model reference adaptive control of a one link flexible arm. *Proceedings of 25th Conference on Decision and Control*, 3, pp. 91-95, Athens, (Greece).
- Siciliano, B. & Villani, L. (1999). *Robot force control*, Springer, ISBN: 0792377338.
- Siciliano, B. & Villani, L. (2000). Parallel force and position control of flexible manipulators. *IEE Proceedings: Control Theory Applications*, Vol. 147, No. 6, pp. 605-612.
- Singer, N. C. & Seering, W. C. (1990). Preshaping command inputs to reduce system vibration. *ASME Journal of Dynamic System, Measurement and Control*, Vol. 112, No. 1, pp. 76-82.
- Singh, S. & Schy, A. (1985). Robust torque control of an elastic robotic arm based on invertibility and feedback stabilization. *Proceedings of the 24th IEEE Conference on Decision and Control*, pp. 1317-1322.
- Spong, M. W. (1987). Modeling and control of elastic joints robots, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 109, No. 6, pp. 310-319.
- Smith, O. J. M. (1958). *Feedback Control Systems*. McGraw-Hill Book Co., Inc. New York.
- Somolinos, J. A.; Feliu, V. & Sánchez, L. (2002). Design, dynamic modelling and experimental validation of a new three-degree-of-freedom flexible arm. *Mechatronics*, Vol. 12, No. 7, pp. 919-948.
- Subudhi, B. & Morris, A. S. (2009). Soft computing methods applied to the control of a flexible robot manipulator. *Applied soft computing journal*, Vol. 9, No. 1, pp. 149-158.
- Su, Z. & Khorasani, K. (2001). A neural-network-based controller for a single-link flexible manipulator using the inverse dynamics approach. *IEEE Transactions on Industrial Electronic*, Vol. 48, No. 6, pp. 1074-1086.
- Talebi, H. A.; Khorsani, K. & Patel, R.V. (2009). *Control of flexible-link manipulators using neural networks*, Springer, 978-1-85233-409-3.
- Theodore, R. J. & Ghosal A. (2003). Robust control of multilink flexible manipulators. *Mechanism and Machine Theory*, Vol. 38, No. 4, pp. 367-377.
- Thomas S. & Mija S. J. (2008). A practically implementable discrete time sliding mode controller for flexible manipulator. *International Journal of Robotics and Automation*, Vol. 23, No. 4, pp. 235-241.
- Tian, L.; Wang, J. & Mao, Z. (2004). Constrained motion control of flexible robot manipulators based on recurrent neural networks, *IEEE Transactions on Systems, Man, and Cybernetic, part B*., Vol. 34, No. 3, pp. 1541-1551.
- Tokhi, M. O. & Azad, A. K. M. (2008). *Flexible robot manipulators: modelling, simulation and control*, Springer-Verlag, 0863414486, London.
- Tuttle, T. & Seering, W. (1997). Experimental verification of vibration reduction in flexible spacecraft using input shaping. *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, pp. 658-664.

- Tsytkin, Y. (1968). Self-learning: What is it? *IEEE Transactions on Automatic Control*, Vol. AC-13, pp. 608-612.
- Ueno, N.; Svinin, M. M. & Kaneko M. (1998). Dynamic contact sensing by flexible beam. *IEEE/ASME Transactions on Mechatronics*, Vol. 3, No. 4, pp. 254-264.
- Vaughan, J.; Yano A. & Singhose W. (2008). Comparison of robust input shapers. *Journal of Sound and Vibration*, Vol. 315, No. 4-5, pp. 797-815.
- Vossoughi, G. R. & Karimzadeh, A. (2006). Impedance control of a two degree-of-freedom planar flexible link manipulator using singular perturbation theory. *Robotica*, Vol. 24, No. 2, pp. 221-228.
- Wang, D. (1989). *Modelling and control of multi-link manipulators with one flexible link*. Ph. D. Thesis, Department of Electrical Engineering, University of Waterloo, Canada.
- Wang, W.; Lu, S. & Hsu, C. (1989). Experiments on the position control of a one-link flexible robot arm. *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, pp. 131-138.
- Wang, D. & Vidyasagar, M. (1991). Transfer functions for a single flexible link. *International Journal of Robotics Research*, Vol. 10, No. 5, pp. 540-549.
- Wang, D. & Vidyasagar M. (1992). Passive control of a stiff flexible link, *The International Journal of Robotics Research*, Vol. 11, No. 6, pp. 572-578.
- Wang, Z.; Zeng, H.; Ho, D. W. C. & Unbehauen H. (2002). Multi-objective Control of a Four-Link Flexible Manipulator: A Robust H_∞ Approach. *IEEE Transactions on Systems Technology*, Vol. 10, No. 6, pp. 866-875.
- Wang, F. & Gao Y. (2003). *Advanced studies of flexible robotic manipulators, modeling, design, control and applications*, World Scientific, ISBN: 978-981-279-672-1, New Jersey.
- Whitney, D. E. (1982). Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 104, No. 1, pp. 65-77.
- Whitney, D. E. (1987). Historical perspective and state of the art in robot force control. *International Journal of Robotic Research*, Vol. 6, No. 1, pp. 3-14.
- Yamano, M.; Kim, J.; Konno, A. & Uchiyama, M. (2004). Cooperative control of a 3D dual-flexible arm robot. *Journal of Intelligent and Robotic Systems*, Vol. 39, No. 1, pp. 1-15.
- Yuan, K. & Lin, L. (1990). Motor-based control of manipulators with flexible joints and links, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1809-1814, 0-8186-9061-5, Cincinnati, OH, USA .
- Yang, G. B. & Donath, M. (1988). Dynamic model of a one link robot manipulator with both structural and joint flexibility, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 476-481.
- Yim, J. G.; Yeon, J. S.; Lee, J.; Park, J. H.; Lee, S. H. & Hur, J. S. (2006). Robust Control of Flexible Robot Manipulators. *SICE-ICASE International Joint Conference*, pp. 3963-3968.
- Yoshikawa, T.; Harada, K. & Matsumoto, A. (1996). Hybrid position/force control of flexible-macro/ rigid-micro manipulator systems. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, pp. 633-640.
- Zeng, G. & Hemami, A. (1997). An overview of robot force control. *Robotica*, Vol. 15, No. 5, pp. 473-482.

Zinn, M.; Khatib, O.; Roth, B. & Salisbury, J. K. (2004). Playing it safe. *IEEE Robotics and Automation Magazine*, Vol. 11, No. 2, pp. 12-21.

Fuzzy logic positioning system of electro-pneumatic servo-drive

Jakub E. Takosoglu¹, Ryszard F. Dindorf^{1,2}
and Pawel A. Laski¹

¹Kielce University of Technology

*²AGH University of Science and Technology, Krakow
Poland*

1. Introduction

Development of automation and robotization in manufacturing process stimulates interest in pneumatic servo-systems whose advantages include low manufacturing costs, high dynamics and reliability (Situm et al., 2004). Unsatisfactory positioning accuracy of multiaxis pneumatic servosystems considerably reduces their application in manipulating machines, manipulators and robots. Rapid advance in parallel pneumatic manipulators imposes a lot of demands on controllers of pneumatic servo-drive concerning positioning accuracy, resistance to alternating parameters of state and disturbing signals (Dindorf et al., 2005; Takosoglu & Dindorf, 2005; Schulte & Hahn, 2004). The problem of positioning accuracy of servo-pneumatic systems is difficult to solve when no sufficient information on the process of conversion of the compressed gas energy into mechanical energy of pneumatic cylinder is available (Zhu, 2006; Takosoglu, 2005). Because of that, new control methods based on artificial intelligence, for example, fuzzy logic are introduced (Schulte & Hahn, 2004; Renn & Liao, 2004; Dindorf & Takosoglu, 2005). In traditional control systems of pneumatic servo-drives control algorithms are designed intuitively on the basis of operator's experience. In fuzzy control the knowledge coded in knowledge base is the result of experience, intuition as well as theoretical and practical understanding of control system dynamics which in this case is the dynamics of pneumatic servosystems. Thanks to fuzzy logic the operator's knowledge can be represented by means of mathematical operations. Fuzzy control enables moving from qualitative to quantitative control of pneumatic servo-drive. Application of fuzzy controller makes control of multiaxial pneumatic servosystems possible in manipulators and robots of various kinematic structures: series, parallel or hybrid series/parallel. Advancements in software for rapid prototyping in real time and in hardware-in-the-loop simulations enable to construct and test positioning fuzzy control (Bucher & Balemi, 2006) of pneumatic servo-drives in laboratory conditions. Such an approach minimizes the design costs of control systems of pneumatic servo-drives. Pneumatic servo-drives with teaching/playback control system have considerable practical significance, especially in the control of manipulating machines, manipulators, industrial robots as well as rehabilitation and physiotherapy manipulators.

Pneumatic servo-cylinders used in multi-axis electro-pneumatic systems and referred to as pneumatic axes perform operations and function as supporting structure. Cartesian manipulators with pneumatic axes connected in series are classified as open-loop chain kinematic mechanisms. In serial kinematic chain elastic strains accumulate on particular pneumatic axes, which lowers the positioning accuracy of pneumatic manipulators. Parallel mechanism is a closed-loop mechanism in which the moving platform is connected to the fixed base by independent kinematic chains. Kinematic structure in the form of a closed-loop chain finds application in parallel kinematic robot (PKR) and parallel kinematic machine (PKM). Manipulators based on parallel kinematics structure can achieve better accuracy of repeatability and they can apply larger forces than conventional serial manipulators because of the higher stiffness of their mechanical structure. By using parallel kinematics in machine tools high stiffness and high machine dynamics is achieved. With Stewart-Gough Platform as a base numerous kinematic structures of parallel manipulators (Nonapod, Hexapod, Tripod) and hybrid manipulators (Tricept, Dyna-M, LinaPod) were formed. The names of kinematic structures of parallel manipulators are related to the kind of kinematic joint and the number of degrees of freedom (DoF). To calculate degrees of freedom of parallel manipulators the formula proposed by Tsai is used. The family of parallel manipulators includes translational parallel manipulators (TPM) based on three degrees of freedom (3-DoF) and containing at least one prismatic joint. In the group of 3-DoF TPM manipulators the most common are – spatial parallel mechanism of the structure: 3-PUU, 3-UPU, 3-UPS, 3-CPU, 3-PUS, 3-PCRR and planar parallel mechanism of the structure: 3-RPR, 3-PRR, 3-PPR, 3-RRR (Merlet, 2006).

2. Simulation model of electro-pneumatic servo-system

The object of research is control system of pneumatic servo-drive which is presented in Fig. 1.

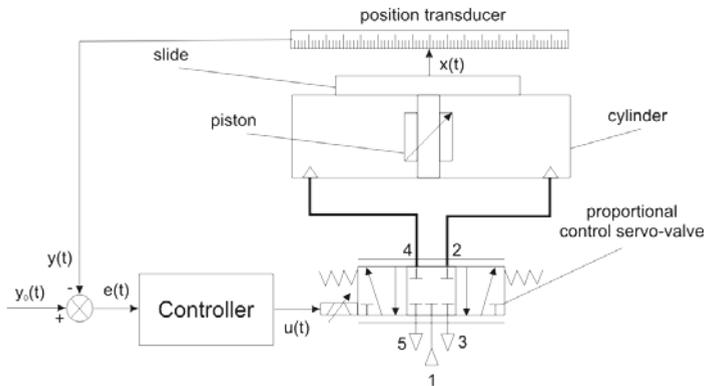


Fig. 1. The diagram of electro-pneumatic servo-drive control system

The controller of pneumatic servo-system processes the position error $e(t) = y_o(t) - y(t)$ into voltage signal $u(t)$ in the solenoid of proportional directional control valve controlling the slide of the rodless cylinder, where $y_o(t)$ is the input signal, and $y(t)$ the signal generated by displacement transducer of the rodless cylinder's linear slide of $x(t)$ coordinate. The functional model of the analyzed pneumatic servo-drive with rodless cylinder controlled by

proportional directional control valve is presented in Fig. 2a and Fig. 2b and its nonlinear dynamic model is written as the following set of differential equations (Takosoglu & Dindorf, 2005):

- equation of motion for the piston – slide of rodless cylinder:

$$\frac{d^2x}{dt^2} = \frac{1}{M+m} \left\{ A \Delta p - \left[f_l v + F_k \operatorname{sgn}(v) + F_{pr} e^{\left(\frac{v}{v_k}\right)} \operatorname{sgn}(v) + k_p |\Delta p| \right] + m g \sin \alpha \right\}, \quad (1)$$

where: x, v – displacement and velocity of piston – slide,

A – area of piston,

Δp – pressure difference between cylinder chambers, $\Delta p = p_1 - p_2$,

p_1, p_2 – absolute pressures in cylinder’s chambers,

M – mass of piston and slide,

f_l – viscous friction coefficient,

F_k – kinetic friction force (Coulomb friction force),

v_k – critical value of velocity, characteristic velocity of the *Stribeck* friction,

F_{pr} – break away force,

k_p – friction coefficient dependent upon seal dimensions,

m – initial load mass,

g – acceleration of gravity,

a – servo-motor inflection angle.

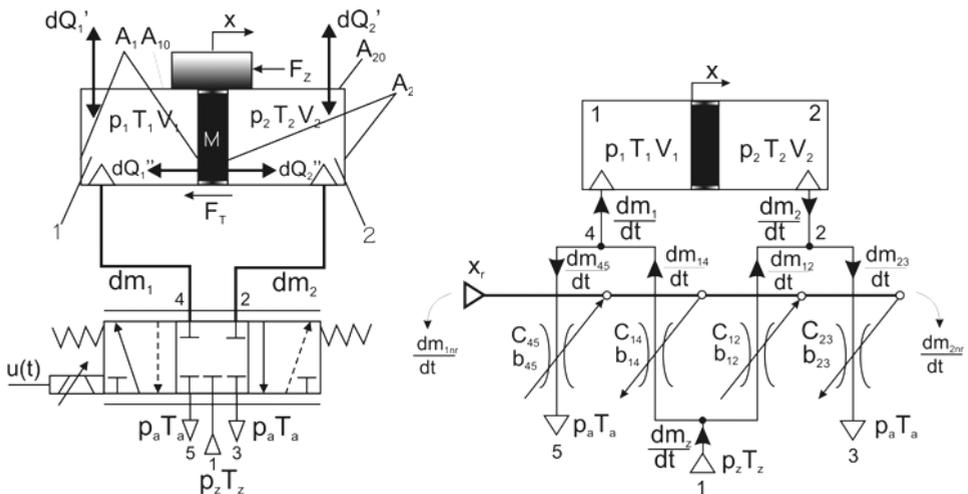


Fig. 2. The diagrams: a) pneumatic rodless cylinder, b) servo-valve system

- equation of motion for the spool of proportional directional control valve:

$$\frac{d^2x_r}{dt^2} = \frac{1}{m_r} \left\{ \frac{k_m}{R_c} (u - n B l_c v_r) \left(1 - e^{-\frac{R_c t}{L}} \right) - f_l v_r - k_s x_r \right\}, \quad (2)$$

where: x_r, v_r – displacement and velocity of valve spool,
 m_r – mass of spool,
 k_m – coefficient of electromechanical force transducer,
 R_c – resistance of the solenoid,
 u – coil voltage,
 n – number of coils,
 B – magnetic induction,
 l_c – length of coil,
 L – inductance of the solenoid,
 f_i – coefficient of viscous friction,
 k_s – spool spring rate.

– equations for pressure in cylinder chambers:

$$\begin{aligned} \frac{dp_1}{dt} &= \frac{\kappa}{A(l_0 + x)} \left\{ R \frac{dm_1}{dt} - p_1 A v - \frac{\kappa - 1}{\kappa} \alpha [A_{10}(T_1 - T_a) + A(T_1 - T_2)] \right\} , \\ \frac{dp_2}{dt} &= \frac{\kappa}{A(l + l_0 - x)} \left\{ -R \frac{dm_2}{dt} + p_2 A v - \frac{\kappa - 1}{\kappa} \alpha [A_{20}(T_2 - T_a) + A(T_2 - T_1)] \right\} \end{aligned} \quad (3)$$

where: κ – adiabatic exponent,
 l_0 – length of dead zone of the pneumatic cylinder,
 l – stroke length of pneumatic cylinder,
 R – specific gas constant,
 α – overall heat-transfer coefficient,
 T_1, T_2 – temperature in cylinder chambers,
 T_a – ambient temperature,
 A_{10}, A_{20} – heat transfer surface,
 $\frac{dm_1}{dt}, \frac{dm_2}{dt}$ – mass flow rate,

– equations for mass flow rate through proportional control valve:

$$\begin{aligned} \frac{dm_1}{dt} &= C_{14} \rho_0 \sqrt{\frac{T_0}{T_z}} p_z w_{14} - C_{45} \rho_0 \sqrt{\frac{T_0}{T_1}} p_1 w_{45} , \\ \frac{dm_2}{dt} &= C_{23} \rho_0 \sqrt{\frac{T_0}{T_2}} p_2 w_{23} - C_{12} \rho_0 \sqrt{\frac{T_0}{T_z}} p_z w_{12} \end{aligned} \quad (4)$$

where: ρ_0 – air density in the normal reference atmosphere (ANR),
 T_0 – normal ambient temperature,
 p_z – air supply pressure,
 T_z – air supply temperature,
 $C_{14}, C_{45}, C_{12}, C_{23}$ – sonic conductance consistent with the standard ISO 6358-1989 for critical pressure ratio,
 $w_{14}, w_{45}, w_{12}, w_{23}$ – nonlinear flow function (sonic flow and subsonic flow) depending on the pressure ratio and on the critical pressure ratio,

$$w_{14} = \begin{cases} 1 \\ \sqrt{1 - \left(\frac{p_1 - b_{14}}{p_z}\right)^2} \end{cases} \quad \text{and} \quad \begin{cases} 0 \leq \frac{p_1}{p_z} \leq b_{12} \\ b_{12} \leq \frac{p_1}{p_z} \leq 1 \end{cases}, \quad (5)$$

$$w_{45} = \begin{cases} 1 \\ \sqrt{1 - \left(\frac{p_a - b_{45}}{p_2}\right)^2} \end{cases} \quad \text{and} \quad \begin{cases} 0 \leq \frac{p_a}{p_2} \leq b_{45} \\ b_{45} \leq \frac{p_a}{p_2} \leq 1 \end{cases}, \quad (6)$$

$$w_{12} = \begin{cases} 1 \\ \sqrt{1 - \left(\frac{p_1 - b_{12}}{p_z}\right)^2} \end{cases} \quad \text{and} \quad \begin{cases} 0 \leq \frac{p_1}{p_z} \leq b_{12} \\ b_{12} \leq \frac{p_1}{p_z} \leq 1 \end{cases}, \quad (7)$$

$$w_{23} = \begin{cases} 1 \\ \sqrt{1 - \left(\frac{p_a - b_{23}}{p_2}\right)^2} \end{cases} \quad \text{and} \quad \begin{cases} 0 \leq \frac{p_a}{p_2} \leq b_{23} \\ b_{23} \leq \frac{p_a}{p_2} \leq 1 \end{cases}, \quad (8)$$

where: $b_{14}, b_{45}, b_{12}, b_{23}$ – critical pressure ratio.

The simulation model of pneumatic servo-drive represented by equations (1) – (8), was used for selection and initial verification of fuzzy controller. The model of pneumatic servo-drive was implemented for Matlab-Simulink package with Fuzzy Logic Toolbox. The block diagram of the model of pneumatic servo-drive with fuzzy controller is presented in Fig. 3.

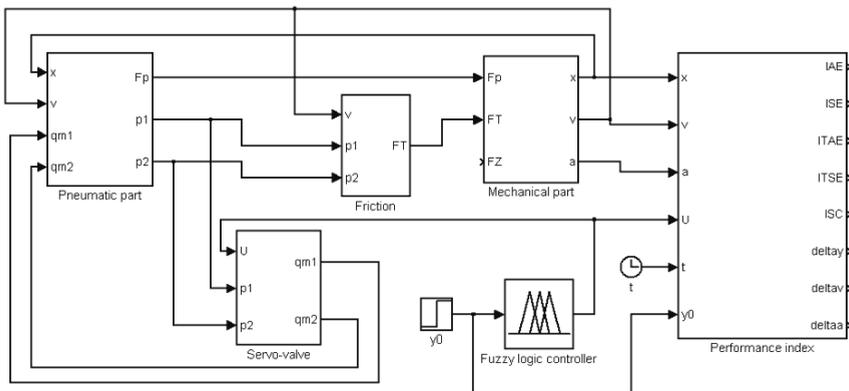


Fig. 3. The simulink block diagram of simulation model of electro-pneumatic servo-drive with fuzzy controller

Fuzzy PD controller constructed in Fuzzy Logic Tollbox of Matlab Simulink package was suggested for the purpose of controlling pneumatic servo-drive. The pneumatic servo-drive together with fuzzy PD controller constitute a system of MISO type with two inputs: position error $e(t)$ and change of position error $\Delta e(t)$ and one output: proportional valve coil voltage $u(t)$. Output and input signals underwent fuzzification process with regular distribution of 7 fuzzy sets of triangular and trapezoid membership functions) (Takosoglu & Dindorf, 2005). The knowledgebase rules of fuzzy controller are 49 Mac Vicar-Whelen rules described in the table entered to Fuzzy Logic Toolbox. In the inference process the firing degree was determined by means of MIN operator, implication operator and all the inputs of particular rules were aggregated by MAX operator. In the defuzzification process the center-of-gravity-method (COG) was applied. The dialogue window "Rule Viewer" of Fuzzy Logic Toolbox is a kind of diagnostic device which enables tracing which fuzzy rules were activated on particular states of input. It also enables observation of fuzzy system output value. The fuzzy logic controller of PD type was tuned by means of Simulink Response Optimization Toolbox of Matlab-Simulink package.

3. Research stand

The view and diagram of the test stand used to conduct experiments on fuzzy logic control of pneumatic servo-drive are presented in Fig. 4 and Fig. 5.

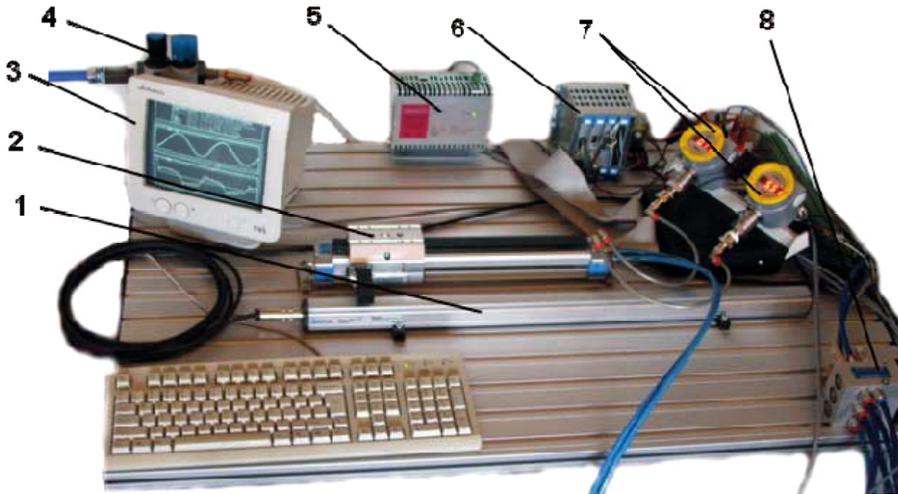


Fig. 4. The test stand of electro-pneumatic servo-drive control system: 1 - position transducer, 2 - rodless cylinder, 3 -xPC Target computer screen, 4 - pneumatic F.R. unit, 5 - power supply, 6 - axis controller, 7 - pressure transducer, 8 - proportional control valve

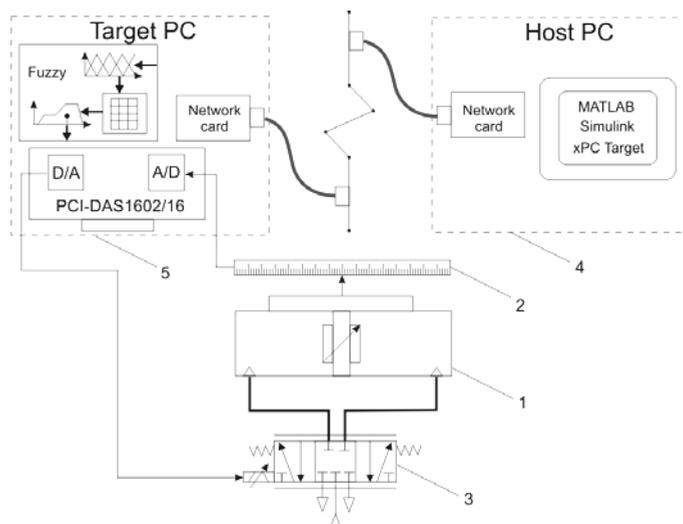


Fig. 5. The diagram of pneumatic servo-drive control system: 1 - rodless cylinder, 2 - position transducer, 2 - proportional control valve, 4 - computer Host, 5 - computer Target

The test stand consists of the following elements:

- pneumatic rodless cylinder (Festo DGPL-25-224) with piston diameter of 25 mm and stroke length of 224 mm,
- proportional 5/3 directional control valve (Festo MPYE-5-1/8-HF-010-B) controlled by 0-10 V voltage of nominal flow rate 700 l/min and switching frequency 80 Hz,
- non-contact micropulse displacement transducer (Balluff BTL5-A11-M0600-P-S32), analog output signal - voltage 0-10 V,
- 16-bit measurement card (Measurement Computing Corporation AD/DA PCI-DAS1602/16) with 8 analog inputs and 2 outputs,
- PC computers Host and Target.

The proposed pneumatic servo-drive control system contains two PC computers Target and Host where the first computer directly controls the pneumatic servo-drive and the second functions as the operator towards the direct control layer. In the computer marked as Host in Fig. 5 the software Matlab-Simulink together with xPC Target for rapid prototyping and real time control were installed. Target possesses an analog I/O card and a Real-Time xPC Target system which activates measurement data and controls pneumatic servo-drive. Host and Target communicate with each other by means of the TCP/IP protocol. Our work with the package for rapid prototyping consisted in construction and compilation of Simulink model, and sending the compiled model onto Target which together with analog I/O card and Real-Time xPC Target system functioned as real controller. Thanks to xPC Target software, the visualization of the analyzed control process was possible. The diagram showing how xPC Target works and communicates with Host computer is presented in Fig. 6.

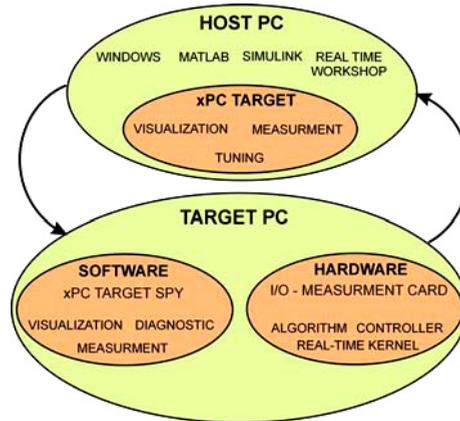


Fig. 6. Configuration of a link between Host and Target computers

In order to control pneumatic servo-drive, fuzzy PD controller was designed. The controller had two inputs $e(t)$ – position error and $\Delta e(t)$ – change of position error and one output $u(t)$ – voltage of coil valve obtained by means of Fuzzy Logic Toolbox of Matlab-Simulink package. The fuzzy controller operates on knowledge base containing IF-THEN rules (McNeill, 1994; Wolkenhauer, 2001; Kandel & Langholz, 1991; Yager et al., 1994; Driankov et al., 1996; Spooner et al., 2002) for undetermined predicates and fuzzy control mechanism (Driankov et al., 1996):

$$u(k) = F[e(k), e(k-1), \dots, e(k-v), u(k-1), u(k-2), \dots, u(k-v)], \quad (9)$$

where: $u(k)$ – control signal describing relation between controller's input and output,

$e(k)$ – position error between input signal $y_0(k)$ and output signal $y(k)$,

k – discrete time (sampling instant), $k=t/T$,

t – continuous time,

T – sampling period,

v – parameter determining controller's order,

F – nonlinear function describing knowledge base of FLC (Driankov et al., 1996).

The FLC describes relations between the change of control signal $\Delta u(k)$ from one side and position error $e(k)$ and change of position error $\Delta e(k)=e(k)-e(k-1)$ from the other side. Thus fuzzy logic control relation for $v=1$ can be written as follows:

$$\Delta u(k) = F[e(k), \Delta e(k)]. \quad (10)$$

The real output of $u(k)$ controller is obtained from the past control value $u(k-1)$ and its updating $\Delta u(k)$ as follows:

$$u(k) = u(k-1) + \Delta u(k). \quad (11)$$

The fuzzy controller of this type was proposed for the first time by Mamdani and Assilian,

and was called fuzzy logic controller of Mamdani type (Driankov et al., 1996). The Control Law of Fuzzy PD Controller can be written as follows:

$$u(k) = F[e(k), \Delta e(k)]. \tag{12}$$

The inference algorithm transforms the Control Law (12) into non-fuzzy control algorithm which resembles the equation of traditional PD controller) (Wolkenhauer, 2001; Kandel & Langholz, 1991; Driankov et al., 1996):

$$u(k) = k_p \cdot e(k) + k_D \cdot \frac{\Delta e(k)}{T} = k_p \cdot e_P(k) + k_D \cdot e_D(k)' \tag{13}$$

where k_p and k_D represent gain of the proportional and derivative controller. The rules for fuzzy PD controller are written as follows (Yager et al., 1994; Driankov et al., 1996):

IF $e_P(k)$ is <linguistic label> AND $e_D(k)$ is <linguistic label> THEN $u(k)$ <linguistic label>.

$\Delta e \backslash e$	NB	NS	Z	PS	PB
NB	NB	NB	NB	NB	NB
NS	NB	NS	NS	NS	PS
Z	NB	NS	Z	PS	PB
PS	NS	PS	PS	PB	PB
PB	PB	PB	PB	PB	PB

Table 1. Knowledge base

The position error $e(k)$ changed in the range from -3,7 V to 3,7 V which was the range of displacement of slide cylinder. For extreme values Γ and L sets were used which enabled to compensate the incomplete knowledge base for high amplitude values of $e(k)$ signal. Additionally, for position error $e(k)$ approaching zero a set of trapezoid type was used which enabled to determine the value of the assumed static error 2δ and to avoid oscillation around the zero error. For $\Delta e(k)$ signal the domain was determined in the range from -25 V/s to 25 V/s which corresponded to the range of displacement of slide cylinder. With the above assumptions taken into account the input signal $e(t)$ underwent fuzzyfication process with distribution of fuzzy sets while the input signal $\Delta e(t)$ underwent defuzzyfication process with distribution of fuzzy sets.

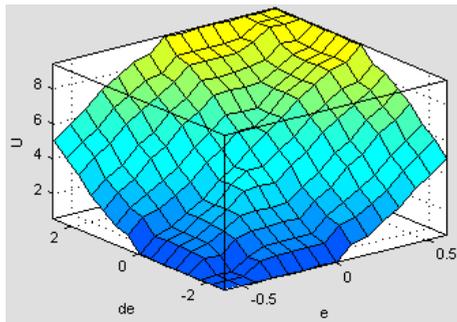


Fig. 7. Output control surface

The measurements showed that flow rate characteristics of the proportional control valve are asymmetrical. Therefore, in the central position of spool valve where the voltage was 5 V the set of sigmoidal type shifted towards right (in accordance with the asymmetrical character of the valve) was used. For extreme fuzzy sets the sets of Γ and L type were applied as it was the case for input signals. Knowledge base contained 25 fuzzy rules included in table 1. As the number of fuzzy rules and consequently the number of fuzzy sets was limited to 25 the sets were concentrated/ refined near zero error. This enabled smooth change of output signal near zero error, which implies, small jumps of the signal while passing from one fuzzy rule to the other. The surface control of the fuzzy controller is presented in Fig. 7. In the fuzzy inference process the firing degree of MIN type, fuzzy implication of MIN type and aggregation of particular outputs of the rule of MAX type were determined. In order to obtain crisp value the method of the Centre of Gravity was used.

4. Results of simulations and experimental tests of electro-pneumatic servo-drive

The performance control of the pneumatic servo-drive with fuzzy controller was checked by means of standard performance index including: settling time t_R , overshoot $\delta_p = y_m - y_0$, positioning error $e = y_0 - y(t)$ as well as integral performance indexes: IAE, ITAE, ISE, ITSE (Takosoglu & Dindorf, 2005). The additional performance criteria for follow-up control comprised absolute position error Δx and absolute velocity error) Δv :

$$\Delta x = \frac{\sum_{i=1}^N |x_0[i] - x[i]|}{N}, \quad (14)$$

$$\Delta v = \frac{\sum_{i=1}^N |v_0[i] - v[i]|}{N}, \quad (15)$$

where: N - number of measuring points.

Simulations and experimental tests were conducted for the following parameters:

- pneumatic unit: $p_z = 0,6$ MPa, $T_z = 293,15$ K,
- normal air: $T_0 = 293,15$ K, $p_0 = 0,6$ MPa, $\rho_0 = 1,205$ kg/m³, $R = 288$ N m/kg K, $\kappa = 1,4$,
- proportional directional control valve: $C_{14} = C_{12} = C_{23} = C_{45} = 1,462 \cdot 10^{-8}$ m⁴ s/kg, b_{14} , b_{12} , b_{23} , $b_{45} = 0,28$,
- pneumatic rodless cylinder: $A = 49 \cdot 10^{-5}$ m², $l_0 = 0,02$ m, $l = 0,2$ m, $M = 1$ kg, $f_l = 250$ N s/m, $k_p = 3$ N/Pa, $F_k = 100$ N, $F_{pr} = 200$ N, $v_k = 0,1$ m/s.

The simulation and experimental tests of pneumatic servo-drive were conducted mainly to check the operation of the designed control system with Fuzzy PD Controller during positioning of slide cylinder at various load mass. In accordance with the earlier assumption FLC with feedback caused by the displacement of slide cylinder was constructed. The displacement of pneumatic rodless cylinder slide was analyzed in the range $x = 0-200$ mm, at velocity v corresponding to 10%, 30%, 50%, 70% and 100% of maximum velocity v_{max} and mass load $m = 0-16,5$ kg. The tasks to be performed by fuzzy controller included transpose control and follow-up control of pneumatic servo-drive. The

input signals for transpose control were *step* signals and for follow-up control *ramp* and *sin* signals. The paper compares the selected results of simulation and experimental tests of position error displacement and velocity error of pneumatic servo-drive slide obtained in the same operating conditions and for the same input signal. In Fig. 8a and Fig. 8b the position, position error, velocity of cylinder slide obtained during transpose control for input signal of *step* type from the position $x = 0$ mm to the position $x = 80$ mm without mass load ($m = 0$) are compared.

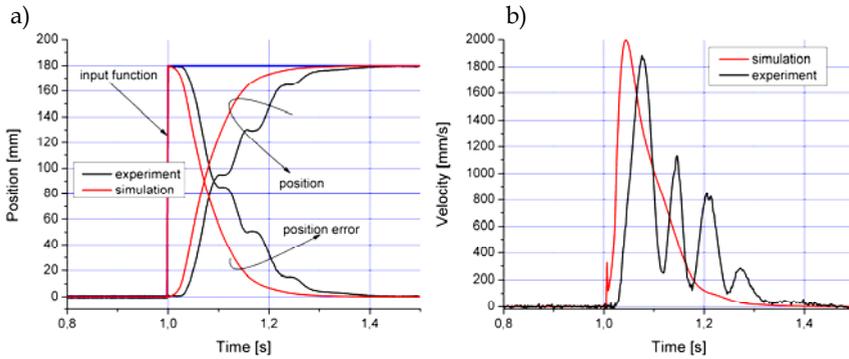


Fig. 8. Experimental and simulation results of position and position error a) and velocity b) of cylinder slide for input signal of *step* type

In Fig. 9a and Fig. 9b the position, position error, velocity and velocity error of cylinder slide obtained during follow-up control for input signal of *ramp* type from the position $x = 12$ mm to the position $x = 180$ mm with velocity 0,5 m/s and mass load $m = 3,67$ kg are presented.

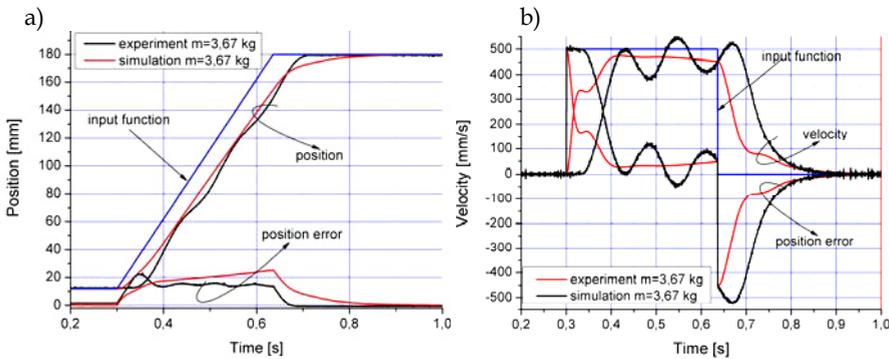


Fig. 9. Experimental and simulation results of position and position error a) and velocity b) of cylinder slide during follow-up control for input signal of *ramp* type

Fig. 10a and Fig. 10b shows the changes of absolute follow-up error of position signal and absolute follow-up error of velocity signal for follow-up control with input signal of *ramp* type.

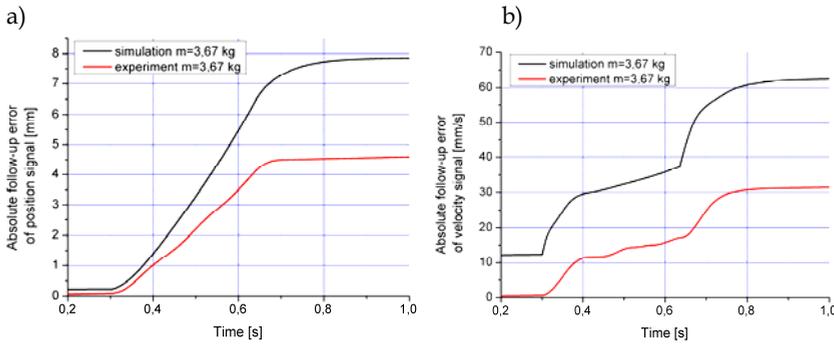


Fig. 10. Experimental and simulation results of absolute follow-up error of position signal Δx a) and velocity signal Δv b)

In Fig. 11a and Fig. 11b the position, position error, velocity and velocity error of pneumatic servo-drive cylinder slide for follow-up control with input signal of *sin* type and frequency 0,5 Hz without mass load ($m = 0$) is presented.

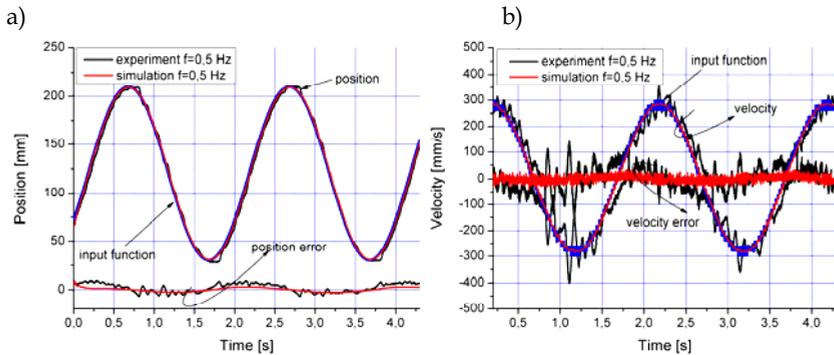


Fig. 11. Experimental and simulation results of position and position error a) and velocity b) of cylinder slide for *sin* input signal with frequency 0,5 Hz

In Table 2 average performance indexes of fuzzy control obtained during experimental tests for transpose control with input signal of *step* type and follow-up control with input signal of *ramp* and *sin* type are compared.

t_R [s]	δ_P [mm]	IAE	ISE	ITAE	ITSE	Δx mm	Δv mm/s
Step input							
0,32	0,5	22,4	3301	3,8	490,1	-	-
Ramp input							
0,17	0,6	7,2	343,7	0,7	26,43	9,5	230,1
Sin input							
-	-	23,5	64,2	116,3	320,8	2,4	62,6

Table 2. Performance indexes

On the basis of the results of experimental research and simulations it may be concluded that the changes of position and velocity of pneumatic servo-drive cylinder slide obtained during experimental tests and simulations are convergent. From experimental research it follows that in the initial phase of the cylinder slide motion a small delay (approx. 0,5 s) in relation to input signal may be caused by break-away friction force in cylinder seals. During cylinder slide motion the correcting effect of FLC is clearly visible. In the next phase of cylinder slide motion the simulation and experimental results are almost the same. The runs of absolute follow-up error of position signal and velocity are also similar and the differences result from the quality of performance control. However, for bigger load mass ($m = 3,67$ kg) the oscillatory motion of servo-motor slide is observed. The differences between simulation and experimental results are also caused by measurements noise in displacement transducer (for simulation tests the measurement noise was not taken into account). The designed fuzzy PD controller efficiently carries out the transpose and follow-up control of pneumatic servo-drive.

5. Teaching/play-back control of electro-pneumatic servo-drive

A teaching/play-back control system of electro-pneumatic servo-drive was constructed. Its input signal introduced manually by the operator by means of linear potentiometer was reproduced by pneumatic cylinder of pneumatic servo drive. In the teaching/play-back control system the software with PD Fuzzy PD Controller performing the task of teaching the pneumatic servo-drive motion was used. The suggested control system operates as follows: the operator by manually moving the linear potentiometer sets an optional motion trajectory and next the slide of rodless cylinder of pneumatic servo-drive reproduces this trajectory in real time. The view of the research stand of pneumatic servo-drive play-back control system is presented in Fig. 12 and its control diagram in Fig. 13. The research stand was constructed of the following elements:

- pneumatic rodless cylinder (Festo DGPL-25-224) with piston diameter of 25 mm and stroke length of 224 mm,
- proportional 5/3 directional control valve (Festo MPYE-5-1/8-HF-010-B) of nominal flow rate 700 l/min and switching frequency 80 Hz controlled by analog voltage signal 0-10 V,
- non-contact micropulse displacement transducer (Balluff BTL5-A11-M0600-P-S32), analog output signal - voltage 0-10 V,
- linear potentiometer (Festo MLO-POT-225-TLF), supply voltage 13-30 V, output voltage 0-10V, resistance 5 k Ω , frequency 5 Hz-2 kHz,
- 16-bit measurement card (Measurement Computing Corporation AD/DA PCI-DAS1602/16) with 8 inputs and 2 analog outputs,
- PC computers Host and Target.

During the experimental tests optional motion trajectories were set by means of linear potentiometer and motion trajectories reproduced by pneumatic servo-drive were recorded in real time. Each trajectory could be reproduced unlimited number of times in real time or recorded in the control program. The recorded motion trajectories were then reproduced by pneumatic servo-drive.

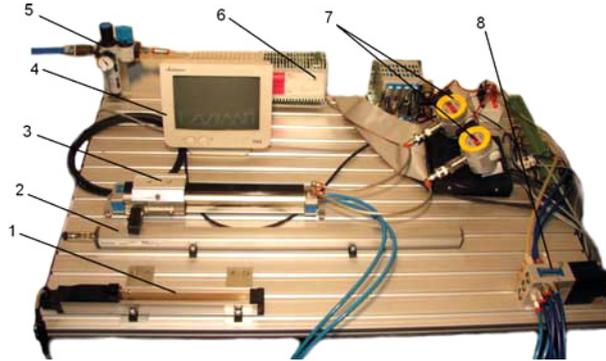


Fig. 12. The research stand of electro-pneumatic servo-drive motion teaching/play-back control system: 1 - linear potentiometer, 2 - position transducer, 3 - rodless cylinder, 4 - Target computer screen, 5 - pneumatic F.R. unit, 6 - power supply, 7 - pressure transducer, 8 - proportional control valve

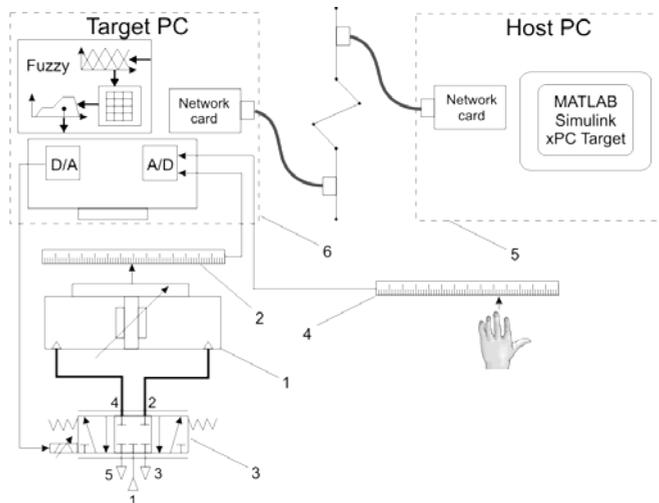


Fig. 13. The diagram of pneumatic servo-drive motion teaching/play-back control system: 1 - rodless cylinder, 2 - position transducer, 2 - proportional control valve, 4 - linear potentiometer, 5 - Host computer, 6 - Target computer

The quality of teaching/play-back control with FLC was checked by means of standard performance indexes including position error $\delta_x = |x_0 - x(t)|$, velocity error $\delta_v = |v_0 - v(t)|$ and acceleration error $\delta_a = |a_0 - a(t)|$. In follow-up control the additional quality criteria comprised: absolute error signal $\Delta x(t)$ of velocity $\Delta v(t)$ and acceleration position $\Delta a(t)$:

$$\Delta x(t) = \frac{\sum_{i=1}^N |x_0[i] - x[i]|}{N}, \tag{1}$$

$$\Delta v(t) = \frac{\sum_{i=1}^N |v_0[i] - v[i]|}{N}, \tag{2}$$

$$\Delta a(t) = \frac{\sum_{i=1}^N |a_0[i] - a[i]|}{N}, \tag{3}$$

where: N – number of measurement points.

The results of experimental tests on displacement, position error, velocity and velocity error as well as acceleration and acceleration error of cylinder slide of pneumatic servo-drive during reproduction of optional motion trajectory are presented in Fig. 14.

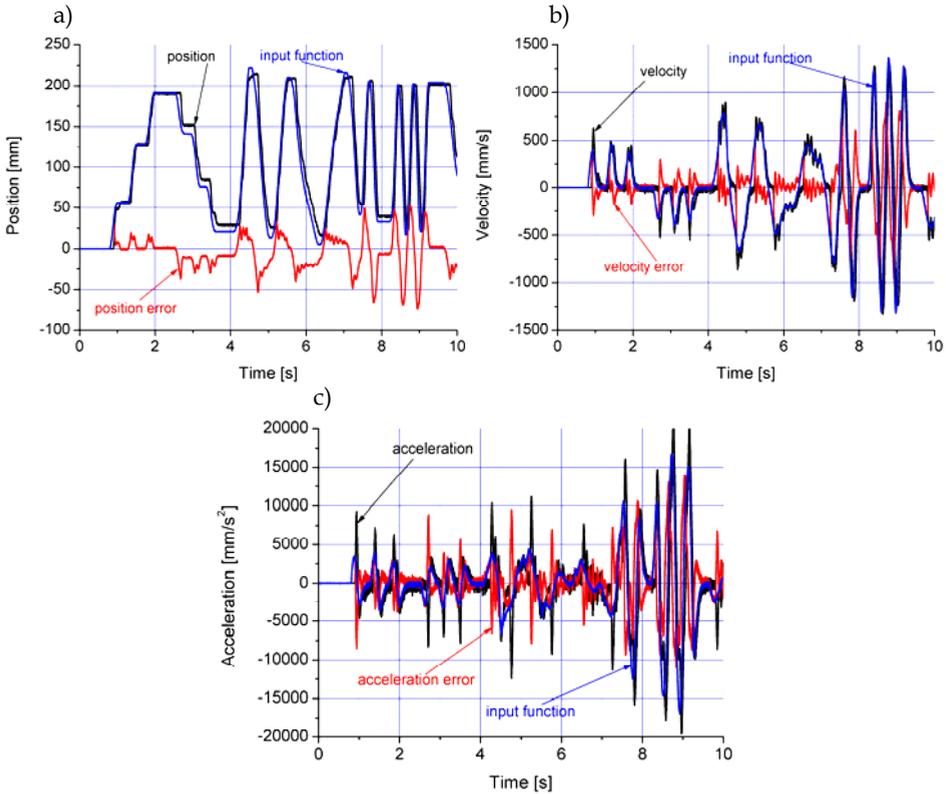


Fig. 14. Experimental results of position a), velocity b) and acceleration c) of cylinder slide

The changes of absolute follow-up error signal of displacement, velocity and acceleration of pneumatic servo-drive cylinder slide are presented in Fig. 15.

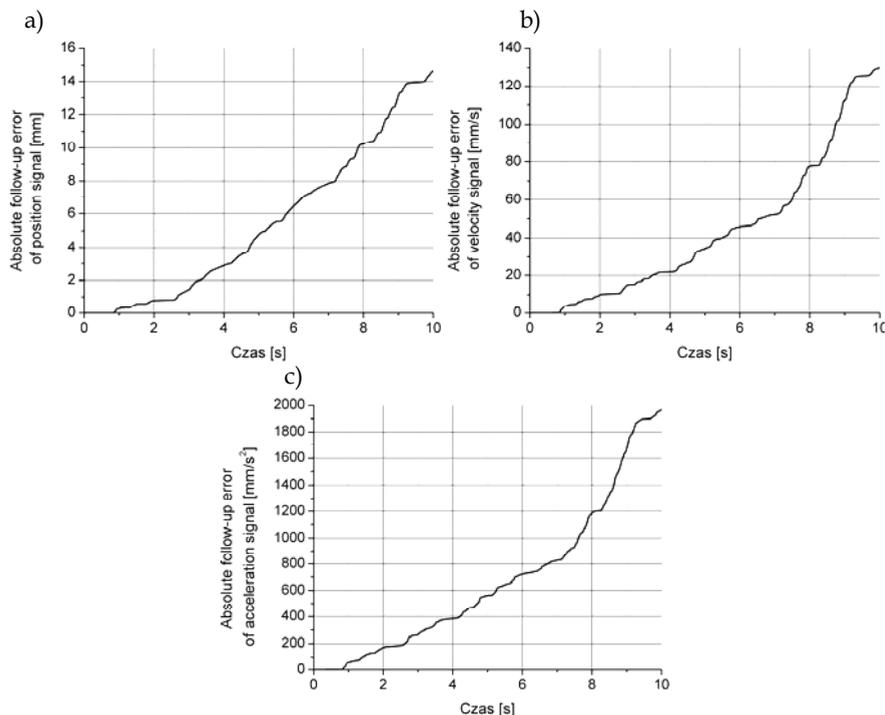


Fig. 15. Experimental values of absolute follow-up error signal of position a), velocity b) and acceleration c)

6. Prototype of electro-pneumatic parallel 3-UPRR tripod manipulator

In the Division of Mechatronics (Kielce University of Technology, Poland) a prototype of pneumatic translational parallel manipulator (PTPM) of tripod kinematic structure was constructed (Dindorf et al., 2005; Laski & Dindorf, 2007). The prototype of tripod parallel manipulator with Festo servopneumatic precision positioning systems is presented in Fig. 1a. The manipulator possesses a supporting structure, fixed base, moving platform and three pneumatic linear motions (servopneumatic axis). Each servopneumatic axis consists of: rodless pneumatic cylinder type DGPIL-25-600 with integral feedback transducer (built-in 'Temposonic' encoders for continual positioning feedback to the master control unit), 5/3 servopneumatic valve (proportional directional control valve) type MPYE-5-1/8-HF-010B, axis interface type SPC-AIF, positioning axis sub-controller type SPC-200 (the use of a sub-controller card permits control of up to four axes) and Ethernet/Can Bus interface. According to the systematics the prototype of 3-DoF pneumatic translational parallel manipulators is of 3-UPRR kinematic structure (Fig. 1b). Each of the three identical closed-loop chains of the manipulator consists of serial kinematic chains: universal cardan joint (U), prismatic joint (P), formed by a rodless pneumatic cylinder and two revolute joints (2R) formed after universal cardan had been parted. The slide of rodless cylinder was connected with fixed base by means of articulated joints of U cardan and the end cap of cylinder were

connected by revolute joint R to the moving platform. The second revolute joint R was placed in tool center point (TCP) of the moving platform. The presented construction of the parallel manipulator ensures parallel position of the moving platform to the fixed base for optional position of pneumatic cylinder. The kinematic structure of a new prototype of 3-UPRR pneumatic parallel manipulator is an interesting solution expanding the architecture of parallel manipulators, type 3-DoF TPM.

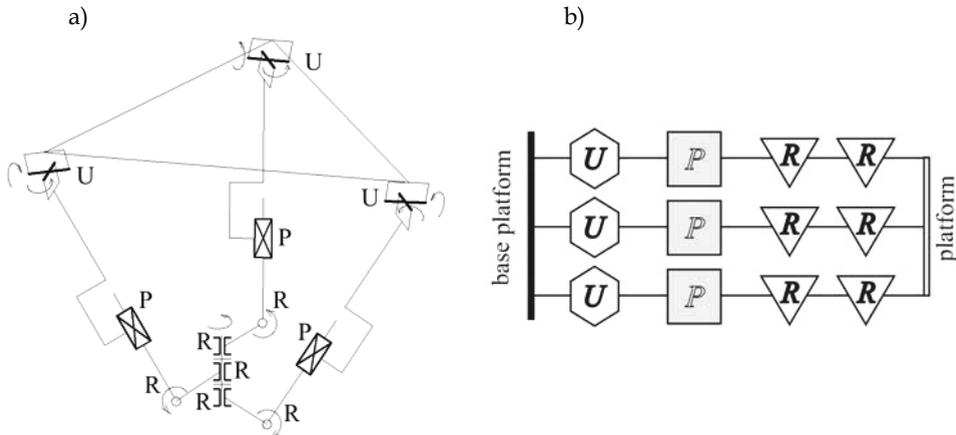


Fig. 16. Pneumatic translational parallel manipulator: a) prototype, b) kinematics scheme

7. Model research of electro-pneumatic parallel 3-UPRR tripod manipulator

CAD software (SolidWorks, Mechanical Desktop, Solid Edge) commonly used by constructors enables designing solid models of complex mechanisms of parallel kinematics. A solid model of 3-UPRR pneumatic parallel manipulator obtained by SolidWorks is presented in Fig. 17a. To record geometric and kinematic relations holding for pneumatic parallel manipulator of 3-UPRR kinematics its kinematic model presented in Fig. 17b was used.

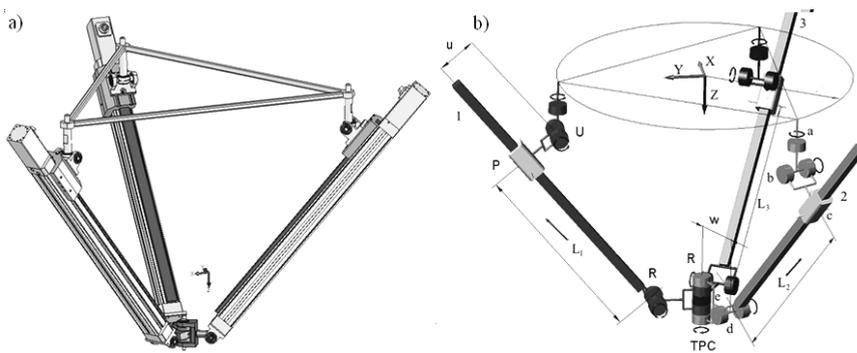


Fig. 17. Solid model a) of pneumatic parallel manipulator b) and kinematic model

By means of Dynamic Designer Motion, which possesses graphic interface SolidWorks the simulation of pneumatic parallel manipulator's motion was conducted. In order to simulate the manipulator's motion it was necessary to define the basic parameters, kinematic joints and motion restrictions. For solid model a few composite relations were defined which enabled assigning them kinematic joints. In some cases it was necessary to introduce joints describing the construction's stiffness. Basing upon material properties and the shape of particular solids the mass of the solid model was calculated. The simulation of manipulator's parallel mechanism motion was saved in .avi format. The simulations conducted on a solid model aimed at position analysis of TCP point of the moving platform. The position of TCP point results from linear motion of pneumatic rodless cylinder, independently controlled by servo-valves.

Since the application of SolidWorks in modeling kinematics and dynamics of parallel manipulators is restricted further simulation was carried out by means of SimMechanics library of Matlab-Simulink package. The library enables the construction of complex mechanisms of parallel manipulators excluding mathematical descriptions of their kinematics and dynamics. The kinematic model of 3-UPRR manipulator obtained by means of SimMechanics library is presented as block diagram in Fig. 18.

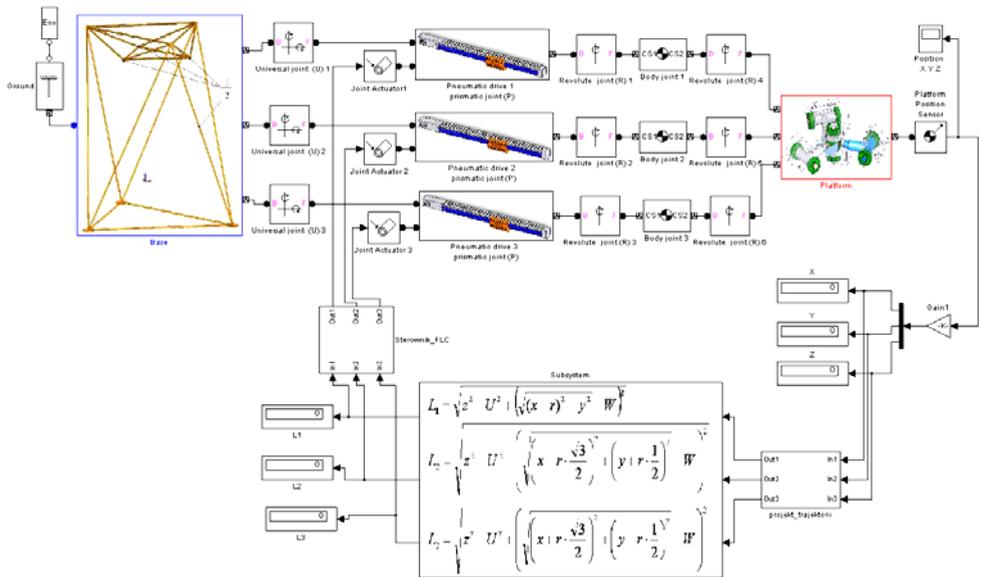


Fig. 18. The block-diagram of kinematic model of electro-pneumatic parallel tripod manipulator

On the basis of this block-diagram the equivalent model of pneumatic parallel manipulator was worked out (Fig. 19a). In simulations based upon SimMechanics library an equivalent model of pneumatic tripod manipulator with its spatial orientation indicated was constructed. In SimMechanics library all the solid elements of the manipulator were described by substitute geometry by means of ellipsoids and assigned both masses and inertial tensors. In Matlab-Simulink environment tripod-based parallel kinematic

manipulator was connected with its control system. The equivalent model retains kinematic joints and spatial orientation defined in solid model in SolidWorks. To create the equivalent model it was necessary to define the gravity centre of solids in central and local coordinates. The kinematic model was used to TCP trajectory analysis. The TCP trajectory of pneumatic parallel manipulator in Cartesian coordinates is shown in Fig. 19b.

The research on the model was supplemented with the analysis of servo-pneumatic axis control applied in 3-UPRR pneumatic parallel manipulator. By means of simulation model and experimental setup transpose control, follow-up control, trajectory motion control and fuzzy control of single servo-pneumatic axis were investigated (Takosoglu 2005). To control the servo-pneumatic axis a controller FLC (Fuzzy Logic Controller) of PD type was used. In fuzzyfication process conditionally firing rules of type MIN, implication operator of type MIN and aggregation of output rules of type MAX were employed. Twenty five FLC's knowledge base forming FLC's control surface were used. To obtain fuzzy output value the center of gravity function (COG) was used.

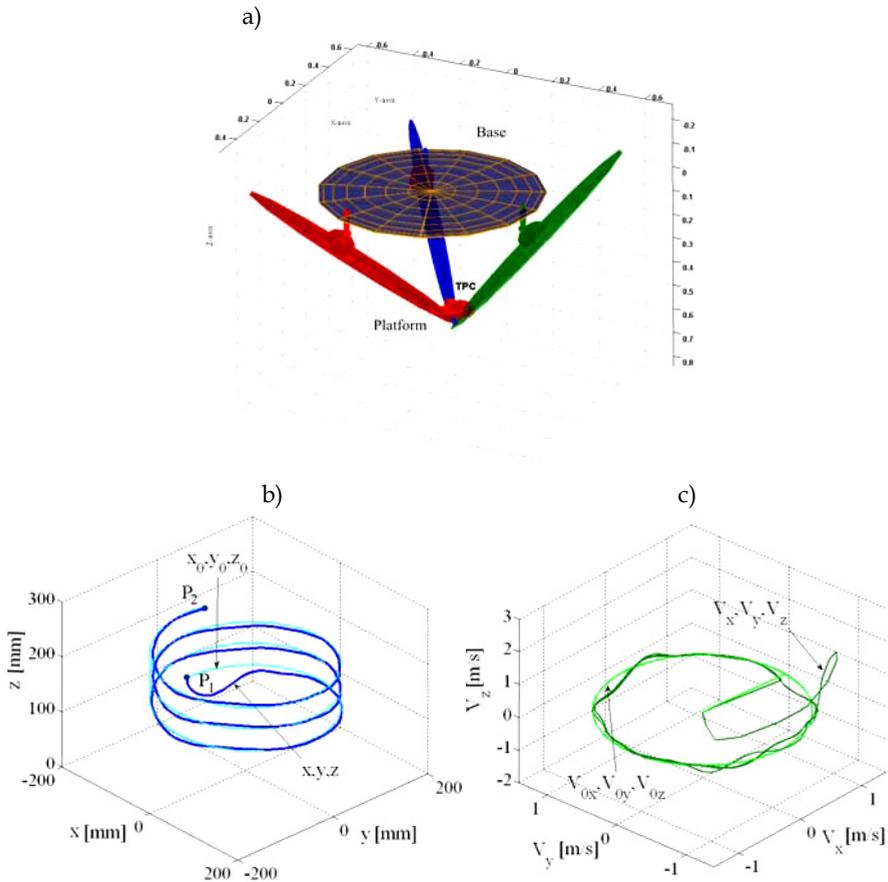


Fig. 19. The equivalent model a), TCP trajectory b) and velocity of electro-pneumatic parallel tripod manipulator

Application of FLC controller improved dynamics and positioning accuracy of servopneumatic axis and eliminated disturbances in its control system. On the basis of the research the control of servopneumatic axis using fuzzy logic for trajectory planning of parallel manipulators can be established. The research proves applicability of fuzzy logic in control of pneumatic parallel manipulators with different kinematic chain structure. Advanced servopneumatic positioning contributes to a new generation of parallel manipulators. Especially parallel manipulators actuated by servopneumatic axis enable realization of very fast pick and place in 3-D workspace.

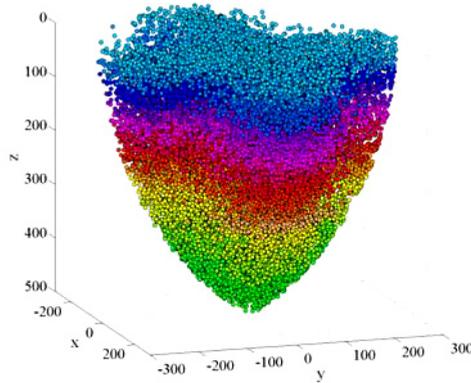


Fig. 20. Working space of pneumatic parallel manipulator

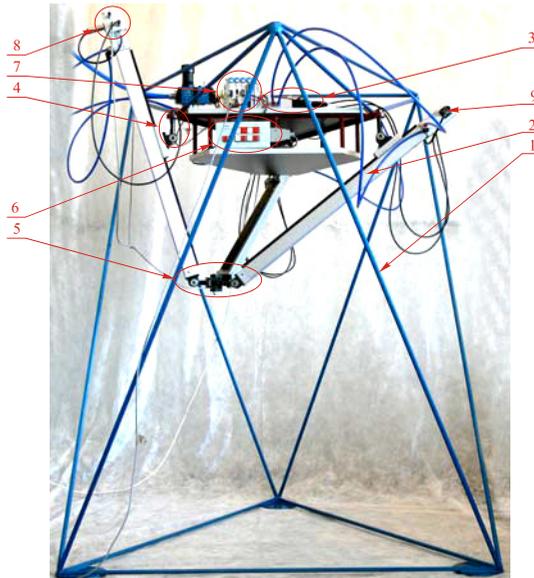


Fig. 21. Component elements of electro-pneumatic parallel tripod manipulator: 1 - basis, 2 - cylinder, 3 - servo-valve 5/3, 4 - universal Cardan joints, 5 - working platform, 6 - control panel, 7 - the driver the SPC -200, 8 - the interface of communicate the network SPC AIF MTS, 9 - the connector communication

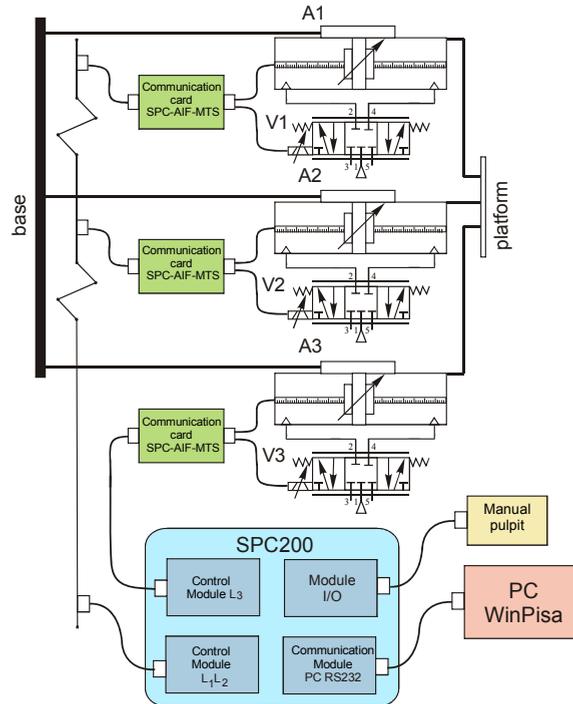


Fig. 22. Schematic diagram of pneumatic servo-drive parallel manipulator

8. Conclusion

The results of simulation and experimental tests conducted for pneumatic servo-drive with FLC are presented. For positioning control of pneumatic servo-drive a fuzzy PD controller was designed and constructed by means of xPC Target software of Matlab-Simulink package for rapid prototyping and hardware-in-the-loop simulation. The non-linear simulation model of pneumatic servo-drive was constructed and used to tune fuzzy PD controller by means of Fuzzy Logic Toolbox of Matlab-Simulink package. The research stand consisted of two computers: Host and Target with the first of them being the master and performing the function of the operator towards the direct control layer and the second directly controlling the pneumatic servo-drive. The fuzzy logic PD controller enables precise positioning of pneumatic servo-drive with the precision specified for industrial manipulators. A lot of simulation and experimental tests were carried on pneumatic servo-drive with fuzzy controller which was used for its transpose and follow-up control. The designed fuzzy system is efficient, stable and resistant to disturbances and can be applied in any configurations of pneumatic servo-drive without necessity to tune the regulator, apply signal filtration or additional operations in track control or restrict the signals generated.

The analysis of displacement and velocity characteristics show that their runs are similar. The position delay (approx. 0,5 s) on the experimental characteristics in relation to input signal is caused by break away friction force. In the process of servo cylinder's motion correcting effect of FLC leading to rapid minimization of displacement error is observed. In

the next motion phase the simulation and experimental characteristics are almost the same. The runs of absolute follow-up error of position signal and velocity are also similar and the differences result from the quality of performance control. Some oscillations of transient response most probably caused by time delay, stick-slip effect in seals and strip of pneumatic rodless cylinder are observed. In the the mathematical model of the cylinder Stribeck friction force was taken into account. Including LuGre (Lund-Grenoble) model in the friction would considerably improve the simulation results but would also make the numerical solutions of simulation model much more complex. It seems that other simplifications of mathematical model do not influence the difference between simulation and experimental results. It should be noted however, that differences between simulation and experimental results are affected by measurement noise in displacement transducer. In simulations measurement noise was not taken into account.

The teaching/play-back control system using fuzzy logic control was constructed and practically applied in various servo-pneumatic systems used in production automation. Basing upon the presented control/teaching/play-back system the prototype of physiotherapy manipulator facilitating the movement of hand and leg is being constructed (Takosoglu, 2005).

The research on models shortened the construction process of the prototype of 3-UPRR electro-pneumatic parallel manipulator. The analysis of geometric and kinematic properties of the prototype resulted in numerous changes and modifications of its construction made in order to obtain the biggest workspace without collision with pneumatic linear motion. The research enabled drawing the conclusions on construction optimization and control of 3-UPRR pneumatic parallel manipulator. Our further research will focus on dynamic analysis and dynamic synthesis as well as on 3-UPRR pneumatic parallel manipulator's programming. The presented novel 3-UPRR parallel mechanism will find its application in manufacturing manipulators and rehabilitation manipulators. Thanks to application of parallel kinematics in construction of electropneumatic manipulators higher rigidity of the whole pneumatic structure has been obtained and both positioning precision and dynamic properties have been improved. The closed mechanical chains make the dynamics of parallel manipulators coupled and highly nonlinear.

9. References

- Bucher R.; Balemi S. (2006). Rapid controller prototyping with Matlab/Simulink and Linux. *Control Engineering Practice*, Vol. 14, (May 2006), pp. 185-192
- Dindorf R.; Laski P.; Takosoglu J. (2005). Control of electro-pneumatic 3-DOF parallel manipulator using fuzzy logic. *Hydraulika a Pneumatyka*, Vol. 1-2, (January 2005), pp. 56-59, ISSN 1335-5171
- Dindorf R.; Laski P.; Takosoglu J. (2008). Solid modeling of pneumatic elements and driving systems, *Book of Extended Abstracts of the 12th International Scientific Seminar on Developments in Machinery Design and Control*, pp.27-28, ISBN 978-83-87982-08-9, Cerveny Klastor, September 2008, University of Technology and Live Sciences, Bydgoszcz
- Dindorf R.; Takosoglu J. (2005). Analysis of pneumatic servo-drive control system using fuzzy controller. *Pneumatyka* Vol. 1 (January-February 2005), pp. 51-53, ISSN 1426-6644

- Driankov, D.; Hellendoorn, H.; Reinfrank, M. (1996) *An introduction to fuzzy control*, WNT, ISBN 83-204-2030-x, Warsaw
- Kandel A. (1991). *Fuzzy Expert Systems*, CRC Press, Inc., ISBN 08-493-4297-x, Boca Raton, Florida
- Kandel A.; Langholz G. (1993). *Fuzzy Control Systems*, CRC Press, Inc., ISBN 08-493-4496-4, Boca Raton, Florida
- Laski P.; Dindorf R. (2007). Prototype of pneumatic parallel manipulator. *Hydraulika a Pneumatyka*, Vol. 1, (January 2007), pp. 22-24, ISSN 1335-5171
- Laski P.; Dindorf R. (2007). Prototyping of tripod-type pneumatic parallel manipulator, *Book of Extended Abstracts of the 11th International Scientific Seminar on Developments in Machinery Design and Control*, pp.49, ISBN 83-87982-42-3, Cervený Kláštor, September 2007, University of Technology and Live Sciences, Bydgoszcz
- McNeill F. M. (1994). *Fuzzy Logic A Practical Approach*, Academic Press Professional, Inc., ISBN 0-12-485965-8, Boston
- Merlet J. P. (2006). *Parallel robots*, Springer, ISBN 1-4020-4132-7, Dordrecht
- Murray R. M.; Li Z.; Sastry S. S. (1994). *A mathematical introduction to robotic manipulation*, CRC Press, Inc., ISBN 0-8493-7981-4, Boca Raton, Florida
- Renn J. C.; Liao C. M. (2004). A study on the speed control performance of a servo-pneumatic motor and the application to pneumatic tools. *The International Journal of Advanced Manufacturing Technology*, Vol. 23, (February 2004), pp. 572-576, ISSN 1433-3015
- Sandler B. Z. (1999). *Robotics: Designing the mechanisms for automated machinery*, Academic Press, ISBN 0-12-618520-4, California
- Schulte H.; Hahn H. (2004). Fuzzy state feedback gain scheduling control of servo-pneumatic actuators. *Control Engineering Practice*, Vol. 12, (May 2004), pp. 639-650
- Situm Z.; Pavkovic D.; Novakovic B. (2004). Servo pneumatic position control using fuzzy PID gain scheduling. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 126, (June 2004), pp. 376-387
- Spooner J. T.; Maggiore M.; Ordóñez R.; Passino K. M. (2002). *Stable adaptive control and estimation for nonlinear systems: Neural and fuzzy approximator techniques*. John Wiley & Sons, Inc., ISBN 0-471-22113-9, New York
- Takosoglu J.; Dindorf R. (2005). Fuzzy control of pneumatic servo-drive. *Proceedings of the 15th National Conference of Automatics*, pp. 117-120, ISBN 83-89475-01-4, Warsaw, June 2005, Systems Research Institute Polish Academy of Science, Warsaw
- Takosoglu J.; Dindorf R. (2006). Rapid prototyping a fuzzy control of electro-pneumatic servo-drive in real time. *Scientific Bulletin of the College of Computer Science*, Vol. 5, No. 1, pp.57-70,
- Takosoglu J.; Dindorf R. (2007). Positioning and teaching/play-back fuzzy control of electro-pneumatic servo-drive in real time, *Proceedings of the 7th European Conference of Young Research and Science Workers Transcom 2007*, pp. 199-202, ISBN 978-80-8070-694-4, Zilina, June 2007, University of Zilina, Zilina
- Takosoglu J.; Dindorf R. (2007). Positioning control and teaching/play-back control of electro-pneumatic servo-drive, *Book of Extended Abstracts of the 11th International Scientific Seminar on Developments in Machinery Design and Control*, pp.89, ISBN 83-87982-42-3, Cervený Kláštor, September 2007, University of Technology and Live Sciences, Bydgoszcz

- Takosoglu, J. (2005). *Analysis and synthesis of pneumatic multi-axis servo-drive control system using fuzzy controller*, Dissertation, Kielce University of Technology, Kielce
- Takosoglu, J. E.; Dindorf, R. F.; Laski, P. A. Rapid prototyping of fuzzy controller pneumatic servo-system. *The International Journal of Advanced Manufacturing Technology*, Vol. 40, No. 3-4, January 2008, pp. 349-361, ISSN 0268-3768
- Takosoglu, J.; Dindorf, R. (2005) Fuzzy control of pneumatic servo-drive, *Proceedings of the 15th National Conference of Automatics. Systems Research Institute Polish Academy of Science*, pp. 117-120, ISBN 83-89475-01-4, Warsaw, June 2005, Systems Research Institute Polish Academy of Science, Warsaw
- Tsai L. W. (1999). *Robot analysis: The mechanics of serial and parallel manipulators*, John Wiley & Sons, Inc., ISBN 0-471-32593-7, New York
- Wolkenhauer O. (2001). *Fuzzy mathematics in systems theory and data analysis*, John Wiley & Sons, Inc., ISBN 0-471-22434-0, New York
- Yager, RR.; Filev, DP. (1994) *Essentials of fuzzy modeling and control*, WNT, ISBN 83-204-1909-3, Warsaw
- Zhu Y. (2006). *Control of pneumatic systems for free space and interaction tasks with system and environmental uncertainties*, Dissertation, Vanderbilt University, Nashville, Tennessee

Teleoperation System of Industrial Articulated Robot Arms by Using Forcefree Control

Satoru Goto
Saga University
Japan

1. Introduction

Recently, network robotics attracts many researchers' attention and a lot of software and hardware on communication technologies are developed for network robotics (Chong et al., 2003; Rogers, 2001; Sanfeliu et al., 2008; Sheridan, 1995; Stassen, 1997). Teleoperation techniques of robot have been developed for many purposes such as ball catching task (Smith et al., 2008), remote handling of dangerous materials in a nuclear environment (Geeter et al., 1999), undersea operation, explosive material disposals, robot-assisted surgery (Challacombe, 2003; Marohn, 2004; Park, 2006) and manipulation systems for planetary exploration (Nickels et al., 2001). Performance of a variety of elements and factors for the telemanipulation system have been investigated by an experimental study (Mora, 2007) and the Internet based teleoperation systems are also eagerly developed (Bambang, 2008; Slawiński et al., 2007; You et al., 2001). For a point of view of control, control techniques of teleoperation system have been investigated such as bilateral control (Aziminejad et al., 2001; Hokayem et al., 2001; Slawiński et al., 2007) and nonlinear adaptive control (Hung, 2003). Explosively grown network technology and robot technology are inextricable relation and expectation on the network robotics becomes large.

In usual teleoperation systems, the operational side and the working side are determined definitely in advance, and the robot in the working side moves according to the command from the operational side. Moreover, in order to operate the robot in the working side, special skill for the operation of the equipment in the operational side is usually required. On the other hand, many kinds of the industrial robot arms have been operated in factories. If these robot arms can be applied both to the operational side and the working side of the teleoperation system, the handleability of the teleoperation system will be remarkably improved. For example, similar mechanism between the operational side and the working side is preferable for intuitive operation of the teleoperation system.

In order to realize passive motion of the industrial robot arms, the forcefree control had been proposed (Goto, 2007). The forcefree control realises the passive motion of the robot arm according to the external force under the zero friction and zero gravity condition. Moreover, the forcefree control was expanded to the forcefree control with independent compensation (Goto et al., 2007). With the forcefree control with independent compensation, the robot arm moves passively according to the external force as in the circumstance of the assigned friction,

the assigned gravity and the assigned inertia. The forcefree control can be applied to the direct teaching (Kushida et al., 2001) and pull-put work of the industrial robot arms (Kushida et al., 2003).

In this research, the teleoperation system is proposed by applying the forcefree control to the robot in the operational side and the position control to the robot in the working side. The method can realize alternation of the roles in the operational side and the working side only by changing the control techniques. The effectiveness of the proposed teleoperation system is confirmed by the experimental results using actual robot arms.

2. Teleoperation System by Using Forcefree Control and Position Control

2.1 Handleability of Teleoperation

Figure 1 shows the concept of the proposed teleoperation system. Both of the operational side and of the working side, any types of industrial robot arms can be used. In the operational side, the forcefree control technique is adopted in order to realize the passive motion due to the influence of external force. When an operator impresses a force upon the robot arm in the operational side by his hand, the robot move according to the applied force. The information of the motion of the robot in the operational side is transmitted to the working side through network. In the working side, the position control is adopted for realization of the same motion of the robot arm in the operational side.

In order to realize the teleoperation system, the dedicated equipment especially for the operational side is usually adopted, and the specific transmission line is usually used. Various equipment is required for various purpose of the teleoperation system, however, the development of the dedicated equipment is costly. If various robot arms can be used both for the operational side and the working side, the development of the dedicated equipment is not required.

The main advantage of the proposed teleoperation system is the usage of the existing equipment and technology. Both for the operational side and for the working side, any kind of robot arms are utilizable. Moreover, the Internet technology is used for the data transmission between the operational side and the working side. Thereby, the most preferable mechanism of the robot arms both for the operational side and for the working side can be selected and the teleoperation system can be constructed freely if the Internet is available.

In addition, the operational side and the working side can be replaced freely by putting both the operational program and the working program in the computer that controls the robot arms. The operational role can be replaced with the working role only by executing the working program, and the working role can be replaced with the operational role only by executing the operational program.

2.2 Configuration Procedure of Teleoperation System

The block diagram of the proposed teleoperation system is shown in Fig. 2. The left side of the Fig. 2 shows the block diagram of the operational side and the hand side shows that of the working side. The operational side and the working side are connected by the network.

A servo controller of industrial robot arm includes a position loop and a velocity loop (Kyura, 1996; Nakamura et al., 2004). Input to the industrial robot arm is usually the joint position of each link. Hence, the industrial robot arms should be considered as the combination of the mechanism of the robot arm and the servo controller. The control loop of the servo controller is shown both on the left side and on the right side of Fig. 2. In the operational side, the

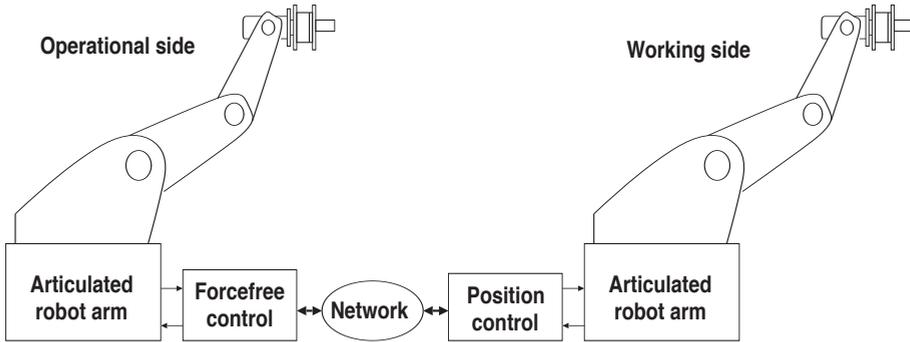


Fig. 1. Schematic diagram of teleoperation system

forcefree control is added to the robot arm and the passive motion according to the external force can be realized.

The tip position of the robot arm in the working side must coincide with that of the robot arm in the operational side. If the robot arm mechanisms between the operational side and the working side are exactly the same, the position output of the robot arm in the operational side can be directly used for the reference input of the robot arm in the working side. However, the robot arm mechanism in the working side is generally different from that in the operational side. Hence, the compensation of mechanism difference is required. The compensated reference input is transmitted to the robot arm in the working side through the network. Then, the robot arm in the working side moves according to the robot arm in the operational side.

2.2.1 Operational Side Control (Forcefree Control)

In the operational side control, the forcefree control is adopted in order to realize the passive motion of the robot arm. Figure 3 shows the concept of the forcefree control. In industrial robot arms, the servo controller is adapted to control the motion of the robot arm, and the robot arm moves according to the position reference of each joint. The external force impressed upon the robot arm is treated as disturbance and the servo controller compensates such disturbance. Hence, the external force never move the industrial robot arm. The forcefree control can achieve the passive motion of the industrial robot arms under virtual circumstances of zero gravity and zero friction without any change of the built-in controller. By use of the forcefree control, the robot arm moves passively according to the external force directly as if it were under the circumstances of zero friction and zero gravity.

The entire dynamics of the industrial robot arms controlled by the forcefree control is described as

$$H^o(q^o)\ddot{q}^o + h^o(q^o, \dot{q}^o) = \tau_f^o \tag{1}$$

where $H^o(q^o)$ is the inertia matrix, $h^o(q^o, \dot{q}^o)$ is the coupling nonlinear term, τ_f^o is the joint torque corresponding to the external force f on the tip of robot arm.

Dynamics of an articulated robot arm is expressed by

$$H^o(q^o)\ddot{q}^o + D^o\dot{q}^o + N_{\mu}^o f_s^o(\dot{q}^o) + h^o(q^o, \dot{q}^o) + g(q^o) = \tau_s^o + \tau_f^o \tag{2}$$

where $D^o\dot{q}^o + N_{\mu}^o f_s^o(\dot{q}^o)$ is the friction term, $g(q^o)$ is the gravity term, q^o is the position of joint angle, τ_s^o is the torque input to the robot arm. The dynamic equation of an industrial

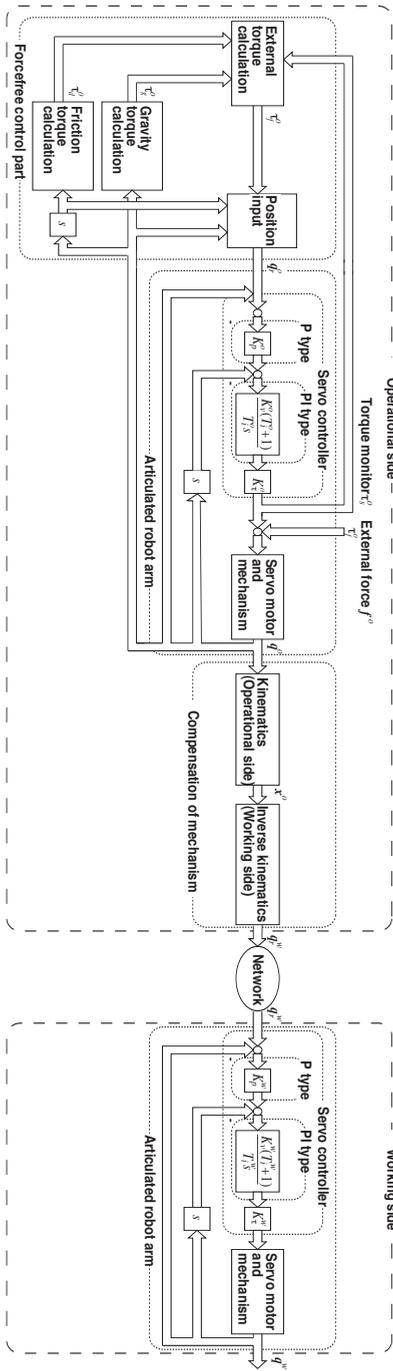


Fig. 2. Block diagram of the teleoperation system

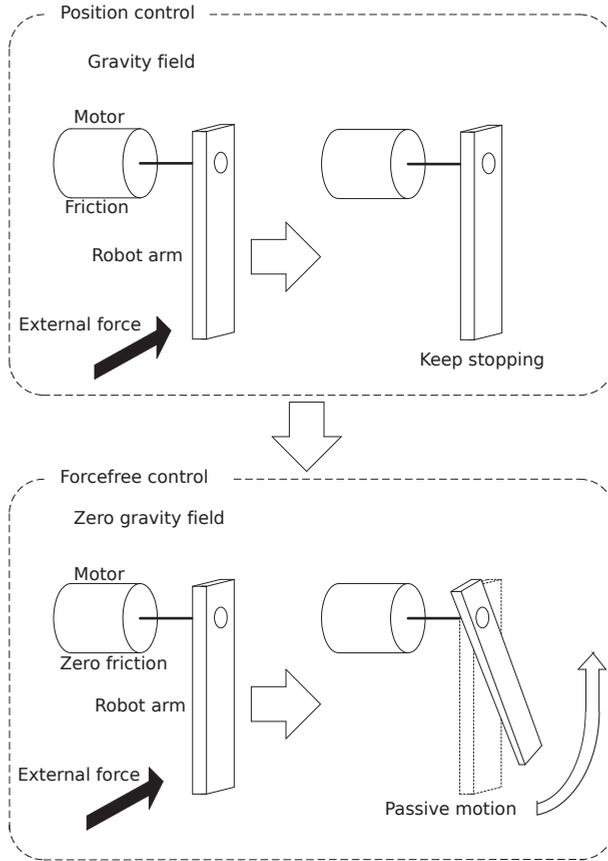


Fig. 3. Concept of the forcefree control

articulated robot arm in the operational side including the servo controller is given by

$$H^o(\mathbf{q}^o)\ddot{\mathbf{q}}^o + \mathbf{h}^o(\mathbf{q}^o, \dot{\mathbf{q}}^o) = K_\tau^o [K_v^o \{K_p^o (\mathbf{q}_r^o - \mathbf{q}^o) - \dot{\mathbf{q}}^o\}] \quad (3)$$

where \mathbf{q}_r^o is the position reference of joint angle, K_p^o , K_v^o and K_τ^o are position loop gain, velocity loop gain and torque constant for the robot in the operational side, respectively

In order to realize the entire dynamics of the industrial robot arms (1), the inputs of joint angle (\mathbf{q}_r^o) for the forcefree control is given by

$$\mathbf{q}_r^o = (K_p^o)^{-1} \{ (K_v^o)^{-1} (K_\tau^o)^{-1} \boldsymbol{\tau}_f^o + \dot{\mathbf{q}}^o \} + \mathbf{q}^o \quad (4)$$

where $\boldsymbol{\tau}_f^o$ is the joint torque corresponding to the external force \mathbf{f} on the tip of robot arm as

$$\boldsymbol{\tau}_f^o = -(\boldsymbol{\tau}_s^o - \boldsymbol{\tau}_d^o - \boldsymbol{\tau}_g^o) \quad (5)$$

where $\boldsymbol{\tau}_d^o$ is the friction torque described by

$$\boldsymbol{\tau}_d^o = D^o \dot{\mathbf{q}}^o + N_\mu^o f_s^o(\dot{\mathbf{q}}^o) \quad (6)$$

and τ_g^o is the gravity torque described by

$$\tau_g^o = \mathbf{g}(\mathbf{q}^o). \quad (7)$$

2.2.2 Compensation of Mechanism

In order to coincide with the tip position of the robot arm in the working side to that in the operational side, the compensation of the mechanism difference between the operational side and the working side is required. The tip position of the robot arm in the operational side (\mathbf{x}^o) is calculated from the position output (\mathbf{q}^o) by using the kinematics of the robot arm in the operational side as

$$\mathbf{x}^o = \mathbf{f}^o(\mathbf{q}^o) \quad (8)$$

where \mathbf{f}^o means the kinematics of the robot arm in the operational side. The inputs of joint angle (\mathbf{q}_d^w) for the robot arm in the working side is given by using the inverse kinematics of the robot arm in the working side as

$$\mathbf{q}_r^w = (\mathbf{f}^w)^{-1}(\mathbf{x}^o) \quad (9)$$

where $(\mathbf{f}^w)^{-1}$ means the inverse kinematics of the robot arm in the working side. Thereby, the tip position of the robot arm in the working side coincides with that in the operational side.

2.2.3 Working Side Control (Position Control)

In the working side control, the usual servo controller for industrial robot arms is adopted as a position control. The position control can realize the following motion of the position reference of the robot arm.

The dynamic equation of an industrial articulated robot arm in the working side including the servo controller is given by

$$\mathbf{H}^w(\mathbf{q}^w)\ddot{\mathbf{q}}^w + \mathbf{h}^w(\mathbf{q}^w, \dot{\mathbf{q}}^w) = \mathbf{K}_\tau^w [\mathbf{K}_v^w \{ \mathbf{K}_p^w (\mathbf{q}_r^w - \mathbf{q}^w) - \dot{\mathbf{q}}^w \}] \quad (10)$$

where $\mathbf{H}^w(\mathbf{q}^w)$ is the inertia matrix, $\mathbf{h}^w(\mathbf{q}^w, \dot{\mathbf{q}}^w)$ is the coupling nonlinear term, \mathbf{q}^w is the position of joint angle, \mathbf{K}_p^w , \mathbf{K}_v^w and \mathbf{K}_τ^w are position loop gain, velocity loop gain and torque constant for the robot in the working side, respectively.

2.3 Communication Procedure

The Internet technology is used for the communication of the teleoperation because the main advantage of the proposed teleoperation is the usage of the existing technology and the Internet is easily available for the communication channel of the teleoperation system. Concretely, the Socket communication via TCP/IP is applied for communication technique of the teleoperation system. Table 1 shows the data format of the communication. The transmit data from the operational side to the working side are the position reference of the robot arm in the working side and the received data of the operational side from the working side are the position output of the robot arm in the working side.

Figure 4 shows the time chart of the teleoperation system. The robot arms both of the operational side and of the working side are controlled by the real time tasks at the constant sampling interval. On the other hand, the real time property can not be fulfilled by the Socket communication via TCP/IP, then the communication must be operated by using the non real time task.

Transmit data	Time[s]	Position reference q_{r1}^{sw} [rad]	Position reference q_{r2}^w [rad]
Received data	Time[s]	Position output q_1^o [rad]	Position output q_2^o [rad]

Table 1. Data format of the communication between the operational side and the working side

	MK.3	SCARA
Resolution	8192	8000
Link2 gear ratio	160	1
Link3 gear ratio	160	1
Link2 Length[m]	0.25	0.3
Link3 Length[m]	0.215	0.3

Table 2. Schematic parameters of Performer MK3 and SCARA

Concerning about the communication, the position reference generated in the operational side is transmitted to the working side. After receiving of the position reference, the position reference is sent to the real time task of the robot arm control in the working side. Then, the robot arm in the working side is moved according to the received position reference. As a result, even if the time intervals between the successively received position references in the working side are varying, the teleoperation system works well.

The flow of the teleoperation system is explained as follows;

1. The start command is transmitted from the operational side to the working side through the Socket communication via TCP/IP.
2. In the operational side, the robot arm is controlled by the forcefree control at the constant sampling time interval.
3. In the operational side, the position request is sent to the real time task, then the position response of the robot arm in the operational side is received.
4. In the operational side, the position reference of the working side is calculated from the position response of the robot arm in the operational side.
5. The position reference of the working side is transmitted from the operational side to the working side through the Socket communication via TCP/IP.
6. The position output of the working side is transmitted from the working side to the operational side through the Socket communication via TCP/IP.
7. In the working side, the received position reference is sent to the real time task of the robot arm control and the robot arm is controlled at the constant sampling time interval, then the position response of the robot arm in the operational side is received.

3. Validation of the Proposed Teleoperation System

3.1 Experimental Condition

In order to assure the effectiveness of the proposed teleoperation system, an experimental study was carried out using actual robot arms connected with LAN. Figure 5 shows the experimental setup. In order to conform that the proposed teleoperation system is applicable to various types of the robot arms, two different types of the articulated robot arms were used for experiments. One was a vertical articulated robot arm, Performer MK3 (Yahata Electric Machinery Mfg. Co. Ltd.) and another was a SCARA (Selective Compliant Articulated Robot

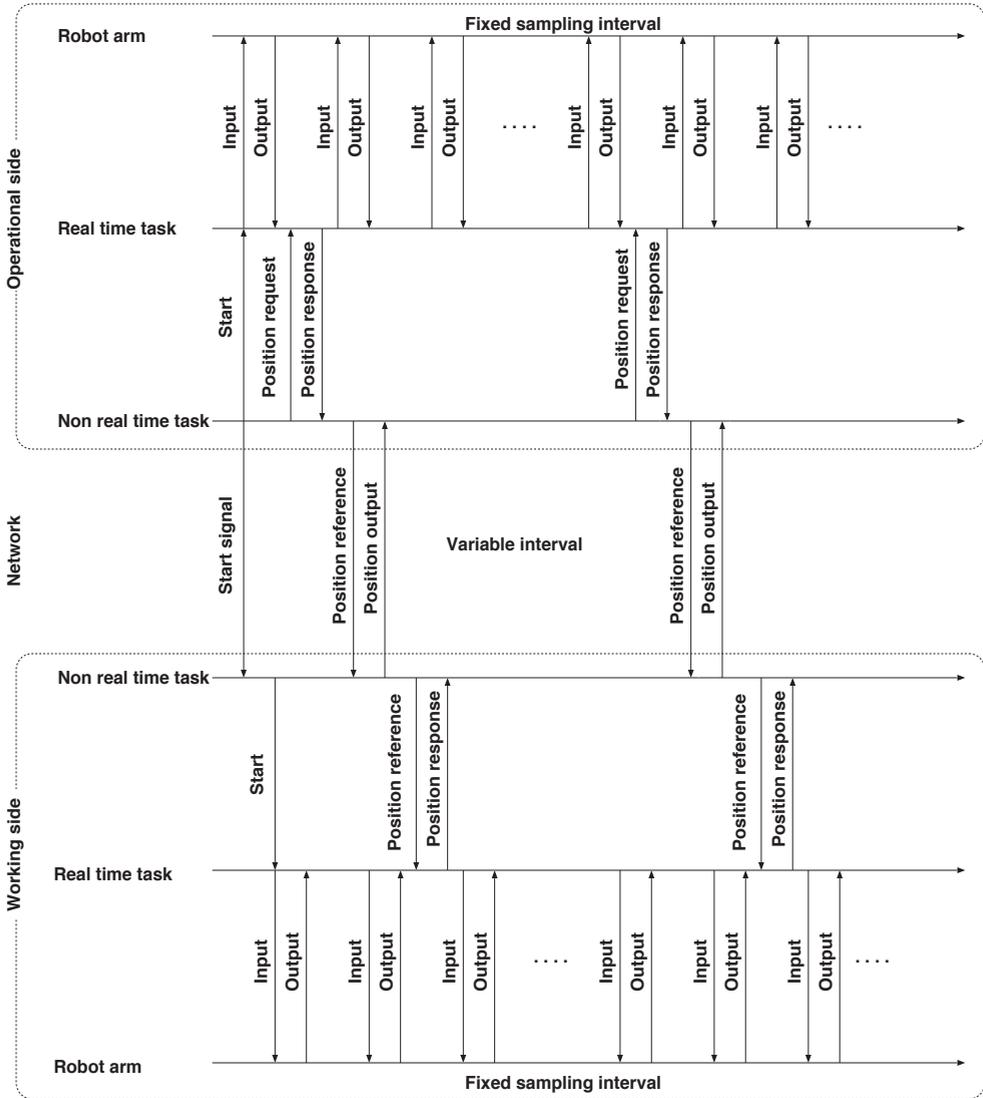


Fig. 4. Time chart of the teleoperation system

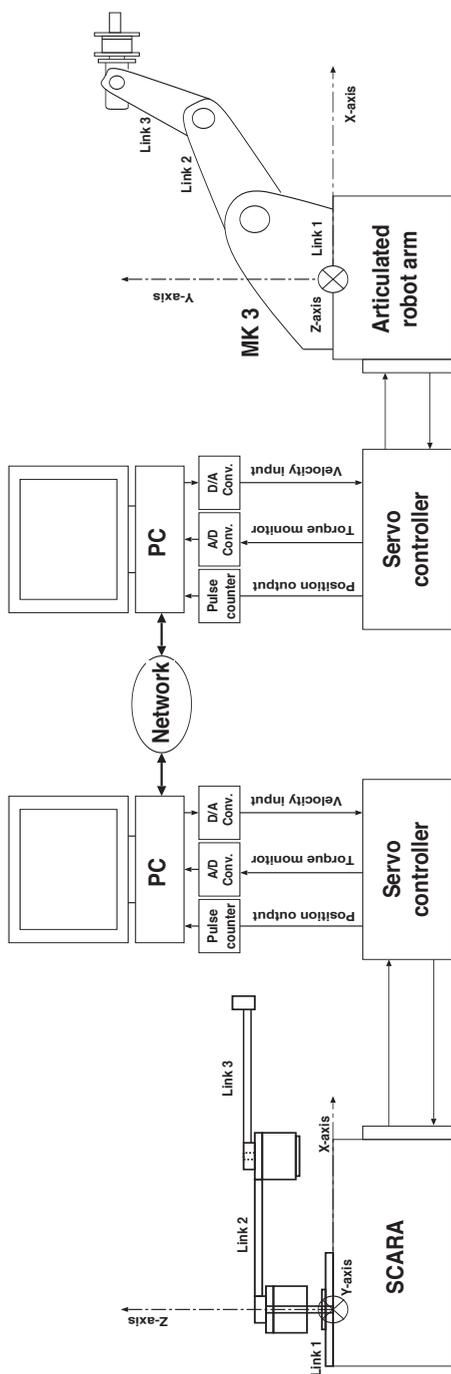


Fig. 5. Experimental setup

Arm). The schematic parameters of these robots are shown in Table 2. The position loop gain was given as $K_p = 25$ [1/s] and the velocity loop gain was given as $K_v = 150$ [1/s] for Performer MK3 and the position loop gain was given as $K_p = 2$ [1/s] and the velocity loop gain was given as $K_v = 120$ [1/s] for SCARA. The sampling interval of the real time task for the robot arm control was 4 [ms], and the time interval of the position reference generation in the non real time task of the operational side was approximately 50 [ms]. Two links of the link2 and the link3 were used both for Performer MK2 and for SCARA. The robot arm in the operational side was moved passively according to the external force applied by a human hand.

3.2 Experimental Result by Using Actual Industrial Robot Arms

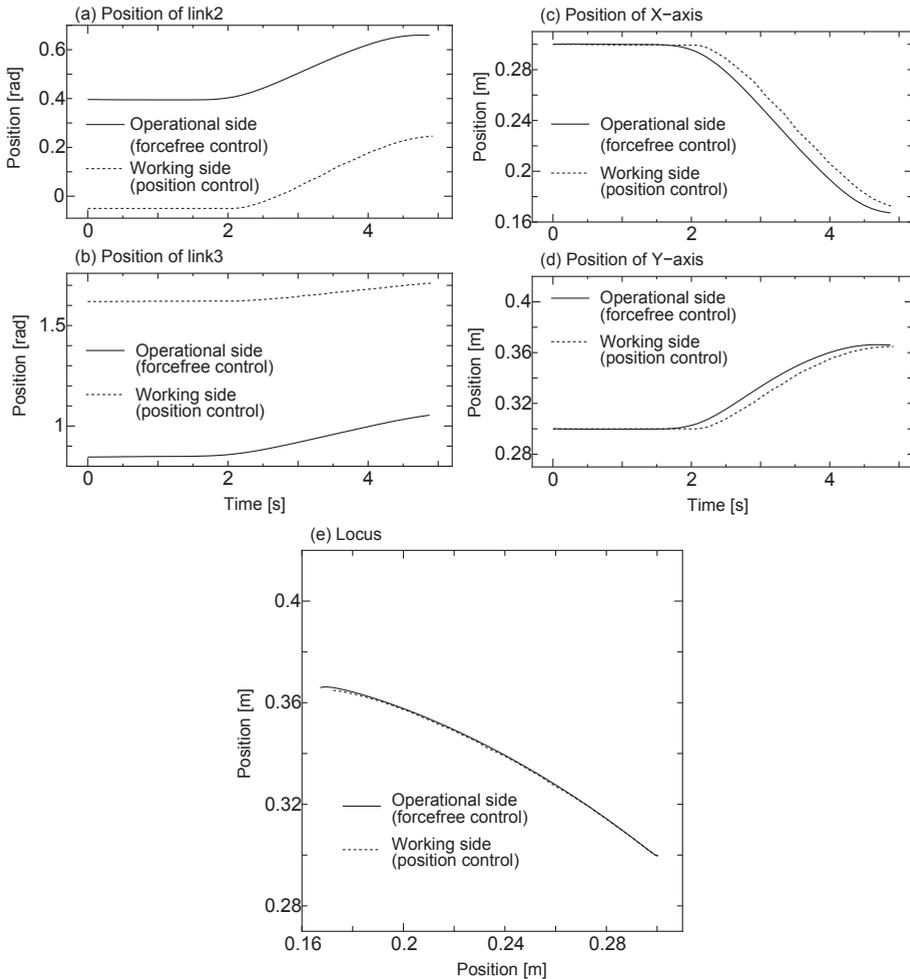


Fig. 6. Experimental result of the teleoperation when the robot arm in the operational side was Performer MK3 and that in the working side was SCARA

First, Performer MK3 was used as the robot arm in the operational side and SCARA was used as the robot arm in the working side. Experimental result is shown in Fig. 6 (a) the time trajectory of the joint position of link2, (b) the time trajectory of the joint position of link3, (c) the time trajectory of the tip position of X-axis, (d) the time trajectory of the tip position of Y-axis and (e) the tip position locus. As shown in Fig. 6 (a) and (b), the joint position in the working side is different from that in the operational side. This is caused by the difference of the mechanism between the working side and the operational side. The tip position in the working side, however, is almost the same as that in the operational side as shown in Fig. 6 (c) and (d) because of the appropriate mechanism compensation. The communication delay was negligible small because LAN was used for the communication channel. The delay about 200[ms] of the working side from the operational side was caused by the dynamics of the robot arm in the working side. As shown in Fig. 6, the robot arm in the working side follows the motion of that of the operational side. The result shows that the teleoperation system by using the forcefree control can be achieved.

3.3 Experimental Result of Alternation of Operational Side and Working Side

Next, the roles of the two robot arms were alternated. SCARA was used as the robot arm in the operational side and Performer MK3 was used as the robot arm in the working side. Experimental result is shown in Fig. 7 (a) the time trajectory of the joint position of link2, (b) the time trajectory of the joint position of link3, (c) the time trajectory of the tip position of X-axis, (d) the time trajectory of the tip position of Y-axis and (e) the tip position locus. As shown in Fig. 7, the robot arm in the working side followed the motion of that of the operational side. The delay about 16[ms] of the working side from the operational side was caused by the dynamics of the robot arm in the working side. The result shows that the teleoperation system by using the forcefree control can be achieved when the operational side and the working side are alternated.

4. Discussion

4.1 Handleability

The proposed teleoperation system can realize the teleoperation as if the operator were in the working side. In the proposed teleoperation system, any types of the industrial robot arms are applicable both for the operational side and for the working side. The experimental study showed that both of the vertical articulated robot arm and SCARA can be applied to both of the operational side and of the working side in the proposed teleoperation system. The servo controller of the industrial robot arm is without change and the additional software of the forcefree control and communication program is enough for the realization of the teleoperation system. The advantage brings flexible teleoperation system construction by use of the appropriate mechanism selection both for the operational side and for the working side.

4.2 Effects on Communication Delay and Data Loss

The Internet technology is used for the proposed teleoperation system. The Socket communication via TCP/IP may include communication delay and data losses. With respect to the communication delay, the influence may appear as the delay of the robot arm motion in the working side from the motion in the operational side because the position reference generated in the operational side is transmitted to the working side, and the robot arm in the work side is moved according to the received position reference with communication delay. With respect to the data loss, the influence may appear as an awkward robot arm motion in the working

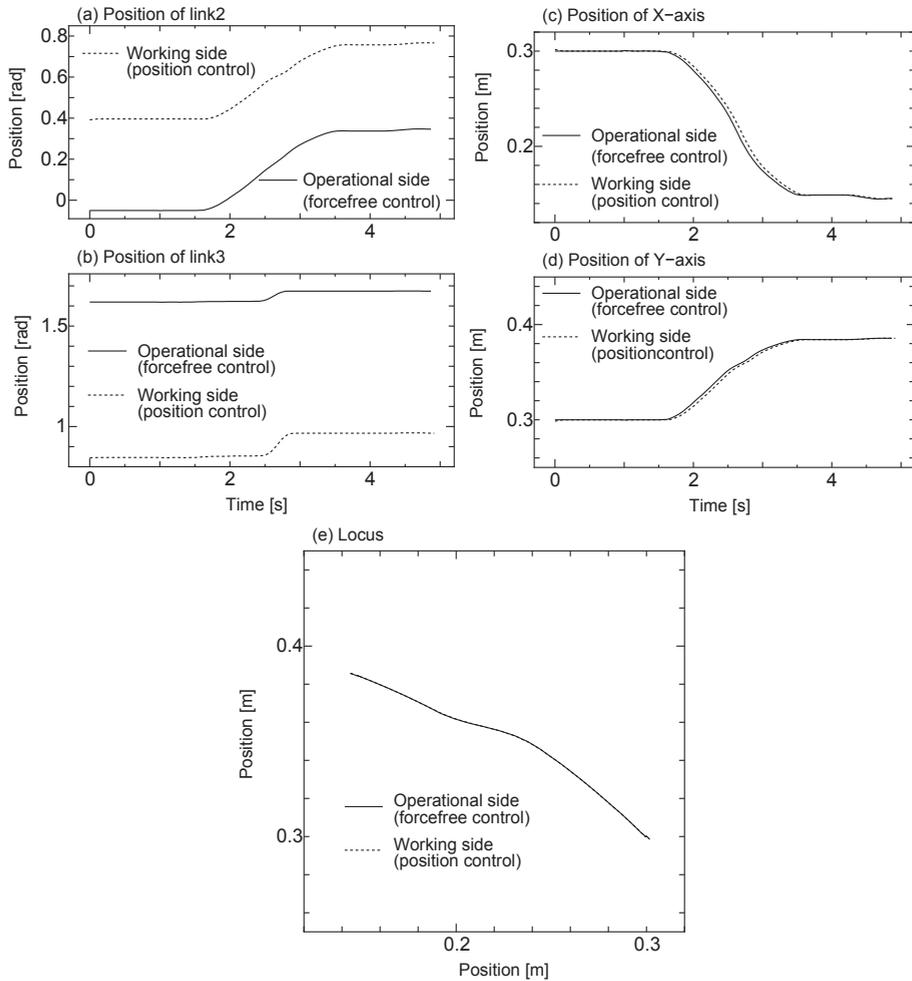


Fig. 7. Experimental result of the teleoperation when the robot arm in the operational side was SCARA and that in the working side was Performer MK3

side because the reference position corresponding to the lost data is vanished. However, the teleoperation system will not become unstable caused by the communication delay or the data loss because the information from the working side is fed back to the operational side.

5. Conclusion

The teleoperation system of the robot arm by using the forcefree control and the position control was proposed. The Internet technology was applied to the communication channel of the teleoperation system. In the proposed teleoperation system, the existing robot arms can be used both for the operational side and for the working side. The experimental results show the effectiveness of the proposed teleoperation system. In the future, further teleoperation system for industrial robot arms considering position, force and visual feedback will be investigated.

6. References

- Aziminejad, A.; Tavakoli, M.; Patel, R.V. & Moallem, M. (2008). Stability and performance in delayed bilateral teleoperation: Theory and experiments, *Control Engineering Practice*, Vol. 16, 1329-1343.
- Bambang, R. T. (2007). Development of Architectures for Internet Telerobotics Systems, *Journal of Bionic Engineering*, Vol. 4, 291-197.
- Challacombe, B. & Dasgupta, P. (2003). Telemedicine- the future of surgery, *The Journal of Surgery*, Vol. 1, No. 1, 15-17.
- Chong, N. Y.; Kotoku, T.; Ohba, K.; Komoriya, K.; Tanie, K.; Oaki, J.; Hashimoto, H.; Ozaki, F.; Maeda, K. & Matsuhira, N. (2003). A collaborative multi-site teleoperation over an ISDN, *Mechatronics*, Vol. 13, 957-979.
- Geeter, J. D.; Decrrton, M. & Colon, E. (1999). The challenges of telerobotics in a nuclear environment, *Robotics and Autonomous Systems*, Vol. 28, 5-17.
- Goto, S. (2007). Forcefree control for flexible motion of industrial articulated robot arm, *Industrial Robotics: Theory, Modeling and Control*, Advanced Robotic Systems International, Chapter 30, 813-840, pro literatur Verlag
- Goto, S.; Usui, T.; Kyura, N. & Nakamura, M. (2007). Forcefree control with independent compensation for industrial articulated robot arms, *Control Engineering Practice*, Vol. 15, No. 6, 627-638.
- Hokayem, F. P & Spong, M. W. (2006). Bilateral teleoperation: An historical survey, *Automatica*, Vol. 42, 2035-2057.
- Hung, N.V.Q.; Narikiyo, T. & Tuan, H.D. (2003). Nonlinear adaptive control of master-slave system in teleoperation, *Control Engineering Practice*, Vol. 11, 1-10.
- Kushida, D.; Nakamura, M.; Goto, S. & Kyura, N. (2001). Human direct teaching of industrial articulated robot arms based on forcefree control, *Artificial Life and Robotics*, Vol. 5, 26-32.
- Kushida, D.; Nakamura, M.; Goto, S. & Kyura, N. (2003). Flexible motion realized by force-free control: Pull-out work by an articulated robot arm, *International Journal of Control, Automation, and Systems*, Vol. 1, No. 4, 464-473.
- Kyura, N. (1996). The development of a controller for mechatronics equipment, *IEEE Trans. on Industrial Electronics*, Vol. 43, 30-37.
- Marohn, C. M. R. & Hanly, C. E. J. (2004). Twenty-first century surgery using twenty- first century technology: Surgical robotics, *Current Surgery*, Vol. 61, No. 5, 466-473.

- Mora, A. & Barrientos, A. (2007). An experimental study about the effect of interactions among functional factors in performance of telemanipulation systems, *Control Engineering Practice*, Vol. 15, 29-41.
- Nakamura, M.; Goto, S. & Kyura, N. (2004). *Mechatronic Servo System Control*, Springer-Verlag Berlin Heidelberg.
- Nickels, K.; DiCicco, M.; Bajracharya, M. & Backes, P. (2009) Vision guided manipulation for planetary robotics — Position control, *Robotics and Autonomous Systems*, 10.1016/j.robot.2009.07.029.
- Park, B. J.; Flores, R. M. & Rusch, V. W. (2006). Robotic assistance for video-assisted thoracic surgical lobectomy: Technique and initial results, *The Journal of Thoracic and Cardiovascular Surgery*, Vol. 131, No. 1, 54-49.
- Rogers, J. R. (2009). Low-cost teleoperable robotic arm, *Mechatronics*, Vol. 19, 774-779.
- Sanfeliu, A.; Hagita, N. & Saffiottid, A. (2008). Network robot systems, *Robotics and Autonomous Systems*, Vol. 56, 793-797.
- Sheridan, T. B. (1995). Teleoperation, telerobotics and telepresence: A progress report, *Control Engineering Practice*, Vol. 3, No. 2, 205-214.
- Slawiński, E.; Postigo, J. & Mut, V. (2007). Bilateral teleoperation through the Internet, *Robotics and Autonomous Systems*, Vol. 55, 205-215.
- Smith, C.; Bratt, M. & Christensen, H. I. (2008). Teleoperation for a ball-catching task with significant dynamics, *Neural Networks*, Vol. 21, 604-620.
- Stassen, H.G. & Smets, G. J. F. (1997). Telemanipulation and telepresence, *Control Engineering Practice*, Vol. 5, No. 3, 363-374.
- You, S.; Wang, T.; Eagleson, R.; Meng, C. & Zhang, Q. (2001). A low-cost internet-based telerobotic system for access to remote laboratories, *Artificial Intelligence in Engineering*, Vol. 25, 265-279.

Trajectory Generation for Mobile Manipulators

Foudil Abdessemed and Salima Djebrani
*Batna University, Department of Electronics,
Algeria*

1. Introduction

Mobile robot navigation has stood as an open and challenging problem over decades. Despite the number of significant results obtained in this field, people still look for better solutions. Some mobile robots are subject to constraints of rolling without slipping and thus belong to non-holonomic systems. Mobile robots also are subject to navigate in environments cluttered with obstacles. Now, in case the mobile robot presents a non-holonomic constraint, the problem consists of finding a path taking into account constraints imposed both by the obstacles and the non-holonomic constraints. Since non-holonomy make path planning more difficult, many techniques have been proposed to plan and generate paths. May be the most popular is the method of potential field (Khatib, 1986). However, this method may present some problems such as sticking to local minima. Moreover, the kinematic constraint is the other problem that can face trajectory planning. This can make time derivatives of some configuration variables non-integrable and hence, a collision free path in the configuration space not achievable by steering control.. Some researchers worked to find feasible path using different methodologies (Sundar & Shiller, 1997), (Laumond et al, 1994), (Reeds & Shepp, 1990). To deal with obstacles, some researchers decomposed the dynamic motion to static paths and velocity-planning problem (Murray et al, 1994), (Tilbury et al, 1995). In the work of (Qu et al, 2004), the authors treated the problem as a family of curves where the optimal path is found by adjusting a certain polynomial parameter. This idea was raised in many references including (Kant & Zucker, 1988) and (Murray & Sastry,1993), where trajectories are represented by sinusoidal, polynomial or piecewise constant functions. In our recent work, and based on intelligent control, we proposed a fuzzy control methodology to navigate a mobile robot in a cluttered environment with the aim to reach the goal while avoiding static and/or dynamic obstacles (Abdessemed et al, 2004). Concerning robot arm path planning and trajectory generation, considerable efforts have been devoted to make these mechanical systems succeeding in their tasks. If we consider a robot arm with n-joints that move independently, the robot's configuration can be described by a 3-dimentional coordinate: (x_e, y_e, z_e) for the location of the end effector. These coordinates characterize the workspace representation, since they represent exactly the same coordinates of the object it intends to manipulate or to avoid. Although the workspace is well suited for collision avoidance, it happens that we are still

facing some other problems. In fact the task is usually expressed in the workspace coordinates and the question is how to map this space into configuration space. This problem finds its solution in the inverse kinematics. However, calculating the inverse kinematics is hard, and the problem becomes much harder if the robot has many DOFs. Moreover, for a particular workspace coordinates, many distinct configurations are possible. The other problem that may arise when using configuration space is the presence of obstacles. Within the scope of all these problems, many methods to path planning emerged. Among the techniques found in the literature, we state the cell decomposition methods (Russel & Norvig, 1995), Skeletonization methods (e.g. Voronoi graph (Okabe et al, 2000)), potential field. Due to some problems of the applied techniques already mentioned, such as local minima and uncertainties, probabilistic and robust methods have been explored. Demonstration of robustness, fuzzy logic and genetic algorithm encouraged many researchers to use these concepts for path planning and obstacle avoidance. We note that robust methods assume a bounded amount of uncertainties and do not assign any probabilities to values. The robots could replace good number of specific machine tools and could continue to supplant the man in a lot of complex tasks. In spite of all, the robot manipulators on fixed base have a lot of limitations. This is why like a man who has the faculty to move to achieve some tasks at different places or to do continuous tasks requiring a work of the arm during the displacement, the mobile manipulators have been considered. Mobile manipulators received particular attentions these last decades (Zhao et al, 1994); (Pin & Culioli, 1992); (Pin et al, 1996); (Lee & Cho, 1997); (Seraji, 1995). This is mainly due to their analytic problems and their various applications. A mobile manipulator consists of an arm fixed on a mobile platform. Such a configuration leads us toward a kinematically redundant system. Although we can construct non redundant mobile manipulators but, there are some good reasons to make us thinking of these systems as for example: increasing the working space of the arm, avoiding static or dynamic obstacles or, to avoid the robot singularities. From these observations, which allow to increase the working space by the mobility of the platform that a number of applications have been appeared. When these systems are devoted to indoor tasks, they are often equipped with wheels. The arrangement of the wheels and their actuation device determine the holonomic or non-holonomic nature of its locomotion system (Campion et al, 1996). Whereas some wheeled mobile manipulators built from an omni-directional platform are holonomic. Extensive research efforts have led people to plan the collision free paths for mobile manipulators. Many techniques have been proposed for path planning and trajectory generation. A desired task is usually specified in the work space. The first type has been a subject of our previous work [Abdessemed *et al*] where as the second type considers the plat form to be holonomic [Djebrani *et al*]. However, trajectory following control is easily performed in the joint space. Therefore, it is essential to obtain the desired joint space trajectory given the desired Cartesian space trajectory. This is accomplished using the inverse kinematics transformation. Controlling such systems is hard and the problem is more difficult to solve in case where the mobile platform contains some non-holonomic kinematical constraints. However, one should know that such a constraint does not decrease the configuration space reachable by the mobile platform but decreases the velocity space. Therefore, the mobile platform moves only along trajectories having a certain shape. On the other hand, if we choose a holonomic platform, the control would be much easier. In fact, A holonomic platform robot is an omni-directional robot whose mechanical structure enables it to change its displacement at any direction, without waiting

for the reconfiguration of its rolling parts. One of the consequences of the omnidirectionality is that the orientation of a robot becomes independent of the trajectory performed, provided that each "wheel" of the robot has the 3 degrees of mobility (2 translations and 1 rotation). In this book chapter, we try to see the two case studies, namely a holonomic and a non-holonomic mobile manipulator. The approach presented describes the development of the complete kinematic representation of a mobile manipulator. In this case, we present the analysis of the whole mechanical system constituted of a mobile platform over which a robot manipulator is mounted, forming thus the mobile manipulator (Fig. 1); the arm and the mobile platform are considered as a unique system. In this part of analysis, the mobile manipulator is considered as a unique entity. In order to have an overall study, we consider the two types of mobile platforms. The mobility introduced by the mobile platform is exploited to solve problems like collision avoidance and joints saturations. The results obtained demonstrate the effectiveness of the approaches for simple situations as well as for complex situations when obstacles are encountered.

2. Mobile manipulator with non-holonomic platform

2.1 Analysis

In this case a mobile manipulator with non-holonomic platform is viewed. Figure. 2 shows the four main reference axes: The stationary reference axis, the reference axis attached to the mobile platform, the reference axis attached to the base of the robot manipulator, and finally the reference axis attached to the end point of the effector. The homogeneous matrix found by a successive multiplication of the three homogeneous sub matrices can obtain the position and the orientation of the terminal point of the end effector with respect to the stationary reference axis:

$$T_e^o = T_p^o \cdot T_m^p \cdot T_e^m \quad | \quad (1)$$

Such that the matrix T_p^o is determined by a certain matrix $A(q)$, T_m^p is a fixed matrix and

T_e^m is determined by the joint variable vector $\theta = [\theta_1, \theta_2, \dots, \theta_{n_m}]^T$, n_m represents the degree of freedom of the arm manipulator.

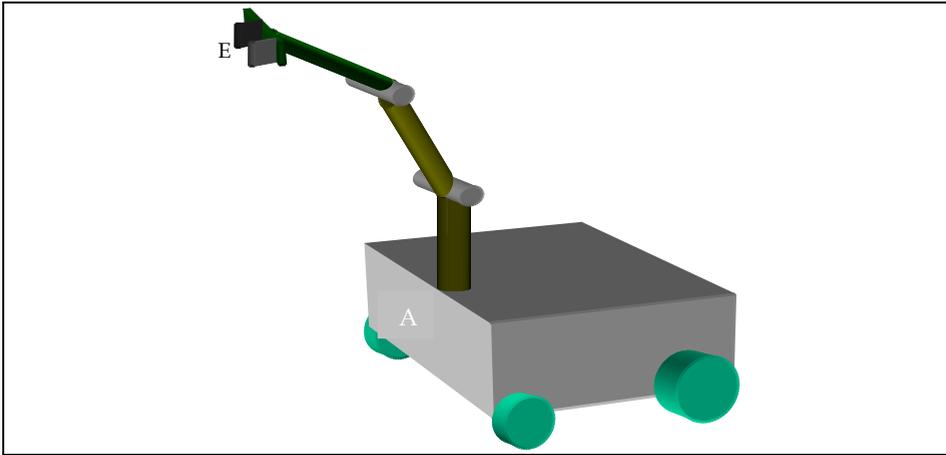


Fig. 1. Mobile Manipulator appearance

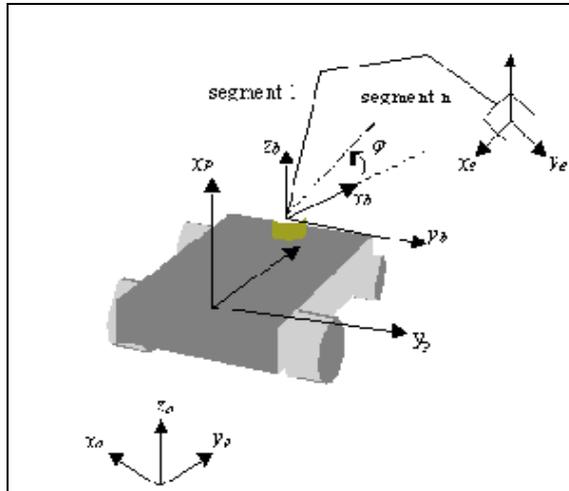


Fig. 2. Mobile Manipulator features

The equations of the geometric model are found to be:

$$\begin{aligned}
 x_e &= x_A + \cos(\theta_1 + \varphi)(l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3)) \\
 y_e &= y_A + \sin(\theta_1 + \varphi)(l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3)) \\
 z_e &= l_1 - l_2 \sin\theta_2 - l_3 \sin(\theta_2 + \theta_3)
 \end{aligned}
 \tag{2}$$

(x_A, y_A, φ) represent the mobile platform coordinates and its orientation in the world frame. $(\theta_1, \theta_2, \theta_3)$ are the three angles of the arm, and l_1, l_2 and l_3 its lengths. (x_e, y_e, z_e) are the coordinates of the end effector in the world frame. As we can see, the position vector

$X_e = (x_e, y_e, z_e)$ of the end effector E with respect to the world coordinate W is a non-linear function of the configuration vector: $q = (p^T, \theta^T)^T \in \mathfrak{R}^n$, ($n=3+n_m$). Now, if the vector X_e^d is the vector of the wanted task then

$$X_e = X_e^d = f(q) \quad (3)$$

If we derive this equation, we get the kinematic equation of the model

$$\dot{X}_e = J_m(q) \cdot \dot{q} \quad (4)$$

where $J_m(q)$ is the $m \times n$ Jacobienne matrix of the mobile manipulator. This equation represents a holonomic kinematic constraint since it can be written as

$$H(q) = 0 \quad (5)$$

If the mobile platform is non holonomic and without slip, then the following kinematic constraint is true:

$$A(p) \cdot \dot{p} = 0 \quad (6)$$

such that :

$$\dot{p} = \begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\phi} \end{bmatrix}^T = \begin{bmatrix} \dot{X}_A \\ \dot{\phi} \end{bmatrix}^T \quad (7)$$

Equivalently, we can write the non-holonomic constraint (5) as:

$$J_v(q) \dot{q} = 0 \quad (8)$$

where: $J_v(q) = [A(p) \quad ; \quad 0]$

\dot{q} cannot be eliminated by integration to give $\frac{d(J_v(q))}{dq} = 0$, this means that the system is non holonomic. Equations (4) and (8) are combined to give the differential kinematic model of the system including the mobile platform and the robot manipulator [1].

$$\begin{bmatrix} J_v(q) \\ \dots \\ J_m(q) \end{bmatrix} \dot{q} = \begin{bmatrix} 0 \\ \dots \\ \dot{X}_e \end{bmatrix} \quad (9)$$

Such that, $q = (p^T, \theta^T) = (x_A, y_A, \phi, \theta_1, \theta_2, \theta_3)^T$. In a compact form, equation (9) can be written as:

$$J(q) \dot{q} = \dot{X} \quad (10)$$

where:

$$J(q) = \begin{bmatrix} J_v(q) \\ \dots \\ J_m(q) \end{bmatrix}, \quad \dot{X} = [0 \quad \dots \quad \dot{X}_e]^T$$

If we assume that the speed of the end-effector is \dot{X}_d , we need then to solve the following differential equation:

$$J(q)\dot{q} = \dot{X}_d \quad (11)$$

The term redundancy designates the determination of admissible control signal for this redundant system. However, the system being undetermined, it is necessary to use some criteria that allow determining one of the infinite solutions of the problem. according to (Liégeois, 1997), the general formulation of the inverse kinematics is expressed by the following equation

$$\dot{q} = J^\# \dot{X}_d + (I - J^\# J)z \quad (12)$$

Where, $J^\#$ is the generalized inverse of the Jacobian matrix. The first term of the equation (12) represents the particular solution used to achieve the desired velocity of the end effector, and the term $(I - J^\# J)$ is an operator of projection that projects an arbitrary vector into the null space of J . Therefore, the term $(I - J^\# J)z$ is the general solution of the homogeneous equation:

$$J\dot{q} = 0 \quad (13)$$

The homogeneous solution contributes only in a motion within the joint space of the mechanical system named the self-motion. In order to find the optimal solutions, let us mention the techniques most commonly used for serial-chain redundant arms:

- Minimization of joint velocities

$$C_v = \dot{q}^T \dot{q} \quad (14)$$

- Minimization of joint acceleration

$$C_a = \ddot{q}^T \ddot{q} \quad (15)$$

- Minimization of kinetic energy

$$C_e = \dot{q}^T H \dot{q} \quad (16)$$

- Minimization of joint torques

$$C_t = \tau^T \tau \quad (17)$$

Some of these techniques can present some problems as: nonzero joint velocities corresponding to zero end-effector velocities, and instability of the motion. Otherwise, if the mobile manipulator is brought to evolve in an environment cluttered of obstacles, then the goal consists to find the solutions to the problem that must take them into consideration, consequently, one can write the relation as a certain function f such that, $\dot{q} = f(q, \dot{x}, \text{obstacles})$

2.2 Geometric solution with no obstacles

For this redundant system the matrix J is of dimension $m \times n$, with $m < n$. We seek a solution to equation (12), which guarantees a minimum value for the norm, in addition to a number of solutions in the null space of J . The solutions in the null space can be used to optimize some tasks as for example: to avoid some obstacles or to warn saturations of the joints. However, it is not recommended in practice to use directly the solution with minimal norm. Indeed, to avoid big velocities values, one can impose a weighting matrix

$W_v = \text{diag}\{w_a, w_b\}$ for the linear and angular velocities, in order to minimize the sum of their norms: $\| \dot{p} \|_{w_a} + \| \dot{\theta} \|_{w_b}$.

such that:

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & w_n \end{bmatrix}, \quad \text{where :} \quad w_i = \frac{b_i}{b_{\max}}, \quad \text{such that:}$$

$$b_i = (q_i)_{\max} - (q_i)_{\min} \quad \text{and} \quad b_{\max} = \max\{b_1, b_2, \dots, b_n\}$$

Thus, the optimal solution is the one that optimises the norm defined by the following expression:

$$\dot{q}^T W \dot{q} \tag{18}$$

While noticing that the matrix W is diagonal, it could be split into two diagonal matrices such that:

$$W = W_o^T W_o \tag{19}$$

In this case the norm could be written as :

$$\dot{q}^T W \dot{q} = \dot{q}^T W_o^T W_o \dot{q} = \|W_o \dot{q}\|^2 \tag{20}$$

and relation (11) becomes:

$$\dot{X}_d = JW_0^{-1}W_0\dot{q} \quad (21)$$

To find \dot{q} , one should solve equation (21), which minimizes the norm (18). The problem becomes an optimization problem, where we consider JW_0^{-1} as a system matrix and the product $W_0\dot{q}$ as the vector whose norm is to be minimized; the solution is thus:

$$\dot{q} = W^{-1}J^T \left(JW^{-1}J^T \right)^{-1} \dot{X}_d + Hz \quad (22)$$

knowing that:

$$H = W_0^{-1} - W^{-1}J^T \left(JW^{-1}J^T \right)^{-1} JW_0 \quad (23)$$

The first term of the equation (22), represent the optimal solution and the second term the homogeneous solution. The vector z is an arbitrary vector that is projected by the matrix H in the null space of J . It can be used to prevent saturation of the manipulator joints or to avoid unforeseen obstacles. To correct any drift of the trajectory of the space of the task, we introduce the error that measures the difference between the wanted vector and the one measured to yield:

$$\dot{q} = W^{-1}J^T \left(JW^{-1}J^T \right)^{-1} \left[\dot{X}_d + K(X_d - X) \right] + Hz \quad (24)$$

However, if we try to solve the equation (24), one can fall on a problem of numeric instability. To overcome this problem, we propose to use the singular value decomposition as a solution. This algorithm is a stable numerical procedure based on the decomposition theorem.

Theorem

Given a matrix A of size $m \times n$, it can be written as a product of three matrices as:

$$A_{m \times n} = U_{m \times m} \cdot \Sigma_{m \times n} \cdot V_{n \times m}^T$$

The matrices U et V are orthogonal matrices and Σ is a diagonal matrix for which the elements on the diagonal are the singular values of the A matrix.

If one applies this theorem to the matrix:

$$A = JW_0^{-1} \quad (25)$$

Then

$$JW_0^{-1} = U\Sigma\Sigma^T \quad (26)$$

In this case, the complete solution becomes:

$$\dot{q} = W_0^{-1} V \Sigma^+ U^T [\dot{X}_d + K(X_d - X)] + W_0^{-1} [I - V \Sigma^+ \Sigma V^T] z \quad (27)$$

Where $\Sigma^+ = \Sigma^T (\Sigma \Sigma^T)^{-1}$. The complete solution makes intervene also the matrix H representing the projection of an arbitrary vector z in the null space of the Jacobian matrix J.

2.3 Geometric solution with obstacle avoidance

Now, if we consider a smooth function $g(q)$, representing a certain criterion to be minimized, then the vector z of the general solution given by equation (22) can be defined as follows:

$$z = -\nabla g(q) \quad (28)$$

where $\nabla g(q) = \left(\frac{\partial g}{\partial q} \right)^T$ is the gradient of g, and the homogeneous solution is obtained by projecting z in the null space of J. However, we can use any function as far as it can be reduced to an expression that involves only terms of generalized joint variables. The method of the gradient is of a very vast use because it allows an easy incorporation of the different performance criteria in the control algorithm. This technique is used in our case to satisfy two objectives; first avoiding obstacles and second avoiding joint limits. Thus, the vector z is composed of two terms:

$$z = z_1 + z_2 \quad (29)$$

where $z_1 = -\beta \frac{\partial V_1}{\partial q}$ and $z_2 = -\beta_2 \frac{\partial V_2}{\partial q}$. V_1 is the potential associated with the joint limits such that: and V_2 is the potential associated with the obstacle presence.

$$V_1(q) = \frac{1}{2} \left(q - \frac{\bar{q} + \underline{q}}{2} \right)^T \left(q - \frac{\bar{q} + \underline{q}}{2} \right) \quad (30)$$

\bar{q}_1 and \underline{q}_1 represent respectively the upper and lower joint limits. Whereas V_2 is the potential associated with the obstacle presence such that:

$$V_2 = \sum_{i=1}^n V_{2i} \quad (31)$$

Such that

$$V_{2i} = \begin{cases} \frac{1}{2}k_1 \left(\frac{1}{d_1(x)} - \frac{1}{d_0} \right)^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases} \quad (32)$$

k_1 denotes some adjusting coefficient, d and d_0 represent respectively the distance of a certain point on the robot to the obstacle and the minimal security distance.

2.4 Simulation results

A series of simulation were conducted in order to illustrate the performance of the method. Fig. 3, shows the terminal point of the end-effector following a reference trajectory in a free obstacle environment in the case where we consider the whole system as unique. In this case, one notes, according to Fig. 4, that the mobile platform follows a non-deformable trajectory in the x-y plane. The curves presented in Figs. 5, 6, 7 and 8 shows the evolution of the corresponding mobile manipulator joints. However, if an obstacle is put on the path of the mobile platform then, one can see that the mobile platform succeeds in getting around the obstacle while maintaining the terminal point on the reference trajectory; this is depicted in Fig. 9, and the new trajectory of the mobile platform in the x-y plane is clearly shown in Fig. 10. The curves presented in Figs. 11, 12, 13 and 14 show the evolution of the new corresponding mobile manipulator joints.

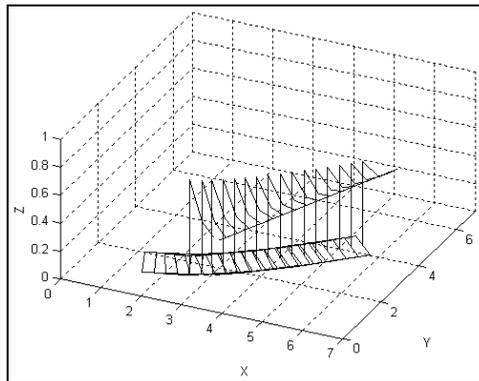


Fig.3. A 3D-view of the arm and the mobile platform evolutions in an obstacle free space.

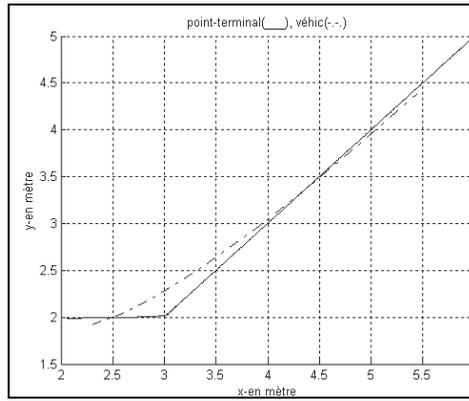


Fig. 4. End-effector and mobile platform trajectories in the x-y plane with no obstacles.

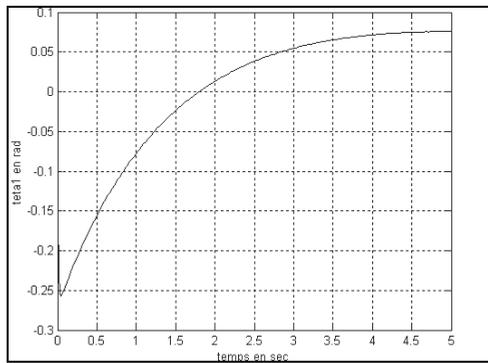


Fig. 5. Articulation θ_1 curve

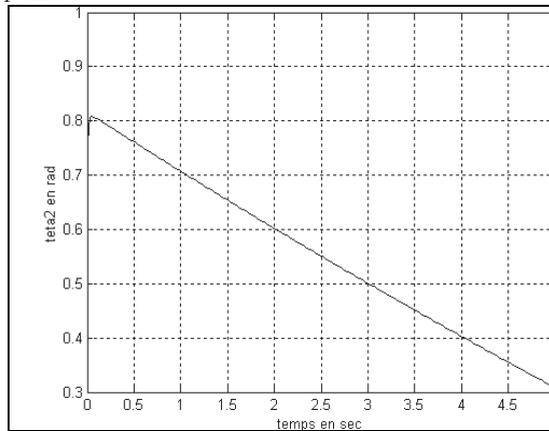


Fig.6. Articulation θ_2 curve

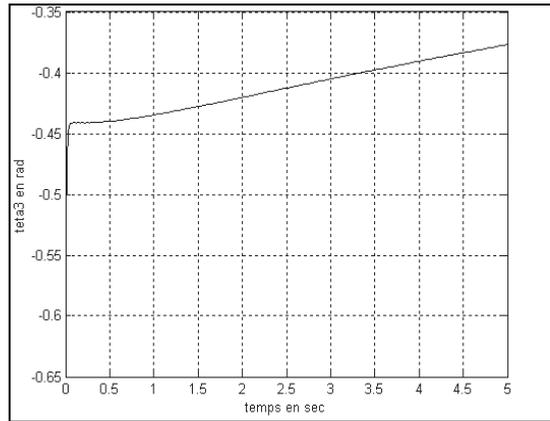


Fig. 7. Articulation θ_3 curve

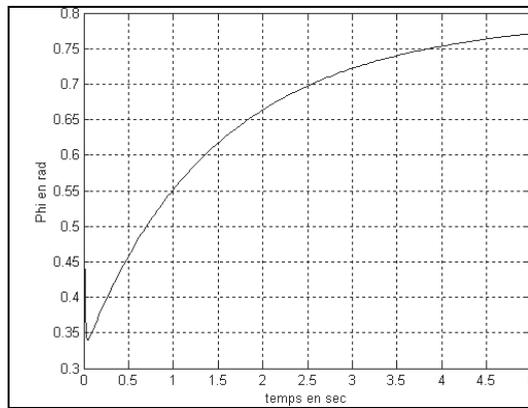


Fig. 8. Articulation ϕ curve

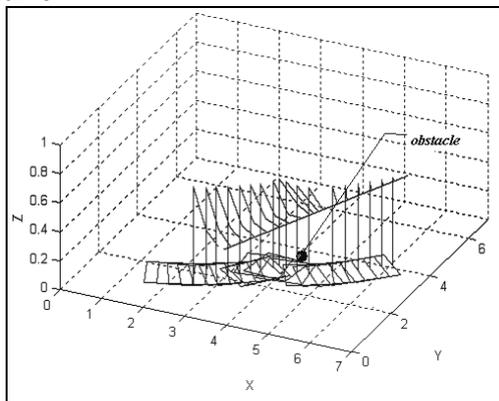


Fig. 9. A 3D-View of the arm and the platform evolutions in presence of obstacles.

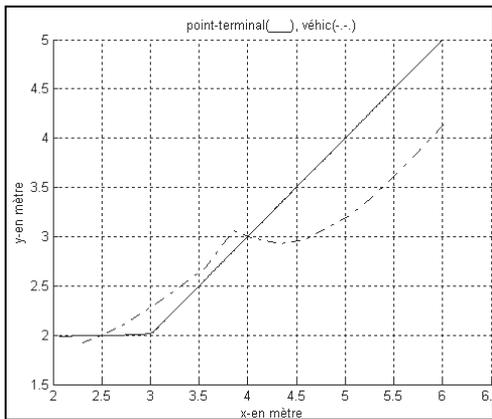


Fig. 10. End-effector and mobile platform trajectories in the x-y plane in presence of obstacles.

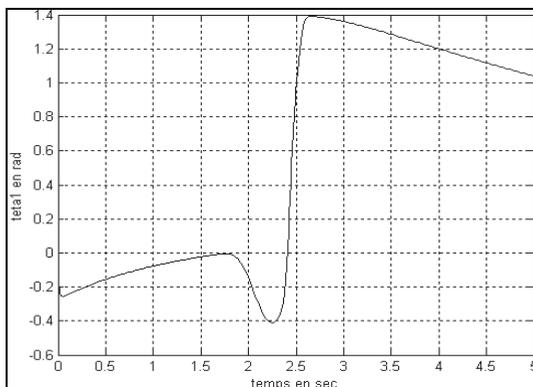


Fig. 11. Evolution curve of the joint θ_1

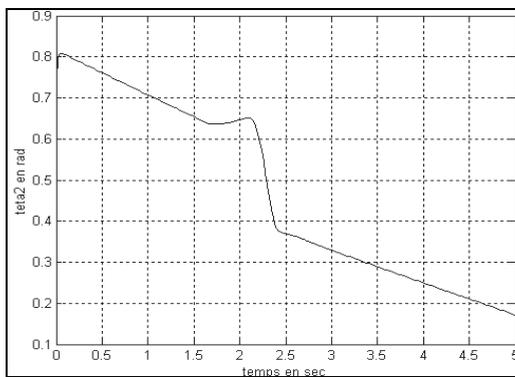


Fig. 12. Evolution curve of the joint θ_2

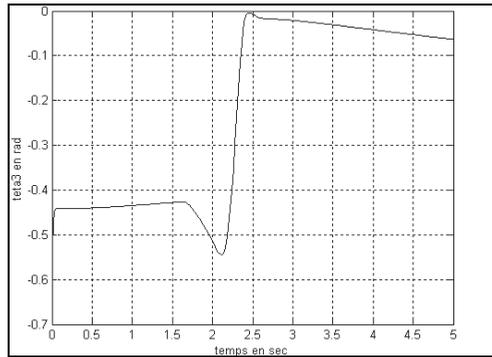


Fig. 13. Evolution curve of the joint θ_3

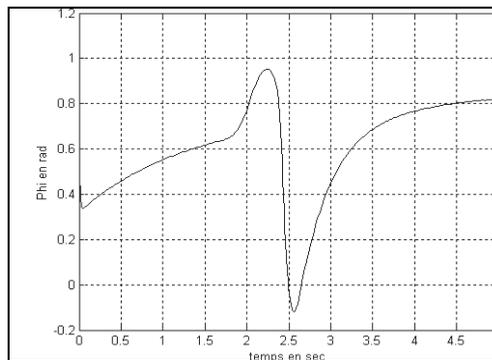


Fig. 14. Evolution curve of the joint ϕ

3. Mobile manipulator with holonomic platform

3.1. Analysis

The other type of mobile platforms that we intend to present in this section is the one with omnidirectional wheels. These particular types of wheels are used to develop a holonomic mobile robot. They enable the robot to move in any direction at any orientation. There is no need to change the orientation of the platform while moving in an arbitrary trajectory. The direction of the linear velocity is independent from the orientation of the mobile platform. We used the particular concept of a wheel formed with 2 truncated spheres intermechanically dependent developed in Mourioux and his colleagues, (Mourioux et al., 2006). Two parallel planes truncate each sphere. An axis enables each sphere to turn on it freely. This axis is maintained by a fork, which can rotate by using a DC motor. We consider here the mobile manipulator shown in Figure 15. The location of the platform is given by a vector $\xi_p = [x, y, \theta]^T$; which defines the position and the orientation of the platform in the fixed frame. The position of the point O_4 in the fixed frame is thus given by its Cartesian coordinates ξ_1 , ξ_2 , and ξ_3 .

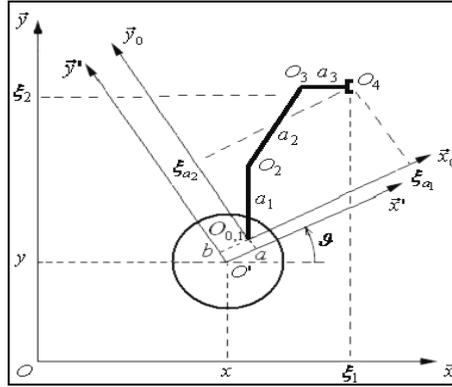


Fig. 15. Mobile manipulator with omni directional platform

3.2 Geometric solution with no obstacles

The control of the mobile manipulator is given by $u = [u_a^T, u_p^T]^T = [\dot{q}_{a1}, \dot{q}_{a2}, \dot{q}_{a3}, \dot{x}, \dot{y}, \dot{\theta}]^T$,

with $u_a = \dot{q}_a = [\dot{q}_{a1}, \dot{q}_{a2}]^T$ being the control of the robotic arm and $u_p = [\dot{x}, \dot{y}, \dot{\theta}]^T$ the control of the platform. The degree of mobility of the mobile manipulator is

$\delta_m = n_a + \delta_{mp} = 3 + 3 = 6$, with δ_{mp} the degree of mobility of the platform. According to Figure 16, the geometric model of this mobile manipulator is:

$$\begin{cases} \xi_1 = x + (a + C_1(a_3C_{23} + a_2C_2))C_\theta - (b + S_1(a_3C_{23} + a_2C_2))S_\theta \\ \xi_2 = y + (a + C_1(a_3C_{23} + a_2C_2))S_\theta + (b + S_1(a_3C_{23} + a_2C_2))C_\theta \\ \xi_3 = z + a_1 - a_2S_2 - a_3S_{23} \end{cases} \quad (33)$$

From (32), we get the instantaneous location kinematics model:

$$\dot{\xi} = J \cdot u \quad (34)$$

Where;

$$\dot{\xi} = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix}, \text{ and } J = \begin{bmatrix} D_3 & D_2 & D_1 & 1 & 0 & D_9 \\ D_6 & D_5 & D_4 & 0 & 1 & D_{10} \\ 0 & D_8 & D_7 & 0 & 0 & 0 \end{bmatrix}$$

With the following intermediate variables:

$$\begin{aligned} C_{g1} &= \cos(\theta + q_{a1}), S_{g1} = \sin(\theta + q_{a1}), C_{12g} = \cos(q_{a1} + q_{a2} + \theta), S_{12g} = \sin(q_{a1} + q_{a2} + \theta), \\ D_1 &= -a_3C_{1g}S_{23}, D_2 = -a_2C_{1g}S_2 + D_1, D_3 = -a_2S_{1g}C_2 - a_3S_{1g}C_{23}, \\ D_4 &= -a_3S_{1g}S_{23}, D_5 = -a_2S_{1g}S_2 + D_4, D_6 = a_2C_{1g}C_2 + a_3C_{1g}C_{23}, \end{aligned}$$

$$D_7 = -a_3 C_{23}, \quad D_8 = -a_2 C_2 + D_7,$$

$$D_9 = -a S_g - b C_g + D_3, \quad D_{10} = a C_g - b S_g + D_6$$

The kinematic control problem is aimed to find the control vector $(u(t) | t \in [t_0, t_f])$ to achieve the desired operational motion $(\xi^*(t) | t \in [t_0, t_f])$ of the end effector in such a way that the error $e(t) = \xi^*(t) - \xi(t)$ approaches zero. Since the system is redundant ($m < \delta_m$), we set m_{add} additional tasks (Seraji, 1998) so that

$$\dot{\xi}_{\text{add}} = J_{\text{add}} \cdot u \quad (35)$$

J_{add} is a matrix of dimension $m_{\text{add}} \times \delta_m$. Also we want to regulate $\xi_{\text{add}}(t)$ to the velocities of the desired additional tasks $\xi_{\text{add}}^*(t)$. Equation (34) and (35) are combined to give the differential kinematics model

$$\dot{\xi}_t = J_t \cdot u \quad (36)$$

Such that: $J_t = \begin{bmatrix} J \\ J_{\text{add}} \end{bmatrix}$ and $\xi_t = \begin{bmatrix} \xi \\ \xi_{\text{add}} \end{bmatrix}$

The problem is now to regulate the actual value of ξ_t to $\xi_t^* = [\xi^*, \xi_{\text{add}}^{*T}]^T$. Let $e_t = [e_t^T, e_{\text{add}}^T]^T = \xi_t^* - \xi_t$ with $e_{\text{add}} = \xi_{\text{add}}^* - \xi_{\text{add}}$. The matrix J_t is of dimension $(m + m_{\text{add}}) \times \delta_m$. Depending on the desired additional tasks this matrix is not necessarily square. If we suppose that J_t is of full rank and if $r = \text{rank}(J_t)$, then $r = \min(m + m_{\text{add}}) \times \delta_m$.

The control vector u is computed by solving the linear system $\dot{\xi}_t = J_t u$ (Bayle et al., 2002).

Now, if we consider only the position of the end effector, then $\xi = [\xi_1, \xi_2, \xi_3]^T$, i.e., $m=3$;

and if $\xi_{\text{add}} = \xi_p = [x, y, \theta]^T$, i.e., $m_{\text{add}}=3$. In this case, we can determine the vector u such that

$$u = J_t^{-1} (\xi_d^* + W_t (\xi_d^* - \xi_t)) \quad (37)$$

Where W_t is an $(m + m_{\text{add}})$ -order definite positive matrix and $\xi_d^*(t) = \xi_t^*(t)$ denotes the desired motion.

3.3. Geometric solution with obstacles

Up to now we have supposed the path of the robot clear from any obstacles.. However, in case of presence of obstacles some modifications have to be done. In this case, we use an approach based on virtual impedance model (Arai & Ota, 1996). This model can be seen as an extension of the potential field concept (Khatib, 1986). This model determines the motion of a robot by means of a desired trajectory ξ_t^* modified by a sum of different forces. These forces consist of three parts: an attractive force named F_{target} , generated to attract the robot toward the objective, a repulsive force generated between the robot and the obstacles F_{obs} , and a third force generated between the platform and the carried arm manipulator F_{man} (see Fig 16). In this work, only the first two forces are considered. The closed loop dynamical equation is expressed as equation (37).

$$M_d(\ddot{\xi}_t^* - \ddot{\xi}_t) + B_d(\dot{\xi}_t^* - \dot{\xi}_t) + K_d(\xi_t^* - \xi_t) = F_{ext} \quad (38)$$

Where F_{ext} represents all the forces exerted on the mobile robot, such that:

$$F_{ext} = F_{target} + \sum_i F_{obs} + F_{man} \quad (39)$$

From equation (37), we can derive the desired impedance Z_d such that:

$$Z_d = M_d s^2 + B_d s + K_d \quad (40)$$

Where M_d, B_d, K_d are diagonal positive definite desired mass, damping and spring effects. Equation (38) can be expressed in terms of the desired impedance and the trajectory tracking. Let e_d be a new signal error such that

$$e_d = (\xi_t^* - \xi_t) - \frac{F_{ext}}{Z_d} \quad (41)$$

If e_d approaches zero, then equation (38) is realized. The realized trajectory in equation (41)

can be seen as the sum of the desired trajectory ξ_t^* and the force correction $\frac{F_{ext}}{Z_d}$ to give

what we call the desired modification motion:

$$\xi_d^*(t) = \xi_t^*(t) + \frac{F_{ext}}{Z_d} \quad (42)$$

Note that $F_{ext} = 0$ in free motion.

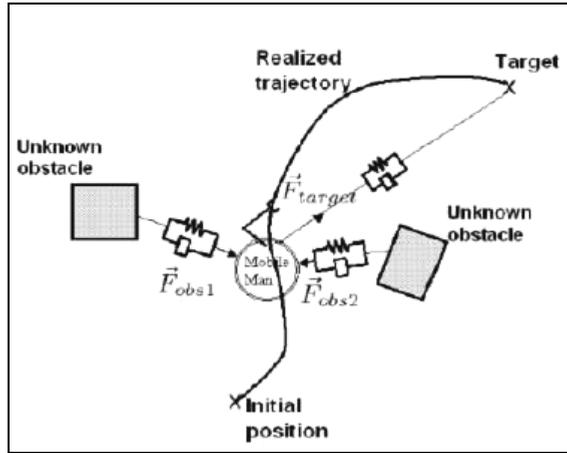


Fig. 16. Virtual impedance model for the mobile platform

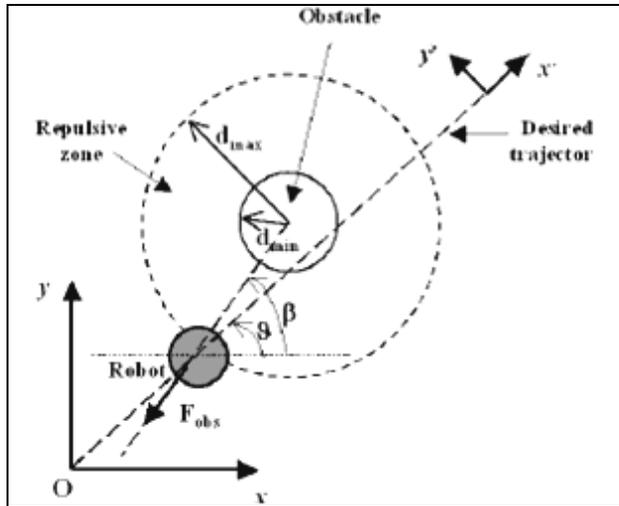


Fig. 17. Obstacle repulsion force

The magnitude F_{obs} is chosen to be (Borenstein & Koren, 1991):

$$F_{obs} = a_{obs} - b_{obs} \left(d(t) - d_{min} \right)^2$$

where a_{obs} and b_{obs} are positive constants satisfying the

condition $a_{obs} = b_{obs} (d_{max} - d_{min})^2$, d_{max} is the maximum distance between the robot and the detected obstacle that causes a nonzero repulsive force, d_{min} represents the minimum distance accepted between the robot and the obstacle and $d(t)$ is the distance measured between the robot and the obstacle $d_{min} < d(t) < d_{max}$ (Fig. 17). Note that the bound d_{max} characterizes the repulsion zone. Which is inside the region where the repulsion force has a non-zero value. Desired interaction impedance is defined as the linear dynamic

relationship $Z_d = B_d s + K_d$ where B_d and K_d are positive constants simulating the damping and the spring effects, respectively, involved in the robot obstacle interaction inside the repulsion zone.

3.4 Simulation results

Simulations are conducted in order to show the performance of the proposed methodology. The numerical example supposes the lengths of the arm are such that $a_1 = 0.6$, $a_2 = 0.4$, $a_3 = 0.3$ and the initial configuration of the mobile manipulator is such

that: $\xi_p = (0.1, 0.1, \pi/6)$ and $q_a = [\pi/4, -\pi/2, \pi/4]^T$. The end effector is supposed to track

the following straight-line trajectory $\xi^*(t) = [\xi_1^*(t), \xi_2^*(t), \xi_3^*(t)]^T = [0.1t, 0.1t, 1 + 0.1t]^T$;

Furthermore, we imposed the following additional tasks to the mobile platform $\xi_p^*(t) = (x^*(t), y^*(t), \vartheta^*(t)) = (t, t, \pi/4)$. Fig. 18 shows the stance of the whole system when the

end effector tracks the reference trajectory. The resulting trajectory of the end effector as well as that of the mobile platform is depicted in Fig. 19. Figures 20, 21, 22 and 23 describe the evolution of the angles of the arm and the orientation of the platform respectively. If the robot finds an obstacle at less than $d_{\max} = 1\text{m}$ the impedance control is activated, and the collision is avoided as it can be seen in Fig. 24.

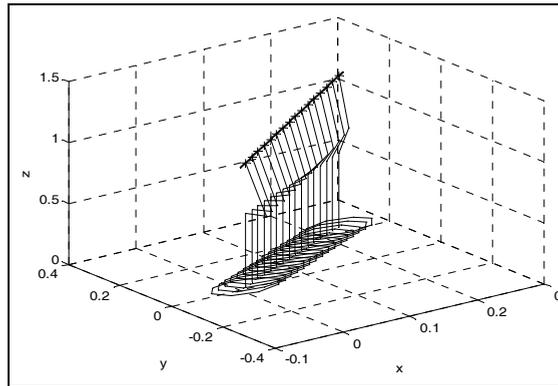


Fig. 18. A 3D-view of the arm and the mobile platform evolutions in an obstacle free space.

The resulted trajectories of the arm as well as of the mobile platform appear in Fig. 26. The corresponding curves showing the evolution of the angles of the arm and the orientation of the platform are depicted in Figs. 27, 28, 29 and 30 respectively.

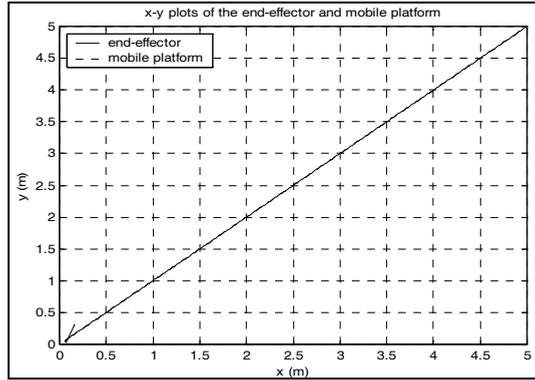


Fig. 19. End-effector and mobile platform trajectories in the x-y plane with no obstacles.

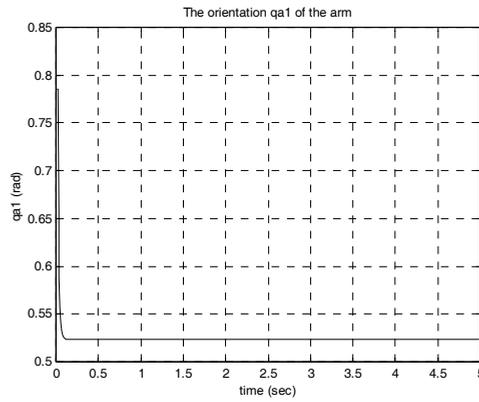


Fig. 20. Articulation q_{a1} curve

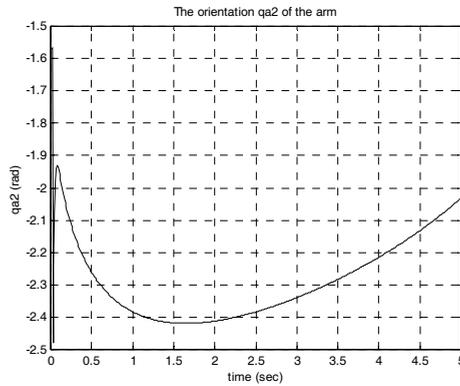


Fig.21. Articulation q_{a2} curve

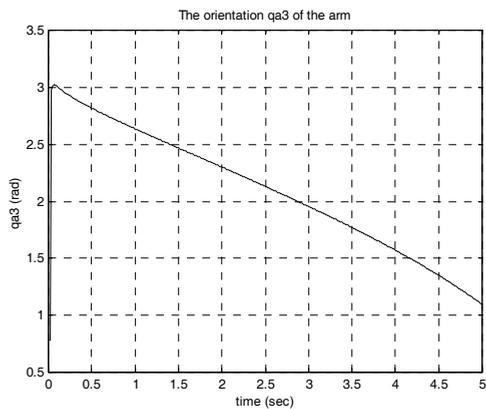


Fig. 22. Articulation q_{a3} curve

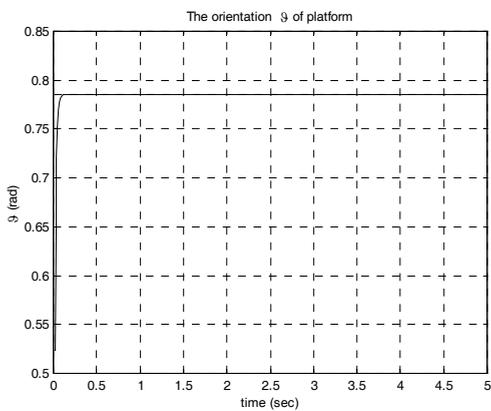


Fig. 23. Articulation ϑ curve

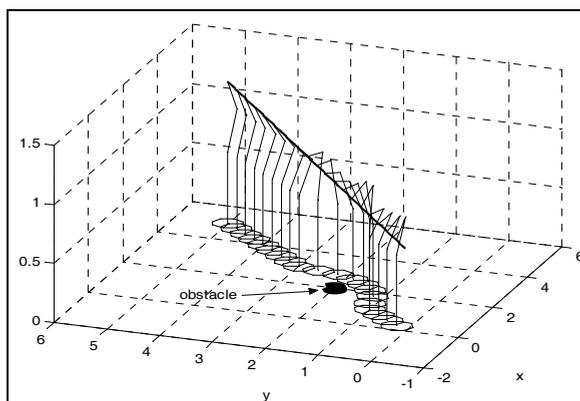


Fig. 24. A 3D-View of the arm and the platform evolutions in presence of obstacles

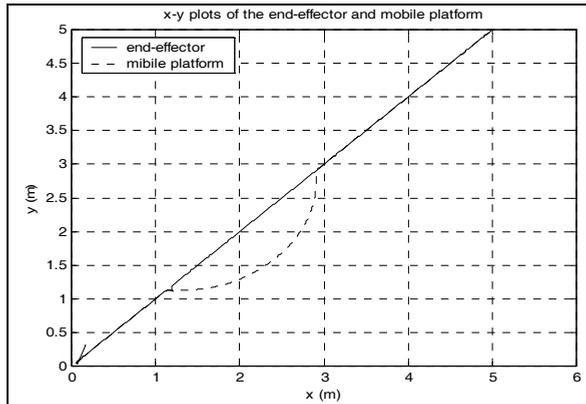


Fig. 25. End-effector and mobile platform trajectories in the x-y plane in presence of obstacles

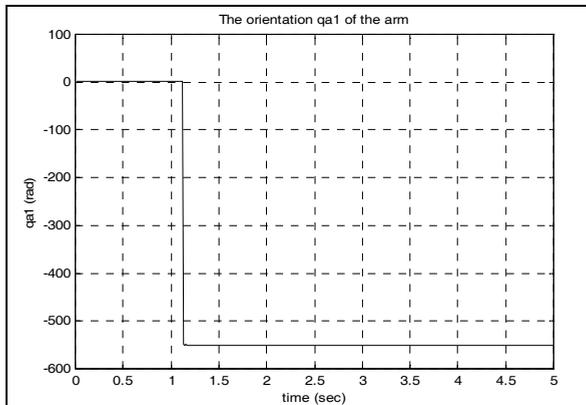


Fig. 26. Evolution curve of the joint θ_1

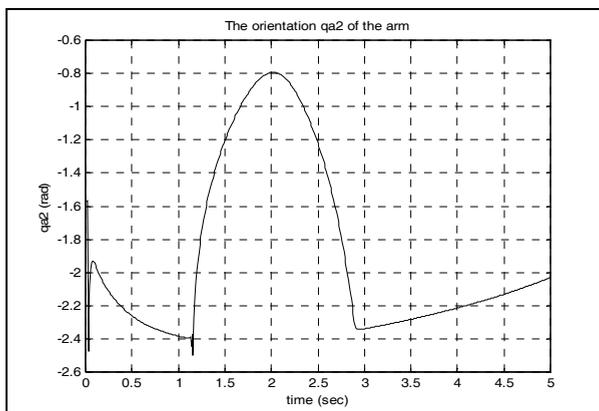


Fig. 27. Evolution curve of the joint θ_2

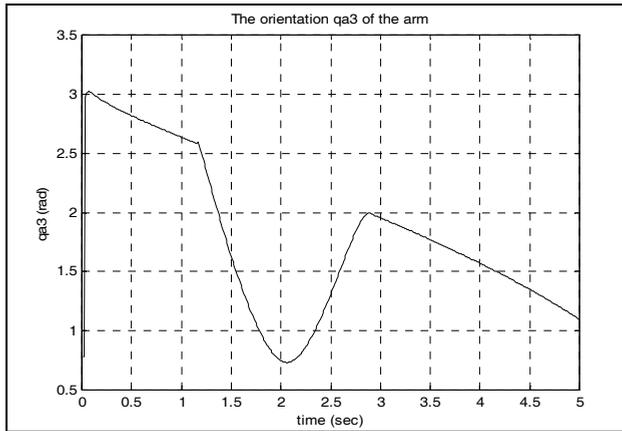


Fig. 28. Evolution curve of the joint θ_3

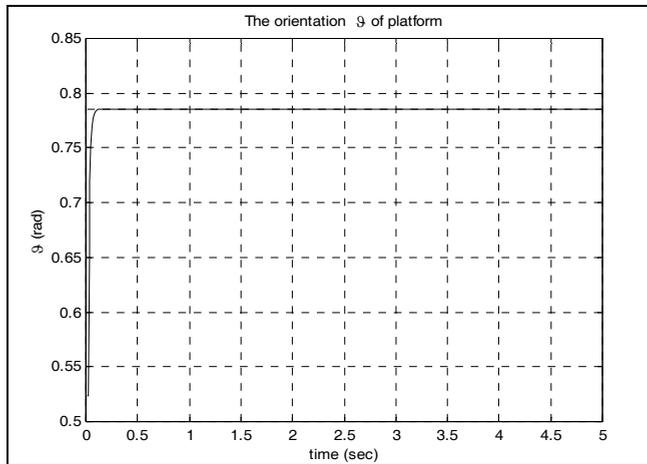


Fig. 29. Articulation φ curve

4. Conclusion

This work proposed two different methodologies to generating desired joint trajectories for both holonomic and non-holonomic mobile manipulators given prespecified operational tasks. The first part considers a non-holonomic platform where the generalized inverses in the resolution of a redundant system are used. The additional degrees of freedom are exploited to avoid unforeseen obstacles and joint limits. In the second part of the work a holonomic platform is used. In this case, the trajectory is generated using a reactive approach based on virtual impedance and additional tasks. When the robot task is about a stationary point, the mobile manipulator showed a good tracking for the manipulator. As perspective an estimate procedure must be conducted in order to estimate the contact forces and the unknown holonomic mobile manipulator parameters driving the system Computer

simulations have validated to show the effectiveness of the two approaches. The reference values obtained by the two methods can be used as inputs to controllers for real motion.

5. References

- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research*, 5(1):90{98.
- Sundar, S. & Shiller, Z. (1997). Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation, *IEEE Trans.on Robotics and Automation*, Vol. 13, pp. 305{310.
- Laumond, J. P., Jacobs, P. E., Taix, M., and Murray, R. M. (1994). A motion planner for nonholonomic mobile robots, *IEEE Trans. on Robotics and Automation*, Vol. 10, pp. 577{593.
- Reeds, J. A. and Shepp, R. A. (1990). Optimal paths for a car that goes both forward and backward, *Pacific J. Math.*, vol. 145, pp. 367-393.
- Murray, R. M.; Li, Z. & Sastry, S. S. (1994). A Mathematical Introduction to Robotic Manipulation. Boca Raton, FL: CRC Press.
- Tilbury, D.; Murray, R. M. & Sastry, S. S. (1995). Trajectory generation for the n-trailer problem using goursatnormal form, *IEEE Trans. Automat. Contr.*, vol. 40, pp. 802-819, May 1995.
- Abdessemed, F. Monacelli, E. & Benmahammed, K. (2008). Trajectory Generation In an Alternated and a Coordinated Motion Control Modes of a Mobile Manipulator, AMSE journal, Modelling, Measurements and Control B, Vol.77, No 1, pp 18-34.
- Djebrani, S. Benali, A. & Abdessemed, F. (2009). Force-position control of a holonomic mobile manipulator, 12 int. Conf. on Climbing & Walking Robots and the support technologis for Mobile Machines Bogazaci Univ. Garanti Culture Center (North Campus).
- Qu, Z.; Wang, J. & Plaisted, C. E. (2004). A New Analytical Solution to Mobile Robot Trajectory Generation in the Presence of Moving Obstacles, *IEEE Tran. on Robotics*, Vol. 20, No. 6.
- Kant, K. & Zucker, S. W. (1988). Planning collision free trajectories in time varying environments: A two-level hierarchy, in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, pp. 1644-1649.
- Murray, R. M. & Sastry, S. S. (1993). Nonholonomic motion planning: Steering using sinusoids, *IEEE Trans. Automat. Contr.*, vol. 38, pp. 700-716.
- Abdessemed, F.; Benmahammed, K. & Eric Monacelli (2004). A Fuzzy Based Reactive Controller for Non-Holonomic Mobile Robot, *Journal of Robotics and Autonomous Systems*, 47 (2004) 31-46.
- Russell, S. & Norvig, P. (2000). Artificial Intelligence: A Modern Approach, *Prentice Hall*, New Jersey, 1995
- A. Okabe, B. Boots, K. Sugihara and S.N. Chiu, Spatial Tesselations and Applications of Voronoi Diagrams, John Wiley & Sons, New York.
- Zhao, M.; Ansari, N. & Hou, E.S.H. (1994). Mobile manipulator path planning by a genetic algorithm, *Journal of Robotic Systems*, 11(3): 143-153.
- Pin, F. G. & Culioli, J. C. (1992). Optimal Positioning of Combined Mobile Platform-Manipulator systems for Material Handling Tasks, *Journal of intelligent and Robotic Systems*. 6: 165-182.

- Pin, F. G.; Morgansen, K. A.; Tulloch, F. A.; Hacker, C. J. & Gower, K. B. (1996). Motion Planning for Mobile Manipulators with a Non-Holonomic Constraint Using the FSP (Full Space Parameterization) Method, *Journal of Robotic Systems* 13(11), 723-736.
- Lee, J. K. & Cho, H. S. (1997). Mobile manipulator Motion Planning for Multiple Tasks Using Global Optimization Approach, *Journal of Intelligent and Robotic Systems*, 18: 169-190.
- Seraji, H. (1995) Configuration control of rover-mounted manipulators, *IEEE Int. Conf. on Robotics and Automation*, pp2261-2266.
- Campion, G.; Bastin, B. & D'Andrea-Novel. (1996). Structural proprieties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. on Robotics and Automation*, 2(1):47{62, February.
- Liegeois, A. (1997). Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Trans. Syst. Man Cybernet.* 7, 842-868.
- Seraji, H. (1993). An on-line approach to coordinated mobility and manipulation, ICRA'93, pp. 28-35, May, 1993.
- Mourioux, G.; Novalés, C.; Poisson, G. & Vieyres, P. (2006). Omni-directional robot with spherical orthogonal wheels: concepts and analyses, *IEEE International Conference on Robotics and Automation*, pp. 3374-3379.
- Seraji, H. (1998). A unified approach to motion control of mobile manipulators, *The International Journal of Robotics Research*, vol. 17, no. 2, pp. 107-118.
- Bayle, B.; Fourquet, J. Y.; Lamiriaux, F. & Renaud, M. (2002). Kinematic control of wheeled mobile manipulators, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1572-1577.
- Arai, T. & Ota, J. (1996). Motion planning of multiple mobile robots using virtual impedance, *Journal of Robotics and Mechatronics*, vol. 8, no. 1, pp. 67-74.
- Borenstein, J. & Koren, Y. (1991). The vector field histogram fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278-288.

Trajectory Control of Robot Manipulators Using a Neural Network Controller

Zhao-Hui Jiang
Hiroshima Institute of Technology
Japan

1. Introduction

Many advanced methods proposed for control of robot manipulators are based on the dynamic models of the robot systems. Model-based control design needs a correct dynamic model and precise parameters of the system. Practically speaking, however, every dynamic model has some degrees of incorrectness and every parameter associates with some degrees of identification error. The incorrectness and errors eventually result positioning or trajectory tracking errors, and even cause the system to be unstable. In the past two decades, intensive research activities have been devoted on the design of robust control systems and adaptive control systems for the robot in order to overcome the control system drawback caused by the model errors and uncertain parameters, and a great number of research results have been reported, for example, (Hsia, 1989), (Kou, and Wang, 1989), (Slotine and Li, 1989), (Spong, 1992), and (Cheah, Liu and Slotine, 2006). However, almost parts of results associate with complicated control system design approaches and difficulties in the control system implementation for industrial robot manipulators.

Recently, neural network technology attracts many attentions in the design of robot controllers. It has been pointed out that multi-layered neural network can be used for the approximation of any nonlinear function. Other advantages of the neural networks often cited are parallel distributed structure, and learning ability. They make such the artificial intelligent technology attractive not only in the application areas such as pattern recognition, information and graphics processing, but also in intelligent control of nonlinear and complicated systems such as robot manipulators (Sanger, 1994), (Kim and Lewis, 1999), (Kwan and Lewis, 2000), (Jung and Yim, 2001) (Yu and Wang, 2001). A new field in robot control using neural network technology is beginning to emerge to deal with the issues related to the dynamics in the robot control design. A neural network based dynamics compensation method has been proposed for trajectory control of a robot system (Jung and Hsia, 1996). A combined approach of neural network and sliding mode technology for both feedback linearization and control error compensation has been presented (Barambones and Etxebarria, 2002). Sensitivity of a neural network performance to learning rate in robot control has been investigated (Clark and Mills, 2000).

In the following, we present a simple control system consisting of a traditional controller and a neural network controller with parallel structure for trajectory tracking control of

industrial robot manipulators. First, a PD controller is designed. Second, a neural network with three layers is designed and added to the control system in the parallel way to the PD controller. Finally, a learning scheme used to train the weights of each layer of the neural network is derived by minimizing a criterion prescribed in a quadratic form of the error between a planned trajectory and response of the robot. Control system implementation issue is discussed. Both the motivation function of the neural network and dynamic model used in the calculation of the learning law are simplified to meet practical needs. An industrial manipulator AdeptOne is adopted as an experimental test bed. Trajectory tracking control simulations and experiments are carried out. The results demonstrate effectiveness and usefulness of the proposed control system.

2. Dynamic models of robot manipulators

2.1 Torque-based dynamic model

A torque-based dynamic model of robot manipulator describes relationship between motion and joint torque of the robot without concerning what generates the torque and how. This class of dynamic formulation is most popular and widely used in the control design and simulation of the robot manipulator. Usually, a torque-based dynamic model can be systematically derived by using the Lagrange method as follows

$$M(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \quad (1)$$

where, $\theta \in R^n$ and $\tau \in R^n$ are joint variable and torque, $M(\theta) \in R^{n \times n}$ is inertia matrix, $H(\theta, \dot{\theta}) \in R^n$ contains Coriolis and centrifugal forces, and $g(\theta) \in R^n$ denotes gravitational force.

Remarks: In motion equation (1), $M(\theta)$ is a symmetric matrix, and $\dot{M}(\theta) - 2H(\theta, \dot{\theta})$ is a skew symmetric matrix. These properties of robot dynamics allow one to design the control system on the basis of dynamic model in an easier way.

2.2 Voltage-based dynamics model

In the almost cases of industrial robot manipulators, the torque-based dynamic model cannot be used directly because most industrial manipulators are not functionally designed on the basis of torque/force control but servo control. In the other words, as actuators almost all robot manipulators are equipped with servo motors that are controlled by input voltage not by current. The former results the so-called velocity servo, and the later meets the needs of the torque-based control that may require the torque-based dynamic model.

For, the robot with servo-controlled motors, we need to take the characteristics of the motors and servo-units into consideration in the dynamic modeling, parameter identification and control design. Generally, the dynamic model of the motor with a servo unit can be given as follows.

$$\frac{L_i}{a_i} \dot{i} + \frac{R_i}{a_i} i + f_{vi} \dot{\theta}_i + D_i(\dot{\theta}_i) = u_i \quad (2)$$

For the n-link robot manipulator, the dynamic characteristics of the motors and servo units can be rewritten in a compact form as

$$La^{-1}\ddot{\tau} + Ra^{-1}\dot{\tau} + f_v\dot{\theta} + D(\dot{\theta}) = u \quad (3)$$

Though rates of amplifiers of the servo units are included in the parameters in the above equation, in the following, we rather like to use the nominal terms of parameters that are often referred directly to a servo motor. $u = \text{diag}(u_1, u_2, \dots, u_n)$ denotes the input voltage of the servo units; $L = \text{diag}(L_1, L_2, \dots, L_n)$, $R = \text{diag}(R_1, R_2, \dots, R_n)$ are matrices of inductance and resistance; and $a = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n)$ is the matrix with the elements being the back electromotive constant of each servo motor; $f_v = \text{diag}(f_{v1}, f_{v2}, \dots, f_{vn})$ denotes matrix of viscous friction constants, and $D(\dot{\theta})$ is a diagonal matrix that diagonal elements indicate the constants of Coulomb frictions and electrical dead zones of the motors. Combining (1) and (3) together, after some simple manipulations we obtain

$$\hat{L}(\theta)\ddot{\theta} + \hat{R}(\theta, \dot{\theta})\dot{\theta} + \hat{H}(\theta, \dot{\theta}) = u \quad (4)$$

where

$$\hat{L}(\theta) = La^{-1}M(\theta) \quad (5)$$

$$\hat{R}(\theta, \dot{\theta}) = La^{-1}(\dot{M}(\theta) + H(\theta, \dot{\theta})) + Ra^{-1}M(\theta) \quad (6)$$

$$\hat{H}(\theta, \dot{\theta}) = La^{-1}(\dot{H}(\theta, \dot{\theta})\dot{\theta} + \dot{g}(\theta)) + Ra^{-1}(H(\theta, \dot{\theta})\dot{\theta} + g(\theta)) \quad (7)$$

3. Control problem statement

Standing on the theoretical view point, the dynamic model given by (4) can be used in model-based control system design with a kind of computed-torque like control method. Implementation of such a control system, however, is difficult to carry out since either acceleration sensors or numerical derivative approaches are necessary for calculating the control input that contains acceleration feedback. Acceleration sensors are not available in industrial robots, and the numerical derivative approaches would result high frequency noises and phase lag. On the other hand, every dynamic model contains more or less modeling errors and/or parameter uncertainties that cause imprecise trajectory tracking in the control based on the dynamic model.

Although what we are discussing here is about high-performance advanced control methods, the practical world that we have to face is that all commercialized industrial robot manipulators associate with built-in traditional PID controllers. However, a significant drawback of the PID control system is that it cannot guarantee a precise tracking result for given dynamic trajectories since such the control system is essentially driven by trajectory error itself.

From the above discussion, we clarified the problems in dynamic trajectory control of robot manipulators, and found two key points for the problem-solving in the dynamic trajectory tracking control system design: one is how to utilize the built-in PID controller of the robot system; another one is how to take the dynamic characteristics of the robot into

consideration in the trajectory control. The neural network provides us with some new options in the control design in many ways: to approximate dynamics and/or inverse dynamics, to compensate dynamic effects, to be a controller itself, etc. In this chapter, we combine the built-in controller and a neural network together to design a new control system for trajectory control of the robot. We aim at high precision trajectory tracking control of the industrial robot manipulators using simple and applicable control method. We design a control strategy with both technologies of PID control a neural network for taking the advantages of both simplicity on design and implementation of a PID controller, and learning capacity of neural network control. The main idea is to establish a control system with the PID controller and a neural network control scheme which are parallel to each other in structure for achieving precise tracking control of dynamic trajectories. The detail description of the control system design yields to the next section.

4. Structures of the robot control system using neural network

It is usual that the neural network controllers are structurally defined as the feedback controllers in the control system. The neural networks are trained such that the trajectory tracking error e converge to zero. Fig.1 and Fig.2 show two kinds of block structures of the

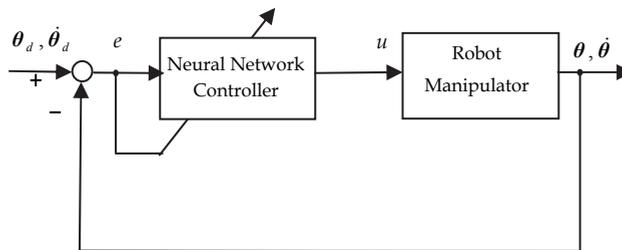


Fig. 1. Structure of a neural network control system with the trajectory error being the input of the neural network.

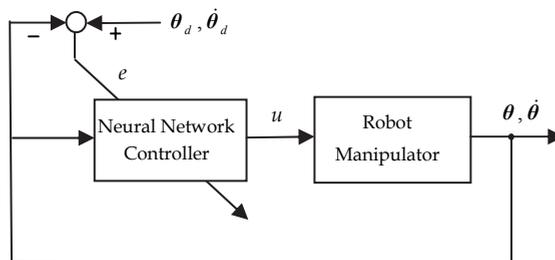


Fig. 2. Structure of a neural network control system with the state variable being the input of the neural network.

neural network system. In Fig.1, the neural network is driven by the trajectory tracking error, whereas in Fig.2 the neural network is driven by the states of the robot system.

To combine a neural network controller and a built-in PID controller together in parallel, we have two ways according to two structures shown in Fig.1 and Fig.2. In detail, the control system block diagrams are given in Fig. 3 and Fig.4.

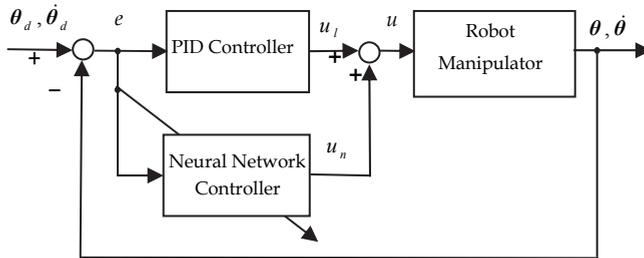


Fig. 3. Structure of robot control system with the trajectory error being the input of the neural network.

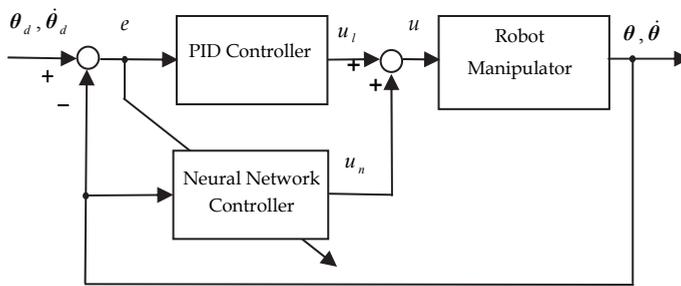


Fig. 4. Structure of robot control system with the state variable being the input of the neural network.

In control system of Fig.3, what role the neural network controller plays is no more than a controller since the neural network's input is trajectory error. On the other hand, the neural network controller given in Fig.4 has possibility to work as not only a controller but also a dynamic compensator. The later generates the forces/torques to compensate the gravity and other dynamic forces/torques according to the dynamic trajectories so that the trajectory tracking may be more accurately achieved. In the rest part of this chapter, we will mainly discuss the design of control system that the structure is shown in Fig.4.

5. Control system design

5.1 The control strategy

In the control system shown in Fig.4, the total control scheme is given as follows.

$$\mathbf{u} = \mathbf{u}_l + \mathbf{u}_n \quad (8)$$

\mathbf{u}_l is control input of the PID controller, and can be simply described as below.

$$u_l = -k_v(\dot{\theta} - \dot{\theta}_d) - k_p(\theta - \theta_d) - k_i \int_0^t (\theta - \theta_d) dt \quad (9)$$

where θ_d and $\dot{\theta}_d$ are planned trajectories of joint displacements and velocities, k_v , k_p , and k_i are gain matrices.

u_n is the control input of the neural network controller being designed. The structure of neural network controller is shown in Fig. 5. The detail mathematical description of the neural network is given by

$$u_n = V f(Wq) \quad (10)$$

where $q = [\theta_1, \theta_2, \dots, \theta_n, \dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n]^T \in R^{2n}$ denotes input vector with elements being each joint variable, velocity; $u_n = [u_{n1}, u_{n2}, \dots, u_{nm}] \in R^m$ is output vector, $W \in R^{2n \times l}$ and $V \in R^{l \times m}$ with their elements being expressed by w_{ij} and v_{jk} , are weight matrices from input nodes to the hidden layer and from hidden layer to the output layer; $f(*) \in R^l$ is an activation function vector of the hidden layer with elements being selected as a saturation function, such as a sigmoid function; l is the number of hidden nodes. Though the dimension of robot joint inputs equals joint numbers n , here we denote it as m in order to describe the network controller design clearer.

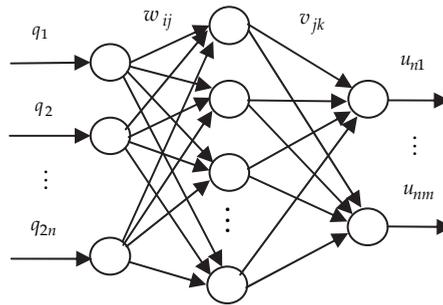


Fig. 5. Multilayer neural network controller.

2.2 Detail design of the neural network controller

For tracking control of a robot with a designed dynamic trajectory, only the PD controller is not enough to ensure a proper tracking precision. For this reason, we design the neural network controller such that it takes the important part on which the PD controller has shown its limitation and/or powerlessness. In doing so, the neural network controller should be trained in such the way: the trajectory tracking error getting smaller and smaller while training. First, we choose a performance criterion of the whole control system with a quadratic form of the trajectory tracking error and velocity tracking error, as follows.

$$\begin{aligned}
E &= \frac{1}{2}(\theta_d - \theta)^T(\theta_d - \theta) + \frac{1}{2}(\dot{\theta}_d - \dot{\theta})^T(\dot{\theta}_d - \dot{\theta}) \\
&= \frac{1}{2}(\mathbf{q}_d - \mathbf{q})(\mathbf{q}_d - \mathbf{q})
\end{aligned} \tag{11}$$

The weights' learning algorithm is derived based on the back-propagation approach. The tuning law is to give weights' increments to be proportional to the negative gradient of the performance criterion with respect to the weights. For updating of the weights between the hidden layer and the output layer, we define an increment as

$$\Delta v_{jk} = -\gamma_{jk} \frac{\partial E}{\partial v_{jk}} \quad (j=1,2,\dots,l; k=1,2,\dots,m) \tag{12}$$

where, j and k indicate the one between j th node of the hidden layer and k th node of the output layer, and γ_{jk} is a constant of proportionality, to be designed as a learning rate.

Whereas for the weights between the input layer and hidden layer, we give

$$\Delta w_{ij} = -\lambda_{ij} \frac{\partial E}{\partial w_{ij}} \quad (i=1,2,\dots,2n; j=1,2,\dots,l) \tag{13}$$

where λ_{ij} is a learning rate to be designed by the user.

Using the chain rule and noting that the weights are independent with u_1 , the partial derivative of (12) can be expressed as follows,

$$\frac{\partial E}{\partial v_{jk}} = \frac{\partial E}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial u_{nk}} \frac{\partial u_{nk}}{\partial v_{jk}} \tag{14}$$

In detail, $\partial E / \partial v_{2k}$ and $\partial v_{2k} / \partial u_{jk}$ can be given as

$$\frac{\partial E}{\partial \mathbf{q}} = -(\mathbf{q}_d - \mathbf{q})^T = -\mathbf{e}^T \in R^{1 \times 2n} \tag{15}$$

$$\frac{\partial \mathbf{q}}{\partial u_{nk}} = \mathbf{b}_k \in R^{2n \times 1} \tag{16}$$

and

$$\frac{\partial u_{nk}}{\partial v_{jk}} = f_j \tag{17}$$

where f_j is the output of j th node of the hidden layer.

Similarly, one can use the chain rule to (13) to obtain

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \mathbf{q}^T} \frac{\partial \mathbf{q}}{\partial \mathbf{u}_n} \frac{\partial \mathbf{u}_n}{\partial f_j} \frac{\partial f_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} \quad (18)$$

where z_j is the summation of input signals to j th node of the hidden layer, i.e.

$z_j = \sum_{s=1}^{2n} w_{sj} q_s$. In detail, each undetermined terms are given as

$$\frac{\partial \mathbf{q}}{\partial \mathbf{u}_n} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m] \equiv \mathbf{b} \in R^{2n \times m} \quad (19)$$

$$\frac{\partial \mathbf{u}_n}{\partial f_j} = [v_{j1}, v_{j2}, \dots, v_{jm}]^T \equiv \mathbf{v}_j \in R^m \quad (20)$$

$$\frac{\partial z_j}{\partial w_{ij}} = q_i \quad (21)$$

In (19), \mathbf{b} is a matrix depended on the dynamics of the robot, and will be specified in the next subsection. \mathbf{b}_k in (16) is the k th column vector of \mathbf{b} .

The fourth partial derivative term of right side of (18) can be determined directly using partial derivative $\partial f_j / \partial z_j = \partial f_j(z_j) / \partial z_j$ for a designed activation function $f_j(z_j)$.

5.3 An implementation issue

In the design of the neural network controller, since we aimed at trajectory tracking performance of the system, we designed the performance criterion using error's quadratic form of the inputs of the neural network other than using error's quadratic form of the outputs of the neural network, though the later is much usual in neural network design. It eventually results the use of dynamics of the system in deriving the learning law with back propagation method since the inputs and outputs of the robot system and neural network controller are contrary to each other. Ignoring the small parameters, usually dynamics (3) can be simplified as

$$\bar{L}(\theta)\ddot{\theta} + \bar{h}(\theta, \dot{\theta}) = u \quad (22)$$

Using $\mathbf{q} = [\theta^T, \dot{\theta}^T]^T$ as a state variable vector, above expression can be rewritten as

$$\dot{\mathbf{q}} = \mathbf{p} + \mathbf{B}u \quad (23)$$

where

$$\mathbf{p} = \begin{bmatrix} \dot{\theta} \\ -\bar{L}^{-1}(\theta)\bar{h}(\theta, \dot{\theta}) \end{bmatrix} \quad (24)$$

$$B = \begin{bmatrix} \mathbf{0} \\ \bar{L}^{-1}(\theta) \end{bmatrix} \quad (25)$$

Generally, the solution of (23) can be given by

$$q = \int_0^t p d\tau + \int_0^t B u d\tau \quad (26)$$

One can numerically calculate b in a real time control process as

$$b = \frac{\partial q}{\partial u_n} \cong \int_0^t B d\tau \cong \sum_{i=1}^k (t_i - t_{i-1}) B \quad (27)$$

where t_i indicates the i th sampling time.

6. Simulation and experimental studies

6.1 The test bed

The experimental test bed used in this research is an AdeptOne XL robot manipulator shown in Fig.6. It is a SCARA type high performance Direct Drive (DD) industrial robot manipulator possessed with 4 joints. Except the third joint being a prismatic joint, all joints are revolute. Though a closed-loop servo system is built-in by Adept Technology Corporation on the basis of servo units and servo motors, using the Advanced Servo Library the user is allowed to access the D/A converter directly to establish a user-designed close-loop servo system for the development of more advanced control system by V+ language. We developed control software on such the software and hardware environment. Since the third joint is prismatic and dynamically independent with other joints, control subsystem for the third joint can be designed independently and easily.

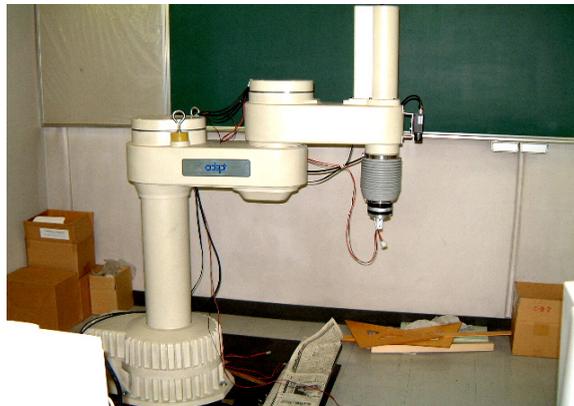


Fig. 6. AdeptOne robot manipulator

Focusing on control of the most complex part of the robot, we do not take the third joint into consideration in the control design. The fourth joint is extremely light-weight designed comparing with other joints and its link length is zero. Fourth joint does not cause dynamic coupling to the others. Therefore, we do not take it into consideration as well in control design and experiments.

6.2 Trajectory tracking control simulations

The joint trajectory tracking control simulations were carried out based on a simplified dynamic model of (3). The neural network controller was designed with three layers, four nodes for the input layer and hidden layer respectively, and two nodes for the output layer. The learning scheme was designed using the method given in section IV. The desired joint trajectories are designed using triangle functions with amplitudes to be 45 and 30 degrees for joint1 and joint2. The feedback gain matrices of the PD controller were determined as $k_p = \text{diag}(0.6, 0.1)$, $k_v = \text{diag}(0.8, 0.3)$. Learning rates in (12) and (13) were chosen as $\gamma_{j1} = 0.07$, $\gamma_{j2} = 0.04$ ($j = 1, \dots, 4$), $\lambda_{ij} = 0.01$ ($i = 1, \dots, 4; j = 1, \dots, 4$). Simulations were taken place under Matlab environment.

Fig.7 ~ Fig.10 show an example of the simulations. Fig.7 gives the planned joint trajectories and tracking control results. The broken lines indicate the planned trajectories which are not easy to be seen since they are almost completely covered by the thick lines i.e. the tracking results in fourth time learning. The dotted lines indicate results according PD control only, and the thin lines are first learning results.

Fig.8 gives velocity tracking results with the lines' types being the same meaning as described for Fig.7. Fig.9 shows control inputs of joint 1, (b) and (c) are control input generated by PD controller and neural network controller, respectively. (a) is the whole control input, i.e. the summation of (b) and (c). Fig.10 shows control inputs of joint 2.

From the simulation results it is seen that using the combined control system with PD controller and neural net work controller high precise joint trajectory tracking performance can be achieved under learning process of the weights of the neural network.

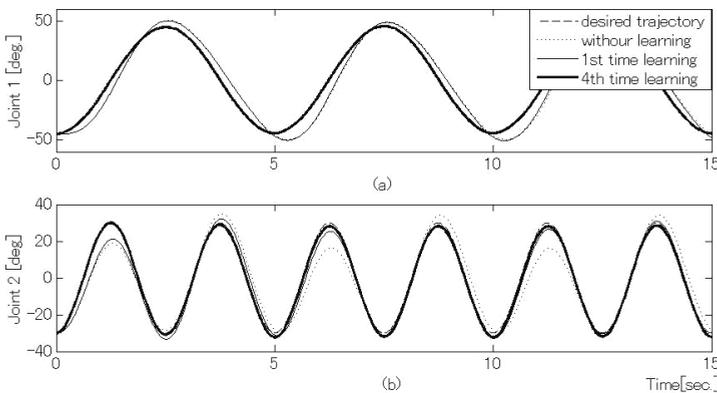


Fig. 7. Simulation results: planned joint trajectories and tracking results.

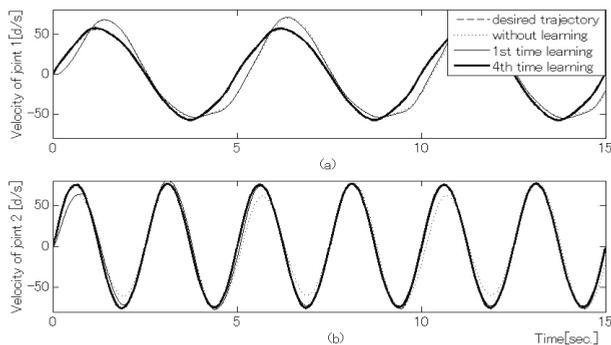


Fig. 8. Simulation results: planned joint velocity trajectories and tracking results.

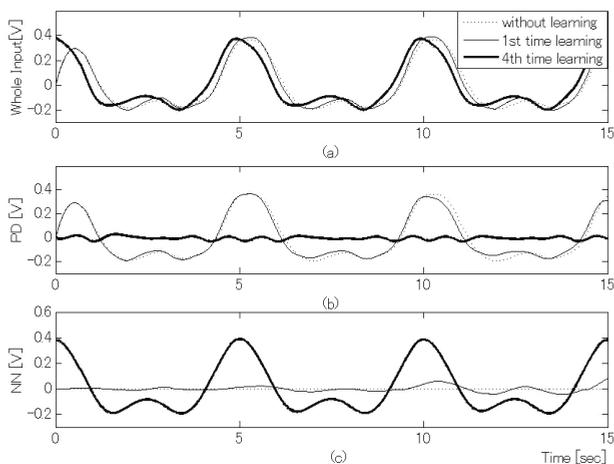


Fig. 9. Simulation results: control inputs of joint 1.

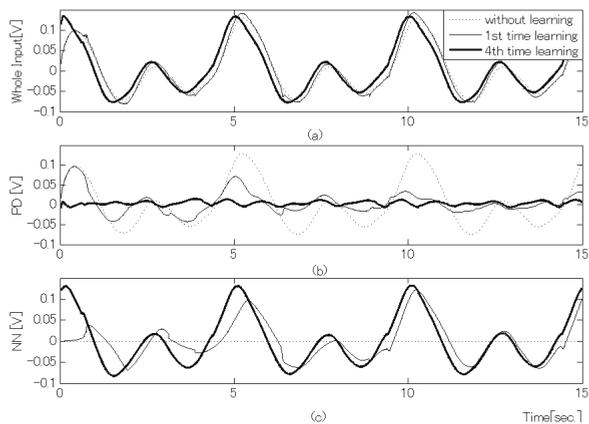


Fig. 10. Simulation results: control inputs of joint 2.

6.3 Trajectory tracking control experiments

The joint trajectory tracking control experiments were carried out under almost same conditions of the simulation except the feedback gain matrices were chosen as $k_p = \text{diag}(1.5, 0.6)$, $k_v = \text{diag}(1.3, 0.4)$, and amplitudes of the trajectories of joint 1 and 2 are planned as 25 and 20 degrees.

Fig.11~Fig.14 show the experimental results. Meaning of each figure stands for the same corresponding to the simulation results shown in the last subsection, as well as the lines in figures.

From the experimental results, it can be seen that though the trajectory tracking accuracy is a little bit lower comparing with the simulation results, the trajectory tracking error becomes less and less when learning time increases. It confirms the effectiveness and usefulness of the proposed control method.

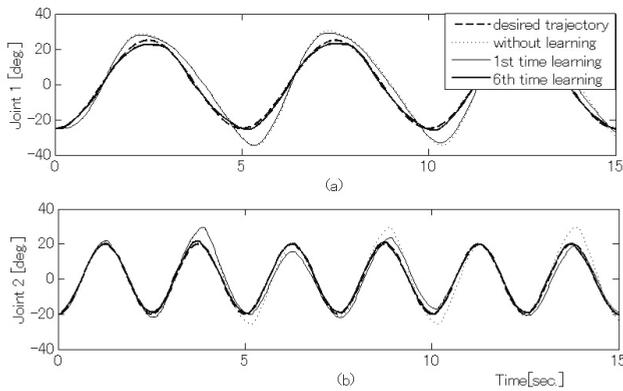


Fig. 11. Experimental results: planned joint trajectories and tracking results.

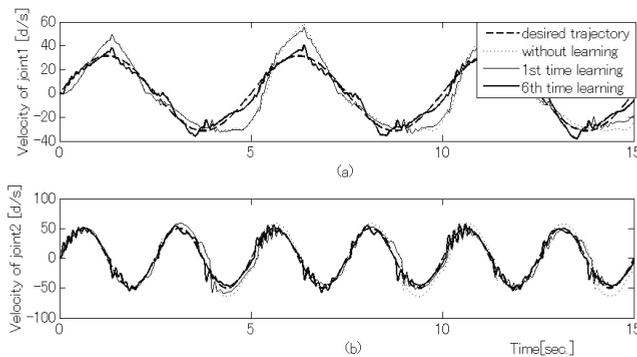


Fig. 12 Experimental results: planned joint velocity trajectories and tracking results.

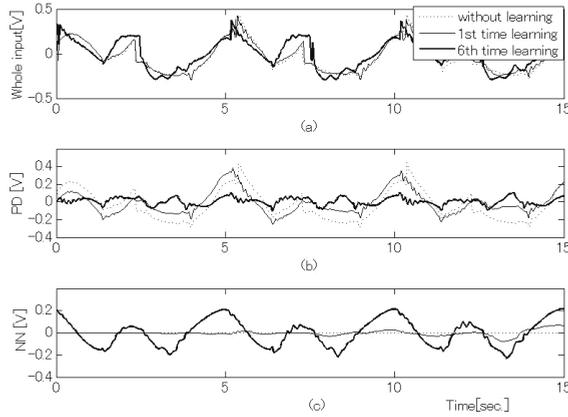


Fig. 13. Experimental results: control inputs of joint 1.

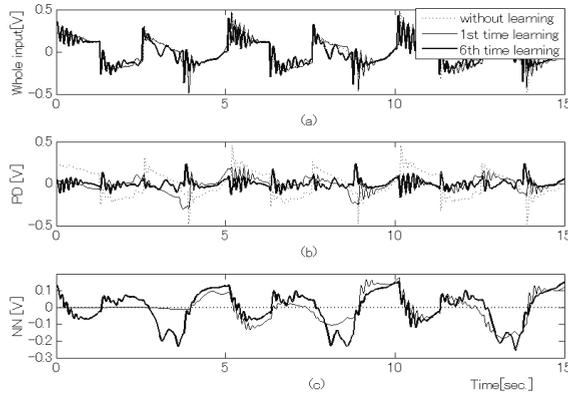


Fig. 14. Experimental results: control inputs of joint 2.

6.4 Discussion

PID controller controlled robot system is essentially driven by position error or trajectory tracking error. In dynamic trajectory tracking of a robot under PD control, the PD controller plays two important roles: one is motion regulation for guaranteeing stability of the robot system; another one is to generate force/torque required by the dynamic trajectory to drive the robot such that it would follow the trajectory. The latter needs a big enough tracking error in order to generate actuating force/torque required by the trajectory for the robot.

From the simulation results one can see that the tracking error significantly decreases as learning time increases while the control inputs generated by the neural network controller. Comparing Fig.9 (b) with (c) or Fig.10 (b) with (c) it can be interestingly found that learning for four times the neural network controller took PD controller over and played the main role in generating actuation voltages for the robot. On the other hand, in the experimental

results Fig.13 and Fig.14 “the roles changing” seems not as evident as in their simulation counterparts. The reason lies on a fact that we had added the dead-zone compensating inputs into PD control inputs.

Standing on the dynamics point of view, with the same tracking accuracy to a designed dynamic trajectory, the whole control input should be the same judging with the same unit of the input (whatever it counted by torque/force or voltage) regardless what kind of control method is adopted. In robot control with neural network, it is a popular way to use a neural network to approximate dynamics of the robot rather than use it as a controller itself. From the simulation and experimental results, it can be concluded that the neural network controller proposed in this paper not only plays the role as a controller but also play the role to generate force/torque required by dynamic trajectories just as an approximated dynamic model using neural network in computed torque control.

Though the results given here are limited on 4th time learning for the simulation and 6th time learning for the experiment, we carried out much more simulations and experiments, learning for 20 times, for example. The results show that after some specified time the learning effect will remain unchanged.

7. Conclusions

In this article, we presented dynamic trajectory tracking control of industrial robot manipulators using a PD controller and a neural network controller. Some different kinds structures of neural network control systems were discussed. The neural network controller was designed as a three layers feed-forward network. The learning law of weights of the neural network was derived using a simplified dynamic model of the robot and back propagation approach. Dynamic trajectory tracking control simulations and experiments were carried out using an industrial manipulator AdeptOne XL robot. The results showed the effectiveness and usefulness of the proposed control method. From the simulations and experiments, it was seen that according the increase of learning times the neural network controller took over of the PD controller on playing the role in generating actuating force/torque required by the dynamic trajectory. It also was clarified that the learning effect of the neural network has some limitation, i.e. after some specified time of learning, trajectory tracking accuracy remains unchanged.

8. References

- Hsia, T. C .S (1989). A new technique for robust control of servo systems, *IEEE Transactions on Industrial Electronics*, Vol. 36, No.1, pp.1-7
- Kou, C. Y.; Wang, S. P. (1989). Nonlinear robust industrial robot control, *Transactions ASME, Journal of Dynamic Systems, Measurement and Control*, Vol.111, No.1, pp 24-30
- Slotine, J. J. E.; Li, W. P. (1989). Composite adaptive control of robot manipulators, *Automatica*, Vol. 25, No. 4, p 509-519,
- Spong, M. W. (1992). On the Robust Control of Robot Manipulators, *IEEE Transactions on Automatic Control*, Vol. 37, no. 11, pp. 1782-1786

- Cheah, C. C.; Liu C. & Slotine, J. J. E. (2006). Adaptive Tracking Control for Robots with Uncertainties in Kinematic, Dynamic and Actuator Models, *IEEE Transactions on Automatic Control*, Vol.51 No.6, pp.1024-1029
- Kwan, C.; Lewis, F. L. (2000). Robust backstepping control of nonlinear systems using neural networks, *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, Vol. 30, No. 6, pp.753-766
- Sanger, T. D. (1994). Neural network learning control of robot manipulators using gradually increasing task difficulty, *IEEE Transactions on Robotics and Automation*, Vol. 10, No. pp.323-333
- Jung, S.; Yim, S. B. (2001). Experimental Studies of Neural Network Impedance Control for Robot Manipulators, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.3453-3458
- Yu, W. S.; Wang, G. C. (2001). Adaptive Control Design Using Delayed Dynamical Neural Network for a Class of Nonlinear Systems, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.3447-3452
- Kim, Y. H.; Lewis, F. L. (1999). Neural Network Output Feedback Control of Robot Manipulators , *IEEE Transactions on Robotics and Automation*, Vol.15, No. 2, pp.301-309
- Jung, S.; Hsia, T. C. (1996). Neural Network Reference Compensation Technique for Position Control of Robot Manipulator, *IEEE Conference on Neural Network*, pp 1735-1741, June
- Barambones, O.; Etxebarria, V. (2002). Robust Neural Control for Robotic Manipulators, *Automatica*, Vol.38, pp.235-242
- Clark, C. M.; Mills, J. K. (2000). Robotic System Sensitivity to Neural Network Learning Rate: Theory, Simulation, and Experiments, *The International Journal of Robotics Research*, Vol.19,No.10, pp.955-968

Performance Evaluation of Autonomous Contour Following Algorithms for Industrial Robot

Anton Satria Prabuwono*, Samsi Md. Said**,
M.A. Burhanuddin*** and Riza Sulaiman*

**Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia*

***Faculty of Electrical and Automation Technology, TATi University College*

****Faculty of Information & Communication Tech., Universiti Teknikal Malaysia Melaka
Malaysia*

1. Introduction

The current trend towards industrial robotics requires the development of simple programming techniques. Programming by demonstration has emerged as one of the most promising solutions for effective programming of robot tasks (Ikeuchi & Suehiro, 1994; Zollner et al., 2002). Contour following is one of basic task in industrial robot manipulation. In this task, the robot is holding a tool to follow the contour of an object whose shape and pose are often unknown (Mi & Jia, 2004). These applications include part polishing, inspection, sealing, painting, cleaning, modeling, etc. During the following process, the tool is constrained on the surface to maintain contact force while moving along some tangential direction.

In order to use robotics for such application two sequences of step need to be considered, the programming phase and the playback phase. In the programming phase, teaching a group of points is required while for playback phase, the robot Tool Centre Point (TCP) will follow the taught points recorded previously. This programming phase especially for contour following application is quite tedious and time consuming. For example, in order to track an arc, the robot programmer needs to manually use teaching box or teaching pendant to jog (powered motion) the robot Tool Centre Point to three points that enclosed an arc. For a complex contour, several series of three points must be taught, besides finding the optimum process parameter (voltage, current and electrode speed for arc welding application) related to those points. Next, the motion instruction, speed and type of termination that describes the closeness of zoning radial distance to the taught points needs also to be defined. The programmer must iterate the points and process parameters several times until the optimum combination are achieved. In comparison to assembly operation where the programmer just need to teach few points such as approach, insert and depart points, contour following for painting, arc welding and sealing application requires a large number of points recorded and at the best location. After all the best program and process

parameters are achieved for one sample part, the same quality is expected for the subsequent parts in a batch. This expectation alone poses difficult challenges to the industry since parts do vary dimensionally due to inaccuracy in manufacturing and joining operation. For example, in welding job such as gas-metal arc welding (GMAW) process the part expands dimensionally since the volume of molten material in the weld bead is proportional to the heat input (Tomizuka et. al., 1980). Furthermore, the current Flexible Manufacturing System (FMS) concept requires different kind of parts variations for one production run. This means that a great number of robot programming is required to cater parts variations and uncertainties per production run compared to the old days of batch or mass production concept.

This work describes a study on developing several types of simple algorithms in order to automate the manual programming process. The main objective of the study is to present the performance evaluation of those algorithms for autonomous contour following task. The algorithms have been developed and tested using Adept Selective Compliant Assembly Robot Arm (SCARA). These algorithms include adapting gradient method, staircase method, and sweeping radius method .

2. Automation and Enhancement of the Robot Teaching Process

Certain robot application requires complex and tedious teaching procedures where skilled programmer needs to iterate the teaching process several times until the disposal of opposing parameter are achieved for optimum robot contour tracking program such as in the arc welding, windshield glass sealing and painting application. Normally it takes days to teach several optimal arc welding points especially in a flexible, just-in-time, and CAD customized production approach (Samsi & Nazim, 2005).

A lot of efforts have been done in automating and enhancing the teaching process of the discussed applications above. Yuehong et al. (2004) explained that for the application of robot to complex tasks, which often requires its end-effector to come into contact with the unknown surface, it is often necessary to control not only the position but also the force exerted by the end-effector on an object, otherwise the arising contact forces may damage the object and the robot system. Active tracking of unknown surface is also a real problem in the industrial world. Force sensing and control, similar to vision, is a fundamental part of robot. It essentially simulates human tactility. Over 70% information can be obtained by human vision from external environments. However, over 2/3 of human brain and neural system are used to manage the tactility. The senses of force, contact, press, and slide all belong to human tactility. Compared with the others, force sensing is the most synthetic and complex means, because it consist of touching the object with the robotic system while maintaining optimum path. Therefore, active tracking of unknown surface using force sensing and control technique has become one of the most important studies in robotics. But, the strategy of force sensing and control has not been effectively solved because of the acute contradiction between the strict requirement of the force exerted by robot on external environment and the stiffness of position-servo and mechanism of robot in the free space. Cartesian space force control is achieved by controlling every actuators torque in joint space. The simplest torque control is an armature current control loop, which is often called an inner current loop.

Robotics welding process is quite challenging for robot programmers since different parts are to be welded demanding an intelligent robot-welding concept because of the individual part dimension deviations due to heat and low tolerance manufacturing process. The realization of such ambitious goals leads to the use of sensors which provide the robot with the necessary information, so that it can interact within their environment (Hewit, 1996). Further requirements are that the robot must hold the electrode at the correct orientation and distance to the seam and move at a constant velocity so that a constant of material flow into the joint. This problem becomes too complex for three dimensional objects than on flat plates, and often requires geometric modeling to plan the robot motion. Several types of arc welding sensors are under investigations (Prinze & Gunnarson, 1984). The most promising are preview sensing, through the arc sensing, and direct-arc sensing. Variants of these are available commercially (Hanright, 1984). The majority of research into seam tracking focuses on non-contact sensors. The most promising sensing method is optical triangulation, where a rotating mirror scans a thin beam of light across the objects to be welded (Oomen & Verbeck, 1983). Preferentially, the robot should autonomously find and precisely weld metal joining paths in order to fulfill some given manufacturing tasks. This solution can be expanded further to surface following task in painting application and the data measured can be used for machining purpose as in coordinate measuring machine (CMM).

Adolfo et. al. (2001) used predictive look ahead sensor guided technique using CCD camera to capture contour line and approximate them with a polynomial of certain degree to smooth out the curve and neglecting any experimental error. It was shown how a smart tool integrated with sensor that can look ahead and plan the trajectory can be attached and reattached in a flexible manner to robotic manipulators. This solution can automate industrial programming processes in an intelligent manner. Those are objects of intense research efforts in the field of artificial intelligence (AI): to build machines that consider the information captured from the surrounding environment in a proper (intelligent) manner. With the support of sensors the working trajectory of the robot can be obtained within a certain sensor field which will be used here as the minimization of the tracking error. The main contribution of the work was an on-line tracking optimization scheme for sensor guided robotic manipulators by associating sensor information, manipulator dynamics and a path generator model. Experimental results on implementation of a CCD-camera guided hydraulic robot and a welding robot demonstrates the proposed approach.

Intelligence and flexibility are two essential features in a smart mechatronics product. In his research work several problems were addressed such as sensor integration, real-world modeling, trajectory path planning, task-level planning and execution, and the control of the robotic system as a whole. Moreover, building autonomous smart tool that enhances the welding process provides a stringent test bed for new concepts and approaches in both hardware and software which is very near to commercialization. The concentration of the work was on the design and development of an autonomous platform, using mechatronics to implement intelligent behaviors, with the help of a industrial robot controller interfaced with sensors (Gopalakrishnan et. al., 2004).

In the automation of arc welding, it is necessary to find the starting point for the welding and control the welding torch precisely along the welding line. However, in the actual welding, there are a lot of external variations including the change of the weld gap, the movement of work piece due to local thermal expansion, and the existence of obstacles and weld tacks. They are mainly to detect a groove location with high precision by the slit

lighting method (Awahara & Taki, 1979), gap by an edge detection operator (Inoue, 1979), and to detect a weld line on a thin plate by considering lighting condition (Suga et al., 1992). Rasol et al. (2001) successfully created a prototype system that automates the teaching of spot welding process by building a prototype knowledge based expert system software. Two main areas were considered such as automatic setting of spot welding parameters and automated placing of spot welds using a robot fitted with a welding head. The prototype system reduces the teaching time, improves positioning of spot welds location, reducing unnecessary spot weld points and increases the flexibility level of manufacturing system.

Andersson & Johansson (2000) developed method to implement a control strategy for wood carving operations. A control method that improves robot control and supports simpler programming, based on a wrist mounted force/torque-sensor, is proposed for the wood carving process. Their work describes the structure and control algorithm of the system and how different machining parameters affect the cutting forces. The evaluation of the system showed that it is possible to control the cutting depth at speeds up to 7.5 mm/s by adjusting the rake angle of the tool to obtain a nominal force.

All the previous works reviewed require force sensor, vision system, laser sensor and CAD data. These entire devices are quite expensive relatively and the algorithm that support them also quite complex. Majority of the works require contact sensing between end-effector and work piece which is undesirable in term of tool reliability and work piece quality in certain process. These research vacuums justify the unique scope of undertaking research to develop and design a cheap and non contact simpler algorithm program which is being tested using real industrial machines. So, a research that employs a simpler solution system which uses only one cheap discrete sensor with simple algorithm, and taking advantage of common industrial robot controller, is justified.

3. Contour Following Methods

3.1 Adapting Gradient

This method requires a sampling distance at time T where the robot TCP traversing a distance dX defined by user. The sampling distance dX is the critical parameter that defines the contour resolution measured just like sweeping segment radius r as in the method discussed earlier. The longer the distance dX , the coarser the curve modeled, and the smaller the r value the finer the curve is modeled (Prabuwno & Samsi, 2007). Fig. 1 shows the details of adapting gradient formulation.

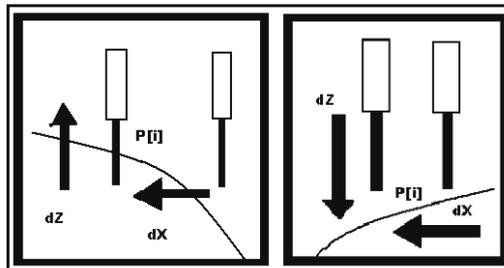


Fig. 1. Adapting gradient formulation.

After the initial sampling distance X , then the sensor on tool tip will sense if the contour is within the sensor sensing range. If the sensor reading is on, that is mean that the tool is inside the contour and need to be brought upward. Fig. 2 shows the adapting gradient algorithm.

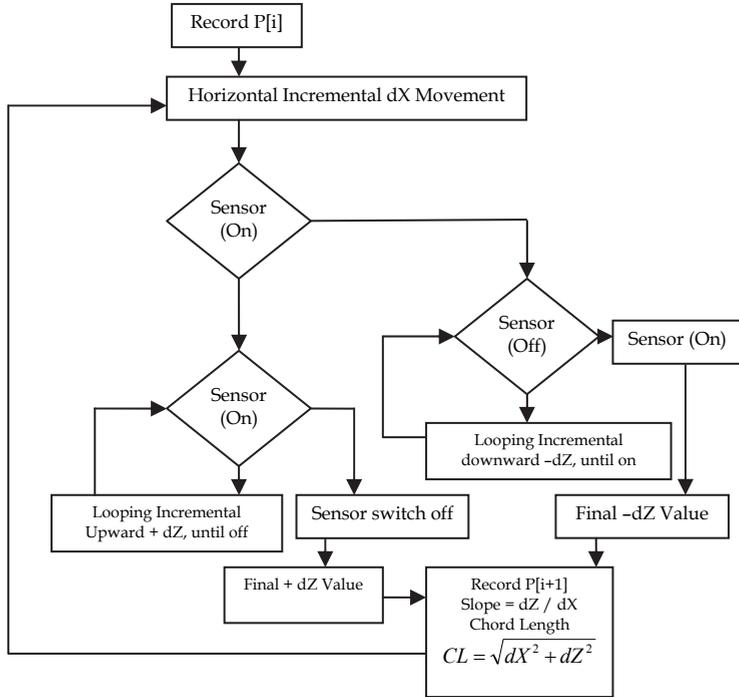


Fig. 2. Adapting gradient algorithm.

On the other hand if the sensor is off then that means the tool is too well upward above the contour and needs to be brought downward incrementally. Incremental downward or upward movement is then employed depending on the sensor status whether on or off . At the end of each cycle the position is stored. The process is being repeated until the whole complex contour is measured. The incremental distances dX and dZ variable are very important in measuring the chord segment of this method. Unlike sweeping chord algorithm that maintains constant chord segment length while measuring and recording the α angle, the adapting gradient method only maintaining constant sampling distance dX while having variable dZ measured and recorded.

The chord length and slope of the ratio of dZ/dX are functions of dX and dZ . The upward adapting motion will approximate incrementally the positive gradient along contour while downward adapting motion will approximate the negative gradient along contour. Using this method the position $p[i]$ after each measuring cycle is stored in the position database and to be used repeatedly in the playback mode. In this way the whole contour is being approximated and the positions stored (refer to Fig. 1 and Fig. 2). The positions can be used for $P_{i+1} = P_i D(i+1)$. The drive function that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & n_A \bullet dX \\ 0 & 1 & 0 & s_A \bullet dX \\ -S(\theta) & 0 & C(\theta) & a_A \bullet dZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Total trajectory point generated at point N as follows:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \quad (2)$$

$$\delta_{XZ} slope = \frac{a_A \bullet dZ}{n_A \bullet dX} \quad (3)$$

The advantage about this method is that the numbers of points measured and stored are constant throughout horizontal contour length regardless of contour complexity (only chord length differs). Anyway, for short sensing range and at high gradient complex contour this method is quite dangerous which can cause collision because it penetrates into the contour solid areas. The decision to measure variable distance dZ is not consistent, from the solid to empty contour edge at positive gradient and from inside the empty contour to the edge of contour, the condition is not uniform. Several precautions must be taken care proactively such as the sampling distance dX and speed \dot{dX} dynamic overshoot for high gradient and short sensing distance.

3.2 Staircase

Staircase method is introduced whereby it requires the robot TCP climbing a constant distance LZ and the sampling distance LX . Then the tool moves downward incrementally until the contour is within the sensor sensing range (Prabuwono et al., 2008). Then it repeats the previous process of climbing upward a constant distance LZ and traversing horizontally a distance LX . At this moment the new position of point $P(i+1)$ is being recorded and the difference $dZ(i)$ of $LZ(i+1)$ and $dZ(i)$ is calculated. This method is maintained constantly dX and dZ . The $dZ(n)$ is the critical parameter that defines the contour resolution measured just like sweeping segment radius r . Fig. 3 shows the staircase method formulation in several sequences.

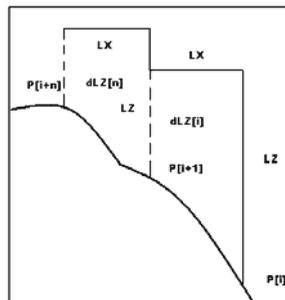


Fig. 3. Staircase method formulation in several sequences.

The longer the distance LX , the coarser the curve modeled, and the smaller the LX values the finer the curve being modeled. In this way the whole contour is being approximated and the positions stored. The positions can be used for $P_{i+1} = P_i D(i+1)$.

The drive function that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & n_A \bullet (LX[i]) \\ 0 & 1 & 0 & s_A \bullet (LX[i]) \\ -S(\theta) & 0 & C(\theta) & a_A \bullet (LZ[i] - ABS[dLZ[i]]) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\delta_{xz} slope = \frac{a_A \bullet (LZ[i] - ABS[dLZ[i]])}{n_A \bullet (LX[i])} \quad (5)$$

Total trajectory point generated at point N as follows:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \quad (6)$$

Unlike sweeping chord algorithm that maintains constant chord segment length and while measuring and recording the α angle, the offset staircase method only maintains constant sampling distance LX and LZ while having variable dZ measured and recorded. The chord length and gradient slope of the ratio of dZ/LX are a function of LX , LZ and $dZ(n)$. The upward staircase (dZ is positive value) will approximate incrementally the positive gradient along contour while downward staircase (dZ is negative value) will approximate the negative gradient along contour. Using this method the contour information is stored in the position database to be used repeatedly in the playback mode. The sampling distance dX is constant throughout the horizontal X axis distance along the complex contour regardless of gradient slope measured.

The measurement variable is only distance dZ where through some manipulations the slope dZ/dX at constant segment $CL = \sqrt{dX^2 + dZ^2}$ is being derived and stored at program database. The problem with this method is the nature of uniform constant horizontal segment regardless of variable slope gradient. The number of positions stored can be calculated depending upon the horizontal contour distance and sampling distance dX . On the other hand the chord length CL will vary according to the measured dZ and constant sampling distance dX . Since dX is constant the slope will be decided by the dZ length, the higher the dZ value, the higher the length of chord segment. The advantage of this method is that the numbers of points measured and stored are constant throughout horizontal contour length regardless of contour complexity. Fig. 4 shows staircase algorithm.

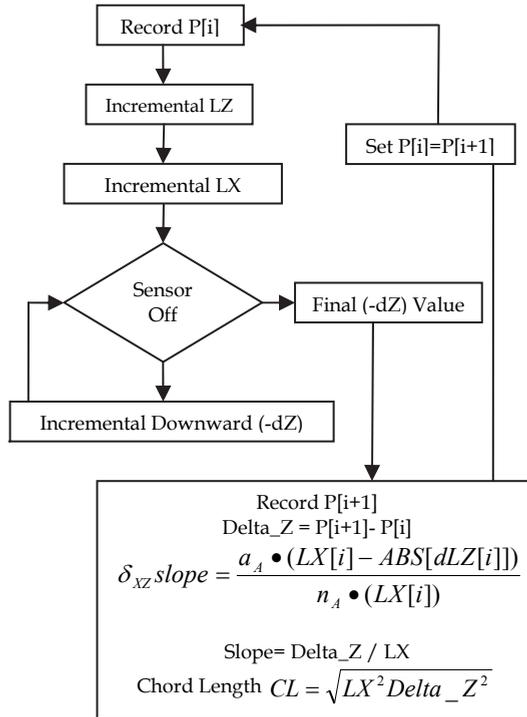


Fig. 4. Staircase algorithm.

3.3 Sweeping Radius

This method is to automate the incremental measuring motion utilizing the gross output of total positions and yaw orientation angles from task planning algorithm. Another important point is the slope gradient measurement at any knot points for correcting the optical sensor reflectance correction factor along the contour positive and negative slope gradient. The complex contour of any different gradient is being approximated by segment of chord distance r . The smaller the r value, the higher the accuracy of contour shapes being measured but at a higher computation cost (refer to Fig. 6 and Fig. 7). The first part of robot program is to measure the incremental position and slope along the contour gradient and store the positions recorded in the database. The stored locations will be used repeatedly for playback purpose in subsequent passes (running a production part program). A low cost digital optical sensor is being fitted into a tool holder as shown in Fig. 5.

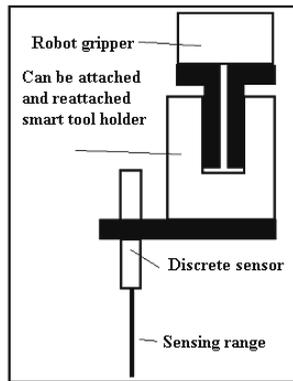


Fig. 5. Smart tool holder with sensor.

This sensor has a sensing range of about 25 millimeter, which is important to avoid tool colliding with the contour surface. The robot TCP started by moving upward in Z axis a distance r and suddenly sweeping downward in radius r from angular step $\alpha = 0$ to $\alpha = 180$ in one degree step. The sweeping motion is terminated when the contour shape is within the digital optical sensing range. In this way, at certain angle α and segment chord length r , the new relative point and slope is measured and recorded.

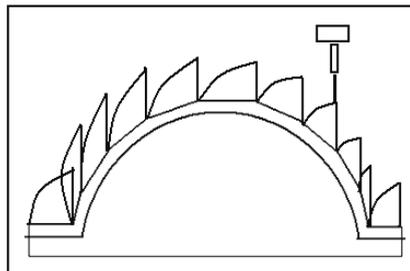


Fig. 6. Semi-circle sweeping radius method.

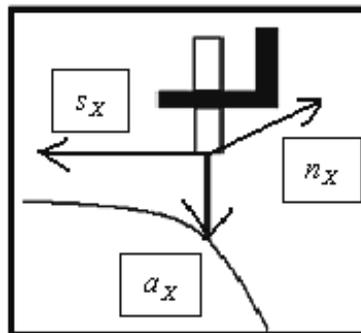


Fig. 7. Tool axes.

Measuring slope is also important for correcting the optical digital sensor reflectance factor. In this way the whole contour is being approximated and the positions stored. The positions can be used for $P_{i+1} = P_i D(i+1)$. The drive function that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & n_A \bullet r(\sin \alpha) \\ 0 & 1 & 0 & s_A \bullet r(\sin \alpha) \\ -S(\theta) & 0 & C(\theta) & a_A \bullet r(\cos \alpha) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Total trajectory point generated at point N for playback purpose in subsequent passes is:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \tag{8}$$

Constant chord segment will follow complex contour regardless of gradient slope measured. The measurement variable is only angle α where through some manipulation the slope at constant segment will be derived and stored at program database. Fig. 8 shows sweeping radius algorithm.

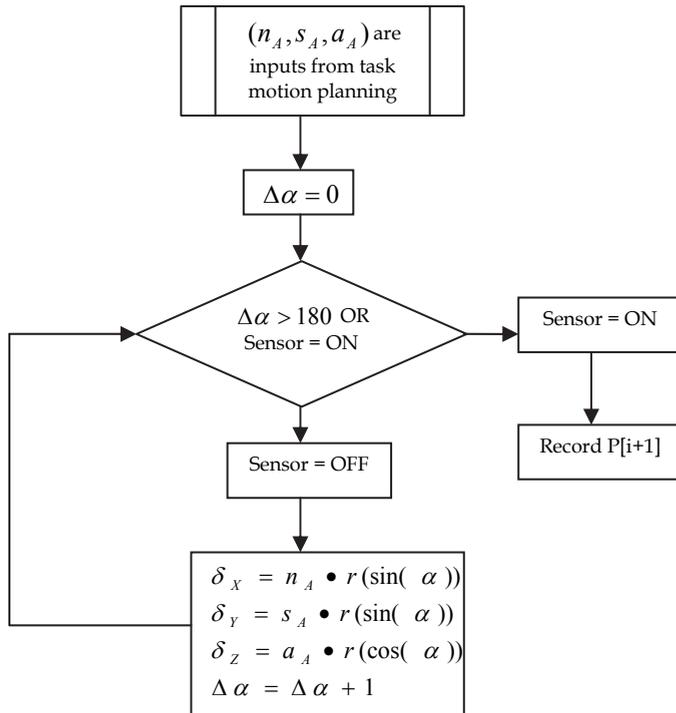


Fig. 8. Sweeping radius algorithm.

The problem with this method is the nature of constant segment will measure the flat surface with the same intensity as high slope gradient (the measurement should vary according to gradient slope which should be less in flat surface contour and a bit high on higher slope gradient contour).

4. Task Planning Formulation

4.1 Cartesian Trajectory Planning

In the path planning process, two choices were given either keying several points of X-Y Cartesian coordinate or manually jog the robot tool centre point (TCP) to the desired initial and final location defining one line segment. By jogging, user will bring the robot end effector TCP using powered motion via a teach pendant (refer to Fig. 9 and Fig. 10). This process will be repeated for multi segment lines defining a closed curve or an open curve. The information from these two initial and final points will be used to adjust the end effector TCP yaw orientation rotation angle from initial point heading to final point.



Fig. 9. Initial line top view position teaching (point recording).

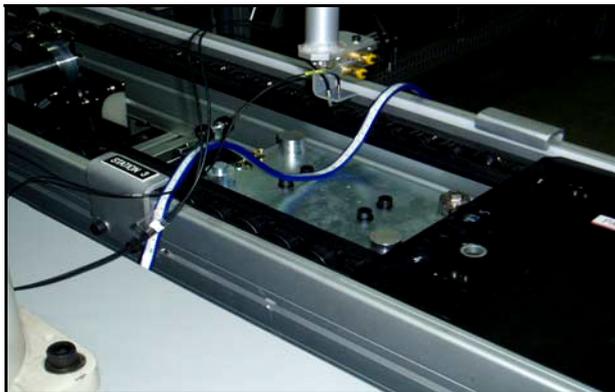


Fig. 10. Final line top view position teaching (point recording).

The initial and final location of robot TCP taught locations are actually homogeneous transformation matrices describing both orientation and position of the TCP with respect to robot base world coordinate system. The homogeneous transformation matrices are shown below which role is to describe the initial and final location of taught points.

$$P_{initial} = B = \begin{bmatrix} n_A & s_A & a_A & p_A \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x^A & s_x^A & a_x^A & p_x^A \\ n_y^A & s_y^A & a_y^A & p_y^A \\ n_z^A & s_z^A & a_z^A & p_z^A \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$P_{final} = A = \begin{bmatrix} n_B & s_B & a_B & p_B \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x^B & s_x^B & a_x^B & p_x^B \\ n_y^B & s_y^B & a_y^B & p_y^B \\ n_z^B & s_z^B & a_z^B & p_z^B \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

The final point is related to initial point through the Gross X-Y plane task planning matrix as follows:

$$P_{final} = P_{initial} T(N) \quad (11)$$

Multiplying by the inverse of $P_{initial}$ for both sides, the gross motion X-Y plane task planning matrices are derived as follows (\bullet indicates dot product of two vectors):

$$T(N) = \begin{bmatrix} n_A \bullet n_B & n_A \bullet s_B & n_A \bullet a_B & n_A \bullet (p_B - p_A) \\ s_A \bullet n_B & s_A \bullet s_B & s_A \bullet a_B & s_A \bullet (p_B - p_A) \\ a_A \bullet n_B & a_A \bullet s_B & a_A \bullet a_B & a_A \bullet (p_B - p_A) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Information from gross X-Y plane task planning matrices can be used to describe the tool position component P_x, P_y, P_z in Cartesian coordinate, the rectilinear distance L and orientation yaw angle β_z as described below:

$$Px = n_A \bullet (p_B - p_A) \quad (13)$$

$$Py = s_A \bullet (p_B - p_A) \quad (14)$$

$$Pz = a_A \bullet (p_B - p_A) \quad (15)$$

$$L = (P_x^2 + P_y^2 + P_z^2)^{\frac{1}{2}} \tag{16}$$

$$\beta_Z = \tan^{-1} \left\{ \frac{[(n_A \bullet a_B)^2 + (s_A \bullet a_B)^2]^{\frac{1}{2}}}{a_A \bullet a_B} \right\} \quad 0 \leq \beta_Z \leq \pi \tag{17}$$

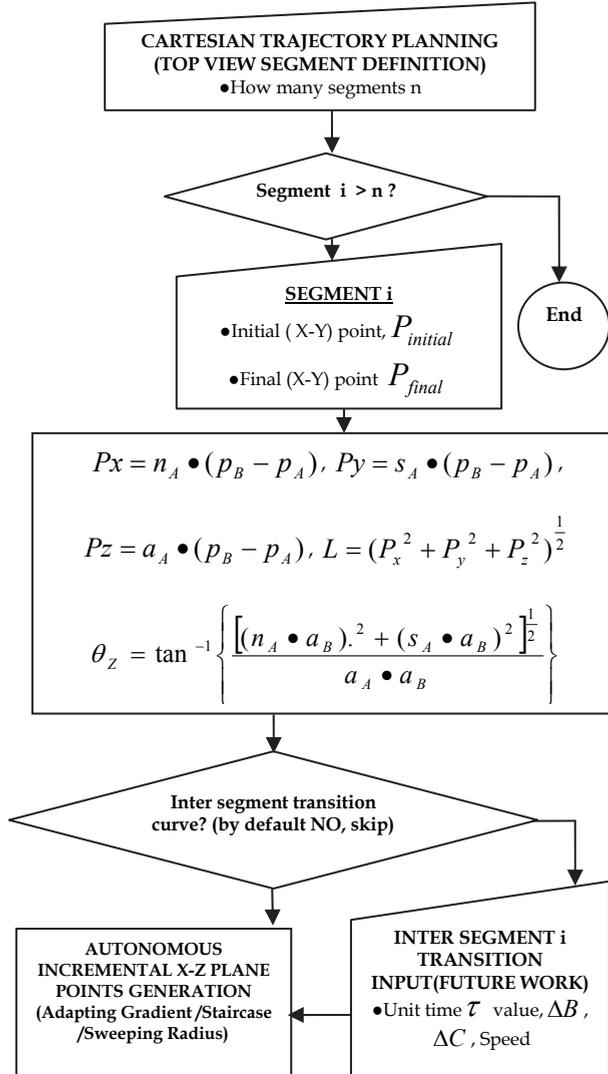


Fig. 11. Hybrid three dimensional path planning.

Fig. 11 shows the hybrid three dimensional path planning. In this diagram, the selected algorithm will replace autonomous incremental X-Z plane points generation. The n_A , S_A , a_A and θ_z will become input into incremental X-Z plane segment differential chord planning which is the major problem to be solved in this work.

4.2 Drive Transform Model

Smart sensor feedback and programming algorithm will guide the TCP to approximate the curve with a straight line segments that knot from points to points in three dimensional Cartesian X-Y-Z plane. The measured knot points and segment slope at any points will be stored in the database and will be used repeatedly in robot part program playback mode. The objectives of automating tedious and time consuming contour tracking programming process will be achieved. Adapting gradient algorithm will further explain the incremental position of δ_x , δ_y and δ_z of general incremental drive transform described in Equation 19. For start, in order to simplify the mathematical formulation, four degree of freedom Adept SCARA robot is used. In future research, a six degree of freedom robot can be used to test the robustness and applicability of the algorithms. Utilizing drive transform equation for four degree of freedom SCARA robot will simplify a lot of things (Paul, 1972; Paul, 1979). For example, only one yaw orientation angle exist. Then, the chord segment relative path transformation drive transform is being decomposed only into one rotation matrix to orientate tool about Z axis and one straight line translation matrix also along tool axis. In order to achieve the motion between two consecutive Cartesian knot points, the derivation of segment drive transform is very useful since motion from i to $i + 1$ is related to drive transform as:

$$T_4(i+1) = C_{workobject} P_i D(i+1) ({}^{tool}T_{i+1})^{-1} \quad (18)$$

$T_4(1 + 1)$ is the transformation stored to the database and contain both tool position and orientation at any points which also becomes input to the inverse kinematics routine in order to get local coordinate of individual robot joint angles (another joint level cubic polynomial trajectory planning or differential Jacobian method which is not discussed here). After some mathematical operation, the position of consecutive knot points at beginning from i to end of segment $i + 1$ is a function of drive transform as $P_{i+1} = P_i D(i+1)$.

The general transformation matrices drive transform that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & \delta_x \\ 0 & 1 & 0 & \delta_y \\ -S(\theta) & 0 & C(\theta) & \delta_z \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (19)$$

The yaw orientation angle θ is actually an input from gross motion task planning that was discussed previously. The detail of incremental position in Cartesian X-Y-Z three dimensional plane δ_x , δ_y and δ_z explained in several individual alternate algorithm developments in the previous section just to automate this differential relative motion. Incremental drive transform will describe the final position at any point N is generated as:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \quad (20)$$

The related transformation at any point N which became input to inverse kinematics routine for joints space trajectory planning as follows:

$$T_4(N) = \prod_{i=1}^N C_{\text{worobject}} P_i D(i+1) ({}^{001}T_{i+1})^{-1} \quad (21)$$

5. Experiment and Results

5.1 Experiment

The V+ programs from those algorithms will be written and will be tested using an Adept SCARA robot. The contours traced by robot TCP on the semicircle object which has radius value of 40 millimeter along the X axis are plotted according to the algorithm used. All actual contour coordinate traced at any point will be compared to the known geometry contour shape equation in order to find tracking error at any point i . The error between actual coordinated traced by robot TCP and known coordinate value of semicircle geometry at any time i is:

$$\varepsilon_i = w_i - z_i \quad (22)$$

Where w_i is actual geometry and z_i is the contour traced.

Two graphs are plotted for individual algorithm such as the actual contour traced versus semicircle geometry, tracking error versus contour geometry (all along X axis). The mean of tracking error will be used to measure the performance index of proposed algorithm for all point N captured as follows:

$$\bar{\varepsilon} = \frac{1}{n} \sum_{i=1}^N \varepsilon_i^2 \quad (23)$$

Another criterion to measure the error distribution is by employing standard deviation as follows:

$$s = \left[\frac{1}{n} \sum (\varepsilon_i - \bar{\varepsilon})^2 \right]^{\frac{1}{2}} \quad (24)$$

Semicircle shape was chosen because it provides an ideal test bed and it contains all ranges of slope gradient that are available in real world. It exhibits infinity value at the very beginning point and progressing down with a finite very high positive slope. The slope decreasing into zero value in the middle of the contour and finally reaches very high negative slope at the other end along the X axis. At the very end of the semicircle contour the infinity slope reappear again. These phenomena cause high reading of Cartesian vector Z for any minute vector X displacement value. These infinity region problems will be

avoided by introducing a safety margin ranging from 0.1-2.5 millimeter at the both ends of the semicircle geometry. The numbers of sampling measurement points depend on the method employed so the sampling points of every method do vary depend on the method employed. It is anticipated that the tracking error value will be quite high in certain slope region of contour gradient (Prabuwono et al., 2009). Fig. 12 shows the four degrees of freedom SCARA robot that used in this study.



Fig. 12. The four degrees of freedom SCARA robot.

5.2 Results

The actual contour traced and the tracking error along contour, matching the semicircle geometry of radius 40 millimeter is plotted. For adapting gradient method, the enlargement of mean of tracking error with the value of - 0.3773 millimeter and the standard deviation of tracking error with the value of 2.3085 millimeter are shown in Fig. 13 and Fig. 14 respectively. The safety margin of 0.1 to 1 millimeter is allowed at the beginning and near to the end of semicircle object in order to avoid measuring the very high slope at those regions. The adapting gradient measuring advance parameter of 1 millimeter is chosen for this contour following experiment. The total sample of good 79 points was collected over 80 millimeter horizontal measuring distance.

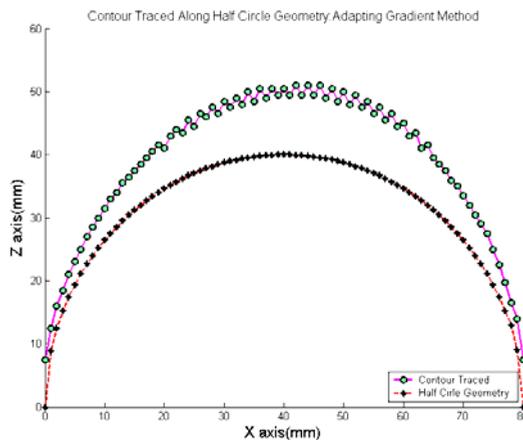


Fig. 13. Contour traced along half circle geometry with adapting gradient method.

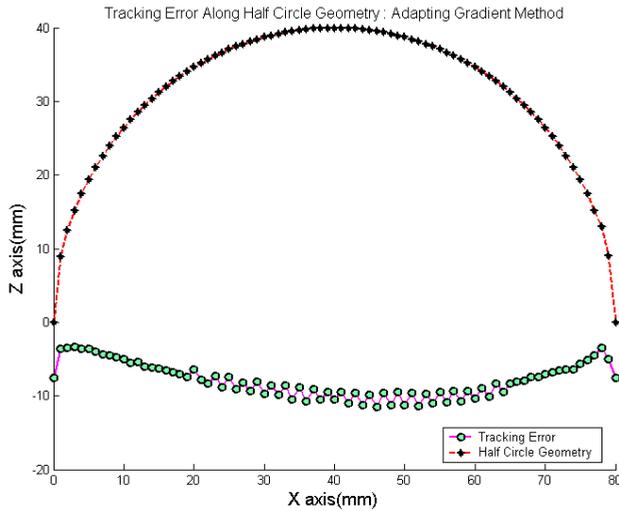


Fig. 14. Tracking error along half circle geometry with adapting gradient method.

For staircase method, the enlargement of mean of tracking error with the value of 3.4011 millimeter and the standard deviation of tracking error with the value of 1.8412 millimeter are shown in Fig. 15 and Fig. 16 respectively. The safety margin of 0.1 to 1 millimeter is allowed at the beginning and near to the end of semicircle object in order to avoid measuring the very high slope at those regions. The staircase measuring advance parameter of 1 millimeter is chosen for this contour tracking experiment. The total good sample of 78 points was collected over 80 millimeter horizontal measuring distance.

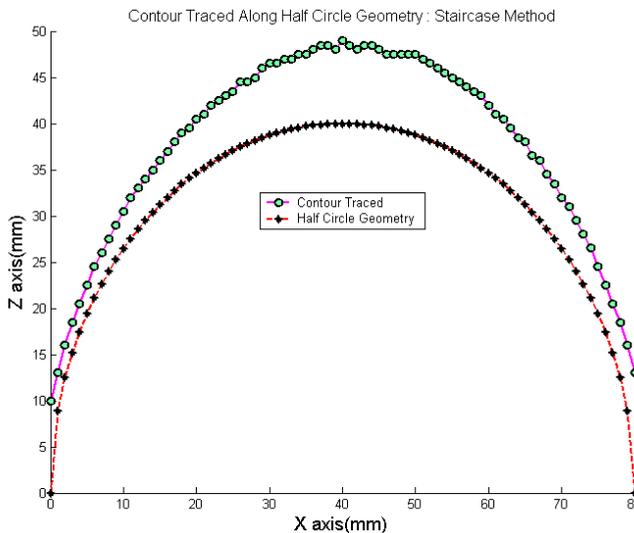


Fig. 15. Contour traced along half circle geometry with staircase method.

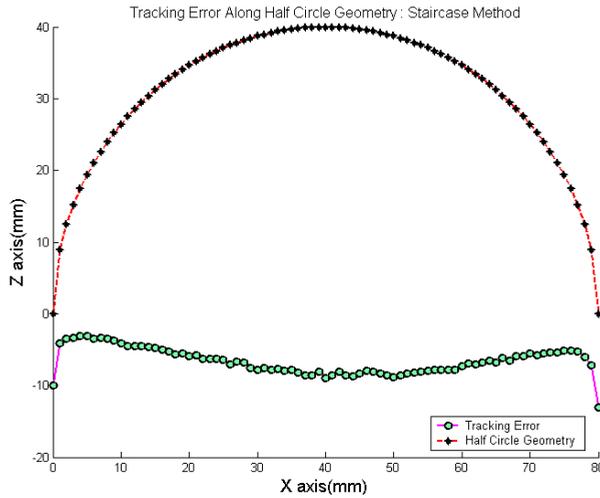


Fig. 16. Tracking error along half circle geometry with staircase method.

For sweeping radius method, the enlargement of mean of tracking error with the value of 0.2101 millimeter and the standard deviation of tracking error with the value of 3.2663 millimeter are shown in Fig. 17 and Fig. 18 respectively. The safety margin of 0.1 to 1 millimeter is allowed at the beginning and near to the end of the semicircle object in order to avoid measuring the very high slope at those regions. The sweeping radius parameter of 1 millimeter is chosen for this contour tracking experiment. The total sample of 67 points was collected over 80 millimeter horizontal measuring distance.

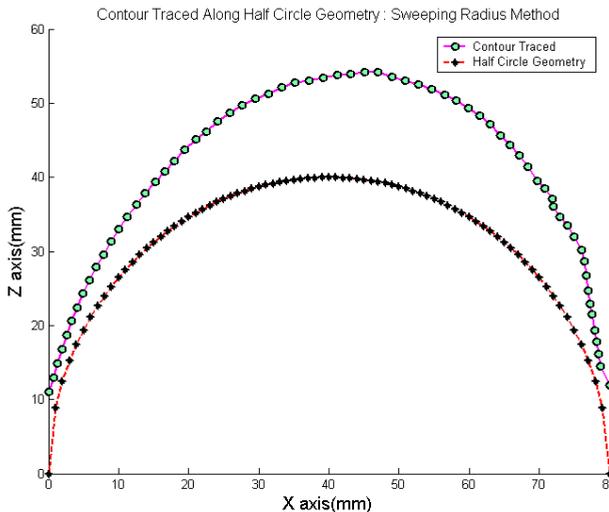


Fig. 17. Contour traced along half circle geometry with sweeping radius method.

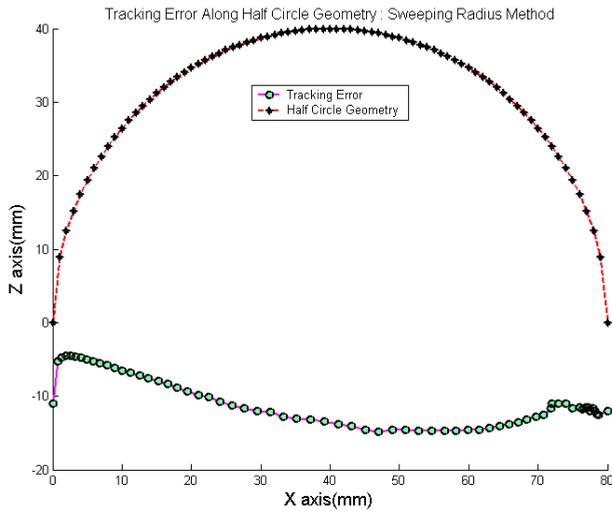


Fig. 18. Tracking error along half circle geometry with sweeping radius method.

6. Performance Evaluation

Fig. 19 summarizes all different methods for path traveling in order to evaluate their efficiency among all algorithms or methods implemented previously. The efficiency is measured with regard to the least tracking error standard deviation value and the shortest distance traveled. The best is assumed to be the least tracking error standard deviation value with the shortest sampling distance. In Fig. 19, the adapting gradient method follows path 1A to 2A, while the sweeping radius method starts from path 1B to 2B. The staircase method is the path that started from 1B to 4D.

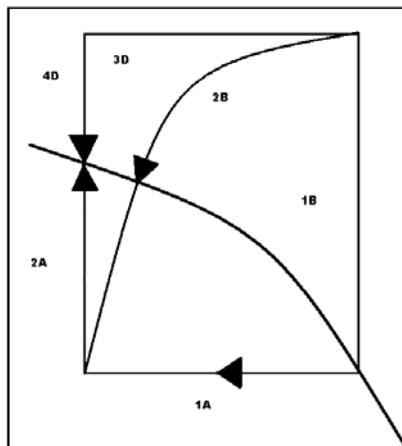


Fig. 19. Path comparison among three different contour following methods.

It is clearly seen in that the staircase method has the longest path followed by the adapting gradient method. The shortest distance is done by the sweeping radius method. With the same speed, it seems that the staircase method takes the longest time while sweeping radius is the fastest of all methods.

All the results are tabulated in Table 1. The adapting gradient method consumes medium teaching time at standard deviation value of 2.3085 millimeter, while the staircase method consumes the longest teaching time at standard deviation value of 1.8412 millimeter. The sweeping radius method is very efficient in term of shortest teaching path but its standard deviation value of 3.2663 is a bit high.

Criteria	Adapting Gradient	Staircase	Sweeping Radius
Mean of Error	-0.3773	3.4011	0.2101
Standard Deviation	2.3085	1.8412	3.2663
Path Length	Medium	Long	Shortest

Table. 1. Summaries of the results for three different contour following methods.

7. Conclusion

In this study, the performance evaluations of autonomous contour following task with three different algorithms have been performed for Adept SCARA robot. A prototype of smart tool integrated with sensor has been designed. It can be attached and reattached into robot gripper and interfaced through I/O pins of Adept robot controller for automated robot teaching operation. The algorithms developed were tested on a semicircle object of 40 millimeter radius. The semicircle object was selected because it exhibits the stringent test bed which provides the changing gradient gradually from steepest positive slope into zero slope of flat curve in the middle and finally to steepest negative slope. The adapting gradient method consumes medium teaching time at reasonable accuracy of standard deviation value of 2.3085 millimeter, while the staircase method consumes the longest teaching time at standard deviation value of 1.8412 millimeter. The sweeping radius method is very efficient in term of shortest teaching path but its standard deviation value of 3.2663 is a bit high. It can be concluded that the staircase method is the most accurate method, while the sweeping radius method has the shortest teaching path.

These tests exhibit the performance of algorithms used which prove its possibility to be applied in the real world application. For the future, automatic curve radius determination between straight line segments can be improved by integrating vision system for the automation of top view (X-Y coordinate) edge finding and path planning. The integration of vision system with the present study will improve the automation level of the project from two to three dimensional capabilities.

8. References

- Adolfo, B.; Sadek, C.A.A. & Leszek, A.D. (2001). Predictive sensor guided robotics manipulators in automated welding cells. *Journal of Materials Processing Technology*, Vol. 109, No. 1-2, February 2001, 13-19, ISSN 0924-0136

- Andersson, J.E. & Johansson, G. (2000). Robot control for wood carving operations. *Mechatronics*, Vol. 11, No. 4, June 2001, 475-490, ISSN 0957-4158
- Awahara, M. & Taki, K. (1979). Tracking control for guiding electrodes along joints by pattern detection of welding groove. *Transactions of the Society of Instrument and Control Engineers*, Vol. 15, 492
- Gopalakrishnan, B.; Tirunellayi, S. & Todkar, R. (2004). Design and development of an autonomous mobile smart vehicle: A mechatronics application. *Mechatronics*, Vol. 14, No. 5, 491-514, ISSN 0957-4158
- Hanright, J. (1984). Selecting your first arc welding robot - a guide to equipment and features. *Welding Journal*, Vol. 1, 41-45
- Hewit, J. (1996). Mechatronics design - the key to performance enhancement. *Robotics and Autonomous Systems*, 135-142, ISSN 0921-8890
- Ikeuchi, K. & Suehiro, T. (1994). Towards an assembly plan from observation, Part I: Task recognition with polyhedral objects. *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 3, 368-385, ISSN 1042-296X
- Inoue, K. (1979). Image processing for on-line detection of welding process (report 1): simple binary image processor and its application (welding physics, processes & instruments). *Transactions of JWRI*, Vol. 8, No. 2, 169-174
- Mi, L. & Jia, Y.B. (2004). High precision contour tracking with joystick sensor. *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, Vol. 1, 804-809, Sendai, Japan, September-October 2004
- Oomen, G.L. & Verbeck, W.J.P.A. (1983). A real-time optical profile sensor for robot arc welding. *Proceedings of the 3rd International Conference on Robot Vision and Sensory Controls*, 659-668, Cambridge, USA, November 1983
- Paul, R. (1979). Manipulator Cartesian path control. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, No. 11, 702-711, ISSN 0018-9472
- Paul, R.P.C. (1972). *Modeling, trajectory calculation and servoing of a computer controlled arm*. Ph.D. Dissertation, Stanford University, CA., USA
- Prabuwono, A.S.; Burhanuddin, M.A. & Samsi, M.S. (2008). Autonomous contour tracking using staircase method for industrial robot. *Proceeding of the 10th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV'08)*, 2272-2276, Hanoi, Vietnam, December 2008
- Prabuwono, A.S. & Samsi, M.S. (2007). Development of adapting gradient method for contour tracking in industrial robot application. *Proceeding of the 10th IASTED International Conference on Intelligent Systems and Control (ISC'07)*, 592-068, Cambridge, USA, November 2007
- Prabuwono, A.S.; Samsi, M.S.; Sulaiman, R. & Sundararajan, E. (2009). Contour following task with dual sensor logic algorithm for Adept Selective Compliant Assembly Robot arm robot. *Journal of Computer Science*, Vol. 5, No. 8, 557-563, ISSN 1549-3636
- Prinze, F.B. & Gunnarson, K.T. (1984). Robotics seam tracking. *Interim Report*, CMU-RI-TR-84-10, Carnegie-Mellon University, Pittsburgh, USA
- Rasol, Z.; Sanders, D.A. & Tewkesbury, G.E. (2001). New prototype knowledge based system to automate a robotics spot welding process. *Elektrika*, Vol. 4, 28-32
- Samsi, M.S. & Nazim, M. (2005). Autonomous and intelligent contour tracking industrial robot. *Proceedings of International Conference on Mechatronics*, 78-86, Kuala Lumpur, Malaysia, May 2005

- Suga, Y.; Takahara, K. & Ikeda, M. (1992). Recognition of weld line and automatic weld line tracking by welding robot with visual and arc voltage sensing system. *Journal of the Japan Society for Precision Engineering*, 1060-1065
- Tomizuka, M.; Dornfield, D. & Purcelli, M. (1980). Applications of microcomputer to automatic weld quality control. *ASME Journal of Dynamics Systems, Measurement and Control*, 62-68
- Yuehong, Y.; Hui, H. & Yanchun, X. (2004). Active tracking of unknown surface using force sensing and control technique for robot. *Sensors and Actuators: A Physical*, Vol. 112, No. 2-3, 313-319, ISSN 0924-4247
- Zollner, R.; Rogalla, O.; Dillmann, R. & Zollner, M. (2002). Understanding users intention: programming fine manipulation tasks by demonstration. *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, 1114-1119, Laussane, Switzerland, September-October 2002

Advanced Dynamic Path Control of the Three Links SCARA using Adaptive Neuro Fuzzy Inference System

Prabu D†, Surendra Kumar‡ and Rajendra Prasad‡
Wipro Technologies†, NJ, USA and Indian Institute of Technology‡, Roorkee, India

1. Introduction

The very precise control of robot manipulator to track the desired trajectory is a very tedious job and almost unachievable to certain limit with the help of adaptive controllers. This task is achievable to certain limit with the help of adaptive controllers but these controllers also have their own limitation of assuming that the system parameters being controlled change relatively very slow. With reference to the tasks assigned to an industrial robot, one important issue is to determine the motion of the joints and the end effectors of the robot. Therefore, the purpose of the robot arm control, as Fu et al (1987) wrote in one classical works on robotics, is to maintain the dynamic response of the manipulator in accordance with some prespecified performance criterion. Among the early robots of the first generation, non-servo control techniques, such as bang-bang control and sequence control were used. These robots move from one position to another under the control or limit switches, relays, or mechanical stops. During the 1970s, a great deal of work was focused on including such internal state sensors as encoders, potentiometers, tachogenerators, etc., into the robot controller to facilitate manipulative operation ((Inoue, H.,(1974) and Wills, et al (1975)) . Since then, feedback control techniques have been applied for servoing robot manipulators. Up till now, the majority of practical approaches to the industrial robot arm controller design use traditional techniques, such as Proportional and Derivative (PD) or Proportional-Integral-Derivative (PID) controllers, by treating each joint of the manipulator as a simple linear servomechanism. In designing these kinds of controllers, the non-linear, coupled and time-varying dynamics of the mechanical part of the robot manipulator system are completely ignored, or dealt with as disturbances. These methods generally give satisfactory performance when the robot operates at a low speed.

However, when the links are moving simultaneously and at a high speed, the non-linear coupling effects and the interaction forces between the manipulator links may degrade the performance of the overall system and increase the tracking errors. The disturbances and uncertainties in a task cycle may also reduce the tracking quality of robot manipulators. Thus, these methods are only suitable for relatively slow manipulator motion and for

limited-precision tasks can be found in the work by Sciavicco (1996). The Computed Torque Control (CTC) is commonly used in the research community. The CTC law has the ability to make the error asymptotically stable if the dynamics of the robot are exactly known. Paul, R.C (1972). However, manipulators are subject to structured and/or unstructured uncertainty. Structured uncertainty is defined as the case of a correct dynamic model but with parameter uncertainty due to tolerance variances in the manipulator link properties, unknown loads, inaccuracies in the torque constants of the actuators, and others. Unstructured uncertainty describes the case of unmodeled dynamics, which result from the presence of high-frequency modes in the manipulator, neglected time-delays and nonlinear friction. It has been widely recognized that the tracking performance of the CTC method in high-speed operations is severely affected by the structured and unstructured uncertainties. To cope with the problem, some adaptive approaches have been proposed to maintain the tracking performance of the robotic manipulator in the presence of structured uncertainty. Dubowsky(1979). To overcome the above mentioned drawback in manipulator motion control, the chapter proposed a Tuned-ANFIS controller for three links Selective Compliant Articulated Robot Arm (SCARA) manipulators. The proposed Tuned-Adaptive Neuro Fuzzy Inference System (ANFIS) controller is designed to overcome the unmodeled dynamics in the presence of structured and unstructured uncertainties of SCARA. The proposed Tuned-ANFIS Controller combines the advantages of fuzzy and neural network intelligence, which helps to improve the overall learning ability, adaptability of the ANFIS controller and also to achieve robust control of SCARA in unmodeled dynamic control. This Tuned-ANFIS Controller has been applied to the Continuous Path Control of SCARA. The result obtained through the tuned ANFIS is encouraging and shows very good tracking performance. The chapter is structured as follows, Section 2 Overview of SCARA robot control system, Section 3 describes the proposed Adaptive Neuro Fuzzy Inference System and Section 4 presents the ANFIS architecture and learning algorithm and simulation of Continuous Path Motion (CPM) of real-world applications of SCARA Robot Manipulator. Finally, conclusions are summarized in Section 5.

Prabu D† was a Master of Technology (M.Tech) graduate student in the Department of Electrical Engineering (with Specialization of System Engineering and Operations Research) of Indian Institute of Technology (IIT) Roorkee, Uttarakhand, 247667, India. This work was done during 2002 through 2004. Currently, He is working with the Wipro Technologies, USA (R&D), Brunswick City, NJ, USA. The proposed book chapter work is not connected with Wipro Technologies, USA. He can be reached for any correspondence of this paper by E-mail: prabud.iitr@gmail.com. He is a member of IEEE, ACM and CMG. Dr Surendra Kumar‡ is a faculty with the Department of Electrical Engineering, IIT Roorkee, Uttarakhand, 247667, India. E-mail: surendra_iitr@yahoo.com. He is a member of IEEE and Chapter President & Director, India Service Region, Olu Olu Institute Consortium for Teaching, Research, Learning & Development, Ruston Louisiana, USA. Dr. Rajendra Prasad‡ is a faculty with the Department of Electrical Engineering, IIT Roorkee, Uttarakhand, 247667, India. E-mail: rpdeefee@iitr.ernet.in.

2. Overview of SCARA Robot Control System

The SCARA acronym stands for Selective Compliant Assembly Robot Arm or Selective Compliant Articulated Robot Arm. SCARA is normally used in industries for pick and place operation, etc.

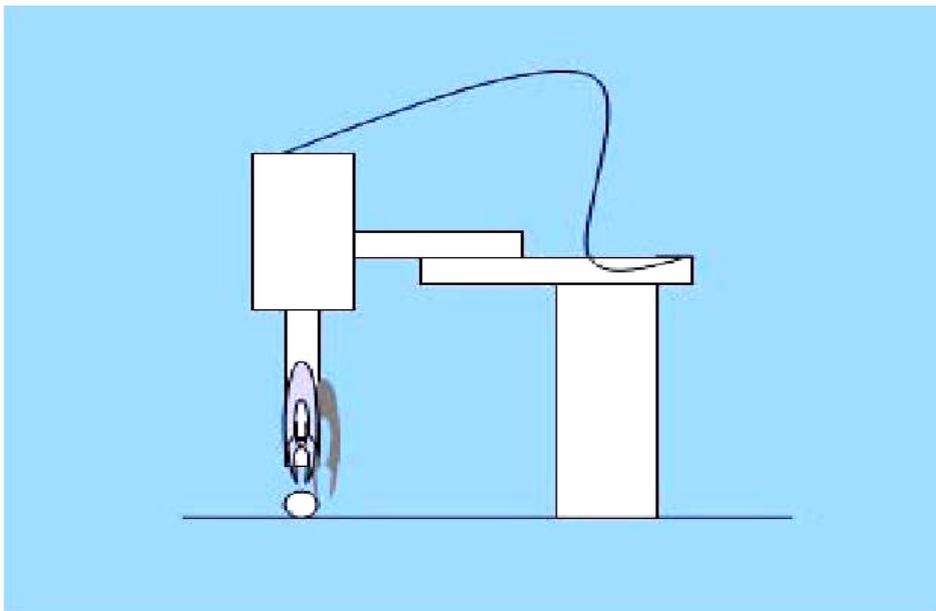


Fig. 1. Shows the SCARA Robot

The figure 1 shows the model picture of SCARA with two vertical revolute joint and one vertical prismatic joint used in this experiment. In this experiment, the dynamical model of SCARA robot is derived using Newton Euler formulation is used for simulating the CPM control using ANFIS and PD Controller. Robot Manipulator control action are exercised in the joint co-ordinates. Moreover, the dynamical model of the three links SCARA is given in many robotics books and papers. The figure 2 shows the basic ANFIS feedback control system for the CPM control of SCARA Manipulator used in this experiment.

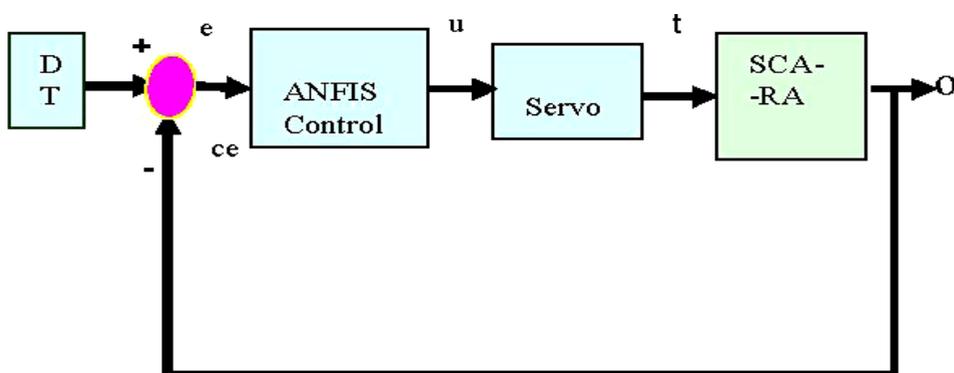


Fig. 2. Shows the ANFIS feedback control system for Continuous Path Motion control of SCARA

The feedback control system consists of ANFIS controller, servo actuating system for the SCARA Robot manipulator system and the SCARA robot manipulator system. The whole feedback system is simulated using Desired Trajectory (DT) generator to achieve minimum tracking error. The ANFIS controller is designed for the two control input viz, error (e) and change in error (ce) and one output as a control signal (u). In order to achieve the feedback control design, the output of the SCARA joint torque angles (θ) is feedback to the system. As a result the error and change error obtained at the adder of the feedback control system is given as the input to ANFIS controller for the SCARA CPM control. The ANFIS output control signals (u) are usually weak signals, which cannot able to drive the SCARA joints directly, so the signal (u) is amplified and actuated by the servo control system for SCARA manipulator joints. The outputs (t) of the servo system are given to individual manipulator links of the SCARA. The simulation model of ANFIS controller architecture is described elaborately in the section 3.

3. Adaptive Neuro Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System (ANFIS) is an artificial intelligence technique, which creates a fuzzy inference system based on the input-output model data pairs of the system. The membership functions of the ANFIS are tuned based on the nature of the input-output obtained from system or system model. The tuning of the ANFIS membership functions are done by using the Back Propagation (BP) algorithm or using least square method in combination with BP algorithm. ANFIS structures with fuzzy IF-THEN-rule based models whose consequent constituents are constants, membership functions, and linear functions as shown in figure 3. The Fuzzy logic can also be used to map complex nonlinear relations by a set of IF-THEN rules. The membership functions are designed by intuitive human reasoning. This causes three different problems. One, for different control applications, a new set of membership functions have to be developed, second, latent stability problem., Rong-Jong Wai(2003) and third, once these membership functions are developed and implemented there is no means of changing them. This means fuzzy logic lacks a learning function. In the past decades, there is a growing interest in Neural-Fuzzy Systems (NFS) as they continue to find success in a wide range of applications. Unfortunately, it has broaden the application spectrum, this paved the way to discover that most existing neural-fuzzy systems. ((Berenji(1992), Jang(1993) and Lin(1996)) exhibit several major drawbacks that may eventually lead to performance degradation. One of the drawbacks is the curse of dimensionality or fuzzy rule explosion. This is an inherent problem in fuzzy logic control systems; that is, too many fuzzy rules are used to approximate the input-output function of the system because the number of rules grows exponentially with the number of input and output variables. Another drawback is their lack of ability to extract input-output knowledge from a given set of training data. Since neural-fuzzy systems are trained by numerical input output data, the cause-effect knowledge is hidden in the training data and is difficult to be extracted. Another drawback is their inability to re-structure the internal structure. i.e. the fuzzy term sets and the fuzzy rules in their hidden layers.

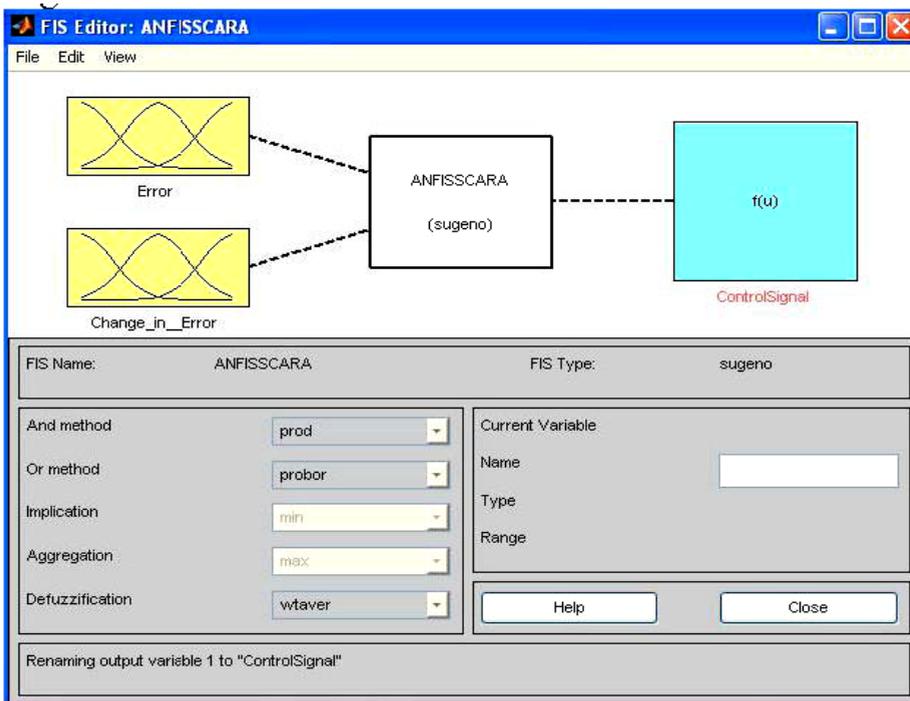


Fig. 3. The architecture of Sugeno Adaptive Neuro Fuzzy Inference System (ANFIS)

In addition, this chapter proposes a systematic approach for establishing a concise ANFIS that is capable of online self-organizing and self-adapting its internal structure for learning the required control knowledge that satisfies the desired system performance. The initial structure of the proposed ANFIS has no rule or term set node. The rule nodes and the term-set nodes are created adaptively and dynamically via simultaneous selforganizing learning and parameter learning procedures. In order to optimize the existing structure, the established rules and term sets are re-examined based on a significance index and similarity measure. Wang(1999). Thus, the rules with the index values below a prespecified threshold are pruned and the highly similar input term sets are combined. The back propagation algorithm and/or the recursive least square estimate are incorporated into the ANFIS to optimally adjust the parameters. This pruning of rule nodes and term-set nodes will result in a more concise ANFIS structure without sacrificing the system performance.

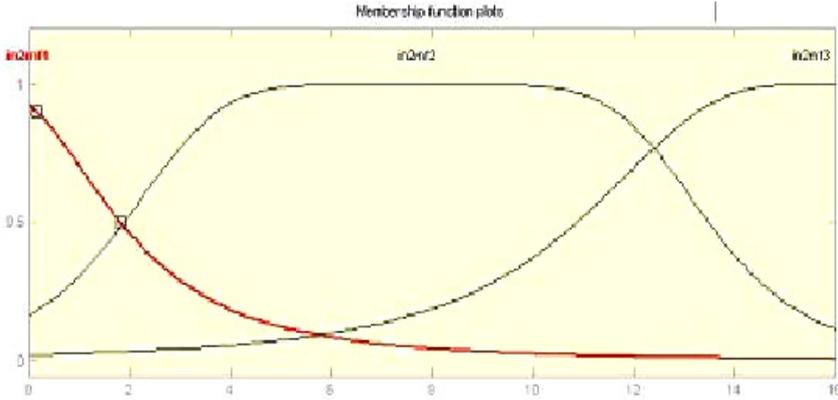


Fig. 4. Membership functions before ANFIS learning

the hybrid learning rule, a computational speedup may be possible by using variants of the gradient method or other optimization techniques on the premise parameters. Since ANFIS and radial basis function networks (RBFNs) are functionally equivalent, a variety of learning methods can be used for both of them. Figure 4 and 5 shows the membership function of the input before training and after training.

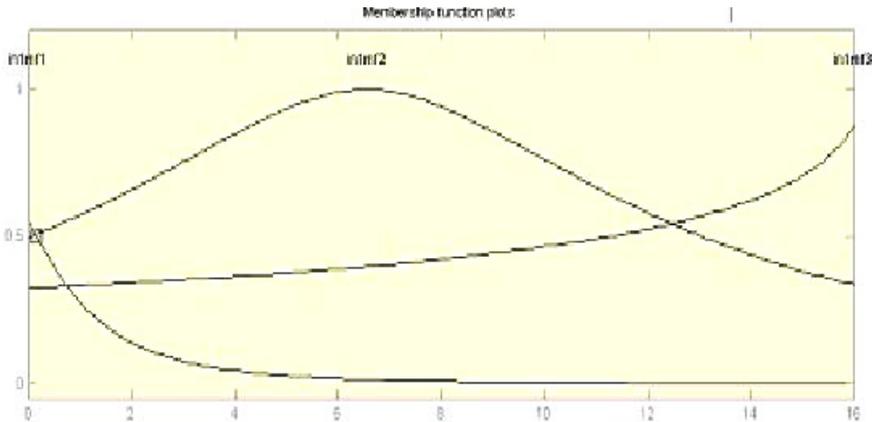


Fig. 5. Membership functions after ANFIS learning

4. Design of ANFIS Controller for SCARA

This section discuss the tracking and adaptability features of the ANFIS control applied to a three-link SCARA manipulator are tested using simulation. Figure 5 shows the architecture of the fuzzy system with the ANFIS approach. The ANFIS methodology is used to estimate the parameters of the membership functions and the consequent functions. In this experiment, ANFIS network is implemented with help of MATLAB, ANFIS toolbox. ANFIS Input variable consist of error (e) and change in error (ce), which has been describes by low, medium and high membership function in the ANFIS network. The training data (control

signal data) is obtained from the dynamic model of SCARA. The designed Sugeno -ANFIS network is trained for SCARA control signal. The back propagation algorithm and/or the recursive least square estimate are incorporated into the ANFIS to optimally adjust(tuned) the parameters (linguistic variables) of the membership function. It is found that there is a significant difference between the ANFIS membership functions before and after training as shown in the figures 4 and 5 respectively.

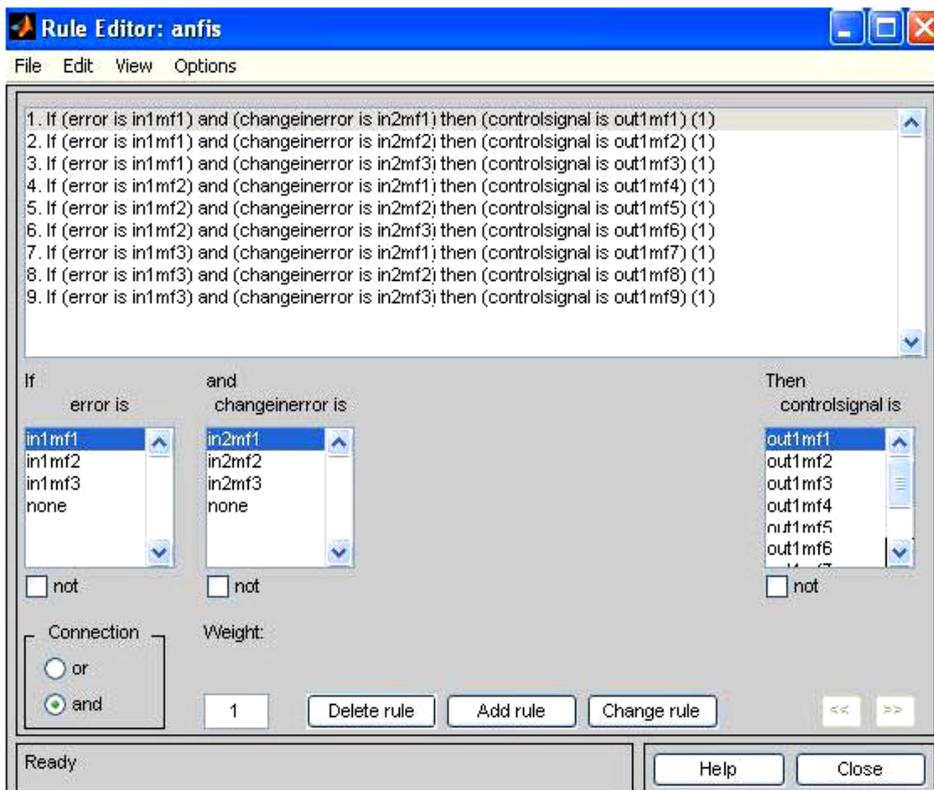


Fig. 6. Generated rule base of the ANFIS structure.

This show the membership functions learns the training data and adjusts its shapes according to the dynamics of the system. The nine rules are used to model the fuzzy part of the ANFIS controller as shown in figure 6 and three membership functions for each linguistic input variable. The fuzzy rules generated by the ANFIS method are shown in figure 6. Figure 7 and 8 shows the loading and training of ANFIS structure using the SCARA dynamic data. The ANFIS structure is trained for 50 epochs, with error tolerance of 0. and the performance Mean Square Error (MSE) is found to be 0.0064759. Figure 9 shows the fuzzy rule viewer of MATLAB, which is used for predetermine the output of the model for specific input values.

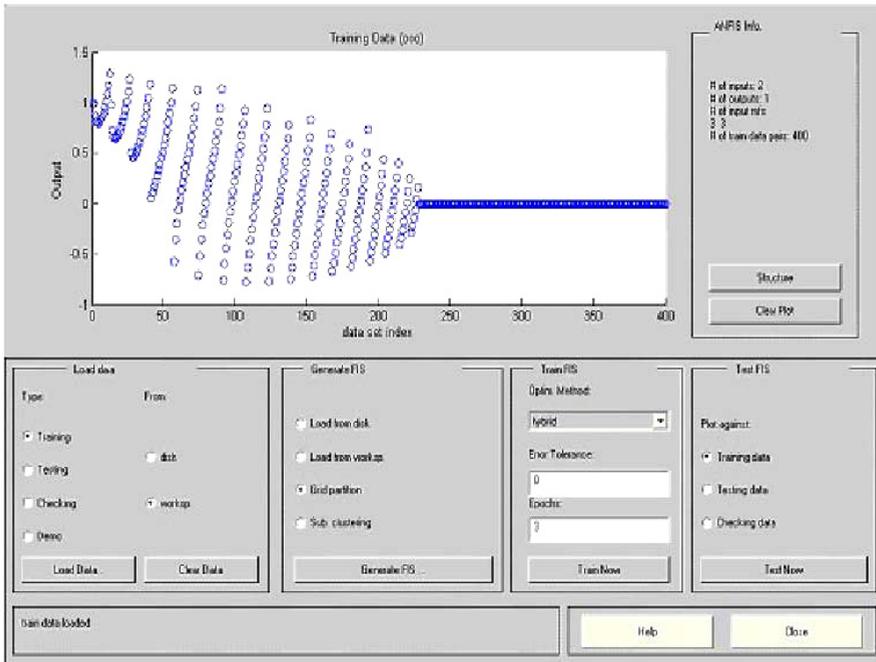


Fig. 7. Loading Training data for ANFIS structure.

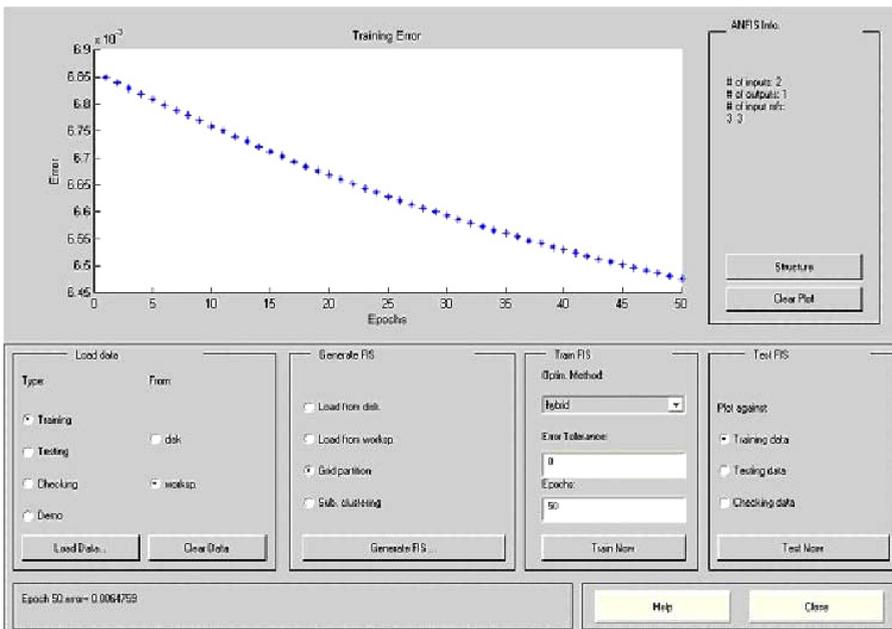


Fig. 8. Training when error tolerance is chosen to be 0 and number of epochs is limited to 50.

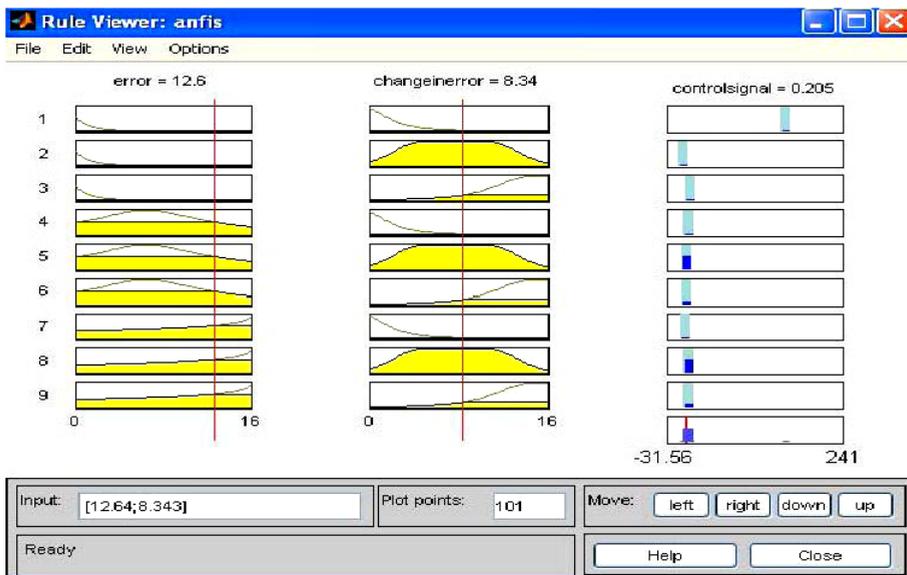


Fig. 9. Rule viewer of ANFIS structure

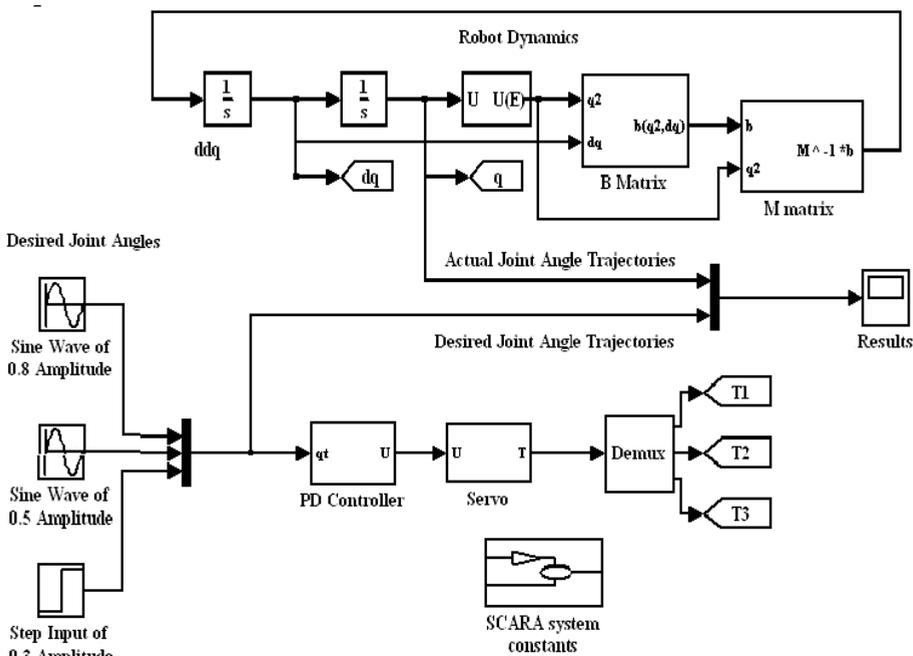


Fig.10. Simulation model of the step/sinusoidal trajectories tracking of three-link SCARA manipulator with PD controller for joint angles ($q_1(t) = 0.8\sin(t)$, $q_2(t) = 0.5\sin(t)$) and joint distance ($q_3(t) = 0.3\text{m}$)

4.1 Continuous Path Control & Experimental Results.

The Continuous Path Motion (CPM), sometimes called controlled-path motion, Schilling (1990). Normally SCARA's are used for pick and place applications in many industries. The positioning and controlling of SCARA End effectors and manipulator are more challenging control problem. The upcoming simulation results with tuned control parameters of ANFIS controller, have achieved a very good tracking performance compared to conventional PD controllers. The figure 10 shows the simulation model of a three-link SCARA manipulator with PD controller for the given joint angle trajectories. This SCARA dynamic model is constructed using MATLAB Simulink software. SCARA is initially tuned for PD values as per the dynamics of the system and its environment. The designed model is experimented with desired trajectories ($q_1(t) = 0.8\sin(t)$, $q_2(t) = 0.5\sin(t)$) and joint distance ($q_3(t) = 0.3\text{m}$) as shown in figure 10. The figure 11 shows good trajectory characteristics at the joint distance, but some tracking error for the sinusoidal trajectories.

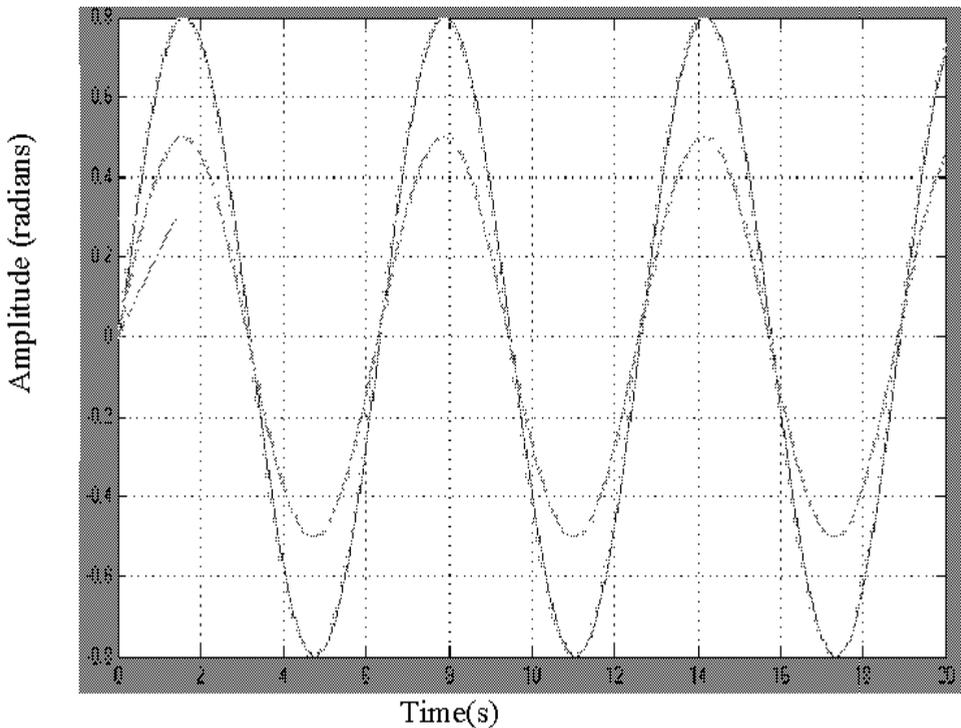


Fig. 11. The step/ sinusoidal trajectories tracking of three-link SCARA manipulator with PD controller for joint angles viz, ($q_1(t) = 0.8\sin(t)$, $q_2(t) = 0.5\sin(t)$) and joint distance ($q_3(t) = 0.3\text{m}$)

The figure 12 depicts the simulation model of three-link SCARA Manipulator with ANFIS controller for joint angles ($q_1(t) = 0.8\sin(t)$, $q_2(t) = 0.5\sin(t)$) and joint distance ($q_3(t) = 0.3\text{m}$). The ANFIS model for SCARA is designed as per the design discussed in section 4 of this chapter. The trained ANFIS network model is shown in figure 12 is modeled by using

Another advantage of this method over classical quantitative controllers is that, it does not require a fixed sampling time. Therefore, the proposed design confirms the fact that ANFIS control is relevant to the control fast of non-linear processes such as robot manipulator controls where quantitative methods are not always appropriate. From the response shown in figure 13 is very clear that ANFIS controller gives no tracking error, i.e. the response of the desired trajectories is almost superimposed with the actual one, Thus the ANFIS controller gave the best results when compare to conventional PD controller. It is very clear from figure 11, the tracking performance of the conventional PD controller is not that appreciable since it is not able cope up with sudden change in the state this leads to some tracking error in its response and also it is not able to follow faithfully as the ANFIS controller does.

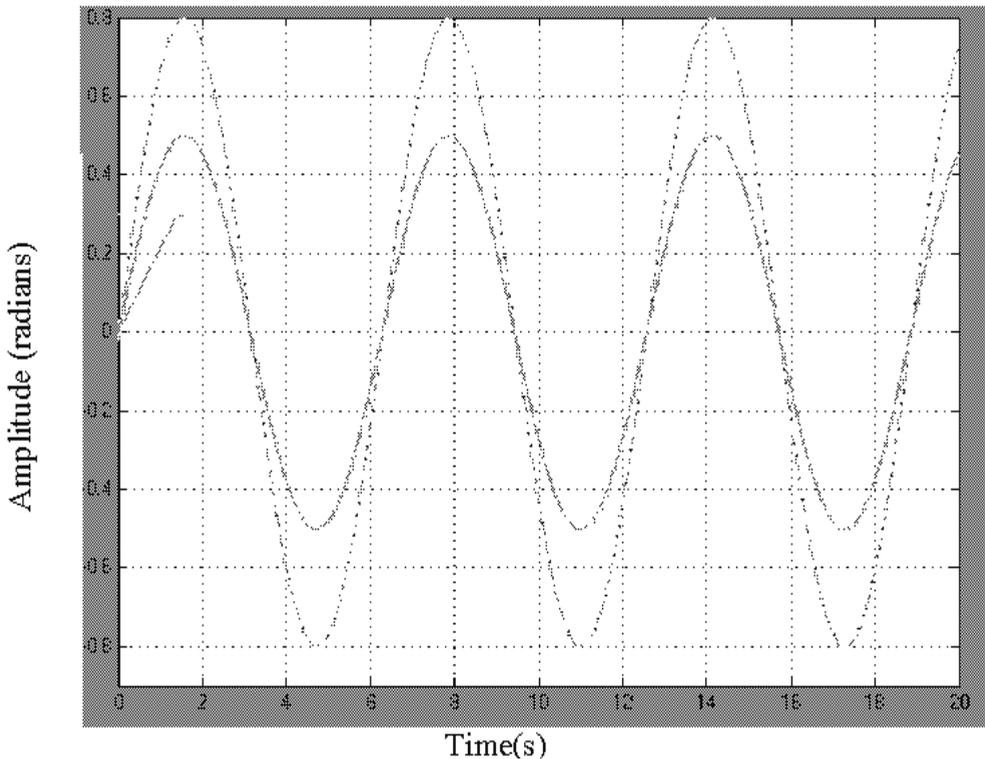


Fig. 13. The step/ sinusoidal trajectories tracking of three-link SCARA manipulator with ANFIS controller for joint angles. With $q_1(t) = 0.8\sin(t)$, $q_2(t) = 0.5\sin(t)$ and joint distance ($q_3(t) = 0.3\text{m}$).

The figure 13 shows the ANFIS controller response of the SCARA for the given desired joint angle trajectories. It is found that actual trajectories of the SCARA are almost merged with the desired trajectories. From this inference, it is concluded that the ANFIS training is completely satisfied and SCARA tracking error is almost nearly zero.

5. Conclusions

In this chapter, the feasibility of ANFIS control for a three link SCARA manipulator has been proved and illustrated by simulation. The best parameters for the fuzzy controller were determined by using the ANFIS methodology and by using simulations of the SCARA manipulator dynamics. ANFIS take only few number of iteration to complete the training of membership functions. A simulation tool (i.e., Neuro-Fuzzy logic toolbox (ANFIS)) was used to validate experimentally the tracking ability and the insensibility to SCARA System parameter changes. The ANFIS controller presented very interesting tracking features and was able to respond to different dynamic conditions. In addition, the fuzzy control computation is very inexpensive, and this regulator could be used for the control of machine tools and robotics manipulators [11] without significantly increasing the cost of the drive. The proposed design confirms the fact that fuzzy control is relevant to the fast control of non-linear processes such as SCARA manipulator control where quantitative methods are not always appropriate. Thus, the results obtained using the ANFIS controllers are encouraging when compared to conventional PD controller.

6. References

- H. R. Berenji and P. Khedkar (1992). Learning and Tuning Fuzzy Logic Controllers through Reinforcements, *IEEE Trans. Neural Networks*, vol.3, no.5, pp.322-320.
- Dubowsky. S. and Desforges, D.T (1979). The application of Model Referenced Adaptive control to Robotic Manipulators, *Transactions of ASME, Journal of Dynamic System*. Vol 101, 193-200
- Fu, K.S.; Gonzalez, R.C. and Lee, C.S.G., (1987). *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw Hill, New York.
- Inoue, H., (1974) Force Feedback in Precise Assembly Tasks, *MIT Artificial Intelligence Laboratory Memo 308*, MIT, Cambridge, Mass.
- J. R. Jang (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. Syst. Man Cybern.*, vol.23, no.3, pp.665-685.
- C. T. Lin and C. S. G. Lee (1996) *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall PTR, 393 Pages.
- Paul, R.C (1972). Modellings, trajectory, Cancellation and Servoing of Computer Controlled Arm, *A.I.Memo 177*, Stanford Artificial Intelligence Lab., Stanford University. California.
- Rong-Jong Wai (2003). Tracking control based on neural network strategy for robot manipulator, *Elsevier, Journal of Neuro computing*. Vol.51, 425- 445.
- R.J. Schilling (1990). *Fundamentals of Robotics*, Prentice-Hall.
- Sciavicco. L. and Siciliano. B (1996). *Modelling and Control of Robot Manipulators*. McGraw-Hill Companies. Inc.
- M. Sugeno (1999). On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents, *IEEE Trans. on Fuzzy Systems*, vol.3, no.2, pp 201-222.
- J. S. Wang, C. S. G. Lee, and C. H. Juang (1999). Structure and Learning in Self-Adaptive Neural Fuzzy Inference System, *Proc. of the Eighth Int'l Fuzzy Syst. Association World Conf.*, Taipei, Taiwan, 935-980, August 13-20.
- Will. P. and Grossnlan. D (1975). An Experimental System for Computer Controlled Mechanical Assembly. *IEEE Trans. EICYY Devices*, Vol 29, 42-48.

Topological Methods for Singularity-Free Path-Planning

Davide Paganelli
University of Bologna
Italy

1. Introduction

Due to their multi-loop architecture, parallel manipulators can reach higher stiffness and load-bearing capability than serial manipulators of equivalent weight. This feature has made them attractive for many applications, including high-precision machining tools, space robots and high-speed manipulators. Unfortunately, the drawback of parallel architectures is the more entangled kinematics, which causes many a problem during design and operation of parallel machines.

The first problem is that it may be impossible to reach a desired configuration without disassembling the mechanism, even though such configuration satisfies all kinematic constraints. A classical trivial example is Grashof four-bar linkage (see Paul, 1979), such as the one depicted in Fig. 1. If the mechanism is at configuration 1, it is impossible to reach configuration 2 without dismantling the kinematic chain.

The configuration of a mechanism will be henceforth meant as the ordered set containing the actual poses (positions and orientations) of all the links of a mechanism. If the pose of at least one link changes, then the configuration changes.

The configuration space of a mechanism is the manifold containing all allowed configurations of the mechanism. The problem of determining whether or not any configurations can be reached is strictly connected with the number of disjoint regions composing the configuration space. If the configuration space is connected, then any configuration can be reached. On the other hand, if the configuration space is composed of two or more disjoint regions, there will always exist unreachable configurations.

Such disjoint regions were named assembly circuits for single-dof mechanism in (Chase & Mirth, 1993), where an interesting discussion is provided to discriminate circuits from branches. Many authors tackled the problem of counting the different assembly circuits in single-dof mechanisms (see for example (Chase & Mirth, 1993), (Mirth & Chase, 1993), (Midha et al., 1985)).

The denomination assembly configuration (AC henceforth) was introduced in (Foster & Cipra, 1998), to generalize the notion of assembly circuit to multi-dof mechanisms. A criterion was given in (Foster & Cipra, 1998) to determine the number of ACs composing the configuration space of any single-loop planar kinematic chain, which was proved to be at

most two. A counting method for the ACs of two-dof multi-loop mechanisms was given in (Foster & Cipra, 2002) and (Dou & Ting, 1998).

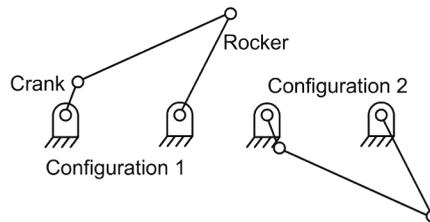


Fig. 1. A Grashof four-bar linkage.

A second hindrance to path-planning toward a target configuration is the presence of singularities. In (Gosselin & Angeles, 1998) all manipulator singularities were classified into three types. Type 1 singularities occur when the inverse kinematics Jacobian matrix is singular. Such singularities are also named serial singularities, because they are typical of serial kinematic chains. Type 2 singularities occur when the direct kinematics Jacobian matrix is singular, whereas type 3 singularities occur when both the afore mentioned conditions are satisfied. Type 2 and 3 singularities are also named parallel singularities, because they are featured by parallel manipulators only.

Serial singularities might cause some loss of dexterity, but are fairly harmless, whereas parallel singularities might trigger the loss of platform control, or the structural break down of the machine. Indeed, the actuator forces required to balance the external actions on the platform might burst to infinity, while crossing a parallel singularity.

It is therefore utterly important to know whether a configuration of the manipulator can be reached or not without meeting a parallel singularity. If, at two different configurations of the manipulator, the Jacobian determinant has opposite signs, then it is impossible to go from one configuration to the other without meeting a singularity, for sooner or later the Jacobian determinant must vanish to change its sign. Nevertheless, if the sign is the same, the existence of a singularity-free path between the two configurations is uncertain.

Many solutions to the challenging problem of singularity-free path-planning are available in the literature, e.g. the geometrical methods proposed in (Dasgupta, and Mruthynjaya, 1998), and (Bhattacharya et al., 1998), or the variational formulation adopted in (Sen et al., 2003). However, the methods hitherto proposed are mainly local, i.e., they might fail to find any singularity-free paths, though some do indeed exist.

The singularity-free path-planning problem is strictly related to the number of disjoint regions into which the configuration space is partitioned by the parallel singularity locus, i.e. the maximal connected regions free of parallel singularities. These disjoint regions will be henceforth named parallel-singularity-free regions (PSFRs). For the purpose of this paper, serial singularities will be ignored, because they are not dangerous for the manipulator.

This paper proposes a method to identify and count all the ACs and PSFRs of a fully-parallel (see Chablat & Wenger, 1998) manipulator. Once this identification process is finished, it is possible to assess whether any singularity-free path connecting any two configurations of the mechanism exists, and whether any path at all exists. The proposed method is based on some elements of differential topology, which will be recalled in the next section. The

developed method will be applied to three classes of parallel manipulators with three degrees of freedom, and numerical examples will show its effectiveness.

2. Morse Theory

Morse theory is an important branch of differential topology. Its aim is to assess the topological properties of a compact manifold through the critical points of a regular function defined on it. In this section, the main definitions and results used in the rest of the paper will be briefly recalled. Further details may be found in (Milnor, 1969).

Let M be a smooth n -dimensional compact manifold and f be a differentiable, real valued function on M . In the neighbourhood of any point P of M it is possible to define a local system of coordinates (x_1, \dots, x_n) . With reference to these coordinates, the gradient of f at P is defined as

$$\nabla f|_P = \left(\frac{\partial f}{\partial x_1} \Big|_P, \dots, \frac{\partial f}{\partial x_n} \Big|_P \right) \tag{1}$$

The points of M where $\nabla f = \mathbf{0}$ are named critical points of f . The property of being critical does not depend on the local coordinate system chosen to calculate the gradient.

The Hessian matrix of f is defined at a point P of M as

$$\mathbf{H}_f|_P = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_P \right] \tag{2}$$

A critical point C of f is said to be nondegenerate if $\mathbf{H}_f|_C$ is nonsingular. The index λ of a nondegenerate critical point is defined as the number of negative eigenvalues of the Hessian matrix $\mathbf{H}_f|_C$. Neither the property of being nondegenerate nor the index depend on the local coordinate system chosen to compute $\mathbf{H}_f|_C$.

For each real value a , let M_a^+ be

$$M_a^+ = f^{-1}[a, +\infty) = \{P \in M : f(P) \geq a\}, \tag{3}$$

the sub-manifold of M where the function f is greater than a . The following two relevant topological results can be stated (see (Milnor, 1969)):

Theorem 1: Let $a < b$ and suppose that the set $f^{-1}[a, b]$, consisting of all points $P \in M$ with $a \leq f(P) \leq b$, contains no critical points of f . Then M_a^+ is diffeomorphic to M_b^+ .

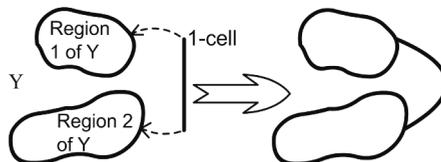


Fig. 2. Attaching a 1-cell to a topological space Y .

Theorem 2: Let c be a real value in the image of f . Suppose that $f^{-1}(c)$ contains a nondegenerate critical point of f . Then, for all sufficiently small $\varepsilon > 0$, $M_{c-\varepsilon}^+$ is homotopic to $M_{c+\varepsilon}^+$ with a k -cell attached. If λ is the index of the critical point and n the dimension of M , then k equals $n-\lambda$.

Rigorous definitions of diffeomorphism and homotopy can be found in (Hirsch, 1976) and (Whitehead, 1978). Connectedness is conserved by homotopy and diffeomorphism: if two sets are diffeomorphic or homotopic they must be composed of the same number of disjoint regions. Theorems 1 and 2 can be thus specialized in these two corollaries:

Corollary 1: Let $a < b$ and suppose that the set $f^{-1}[a, b]$ contains no critical points of f . Then the number of disjoint regions composing M_a^+ equals the number of disjoint regions composing M_b^+ .

Corollary 2: Let c be a real value in the image of f . Suppose that $f^{-1}(c)$ contains one nondegenerate critical point of f . Then, for all sufficiently small $\varepsilon > 0$, $M_{c-\varepsilon}^+$ is composed of the same number of disjoint regions as a topological space obtained by attaching a k -cell to $M_{c+\varepsilon}^+$. If λ is the index of the critical point and n the dimension of M , then k equals $n-\lambda$.

Corollaries 1 and 2 are useful to understand how the number of disjoint regions composing M_a^+ varies as the real value a decreases. As long as the critical points of f contained in M_a^+ remain the same, the number of disjoint regions is constant, by virtue of Corollary 1. As soon as a new critical point is included in M_a^+ , the number of disjoint regions composing it may vary. By virtue of corollary 2 this variation is the same as the one obtained by attaching a k -cell to M_a^+ .

A k -cell is the k -dimensional ball of radius 1. Roughly speaking, to attach a k -cell to a topological space Y means to glue k -cell to the boundary of Y . Fig. 2 shows an example: a 1-cell is glued to the topological space Y . After attaching the cell, the number of disjoint regions of Y changes: it consists of one region only.

Not any variation of the number of disjoint regions composing a topological space can be obtained through the attachment of a k -cell, for the ensuing three corollaries hold (see (Paganelli, 2008)):

Corollary 3: The number of disjoint regions composing a topological space increases when a k -cell is attached to it if and only if k equals 0. In this case only one disjoint region is added.

Corollary 4: If the number of disjoint regions composing a topological space decreases when a k -cell is attached to it, then k equals 1. If a 1-cell is attached to a topological space, the number of disjoint regions composing it may remain the same or be diminished at most by one.

Corollary 5: If k is greater than 1, the number of disjoint regions composing any topological space does not change after a k -cell is attached to it.

Finally, note that corollaries 1 and 2 can be analogously formulated for the set M_a^- , containing all the points of M where $f \leq a$ (see (Paganelli, 2008)).

3. Analysis of Singularity Loci

In most cases, as it will be shown in section 4, the singularity locus of a manipulator is defined on the configuration space by an equation $J=0$, where J is a Jacobian determinant. By using the notation of section 2, there is a compact manifold C , the configuration space, upon which a differentiable real-valued function J is defined. It will be assumed that the manipulator has three degrees of freedom, as the manipulators analyzed in section 4, but all results can be easily generalized for higher mobility manipulators. The aim of this section is to determine how many PSFRs where J is positive exist, i.e. how many disjoint regions compose C_a^+ according to definition (3). The same method will be applied to count the PSFRs in C_a^- , and the number of ACs composing C .

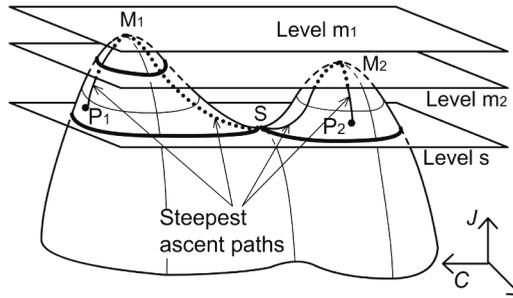


Fig. 3. Generation and joining of disjoint regions.

The evolution of the set C_a^+ is studied as the level a decreases, starting from a level above the absolute maximum, down to zero level. In order to visualize this process, pretend that the manifold C were two-dimensional, and that the graph of J could be plotted as a three-dimensional landscape on C (see Fig. 3). This fictitious lower-dimensional representation is only adopted for visualizing the real process, which occurs on a four-dimensional landscape plotted on the three-dimensional manifold C . Imagine now that the landscape is completely flooded with water. Now let the water level a decrease: as the water level reaches the height of the highest peak, M_1 , an island crops out from the water. The set C_a^+ is the projection on C of the section obtained by cutting the landscape with a plane at height a . As soon as a critical point of J is met, the number of disjoint regions composing C_a^+ varies. Before meeting the absolute maximum M_1 , C_a^+ was empty: it contained zero disjoint regions. After meeting the absolute maximum, the number of disjoint regions composing it changes as if a k -cell were attached to it. The maximum is a critical point of index 3 and the dimension of the manifold C is also 3, thus, k equals 0 (corollary 2). For corollary 3, if a 0-cell is attached to the set one disjoint region is added, thus, after the maximum, the number of disjoint regions is one.

The level of water a keeps on decreasing: as long as it remains between m_1 and m_2 , the heights of the two maxima of Fig. 3, the number of disjoint regions remains equal to one, by virtue of corollary 1. There exists only one island above the water. As soon as maximum M_2 is reached, another island appears and a new disjoint region of C_a^+ is generated. The number of disjoint regions remains equal to two until the saddle point S is reached.

Consider a point P of C_a^+ , with a contained in the open interval (s, m_2) . It is possible to establish whether P belongs to the disjoint region generated by M_1 or to the disjoint region generated by M_2 . The steepest ascent path starting from P must reach one of the two maxima M_1 or M_2 : P belongs to the disjoint region generated by the reached maximum. Thus, the maxima work as "labels" for the disjoint regions: each disjoint region is identified by the maximum contained in it.

As the level a reaches the height of the saddlepoint S , another change of the number of disjoint regions of C_a^+ is expected. Suppose that S is a 2-saddle, i.e. the Hessian matrix has two negative eigenvalues and the index of S is equal to 2, thus the number of disjoint regions composing C_a^+ changes as if a 1-cell were attached to it. For corollary 4 the number of disjoint regions may be diminished by one or remain constant.

To decide whether or not the number of disjoint regions has decreased, it is necessary to find out to which one of the existing disjoint regions the saddle point belongs. The method to reach this goal is identical to that proposed for a noncritical point: the steepest ascent path is followed, starting from the saddle, until a maximum is reached. There are two different steepest ascent paths starting from a 2-saddle. If the steepest ascent paths reach the same maximum, then a disjoint region is joining with itself, and the number of disjoint regions remains constant. If the steepest ascent paths reach two different maxima, the disjoint regions generated by the two maxima join together (Fig. 3). To identify the disjoint region generated by the joining, the maxima inside it can be used: the steepest ascent paths starting from any points inside the new region will lead to one of its maxima.

The procedure is henceforth analogous. Each maximum generates a new disjoint region, and each 2-saddle may connect two existing disjoint regions. Following the two steepest ascent paths as for the first 2-saddle, two maxima are reached: if they belong to two different disjoint regions, such disjoint regions have joined together. If the reached maxima belong to the same disjoint region, the number of disjoint regions remains constant.

As the level a reaches the value zero, the number of disjoint regions that compose C_0^+ is determined. These disjoint regions are the PSFRs with a positive sign of the Jacobian determinant. Each PSFR is provided with a set of maxima which completely characterizes it. Furthermore, all maxima of a PSFR are connected by a network of singularity-free steepest ascent paths. Given any two configurations where the Jacobian determinant is positive (e.g. P_1 and P_2 in Fig. 3), it can be assessed whether or not they belong to the same PSFR: if the steepest ascent paths starting from the two given points reach two maxima of the same PSFR, the two points belong to the same PSFR too, otherwise not. If they do, a singularity-free path is obtained by joining the steepest ascent paths connecting the two points to the maxima and any path in the singularity-free network connecting the maxima of the PSFR.

The positive minima and the positive 1-saddles (i.e. saddle-points with index λ equal to 1) are ignored during the identification of PSFRs. In these two cases, the index λ is lesser than 2, thus only k -cells with k greater than 1 are attached to C_a^+ . Corollary 5 ensures that the number of disjoint regions composing C_a^+ can neither increase nor decrease. Also any singular critical point is irrelevant to classify the PSFRs: two disjoint PSFRs may touch on the boundary at a saddle point, or a singular isolated point appears at a singular maximum, but no regions are generated or joined.

If a degenerate critical point is met, it is not possible to know whether the number of disjoint regions is changing by means of the Hessian matrix only. Higher derivatives have to be considered: the point might be a maximum, thus a new disjoint region is born. Or it might be neither a maximum nor a minimum and two or more disjoint regions could join together (see for example the “monkey-saddle” in (Milnor, 1969)).

An analogous method can be used to count and identify the number of PSFRs where the Jacobian determinant is negative, thus, at the end of this procedure, it is possible to establish to which PSFR any nonsingular point belongs.

Suppose now that the level a keeps on decreasing, below zero level. The process of generation and joining of disjoint regions continues just the same as above zero level: the negative maxima generate new disjoint regions, whereas negative 2-saddles may join existing disjoint regions, but now if the steepest ascent paths starting from negative 2-saddles reach positive maxima, they are not singularity-free anymore. However, they are still feasible paths, even though control might be lost while crossing parallel singularities.

There must exist an absolute minimum of the function J on C , for C is compact and J is continuous. As soon as level a reaches the absolute minimum level, the manifold C_a^+ coincides with the whole configuration space C . Therefore, the disjoint regions composing C_a^+ are indeed the ACs composing the whole configuration space C . As for the PSFR, each AC is endowed with a set of maxima of the function J , which completely defines it. All the maxima contained in the same AC are connected through a network of steepest ascent feasible paths.

In order to assess whether two points belong to the same AC, the steepest ascent paths starting from such points can be followed, until any of the maxima is reached. If the two maxima belong to the same AC, then there exists at least one feasible path connecting them, which can be obtained by joining the steepest ascent paths from the two points to the reached maxima, and any of the feasible paths among the network connecting the two maxima. This path is singularity-free only if the two points belong to the same PSFR, which can be assessed through the method just described.

This process can be analogously repeated for the manifold C_a^+ , letting the level a increase from the absolute minimum to the absolute maximum. This second procedure is redundant for the purpose of determining the ACs, but it might be useful to find out which negative PSFRs belong to which AC, and to cross-check the results hitherto obtained.

The procedure described in the previous two sections can be summarized as follows:

- 1) All critical points of the Jacobian determinant J on the configuration space C are determined.
- 2) The critical points are classified into positive and negative maxima and into positive and negative 1- and 2-saddles.
- 3) The two steepest ascent paths are followed, starting from each positive 2-saddle up to two positive maxima. The two positive maxima, and any maxima belonging to their PSFRs are assigned to the same PSFR. After all the positive 2-saddles have been processed, the positive maxima belonging to each positive PSFR are stored.
- 4) The two steepest ascent paths are followed, starting from each negative 2-saddle up to two maxima. The two maxima, and any maxima belonging to their ACs are assigned to the same AC. After having processed all negative 2-saddles, the maxima of each AC are stored.

- 5) Step 3) is repeated, suitably modified, for the negative PSFRs, to find the negative minima contained in each negative PSFR.
- 6) Step 4) is repeated, suitably modified, for the positive 1-saddles, to find the minima contained in each AC.

4. Case studies

4.1 3UPS Spherical wrists

Spherical wrists are manipulators whose task is to position a rigid body with a fixed point. Thus, by moving the actuators, the orientation of the platform is varied. 3UPS spherical wrists devise a simple parallel architecture to reach this target, which is depicted in Fig. 4.

A 3UPS spherical wrist is composed of a platform, connected to the base by a spherical joint, and three legs, composed of two rigid bodies connected through a prismatic joint. The three legs are connected to the base and to the platform by means of a spherical joint and a universal joint. The universal joint could possibly be replaced by a spherical joint, but the legs would gain a passive rotational degree of freedom which might be undesired.

Let S and S' be two reference frames, attached to the base and to the platform respectively, and with the origin in the centre of the spherical joint between the platform and the base. Let the three points P_1 , P_2 , and P_3 be the centres of the joints between the base and the legs, and the three points Q_1 , Q_2 , and Q_3 be the centers of the joints between the platform and the legs. The kinematic architecture of any 3UPS wrist is identified by the three vectors \mathbf{p}_1 , \mathbf{p}_2 , and \mathbf{p}_3 , containing the coordinates of points P_1 , P_2 , and P_3 relative to frame S , along with the three vectors \mathbf{q}_1 , \mathbf{q}_2 , and \mathbf{q}_3 , containing the coordinates of points Q_1 , Q_2 , and Q_3 relative to frame S' .

3UPS spherical wrists were first studied in (Innocenti & Parenti-Castelli, 1993), where the direct kinematics problem was solved. In (Sefrioui & Gosselin, 1994) the singularity locus of 3UPS spherical wrists was studied, and a representation method was proposed. However, in the following sections a different parameterization and visualization will be adopted.

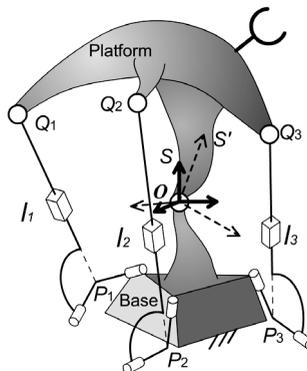


Fig. 4. 3UPS spherical wrist.

The workspace of a 3UPS spherical wrists contains all possible orientations of the platform. According to Euler theorem, any possible orientation of a rigid body with a fixed point can be obtained from a given reference position of the body by rotating of an angle θ about an

axis directed as a unit vector \mathbf{u} and containing the fixed point. Therefore, adopted a reference position where frames S' and S coincide, any orientation of the platform can be defined by way of a unit vector \mathbf{u} and an angle θ . The orientations of the platform associated to \mathbf{u} and θ , and to $-\mathbf{u}$ and θ always coincide, thus the variation range of the angle θ can be restricted to the interval $[0, \pi]$.

A possible visualization of the workspace can be obtained by considering a ball of radius π in the three-dimensional Euclidean space. With reference to Fig. 5., every point P inside the ball represents the orientation of the platform identified by the unit vector directed as the position vector of P , and by the angle θ equal to the length of the position vector of P . Thus, every orientation of the platform with an angle θ lesser than π corresponds to only one point inside the ball, whereas any orientation with $\theta = \pi$ is identified by two diametrically opposite points on the boundary sphere of the ball.

It is useful to introduce a more homogeneous parameterization of the orientation of a rigid body, i.e. Euler parameters, which will enable an easier determination of the critical points of the Jacobian determinant. We consider the vector \mathbf{e} , containing the four Euler parameters (e_0, e_1, e_2, e_3) , such that:

$$e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1 \tag{4}$$

and $e_0 \geq 0$. In this section, Euler parameters will be used for the mathematical representation of the workspace, whereas the visualisation of Fig. 5. will be adopted to show results on a three-dimensional graph.

Each point of the jointspace can be identified via the three length of the legs, i.e. the three distances l_i between points P_i and Q_i of Fig. 4.

The vector $(e_0, e_1, e_2, e_3, l_1, l_2, l_3)$ can be used to identify a configuration of the parallel wrist, yet, not any such vector determines an allowed configuration of the wrist, for three constraints must be satisfied. The equations expressing these constraints are derived by means of Carnot theorem applied to the three triangles P_iOQ_i , as shown in Fig. 6. For each of such triangles one can write the constraint:

$$l_i^2 = \mathbf{p}_i^T \mathbf{R} \mathbf{q}_i \tag{5}$$

where \mathbf{R} is the rotation matrix ruling the coordinate change from S' to S , whereas \mathbf{p}_i and \mathbf{Rq}_i are the column vectors containing the coordinates of points P_i and Q_i in the fixed frame S .

The rotation matrix can be written as a quadratic function of the four Euler parameters, , thus Eq. (5) represents a set of three quadratic equations in Euler parameters and leg lengths. If only positive leg lengths are accepted, which indeed does not exclude any configuration of the wrist, there is only one set of leg lengths for any orientation of the platform. Thus, the workspace alone can be used to represent the whole configuration space of the wrist, and there is only one AC.

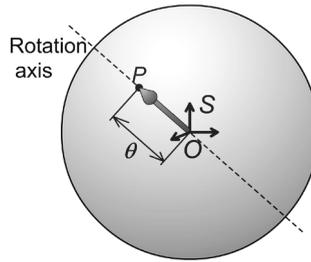


Fig. 5. Workspace of a 3UPS wrist.

In order to determine the singularity locus of the 3UPS spherical wrist, Eq. (5) is differentiated, obtaining the ensuing relationship between the virtual displacement of the platform and the virtual variations of leg lengths:

$$\mathbf{A} \begin{pmatrix} \delta l_1 \\ \delta l_2 \\ \delta l_3 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \delta e_0 \\ \delta e_1 \\ \delta e_2 \\ \delta e_3 \end{pmatrix} \quad (6)$$

where \mathbf{A} and \mathbf{B} are the Jacobian matrices of Eq. (5) with respect to leg lengths and Euler parameters respectively.

Moreover, not any virtual variation of Euler parameters is allowed, for Eq.(4) must hold for first order variations too. Thus, differentiation of Eq.(4) yields the ensuing constraint upon the virtual variations of Euler parameters:

$$e_0 \delta e_0 + e_1 \delta e_1 + e_2 \delta e_2 + e_3 \delta e_3 = 0 \quad (7)$$

If equations (7) and (6) are put together, the ensuing relation is obtained:

$$\begin{pmatrix} \mathbf{A} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta l_1 \\ \delta l_2 \\ \delta l_3 \end{pmatrix} = \begin{pmatrix} \mathbf{B} \\ e_0 & e_1 & e_2 & e_3 \end{pmatrix} \begin{pmatrix} \delta e_0 \\ \delta e_1 \\ \delta e_2 \\ \delta e_3 \end{pmatrix} \quad (8)$$

Parallel singularities occur whenever a nonzero virtual displacement of the platform is allowed by the constraints, although the actuators undergo no virtual displacements.

This implies that the determinant of the matrix at the right-hand side of Eq. (8) vanishes. Thus the parallel singularity locus is defined as the zero level set of a function J on the configuration space, which contains all possible orientations of the platform. The function J can be obtained as:

$$J = \det \begin{pmatrix} \mathbf{B} \\ e_0 & e_1 & e_2 & e_3 \end{pmatrix} \quad (9)$$

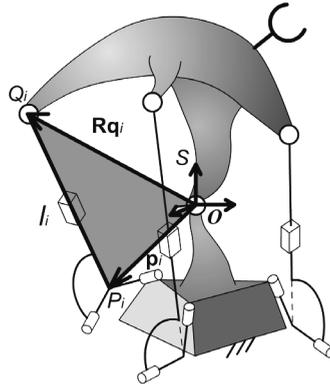


Fig. 6. Constraints of the spherical wrist through Carnot theorem

Each element of matrix \mathbf{B} is linear and homogeneous in the four Euler parameters, therefore J is a fourth-order homogeneous polynomial in the four Euler parameters. The singularity locus $J = 0$ can be represented as a two-dimensional surface cutting the workspace, and the method developed in Section 3 can be used to determine how many PSFRs are partitioned by the singularity locus, and to find out whether it is possible or not to reach a desired position in the workspace without crossing a parallel singularity.

The toughest and most important task is the determination of all critical points of the function J on the configuration space, which coincides in this case with the workspace. For the case at hand, a redundant parameterization is used, because the four Euler parameters, tied by Eq.(4) identify a point of the three dimensional manifold containing all possible orientations of a rigid body.

The most straightforward way to tackle the problem is to resort to Lagrange's multipliers. At the critical points of J , the gradient of J is parallel to the gradient of the constraint, formalized by Eq.(4), i.e.:

$$\frac{\partial J}{\partial \mathbf{e}} = \lambda \mathbf{e} \tag{10}$$

where λ is a Lagrange multiplier, and can be easily eliminated considering the ensuing equation set:

$$\begin{aligned} (\partial J / \partial e_1) e_0 - (\partial J / \partial e_0) e_1 &= 0 \\ (\partial J / \partial e_2) e_0 - (\partial J / \partial e_0) e_2 &= 0 \\ (\partial J / \partial e_3) e_1 - (\partial J / \partial e_1) e_3 &= 0 \end{aligned} \tag{11}$$

which stems from Eq. (10) by multiplying the i^{th} equation by e_j and by subtracting the result from the product of the j^{th} equation by e_i , with a proper choice of i and j .

Eq.(11) is set of three homogeneous fourth-order polynomial equations, in the four Euler parameters. Each solution in the projective space of such equation, when properly normalized, is a set of Euler parameters defining a critical point of J on the workspace, except some extraneous solutions introduced while passing from Eq. (10) to Eq.(11). Such

extraneous solutions are obtained when e_0 or e_1 are posed equal to zero. If $e_0 = 0$, Eq.(11) becomes:

$$\begin{aligned}(\partial J / \partial e_0) &= 0 \\(\partial J / \partial e_0) &= 0 \\(\partial J / \partial e_3) e_1 - (\partial J / \partial e_1) e_3 &= 0\end{aligned}\tag{12}$$

where the first two equations degenerate into the same one. Therefore, Eq.(12) is a set of two homogeneous equations, the first of degree three and the second of degree four, in three unknowns. By virtue of Bezout theorem, Eq.(12) admits 12 solutions, which are extraneous solutions to Eq. (10), that does not admit, in general, solutions with $e_0 = 0$.

Analogously, if $e_1 = 0$ Eq.(11) degenerates again into two equations, that yield twelve additional extraneous solutions. Eq.(11) is a set of three homogeneous equations of degree 4, therefore, by virtue of Bezout theorem, it admits $4^3=64$ solutions in the complex projective space. Since 24 solutions are extraneous for Eq. (10), there are 40 solutions to Eq.(11), and the real ones are critical points of J .

Such forty solutions can be obtained by partial homogenization. First of all, Eq.(11) is transformed into a non homogeneous system of equations, by posing $e_0 = 1$. In this way, any homogeneous solution with $e_0 = 0$ becomes a solution at infinity, included 12 of the 24 extraneous solutions. Then, Eq.(11) is partially homogenized, by posing $e_2 = x_1/x_0$ and $e_3 = x_2/x_0$, and by simplifying the denominators. In this way, Eq.(11) becomes a system of three homogeneous equations of degree four in the three variables x_1 , x_2 , and x_0 , where variable e_1 , that has been left out of partial homogenization, is hidden in the coefficients. Variables x_1 , x_2 , and x_0 can be got rid of by means of classical elimination methods, (see for example (Salmon, 1885)), obtaining a polynomial in the hidden variable e_1 .

Stemming from a homogeneous equation set that should have 64 solutions, the resultant polynomial should be of degree 64. However, since the homogeneous equation set always possesses 12 solutions with $e_0 = 0$, the resultant polynomial must have at least 12 solutions at infinity, and its degree will be at most 52.

Furthermore, since there are always twelve extraneous solutions with $e_1 = 0$, the resultant polynomial will be divisible by the monomial e_1^{12} . By dividing the resultant by e_1^{12} , a final equation of degree 40 is obtained, that is completely purged from extraneous solutions.

The polynomial of degree 40 is solved numerically, and the values of e_2 and e_3 corresponding to each solution in e_1 are easily found (Salmon, 1885). The values obtained are homogeneous solutions with $e_0 = 1$. In order to obtain the four Euler parameters identifying the orientation of the rigid body, the four values just obtained must be normalized, so that Eq.(4) is satisfied. Should there be any critical point with $e_0 = 0$, this would be another solution at infinity to the resultant polynomial, whose degree would be lesser than 40. In this case, the loss of a solution is easily detected by the loss of one degree of the final polynomial, and the lost solution can easily be found by substituting $e_0 = 0$ into Eq. (10).

In this way, all 40 complex solutions to Eq. (10) are found, and the real ones are the critical points of the function J . These critical points must be classified into maxima, minima, 1-saddles and 2-saddles. In order to perform the classification, a local coordinate system could be chosen, and the Hessian matrix could be calculated and analyzed. However, this is not the most straightforward way to proceed, for the parameterization used henceforth is redundant, and represents no local coordinate system. At the same way Lagrange multipliers enable determination of critical points with no need of local coordinate systems, it is possible to intrinsically analyze second order variations of J in the neighborhood of a critical point through the following eigenvalue problem (see (Fletcher, 1987)):

$$\left(\mathbf{H}_{|\tilde{\mathbf{e}}}^* - \alpha \mathbf{I}^*\right) \mathbf{a}^* = 0 \tag{13}$$

where:

- $\mathbf{H}_{|\tilde{\mathbf{e}}}^*$ is the bordered Hessian, i.e. the Hessian matrix of the Lagrangian function, calculated at the critical point $\tilde{\mathbf{e}}$. The Lagrangian function is defined as $L(\mathbf{e}, \lambda) = J(\mathbf{e}) - \lambda \cdot c(\mathbf{e})$, where $c(\mathbf{e}) = \mathbf{e}^T \mathbf{e} - 1$ is the optimization constraint.
- \mathbf{I}^* is equal to the 5x5 identity matrix, save the fifth element of the fifth row, which is equal to zero.
- \mathbf{a}^* is a five-dimensional vector obtained appending a dummy variable to a four dimensional vector, representing a small variation of Euler parameters in the neighbourhood of the critical point.

The steepest increase or decrease directions are the directions for which the ensuing condition is satisfied:

$$\det\left(\mathbf{H}_{|\tilde{\mathbf{e}}}^* - \alpha \mathbf{I}^*\right) = 0 \tag{14}$$

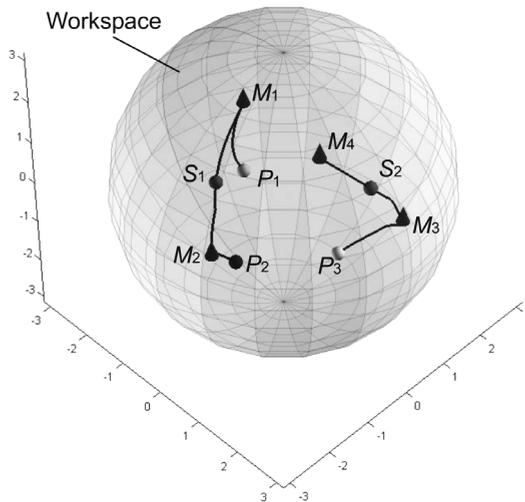


Fig. 7. Critical points in the workspace of W_1

Eq.(14) is always a third order polynomial in the eigenvalue α , therefore three solutions are expected. For each solution, Eq.(13) yields the corresponding direction of steepest variation of J : it is represented by the first four components of the eigenvector \mathbf{a}^* of each eigenvalue α . The eigenvectors corresponding to positive eigenvalues are steepest increase directions, whereas the eigenvectors corresponding to negative eigenvalues are steepest descent directions. The index of the critical point is thus the number of negative solutions to Eq.(14), which enables the classification of any possible critical point of J .

Also the generation of the steepest ascent or descent paths does not require the use of a local coordinate system. A small displacement in the steepest ascent direction just found is used to leave a saddle point. Then, small displacements following the projection of the gradient of J along the constraint surface described by Eq.(4) will build the steepest variation path, ending upon a maximum or a minimum. Whenever, while following a steepest ascent or descent path, a set of Euler parameters \mathbf{e} with $e_0 < 0$ is reached, it is immediately replaced with $-\mathbf{e}$, which is the same position of the platform, in order that e_0 is always greater than zero, as discussed above.

The 3UPS spherical wrist W_1 will be used as a numerical application. The parameters defining manipulator W_1 are reported in Table 1.

\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_3	\mathbf{q}_1	\mathbf{q}_2	\mathbf{q}_3
(1,0,0)	(0,1,0)	(0,1,1)	(-9,2,6)/11	(6,6,7)/11	(-1,1,0)

Table 1. Parameters defining manipulator W_1

Through the elimination method just described, 32 critical points are determined, among which there are 4 positive maxima, 2 positive 2-saddles, 4 negative minima, 12 negative 1-saddles, and 10 singular 2-saddles.

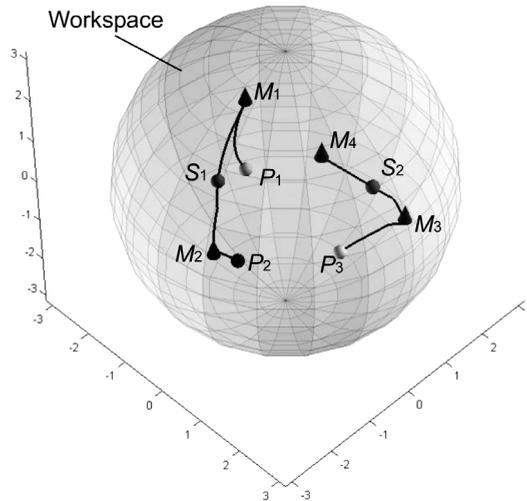


Fig. 7 shows the workspace of the spherical wrist, through the representation proposed in Fig. 5. The four positive maxima M_1, M_2, M_3 and M_4 , are depicted as cones, and the two positive 2-saddles S_1 and S_2 , depicted as spheres.

The steepest ascent paths starting from the two 2-saddles S_1 and S_2 join M_1 to M_2 and M_3 to M_4 respectively, thus there are two positive PSFRs. The steepest ascent paths are represented in

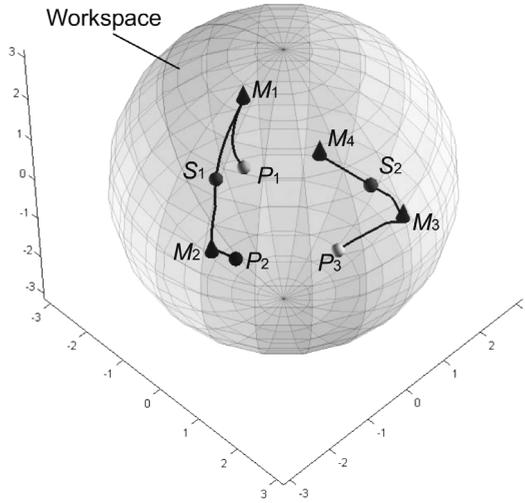


Fig. 7 as black lines.

Given the three points P_1 , P_2 , and P_3 , where J is positive, it can be assessed to which one of the two positive regions they do belong by following the steepest ascent paths (black lines in

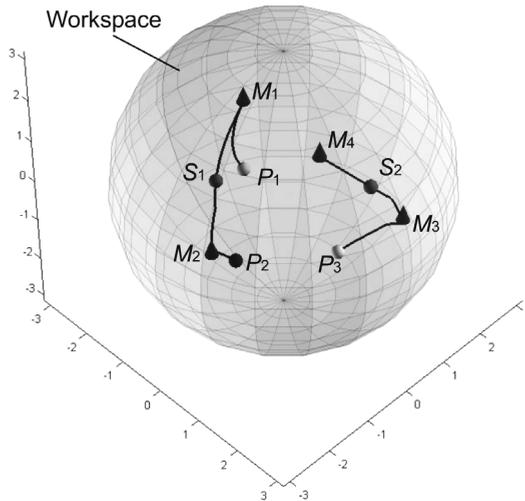


Fig. 7. The steepest ascent paths starting from P_1 , P_2 , and P_3 reach the maxima M_1 , M_2 and M_3 , respectively. Therefore P_1 and P_2 belong to the same region, and the path $P_1 - M_1 - S_1 - M_2 - P_2$, connecting P_1 to P_2 is singularity-free. The steepest ascent path starting from P_3 reaches M_3 , which belongs to a different region, therefore there exists no singularity-free path at all to reach P_1 or P_2 starting from P_3 .

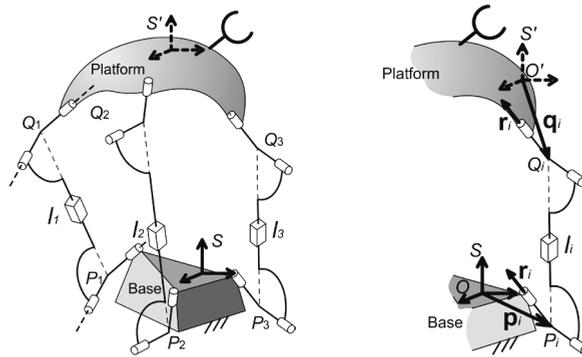


Fig. 8. Kinematic architecture of a 3UPU translational manipulator.

The four minima are all connected through a network of steepest descent paths starting from the negative saddles, therefore there is only one negative singularity-free region.

4.2 3UPU Translational Manipulators

3UPU translational manipulators have been proposed and analyzed in (Tsai, 1996), (Di Gregorio & Parenti Castelli, 1998), and (Parenti Castelli et al., 1998).

The architecture that will be hereafter considered is that proposed in (Di Gregorio & Parenti Castelli, 1998), and sketched in Fig. 8. The platform is connected to the base by means of three legs, consisting of two links connected to each other by a prismatic joint and to the base and the platform through universal joints. The universal joints satisfy the ensuing two geometrical requirements:

- in each leg, the axes of the two revolute joints connected to the base and to the platform are parallel;
- in each leg, the axes of the two middle revolute joints, not connected to the base nor to the platform, are parallel.

It can be proved that this architecture constrains the platform to pure translational motions (see (Di Gregorio & Parenti Castelli, 1998)).

With reference to Fig. 8, the geometry of 3UPU translational manipulators, can be parametrized in the ensuing way:

- two reference frames S and S' with parallel axes, attached to the base and to the platform respectively, are defined;
- on the i^{th} leg, the center P_i of the universal joint attached to the base is identified through its coordinate vector \mathbf{p}_i in frame S ;
- on the i^{th} leg, the center Q_i of the universal joint attached to the platform is identified through its coordinate vector \mathbf{q}_i in frame S' ;
- on the i^{th} leg, the common directions of the axes of the revolute joints attached to the base or to the platform is identified by way of a unit vector \mathbf{r}_i ;

Therefore, the nine vectors \mathbf{p}_i , \mathbf{q}_i , and \mathbf{r}_i , for $i=1,2,3$, completely define the kinematic architecture of 3UPU translational manipulators.

The workspace of a 3UPU translational manipulator is the manifold containing all possible positions of the platform. Each point of the workspace can be identified by means of the

coordinate vector $\mathbf{x} = (x, y, z)$ of a point, for example the origin O' of S' , with respect to the fixed frame S . Therefore the workspace is the whole three dimensional Euclidean space.

Any point of the jointspace is defined by the vector $\mathbf{l} = (l_1, l_2, l_3)$, containing the lengths of the three actuated legs, thus the jointspace is a subset of the three dimensional Euclidean space, too. More specifically, the length l_i is equal to the distance between points P_i and Q_i . No limits will be considered for leg length, thus each l_i can range from zero to infinity.

The vector (l_1, l_2, l_3, x, y, z) identifies a configuration of the manipulator only if the ensuing constraints are satisfied:

$$l_i^2 = (Q_i - P_i)^2 = (\mathbf{q}_i + \mathbf{x} - \mathbf{p}_i)^2 \quad (15)$$

Like the spherical wrists, if only positive lengths are accepted to describe the length of the legs, there exists only one point of the jointspace that defines a configuration along with a given point in the workspace, which means that the workspace and the configuration space can be considered as the same manifold. In other words, the vector (x, y, z) identifies both a position of the platform and a configuration of the manipulator, and there is only one AC.

Eq. (15) can be differentiated, obtaining the ensuing relation:

$$\mathbf{A}\delta\mathbf{l} = \mathbf{B}\delta\mathbf{x} \quad (16)$$

where the i^{th} row of matrix \mathbf{B} is the vector $2(Q_i - P_i)$, which can be written as $2(\mathbf{x} + \mathbf{c}_i)$, where \mathbf{c}_i is a constant vector for the i^{th} row. This means that the determinant of \mathbf{B} is linear in the variables x, y , and z .

The parallel singularity locus is therefore a plane in the three-dimensional Euclidean space, because it is determined by the equation:

$$J_p = \det \mathbf{B} = 0 \quad (17)$$

which is linear in the variables x, y , and z .

Parallel singularities derived by Eq. (17) are not the only dangerous configurations for a 3UPU manipulator. Eq. (17) is based upon the implicit assumption that only translational virtual displacement of the platform are allowed, but there is no direct kinematic constraint enforcing this condition. Translational motion is the result of the particular choice of the U-joint axes, which might be unable to hinder virtual rotations at some singular positions.

These singular positions were named in (Zlatanov et al., 2002) *constraint singularities*, because at such positions some constraints of the parallel architecture are locally lost. Constraint singularities are typical of lower mobility parallel manipulators where the platform possesses less than six degrees of freedom. In such manipulators, some of the six degrees of freedom of the platform are controlled through the actuators of the manipulator, whereas some other (the rotational ones, in the case at hand) are passively constrained through the geometry of the legs. Parallel singularities are always detected by differentiating the equations connecting the input variables of the jointspace to the output variables of the workspace, but constraint singularities may not. In order to detect constraint singularities is always necessary to consider all six degrees of freedom of the platform, and to investigate under which conditions the constraints upon the degrees of freedom that are not controlled by the actuators might fail.

Constraint singularities of translational 3UPU manipulators were studied and rigorously determined in (Parenti Castelli & Di Gregorio, 2002). Constraint singularities are organized in a locus, which is defined by the ensuing equation:

$$J_c = \det C = 0 \quad (18)$$

The i^{th} column of the 3×3 matrix C is the axis direction of the rotation hindered by the i^{th} leg, that can be expressed as follows:

$$t_i = (\mathbf{q}_i + \mathbf{x} - \mathbf{p}_i) - [(\mathbf{q}_i + \mathbf{x} - \mathbf{p}_i)^T \mathbf{r}_i] \mathbf{r}_i \quad (19)$$

Eq. (19) is a third order polynomial in the coordinates x , y and z , identifying the configuration of the manipulator.

Both constraints and parallel singularities are equally dangerous for a 3UPU translational manipulator, and must be avoided while moving from a configuration to another. The surface to be avoided is the zero level set of the function $J_p J_c$ on the workspace of the 3UPU translational manipulator. Unfortunately, the workspace of a translational manipulator is the three-dimensional Euclidean space, which is not compact. Thus, the method developed in Section 3 cannot be straightforwardly applied, because it works on compact manifolds only.

Yet, it is possible to transform the three-dimensional Euclidean space into a compact manifold. First of all, consider the three-dimensional real projective space associated to the three-dimensional Euclidean space, i.e. each vector (x_0, x_1, x_2, x_3) of the projective space is such that $(x, y, z) = (x_1/x_0, x_2/x_0, x_3/x_0)$. Each point of the projective space corresponds to one point of the Euclidean space, except the points with $x_0=0$, i.e. points at infinity, that do not exist in the Euclidean space.

We can imagine the workspace of the 3UPU manipulator as the projective space, where the points with $x_0=0$ must never be crossed, exactly like singularities. Thus, the locus of "forbidden" points, is defined by the ensuing equation in the real projective space:

$$J = x_0 J_p J_c = 0 \quad (20)$$

where J_p and J_c are properly converted to homogeneous coordinates.

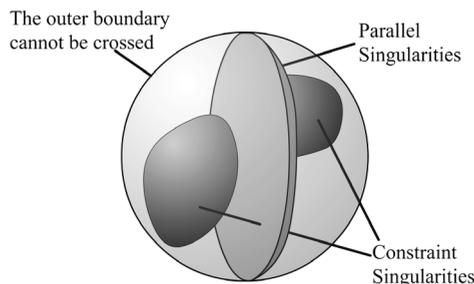


Fig. 9. Singularity locus of the 3UPU translational manipulator.

The real projective space can be represented as a ball, analogous to the manifold containing all orientations of a rigid body. With reference to Fig. 9, a three-dimensional ball with radius 1 is considered. For any point inside the ball, the coordinates of the point are the homogeneous coordinates $x_1, x_2,$ and $x_3,$ of the corresponding point in the projective space. The projective coordinate x_0 is defined as $x_0 = \sqrt{1 - x_1^2 - x_2^2 - x_3^2}$. In this way, all the points of the ball correspond to one point of the projective space. Furthermore, all points on the spherical boundary of the ball are points with $x_0 = 0,$ i.e. points at infinity. Like Euler parameters, we represent a point of the projective space as a four-dimensional vector $\mathbf{x}_* = (x_0, x_1, x_2, x_3)$ with the constraint:

$$c = \mathbf{x}_*^T \mathbf{x}_* = 1 \tag{21}$$

The critical points of J on the configuration space must be found. The configuration space is the three-dimensional real projective space, and the critical points can be found through Lagrange multipliers method:

$$\frac{\partial J}{\partial \mathbf{x}_*} = \lambda \frac{\partial c}{\partial \mathbf{x}_*} \tag{22}$$

Lagrange multiplier λ can be easily eliminated considering the equation set:

$$\begin{aligned} T_1 &= (\partial J / \partial x_1)x_0 - (\partial J / \partial x_0)x_1 = 0 \\ T_2 &= (\partial J / \partial x_2)x_0 - (\partial J / \partial x_0)x_2 = 0 \\ T_3 &= (\partial J / \partial x_3)x_0 - (\partial J / \partial x_0)x_3 = 0 \end{aligned} \tag{23}$$

obtained by extracting λ from the first equation and then substituting it into the remaining three. Eq. (23) is a set of three homogeneous equations of degree five. Any critical point with $x_0 = 0$ must not be considered, because it lies on the locus $J = 0$. Thus $x_0 = 1$ can be substituted into Eq. (23), which is solved in terms of $x_1, x_2,$ and $x_3.$

Since J is divisible by J_c and $J_p,$ Eq. (23) can be written in the ensuing form:

$$\mathbf{M} \begin{pmatrix} J_c \\ J_p \\ J_c J_p \end{pmatrix} = \mathbf{0} \tag{24}$$

Therefore all points where $J_c = J_p = 0$ are critical points of J . These points form in general a curve in the workspace, and, not being isolated, are always degenerate critical points. Fortunately, the critical points where $J_c = J_p = 0$ are all singular, and must be ruled out of the analysis. Thus, only critical points where either J_c or J_p do not vanish must be considered, which, along with Eq. (24) yields the additional equation:

$$T_4 = \det \mathbf{M} = 0 \tag{25}$$

Eq.(25) is a third order equation in $x_1, x_2,$ and x_3 and can be used to reduce the degree of

Eq. (23). Each of the three polynomials T_1 , T_2 , and T_3 can be written as follows:

$$T_i = T_4 Q_i + R_i \tag{26}$$

where Q_i and R_i are the quotient and the remainder of a polynomial division on T_i through the divisor T_4 with respect to a given variable x_j . At every point where all T_i vanish along with T_4 , all remainders R_i must vanish too. Therefore the equation set

$$R_1 = R_2 = R_3 = T_4 = 0 \tag{27}$$

is always equivalent to Eq. (23), along with the condition $T_4 = 0$. If R_1 , R_2 , and R_3 are remainders of polynomial divisions with respect to variables x_2 , x_1 , and x_1 , respectively, R_1 , R_2 , and R_3 are polynomials of degree four in the two variables x_2 and x_3 . Therefore the equation set:

$$R_1 = R_2 = R_3 = 0 \tag{28}$$

can be solved with a method similar to that used for spherical wrists. Variable x_1 can be hidden in the coefficients, and a partial homogenization with respect to x_2 and x_3 yields a set of three homogeneous equations in three unknowns of degree four. A resultant polynomial in x_1 can then be found through classical elimination methods.

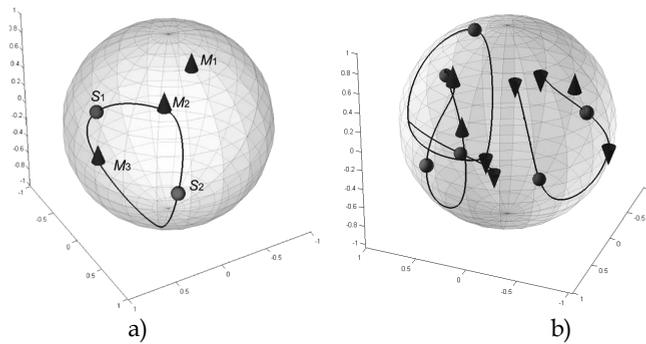


Fig. 10. a) Positive critical points of manipulator T_1 ;
 b) All critical points of manipulator T_1 .

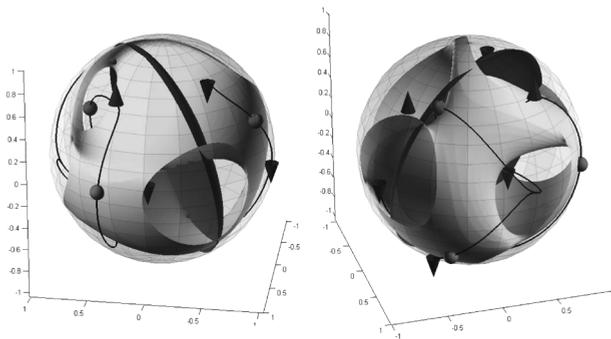


Fig. 11. The steepest ascent and descent paths are all singularity-free.

Unfortunately, in this way the condition $T_4 = 0$ has not been directly imposed: Eq.(28) is not completely equivalent to Eq.(27), which introduces extraneous solutions. The author has found no way to factor out such extraneous solutions from the resultant polynomial, however they can be easily detected, for they do not satisfy the condition $T_4 = 0$.

Once all real solutions have been found by numerically solving the resultant polynomial, and all extraneous solutions have been cancelled, all critical points of J are known. The classification of critical points, and the determination of steepest ascent paths is then analogous to the one proposed in Section 4.1 for spherical wrists.

Manipulator T_1 is now considered as a numerical example. According to the conventional parameterization adopted before, T_1 , is defined by vectors reported in Table 2.

\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_3	\mathbf{q}_1	\mathbf{q}_2	\mathbf{q}_3	\mathbf{r}_1	\mathbf{r}_2	\mathbf{r}_3
(1,0,0)	(0,1,0)	(0,1,-1)	(1,1,1)	(0,1,-1)	(1,1,1)	$(1,0,1)/\sqrt{2}$	$(1,0,1)/\sqrt{2}$	$(1,0,1)/\sqrt{2}$

Table 2. Parameters defining manipulator T_1

In the workspace of T_1 there are three positive maxima and two positive 2-saddles, shown in Fig. 10. a) Positive critical points of manipulator T_1 ; Fig. 10a through the conventional ball visualization proposed in Fig. 9. Fig. 10a also shows the steepest ascent paths (black lines), departing from the positive 2-saddles and reaching the maxima. It can be seen that maxima M_2 and M_3 are joined, while no paths reach maximum M_1 . Therefore there are two positive regions, free of parallel and constraint singularities.

There are five negative minima and four negative 1-saddles, and the network of steepest descent paths is such that there are also two negative regions.

Fig. 10b shows all relevant critical points: the positive maxima are depicted as upward bound cones, the negative minima as downward bound cones, and the saddle points as spheres. The network of singularity-free steepest ascent and descent paths is represented as black lines.

Fig. 11 shows two rotated views of the locus $J = 0$. The outer spherical boundary belongs to the locus, but it has not been plotted, in order for the inside of the ball to be visible. The darker surface inside the ball represents the locus of parallel singularities, whereas the brighter surface the locus of constraint singularities. The intersection curve of the two surfaces is a set of singular degenerate critical points, that have been ruled out from the determination of critical points by means of the polynomial division. It is possible to verify that the steepest ascent and descent paths never cross the spherical boundary, nor the parallel and constraint singularity loci.

4.3 3RRR Planar manipulators

A 3RRR planar manipulator with general structure is depicted in Fig. 12. The platform is connected to the rigid frame through three legs, composed of two connecting rods and three revolute joints, with the middle one actuated.

The center of the i^{th} leg revolute joint on the fixed frame is indicated by P_i , whereas the center of the i^{th} leg revolute joint on the platform is indicated by Q_i . The center of the actuated revolute joint of the i^{th} leg is denoted by R_i .

The kinematic structure of the platform can be determined through the three parameters u_2 , u_3 , and v_3 , defining the coordinates of Q_1 , Q_2 , and Q_3 in the reference frame uQ_1v attached to the platform, as shown in Fig. 12.. Analogously, the kinematic structure of the fixed frame is given by the three parameters a_2 , a_3 , and b_3 , defining the coordinates of P_1 , P_2 , and P_3 in the fixed reference frame xP_1y . The i^{th} leg can be defined through the lengths of the two connecting rods: l_i and m_i (see Fig. 12). Thus twelve parameters are used to define a 3RRR manipulator.

This class of planar manipulators have been widely studied, and often used as an example, due to its simple kinematic architecture. Workspace analysis methods for similar manipulators were proposed in (Pennock & Kassner, 1993) and (Merlet et al.,1998) and the singularity locus of analogous manipulators was defined and studied in (Sefrioui & Gosselin, 1995) and (Wang & Gosselin, 1997).

The workspace of a 3RRR planar manipulator is a subset of the manifold containing all possible positions of the platform in the plane. Each point of the workspace will be identified by the coordinates x and y of point Q_1 in the fixed reference frame xP_1y and by the angle φ between x - and u -axes.

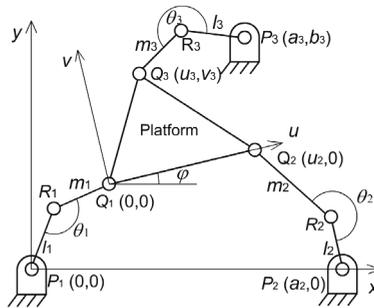


Fig. 12. A 3RRR manipulator.

The position of the i^{th} actuator is given by the angle θ , between the two rods composing each leg. Any point in the jointspace is therefore identified by the three angles $(\theta_1, \theta_2, \theta_3)$.

Any configuration of the manipulator can be represented through the six parameters $(x, y, \varphi, \theta_1, \theta_2, \theta_3)$. However, not any combination of these six parameters identifies a configuration of the manipulator, for the ensuing constraints imposed by the three legs must be satisfied:

$$f = 0 \tag{29}$$

where $f = (f_1, f_2, f_3)$, and

$$f_i = (P_i - Q_i(x, y, \varphi))^2 - l_i^2 - m_i^2 + 2m_i l_i \cos \theta_i, \quad i = 1, 2, 3 \tag{30}$$

Eq.(29) can be easily derived by expressing the coordinates of each Q_i in the fixed reference frame xP_1y , and by applying Carnot theorem to the three triangles $P_iQ_iR_i$.

The configuration space can be represented as the three dimensional manifold C described by Eq. (29) and embedded in the six dimensional manifold containing all the possible vectors $(x, y, \varphi, \theta_1, \theta_2, \theta_3)$.

Unlike the manipulators presented so far, the configuration space of planar 3RRR manipulators does not coincide with the workspace, and might be composed of more than one assembly configuration, therefore the proposed method will be applied to determine also the number of ACs and existence of feasible paths between any two configurations.

In order to derive the equation of the singularity locus the relationship between the first order displacements of the platform and the actuators is needed. Such relationship is obtained by differentiating Eq. (29):

$$\frac{\partial \mathbf{f}}{\partial \mathbf{s}} \delta \mathbf{s} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \delta \mathbf{q} = \mathbf{0} \quad (31)$$

where $\mathbf{s}=(x, y, \varphi)$ and $\mathbf{q}=(\theta_1, \theta_2, \theta_3)$. Parallel singularities occur when the platform can undergo infinitesimal displacements $\delta \mathbf{s}$, even though all actuators are locked, i.e. $\delta \mathbf{q}$ vanishes. Thus all singular points must satisfy the condition:

$$J(x, y, \varphi) = \det \left(\frac{\partial \mathbf{f}}{\partial \mathbf{s}} \right) = 0 \quad (32)$$

The singularity locus is a two-dimensional manifold defined by the zero level-set of the function J , on the three-dimensional configuration space C .

Lagrange's optimization method is used again to find out critical points. The Lagrangian function L can be defined as:

$$L(x, y, \varphi, \theta_1, \theta_2, \theta_3, \lambda_1, \lambda_2, \lambda_3) = J - \lambda_1 f_1 - \lambda_2 f_2 - \lambda_3 f_3 \quad (33)$$

where f_1 , f_2 , and f_3 , are defined by Eq.(30). The critical points of J constrained on C are the points where the gradient of L with respect to all its nine variables vanishes.

By equating to zero the derivatives of L with respect to the i^{th} actuator angle θ_i , the ensuing equations are obtained:

$$\lambda_i \sin(\theta_i) = 0, \quad i=1,2,3. \quad (34)$$

Therefore, the following four cases are given.

Case a): All Lagrange's multipliers λ_i are not equal to zero.

In this case the sine of the three angles θ_i must vanish (Eq.(34)), thus all three legs are completely outstretched or folded-up, for θ_i is equal to 0 or π . Such positions can be obtained by substituting all possible combinations of 0 and π into each θ_i of Eq.(29), which is reobtained as derivatives of L with respect to Lagrange's multipliers. By subtracting the first equation of Eq.(29) from the last two, two linear equations in x and y are obtained. From these linear equations, x and y can be determined as functions of the sine and cosine of φ , and back substituted into the first of Eq. (29), yielding a trigonometric equation in φ , which is easily solved through standard techniques. Lagrange's multipliers, which are useful for the classification of critical points, can be determined through the remaining derivatives of L .

Case b): i^{th} Lagrange's multiplier is equal to zero.

In this case, only the sines of θ_j and θ_k vanish, with j and k different from i . Analogous to the previous case, two equations for x , y , and φ are obtained by substituting all possible

combinations of 0 and π into the cosine of θ_j and θ_k , in the j^{th} and the k^{th} equations of Eq. (29). By subtracting one of such equations from the other, a linear equation in x is obtained:

$$g(y, \varphi)x + h(y, \varphi) = 0 \quad (35)$$

which yields x as a function of y and φ . By equating to zero the derivatives of L with respect to x , y , and φ , the ensuing equation is obtained:

$$\mathbf{A}(x, y, \varphi) \cdot (1, -\lambda_j, -\lambda_k)^T = \mathbf{0} \quad (36)$$

where \mathbf{A} is a 3×3 matrix, whose columns contain the gradients of J , f_j , and f_k with respect to variables x , y , and φ . Eq.(36) implies that the determinant of \mathbf{A} must vanish, which yields the third condition in x , y , and φ . By substituting the expression of x obtained from Eq.(35) into this equation and into the j^{th} equation of Eq.(29), the variable x is eliminated, and two polynomial equations in y and the tangent of $\varphi/2$ are obtained, which can be easily solved through Sylvester dyalitic elimination method (see Salmon, 1885). Among the solutions just obtained, there are some extraneous solutions, which can be easily got rid of, for at such solutions the two coefficients g and h of Eq.(35) vanish. The angles θ_j and θ_k are equal to 0 or π , whilst the angle θ_i can be derived from the i^{th} equation of Eq. (29). The j^{th} and k^{th} Lagrange's multipliers are obtained from Eq. (36), and the i^{th} is obviously zero.

Case c): i^{th} and j^{th} Lagrange's multipliers vanish.

By equating to zero the derivatives of L with respect to x , y , and φ , the ensuing equation is obtained:

$$\mathbf{B}(x, y, \varphi) \cdot (1, -\lambda_k)^T = \mathbf{0} \quad (37)$$

Where \mathbf{B} is a 3×2 matrix, whose columns contain the gradients of J and f_k with respect to variables x , y , and φ . Eq.(37) implies that all the three 2×2 minors of \mathbf{B} are singular, which yields three equations. By considering two of the three conditions just derived, along with the equation obtained by substituting 0 or π into θ_k in the k^{th} equation of Eq. (29), three equations in the variables x , y , and φ are obtained, which can be solved analogously to case b). It is possible to prove that, by equating to zero only two of the three 2×2 minor determinants, some extraneous solutions are introduced, which do not make the third determinant vanish. By imposing this last condition, it is possible get rid of such extraneous solutions.

Case d): All Lagrange's multipliers are equal to zero.

In this case the gradient of J with respect to x , y , and φ must vanish, which yields two linear equations in x and y , and a quadratic equation in x and y . This equation set can be solved by techniques analogous to case a).

Once the critical points are determined, they are all classified by means of the bordered Hessian, as discussed in Section 4.1, and the maximum increase and decrease directions in the neighbourhoods of the saddle points are determined.

Two numerical examples are presented hereafter, manipulators P_1 and P_2 . The kinematical structure of the two examples is summarized in Table 3, according to the parameterization adopted.

	a_2	a_3	b_3	u_2	u_3	v_3	l_1	m_1	l_2	m_2	l_3	m_3
P_1	10	3	10	10	3	3	1	2	10	2	6	7
P_2	10	3	10	10	3	3	1	2	4	2	5	6

Table 3. Parameters defining manipulators P_1 and P_2 .

In manipulator P_1 there are four positive maxima, nine positive 2-saddles, four negative 2-saddles, and no negative maxima. The four positive maxima are shown in Fig. 13a. Maxima M_1 and M_2 are joined by steepest ascent paths starting from some of the positive 2-saddles, while M_3 and M_4 are not connected to any other maximum by any steepest ascent path starting from any positive or negative 2-saddle. There are three ACs: one containing M_1 and M_2 , and the other two containing M_3 and M_4 . Manipulator P_1 was generated by imposing that the loop composed by leg 1, leg 2, the platform and the frame have two ACs, through the condition derived in (Foster & Cipra, 1998), and that leg 3 be able to completely outstretch in one of such ACs, but not in the other. Therefore one of the two ACs of the loop is split into two ACs by the fact that leg 3 can never outstretch, nor fold back.

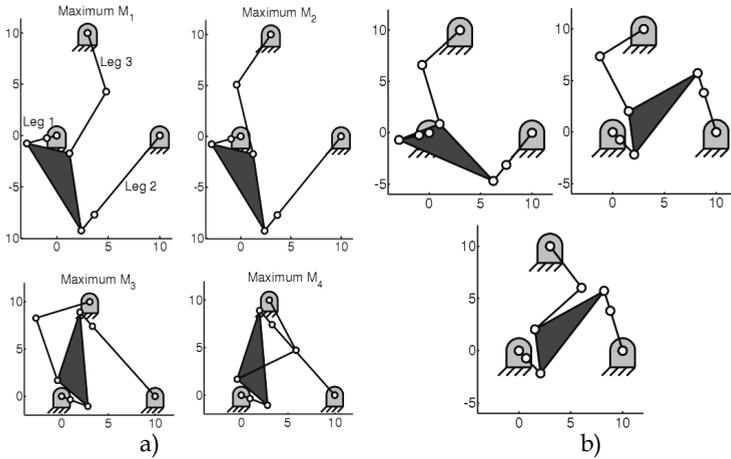


Fig. 13. a) The four maxima of manipulator P_1 .
 b) Three maxima of manipulator P_2 .

The analysis of the negative critical points shows that each of the tree ACs is split into two PSFRs, one positive, and one negative. Therefore, if the sign of the Jacobian determinant is the same at two configurations belonging to the same AC, a singularity-free path connecting them always exists.

However, the ACs are not always split into two PSFRs only, as manipulator P_2 shows. In manipulator P_2 there is only one AC, therefore any configuration of the manipulator is reachable, but this AC is split into four PSFRs, three positive and one negative. Fig. 12b shows three positive maxima belonging to the three positive PSFRs: many feasible paths connect these three configurations where the Jacobian determinant is positive, but none of them is free of parallel singularities.

7. Conclusion

This work presented a numerical method able to count and identify the PSFRs and the ACs carved by the singularity locus in the configuration space of a manipulator, and its application to three types of parallel manipulators.

In principle, this method works for any manipulator, but some very particular cases, where there are degenerate critical points of the Jacobian determinant. The application is rather simple, except the determination of all critical points of the Jacobian determinant on the configuration space. This part of the procedure reduces in most cases to the determination of all solutions to a polynomial equation set, that might be a very hard task in practice, although it is always theoretically possible.

However, if the determination of the critical points of the Jacobian determinant is viable, like the presented examples, the proposed method represents a stable and powerful tool for analyzing the topology of the singularity locus and for planning singularity-free paths.

The proposed method does not take into account the possible reduction of configuration space of a manipulator due to the mechanical interference between the links, or by actuator limits. The analysis of the singularity locus under the additional constraint that no collision between the links takes place is a possible future development of the proposed method, as well as its application to more parallel manipulators with six degrees of freedom.

8. References

- Bhattacharya, S., Hatwal, H., and Ghosh, A., 1998, "Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators," *Mechanism and Machine Theory*, 33(7), pp. 965-974.
- Chablat, D. and Wenger, P., 1998, "Working modes and aspects in fully parallel Leuven, Belgium, pp. 1964-1969.
- Chase, T.R. and Mirth, J.A., 1993, "Circuits and branches of single-degree-of-freedom planar linkages," *Journal of Mechanical Design*, 115(2), pp. 223-230.
- Dasgupta, B. and Mruthunjaya, T., 1998, "Singularity-free path planning for the Stewart platform manipulator," *Mechanism and Machine Theory*, 33(6), pp. 711-725.
- Di Gregorio, R., and Parenti-Castelli, V., 1998, "A Translational 3-dof Parallel Manipulator," *Advances in Robot Kinematics: Analysis and Control* Lenarcic, J., and Husty, M. L., Eds., Kluwer Academic Publishers, pp. 49-58.
- Di Gregorio, R., and Parenti Castelli V., 2002, "Mobility Analysis of the 3-UPU Parallel Mechanism Assembled for a Pure Translational Motion," *Journal of mechanical Design*, Vol. 124, pp. 259-264.
- Dou, X. and Ting, K.L., 1998, "Identification of singularity free joint rotation space of two-dof parallel manipulators," *proceedings of DETC'98*, Atlanta, Georgia.
- Fletcher, R., 1987, *Practical Methods of Optimization*, John Wiley & Sons, Chichester.
- Foster, D.E. and Cipra, R.J., 1998, "Assembly configurations and branches of single-loop mechanism with pin joints and sliding joints," *Journal of Mechanical Design*, 120(3), pp. 387-391.
- Foster, D.E. and Cipra, R.J., 2002, "An automatic method for finding the assembly configurations of planar non-single-input-dyadic mechanisms," *Journal of Mechanical Design*, 124(1), pp. 58-67.

- Gosselin, C. and Angeles, J., 1990, "Singularity analysis of closed-loop kinematic chains," *IEEE Transactions On Robotics and Automation*, 6(3), pp. 281-290.
- Hirsch, M.W., 1976, *Differential Topology*, Springer, New York.
- Innocenti, C., and Parenti-Castelli, V. , 1993, "Echelon form solution of direct Kinematics for the general fully-parallel spherical wrist," *Mechanism and Machine Theory*, vol. 28, No.4, pp. 553-561.
- Kevin Jui C.K. and Qiao Sun, 2005, "Path tracking of parallel manipulators in the presence of force singularity," *Journal of Dynamic Systems, Measurement and Control*, 127, pp. 550-563.
- Merlet, J.P., Gosselin, C.M., and Mouly, N., 1998, "Workspaces of planar parallel manipulators," *Mechanism and Machine Theory*, 33(1/2), pp.7-20.
- Milnor, J., 1969, *Morse Theory*, Princeton University Press.
- Mirth, J.A. and Chase, T.R., 1993, "Circuit analysis of Watt chain six-bar mechanisms," *Journal of Mechanical Design*, 115(2), pp.214-222.
- Midha, A., Zhao, Z.L., and Her, I., 1985, "Mobility conditions for planar linkages using triangle inequality and graphical interpretation," *Journal of Mechanical Design*, 107(3), pp. 394-400.
- Paganelli, D., 2008, "Topological Analysis of Singularity Loci for Serial and Parallel Manipulators," Phd. Thesis, University of Bologna.
- Paul, B., 1979, "A reassessment of Grashof's Criterion," *Journal of Mechanical Design*, 101, pp. 515-518.
- Parenti-Castelli, V., Di Gregorio, R., and Lenarcic, J., 1998, "Sensitivity to Geometric Parameter Variation of a 3-dof Fully-Parallel Manipulator," *Proceedings of the 3rd International Conference on Advanced Mechatronics*, pp. 364-369.
- Pennock, G.R. and Kassner, D.J., 1993, "The workspace of a general geometry planar three-degree-of-freedom platform-type manipulator," *Journal of Mechanical Design*, 115(1), pp. 269-276.
- Salmon, G. , 1885, *Modern Higher Algebra*, Hodges, Figgis & Co., Dublin.
- Sen, S. , Dasgupta, B., and Mallik, A.K., 2003, "Variational approach for singularity-free path-planning of parallel manipulators," *Mechanism and Machine Theory*, 38, pp. 1165-1183.
- Sefrioui, J., Gosselin, C., 1994, "Etude et Representation des Lieux de Singularite des Manipulateurs Paralleles Spheriques a Trois Degres de Liberte avec Actionneurs Prismatiques," *Mechanisms and Machines Theory*, 29, No.4, pp.559-579.
- Sefrioui, J. and Gosselin, C.M., 1995, "On the quadratic nature of the singularity curves of planar three-degree-of freedom parallel manipulators," *Mechanism and Machine Theory*, 30(4), pp.533-551.
- Tsai, L. W., 1996, "Kinematics of a Three-dof Platform With Three Extensible Limbs," *Recent Advances in Robot Kinematics*, Lenarcic, J., and Parenti-Castelli, V., Eds., Kluwer Academic Publishers, pp. 401-410.
- Wang, J., and Gosselin, C.M., 1997, "Singularity loci of planar parallel manipulators with revolute actuators," *Robotics and Autonomous Sustersms*, 21, pp.377-398.
- Whitehead, W., 1978, *Elements of Homotopy Theory*, Springer, New York.
- Zlatanov, D., Bonev I.A., and Gosselin, C.N., 2002, "Constraint Singularities of Parallel Mechanisms," *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, pp. 496-502.

Vision-based 2D and 3D Control of Robot Manipulators

Luis Hernández¹, Hichem Sahli² and René González³

¹Universidad Central de Las Villas (UCLV), ²Vrije Universiteit Brussel (VUB),

³Empresa de Automatización Integral (CEDAI)

¹Cuba, ²Belgium, ³Cuba

1. Introduction

Robotics has been a paradigm for science in the last few decades. At first, scientists' efforts were devoted to the solution of the problem of planning and control of the motion of robot manipulators. However, the motion control of robot manipulators in unstructured environments is today an attractive scientific problem. An interesting solution for motion control is the use of sensor information, such as computer vision, in the system's feedback. Several works and tools have been developed in recent years in this field (Corke; 2005), (Chaumette and Hutchinson; 2006). The more typical approaches consider visual perception for servoing and for the so called *look and move* (Hutchinson et al.; 1996). Visual servoing (Kelly et al.; 2000) can be classified into two approaches: camera-in-hand or camera-to-hand (Flandin et al.; 2000). In camera-to-hand robotic systems, multiple cameras or a single camera fixed in the world-coordinate frame capture images of both, the robot and its environment. The tracking of the object with visual feedback can be made in 2D or 3D. An interesting solution for the visual servoing of camera-in hand robot manipulators in 2D can be found in e. g. (Bonfe et al.; 2002) and (Hernández et al.; 2008) where stability demonstration of a decoupled controller have been presented. For 3D tracking some solutions reported are *look and move* controller, with one camera (Sim et al.; 2002) or more than one camera (Xie et al.; 2005). A 3D visual servoing with stability analysis in continuous time is presented by (Hernández et al.; 2008a) and (Kelly et al.; 2006) present a direct visual servoing with transpose Jacobian control technique for regulation of robot manipulators in the 3D Cartesian space. In a similar vein (Enescu et al.; 2006) present mobile robot navigation for person tracking using a stereo head-camera.

In this chapter, we consider the control problem of camera-in-hand robot manipulators in 2D and 3D. In both cases only one camera is mounted on the robot's arm, which supplies visual information of the environment, with the aim of moving the manipulator by maintaining the image of the tracked object (a sphere) in the centre of the image plane, despite the possible movements of the object. In the 3D control the constant radius is used as a feature too. In this work, the proposed control system considers two loops in cascade, an internal loop solving the robots' joint control, and an external loop implementing a dynamic look and move visual controller. A stability analysis in discrete time is developed under the conditions that it is possible to approximate the dynamic effect of the internal loop as an external loop time delay (Corke; 1996) and (Bonfe et al.; 2002). The more classic presentations of servovisual control are velocity controllers, based in the term *feature Jacobian*, (Chaumette and Hutchinson; 2006)

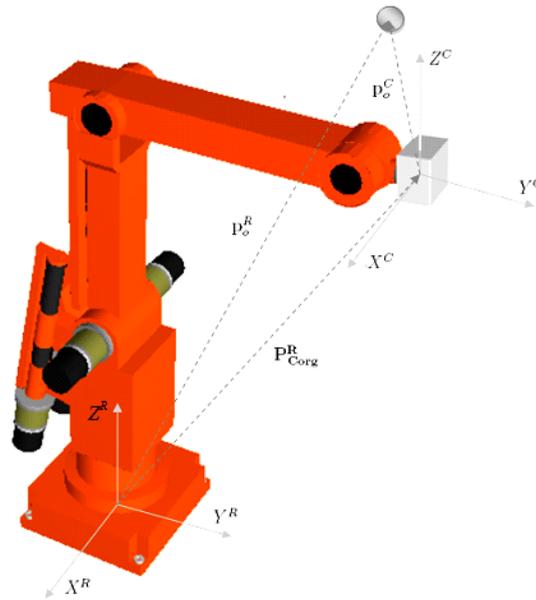


Fig. 1. Camera-in-hand robotic system

(Chaumette and Hutchinson; 2007), but we present the robotic and vision systems modeled for small variations about the operating point for position control and in these conditions the stability of the whole system are balanced.

A particular study is made using an ASEA IRB6 robot manipulator which has mechanically decoupled wrist. This allows keeping the orientation of the camera's optical axis while the arm is moving.

In order to validate the proposed control system, a general stability analysis is presented and an analysis of the step response in a regulator type system is made. The disturbance is interpreted as initial conditions. To illustrate the proposed controller, the control system stability and its performance, both simulation results as well as experimental results using the ASEA IRB6 robot manipulator are presented. Our experimental results confirm the expected step response in the image plane, with good time performance and zero steady-state error.

2. Robotic System Model

As shown in Fig. 1, the robotic system considered has a robot manipulator with a camera in its hand. The basic mathematical description of this system consists of the robot and the camera model.

2.1 Robot Kinematics Model

The kinematics of a manipulator gives the relationship between the joint positions \mathbf{q} and the corresponding tool translational (x, y, z) and angular position (α, β, γ) . For an n -axis rigid link manipulator, the forward kinematic solution, \mathbf{T} , could be computed for any manipulator,

irrespective of the number of joints or kinematic structure (Barrientos et al.; 1997). A generic mathematical representation could be:

$$\begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T = f(q_1, q_2, \dots, q_n) = \mathbf{T} \quad (1)$$

For manipulator path planning, the inverse kinematic solution \mathbf{T}^{-1} gives the joint angles \mathbf{q} required to reach the specified tool's position. In general this solution is non-unique (Barrientos et al.; 1997). A generic mathematical representation could be:

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T = g(x, y, z, \alpha, \beta, \gamma) = \mathbf{T}^{-1} \quad (2)$$

2.2 Robot Dynamics Model

In the absence of friction or other disturbances, the dynamics of a serial n -link rigid robot manipulator can be written as (Kelly and Santibáñez; 2003)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (3)$$

where:

$\mathbf{M}(\mathbf{q})$ $n \times n$ symmetric positive definite manipulator inertia matrix

\mathbf{q} $n \times 1$ vector of joint displacements

$\mathbf{C}(\mathbf{q})$ $n \times 1$ vector of centripetal and Coriolis torques

$\mathbf{g}(\mathbf{q})$ $n \times 1$ vector of gravitational torques

$\boldsymbol{\tau}$ $n \times 1$ vector of applied joint torques

2.3 Camera Model

According to Fig. 1 we consider a vision system mounted on the robot tool, with coordinate frame, Σ_C , which moves in the space $\langle X^R, Y^R, Z^R \rangle$ of the robot coordinate frame Σ_R . The origin of the camera coordinate frame (tool frame) with respect to the robot coordinate frame is represented by the vector $\mathbf{p}_{\text{Corg}}^R$ with coordinates $\begin{bmatrix} p_{xc}^R & p_{yc}^R & p_{zc}^R \end{bmatrix}^T \in \mathfrak{R}^3$.

Points of interest in the workspace are identified as \mathbf{p}_o^C with coordinates $\begin{bmatrix} p_{xo}^C & p_{yo}^C & p_{zo}^C \end{bmatrix}^T \in \mathfrak{R}^3$ in the camera reference system Σ_C , and $\mathbf{p}_o^R = \begin{bmatrix} p_{xo}^R & p_{yo}^R & p_{zo}^R \end{bmatrix}^T \in \mathfrak{R}^3$ in the robot reference system Σ_R .

2.3.1 Camera Model for Visual-Based 2D Control

The image acquired by the camera supplies a two-dimensional (2D) array of brightness values from a three-dimensional (3D) scene. This image may undergo various types of computer processing to enhance image properties and extract image features. In this paper we work with a known spherical object with radius r_o . The centre of gravity is used as object features (state) for the 2D control. We assume that the image features are the projection into the 2D image plane of the 3D characteristics of the scene.

The object moves in a plane parallel to the plane $\langle Y^C, Z^C \rangle$ of the camera coordinate frame Σ_C . The camera's optical axis coincides with X^C axis.

A perspective projection with a focal λ is assumed. The point \mathbf{p}_o^C with coordinates $\begin{bmatrix} p_{xo}^C & p_{yo}^C & p_{zo}^C \end{bmatrix}^T$ in the camera frame projects onto a point (u, v) (pixel) on the image plane.

Let \mathbf{p}_o^C be the position of the object's centre of gravity. According to the perspective projection (Hutchinson et al.; 1996), we have

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -\alpha \frac{\lambda}{p_{x_o}^C} \begin{bmatrix} p_{y_o}^C \\ p_{z_o}^C \end{bmatrix} \quad (4)$$

where α is the scaling factor in pixels per meter due to the camera sampling, λ is the focal length of the camera lens. This model is also called the imaging model (Kelly et al.; 2000).

The object distance $p_{x_o}^C$ along the camera's optical axis X^C is constant.

The orientation of the camera frame with respect to the robot frame in the plane $\langle Y^R, Z^R \rangle$ of Σ_R is denoted by $\mathbf{R}_R^C = \mathbf{R}_R^C(\psi) \in SO(2)$

where

$$\mathbf{R}_R^C(\psi) = \begin{bmatrix} C\psi & -S\psi \\ S\psi & C\psi \end{bmatrix} \quad (5)$$

Where Si is $\sin(i)$ and Ci is $\cos(i)$, with ψ being the mechanical angle between Σ_C and Σ_R . Note that $\mathbf{R}_R^C(\psi)$ is an orthogonal matrix.

Following the configuration of Fig. 1, and taking into account equation (5), it is possible to obtain the y and z components of vector \mathbf{p}_o^C as:

$$\begin{bmatrix} p_{y_o}^C \\ p_{z_o}^C \end{bmatrix} = \mathbf{R}_R^C(\psi)^T \left(\begin{bmatrix} p_{y_o}^R \\ p_{z_o}^R \end{bmatrix} - \begin{bmatrix} p_{y_c}^R \\ p_{z_c}^R \end{bmatrix} \right)$$

and finally, according to (4) and (5) we obtain:

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -h \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} \left(\begin{bmatrix} p_{y_o}^R \\ p_{z_o}^R \end{bmatrix} - \begin{bmatrix} p_{y_c}^R \\ p_{z_c}^R \end{bmatrix} \right) \quad (6)$$

Where $h = \alpha \frac{\lambda}{p_{x_o}^C}$.

2.3.2 Camera Model for Visual-Based 3D Control

In the 3D control the object moves in the space $\langle X^R, Y^R, Z^R \rangle$ of the robot coordinate frame Σ_R ; and the camera's optical axis coincide with the Z^C axis of the camera's coordinate frame Σ_C . We continue work with a known spherical object with radius r_o ; and in this case the object features (state) are the centre of gravity and image radio.

According to this consideration Equation (4) becomes:

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -\alpha \frac{\lambda}{p_{z_o}^C} \begin{bmatrix} p_{x_o}^C \\ p_{y_o}^C \end{bmatrix} \quad (7)$$

In order to estimate the object's distance $p_{z_o}^C$ along the camera's optical axis Z^C it is possible to use a well-known object size and the corresponding apparent size in the image plane (Corke; 1996). Our object is a sphere with radius r_o and the apparent image radius is r . Following Equation (7), $p_{z_o}^C$, r_o and r can be related by,

$$r = -\frac{\alpha \lambda r_o}{p_{z_o}^C} \quad (8)$$

Finally combining Equations (7) and (8) we define the object state (feature) vector:

$$\tilde{\zeta}' = \begin{bmatrix} u \\ v \\ r \end{bmatrix} = -\alpha \frac{\lambda}{p_{z0}^C} \begin{bmatrix} p_{x0}^C \\ p_{y0}^C \\ r_o \end{bmatrix} \quad (9)$$

The orientation of the camera frame, Σ_C , with respect to the robot frame, Σ_R is denoted by $\mathbf{R}'_C = \mathbf{R}'_C(\phi, \theta, \psi) \in SO(3)$ where \mathbf{R}'_C can be described by Euler angles (Barrientos et al.; 1997).

Let (ϕ, θ, ψ) being the given set of Euler angles, the following rotations: Frame rotation by the angle ϕ about axis Z , frame rotation by the angle θ about axis Y' and frame rotation by the angle ψ about axis Z'' .

In this case the rotation matrix is:

$$\mathbf{R}'_C(\phi, \theta, \psi) = \begin{bmatrix} C\phi C\theta C\psi - S\phi S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta \\ S\phi C\theta C\psi + C\phi S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta \\ -S\theta C\psi & S\theta S\psi & C\theta \end{bmatrix} \quad (10)$$

Following the configuration of Fig. 1, and taking into account Equation (10), it is possible to obtain the components of vector \mathbf{p}_o^C as:

$$\begin{bmatrix} p_{x0}^C \\ p_{y0}^C \\ p_{z0}^C \end{bmatrix} = \mathbf{R}'_C(\phi, \theta, \psi)^T \left(\begin{bmatrix} p_{x0}^R \\ p_{y0}^R \\ p_{z0}^R \end{bmatrix} - \begin{bmatrix} p_{xc}^R \\ p_{yc}^R \\ p_{zc}^R \end{bmatrix} \right) \quad (11)$$

2.3.3 Linear Camera Model

According to Fig. 1, the axes Z^R and Z^C are parallel and, for simplicity in the analysis, they are taken with the same direction. In this case, the Euler's angles are $\phi = 0, \theta = 0$ and ψ varies according the rotation of the robot's base. In these conditions Equation (10) becomes:

$$\mathbf{R}'_C(\psi) = \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

According to the assumption A2.1, only a small variation about the operating point will be taken into account. If we set $\psi_0 = 0$ as operating point and take the linear approximation of (12), the variations of Equation (11) can be written as:

$$\begin{bmatrix} \delta p_{x0}^C \\ \delta p_{y0}^C \\ \delta p_{z0}^C \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} \delta p_{x0}^R \\ \delta p_{y0}^R \\ \delta p_{z0}^R \end{bmatrix} - \begin{bmatrix} \delta p_{xc}^R \\ \delta p_{yc}^R \\ \delta p_{zc}^R \end{bmatrix} \right) \quad (13)$$

We can express their perturbation using Taylor series expansion. Since we look for a linear dependency on the variables, we only use the first order terms of the series. With this analysis Equation (9) becomes:

$$\delta \tilde{\zeta}' = \begin{bmatrix} \delta u \\ \delta v \\ \delta r \end{bmatrix} = -\alpha \frac{\lambda}{p_{z0}^C} \begin{bmatrix} \delta p_{x0}^C \\ \delta p_{y0}^C \\ -r_o \frac{\delta p_{z0}^C}{p_{z0}^C} \end{bmatrix} \quad (14)$$

and finally, according to (13) and (14) we obtain:

$$\delta \tilde{\zeta}' = -\alpha \frac{\lambda}{p_{z0}^C} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{r_o}{p_{z0}^C} \end{bmatrix} \left(\begin{bmatrix} \delta p_{x0}^R \\ \delta p_{y0}^R \\ \delta p_{z0}^R \end{bmatrix} - \begin{bmatrix} \delta p_{xc}^R \\ \delta p_{yc}^R \\ \delta p_{zc}^R \end{bmatrix} \right) \tag{15}$$

In the case of $\psi \neq 0$ Equation (15) becomes:

$$\delta \tilde{\zeta}' = -h' \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & -\frac{r_o}{p_{z0}^C} \end{bmatrix} \left(\begin{bmatrix} \delta p_{x0}^R \\ \delta p_{y0}^R \\ \delta p_{z0}^R \end{bmatrix} - \begin{bmatrix} \delta p_{xc}^R \\ \delta p_{yc}^R \\ \delta p_{zc}^R \end{bmatrix} \right) \tag{16}$$

Where $h' = \alpha \frac{\lambda}{p_{z0}^C}$.

3. Control Problem

Recall that, our aim is moving the manipulator maintaining the image of the tracked object (its centre of gravity) coincident to the centre of the image plane, for the 2D control; and the image of the tracked object (its centre of gravity) coincident to the centre of the image plane with a given image radius, for the 3D control.

The control problem is formulated as the design of a controller which computes a control signal Δ corresponding to the movement of the robot's arm in such a way that the actual image object state reaches the desired state.

3.1 2D Control Problem Formulation

For 2D control the desired state, $[u_d \quad v_d]^T$, is the centre of gravity of the object's image. The *state error* being defined as:

$$\tilde{\xi} = \tilde{\zeta}_d - \tilde{\zeta} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix}$$

which could be calculated at every measurement time and used to move the robot in a direction allowing its decrease. Therefore, the control aims at ensuring that

$$\lim_{t \rightarrow \infty} \tilde{\xi} = \lim_{t \rightarrow \infty} [\tilde{u}(t) \quad \tilde{v}(t)]^T = 0 \in \mathbb{R}^2$$

provided that the initial feature error $\tilde{\xi}(0)$ is sufficiently small.

We make the following assumptions for the 2D control problem:

A1.0 The object is static.

A1.1 There exists a robot joint configuration \mathbf{q}_d for which $\tilde{\zeta}_d = \tilde{\zeta}(\mathbf{q}_d)$.

A1.2 ψ is the mechanical angle between Σ_C and Σ_R , and can take different values but will be constant in each experiment.

A1.3 The axes X^R and X^C , Fig. 1, are parallels.

A1.4 The distance p_{x0}^C from the camera to the object is constant.

Assumption A1.0, ensures that only the control problem is evaluated. Assumption A1.1, ensures that the control problem is solvable. Assumption A1.2, stability condition, will be different for each value of ψ . Assumption A1.3 maintains the condition of equation (5).

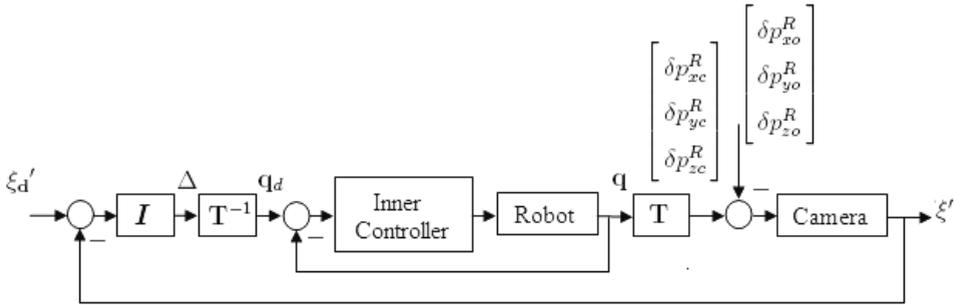


Fig. 2. Control scheme with visual feedback

3.2 3D Control Problem Formulation

In the case of 3D control the desired state object’s image centre of gravity and radius is represented by $[u_d \ v_d \ r_d]^T$. The *state error* is defined as

$$\tilde{\zeta}' = \zeta'_d - \zeta' = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{r} \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \\ r_d \end{bmatrix} - \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

The control aims at ensuring that

$$\lim_{t \rightarrow \infty} \tilde{\zeta}' = \lim_{t \rightarrow \infty} [\tilde{u}(t) \ \tilde{v}(t) \ \tilde{r}(t)]^T = 0 \in \mathbb{R}^3$$

provided that the initial feature error $\tilde{\zeta}'(0)$ is sufficiently small.

We make the following assumptions the 3D control problem:

- A2.0 The object is static for position regulation.
- A2.1 There exists a robot joint configuration \mathbf{q}_d for which $\zeta'_d = \zeta'(\mathbf{q}_d)$.
- A2.2 The axes Z^R and Z^C , Fig. 1, are parallel.
- A2.3 The initial feature error $\tilde{\zeta}'(0)$ is sufficiently small.
- A2.4 The object moves with low velocity in following a simple trajectory.

Assumption A2.0, ensures that the regulation problem is evaluated. Assumption A2.1, ensures that the control problem is solvable. Assumption A2.2, conditions for Euler angles. Assumption A2.3 makes possible the linear analysis about the operating point. Assumption A2.4 conditions of trajectory following.

3.3 Controller With Visual Feedback

For our control problem formulation, the state vector of the object can only be measured through the camera, as such, a direct knowledge of the desired joint position \mathbf{q}_d is not available. Nevertheless, the desired joints position can be obtained as a result of the estimated control signal Δ and the solution of the kinematics problems.

The implemented closed-loop block diagram can be described as shown in Fig. 2. The control system has two loops in cascade, the internal loop solving the robots' joint control, and the external loop implementing a dynamic *look and move* visual controller.

The inner control loop has an open control architecture; in this architecture it is possible to implement any type of controller. One possibility is to use a non-linear controller in the state variables, called torque-calculated (Kelly and Santibáñez; 2003) having the following control equation:

$$\tau = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_{vi}\dot{\tilde{\mathbf{q}}} + \mathbf{K}_{pi}\tilde{\mathbf{q}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

Where $\mathbf{K}_{pi} \in \mathfrak{R}^{n \times n}$ and $\mathbf{K}_{vi} \in \mathfrak{R}^{n \times n}$ are the symmetric positive-definite matrices and $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$. Kelly (Kelly and Santibáñez; 2003) demonstrated that with this configuration the system behaves in a closed loop as a linear multivariable system, decoupled for each robot's joint, suggesting that the matrices could be specified as:

$$\begin{aligned}\mathbf{K}_{pi} &= \text{diag}\{\omega_1^2, \dots, \omega_n^2\} \\ \mathbf{K}_{vi} &= \text{diag}\{\omega_1, \dots, \omega_n\}\end{aligned}$$

In this way each joint behaves as a critically damping second order linear system with bandwidth ω_i . The bandwidth ω_i determines the speed of response of each joint. In such way the dynamic effect of the internal loop could be independent with regard to the external loop, being, according to (Hernández et al.; 2008), under the conditions that:

$$\mathbf{q}(t) = \mathbf{q}_d(t) \quad \forall t > 0 \quad (17)$$

Nevertheless, the vision-based control systems are fully sampled data systems. The feedback sensor has some dynamic characteristics such as: transport delay of pixel camera, image processing algorithms, communication between the vision system and control computer, etc. Åström (Astrom and Wittenmark; 1990) established that the sampling rate of digital control systems should be between 10 and 30 time the desired closed loop bandwidth. For the case of a 20Hz vision system the close loop bandwidth should be between 0.66 to 2 Hz.

With these conditions for the internal and external loop it is possible to make a design in order to avoid the dynamic effect of the internal loop in relation with the dynamics of the external loop (Lange and Hirzinger; 2003), a complete analysis of this topic can be found in (Hernández et al.; 2008).

But, if we analyze the control problem in the field of digital control systems, other dynamic representation of the set robot vision system could be as one or two delay units for the vision set (Corke; 1996) or for the robot (Bonfe et al.; 2002). Using this consideration we modify Equation (17) as,

$$\mathbf{q}(k) = \mathbf{q}_d(k-1) \quad \forall k > 0 \quad (18)$$

In this chapter, we consider a simpler approach, which consists of directly using the image feature vectors $\tilde{\zeta}$ for 2D control or $\tilde{\zeta}'$ for 3D control, being the difference between the centre of the image plane ζ_d or ζ'_d and the centre of gravity of the object in the image space ζ or ζ' ; and for 3D control the difference between the desired radius in the image plane r_d and the actual radius of the object in the image plane r (*image coordinate frame*). This error depends on the object's absolute position, in the task space \mathbf{p}_o^R , and the camera's centre position, \mathbf{p}_{Corg}^R , according to Equations (6) 2D or (16) 3D.

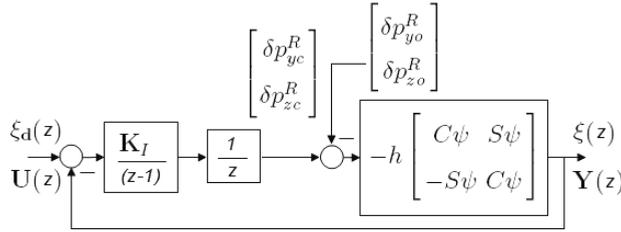


Fig. 3. 2D Vision-based simplified control scheme

4. 2D Vision-based Control. Stability Analysis

A very simple I controller can be used in this control scheme (Hernández et al.; 2008), for that case the control law can be given by:

$$\Delta = \mathbf{K}_I \int \tilde{\zeta} \quad (19)$$

Where $\mathbf{K}_I \in \mathbb{R}^{2 \times 2}$ is the symmetric integral matrix:

$$\mathbf{K}_I = \begin{bmatrix} -K_{I_1} & 0 \\ 0 & -K_{I_2} \end{bmatrix} \quad (20)$$

As in Sim's work (Sim et al.; 2002), Δ can be interpreted as the coordinates increment in the world space as a result of the image feature error $\tilde{\zeta}$. Solving the inverse kinematics problem T^{-1} it is possible to obtain \mathbf{q}_d . The proposed system works as a regulator system, because $\tilde{\zeta}_d$ is constant and can be set = 0.

Taking into account Fig. 2, obtaining the discrete equivalence of the controller of Equation (19) and according to Equations (6) and (18); a simplified diagram can be obtained as shown in Fig. 3.

Making

$$\mathbf{K} = -\alpha \frac{\lambda}{p_{x_0}^c} \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} = -h \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} \text{ and } \mathbf{K}_I = \begin{bmatrix} -K_{I_1} & 0 \\ 0 & -K_{I_2} \end{bmatrix}$$

according to Fig. 3 and if we consider the disturbance $\begin{bmatrix} \delta p_{xc}^R & \delta p_{yc}^R \end{bmatrix}^T$ as the system's initial conditions, the closed loop transfer function, taking a sampling period of 50ms, can be written as:

$$\frac{0.05\mathbf{K}_I\mathbf{K}}{(z^2 - z)} [\mathbf{U}(z) - \mathbf{Y}(z)] = \mathbf{Y}(z) \quad (21)$$

where

$$\mathbf{Y}(z) = \zeta(z)$$

Solving and taking the inverse Z transform we obtain:

$$\mathbf{y}(k+2) - \mathbf{y}(k+1) = -0.05\mathbf{K}_I\mathbf{K}\mathbf{y}(k) + 0.05\mathbf{K}_I\mathbf{K}\mathbf{u}(k) \quad (22)$$

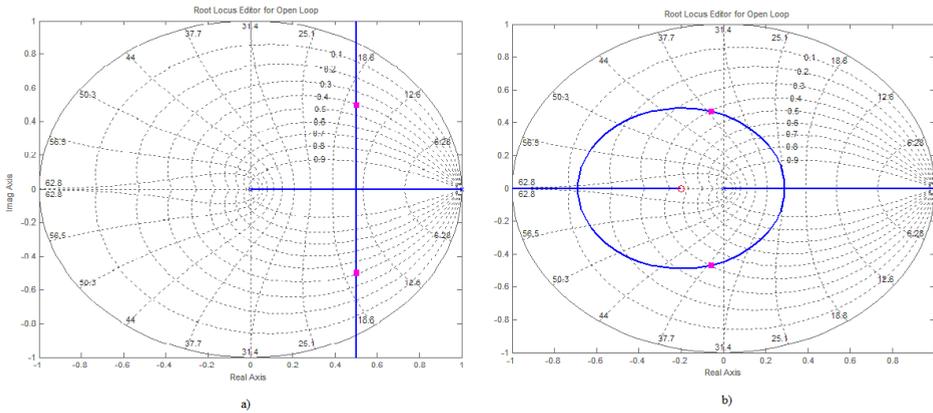


Fig. 4. Root-locus of axis model for: a) I controller with closed poles for gain $hK_I = 10$. b) PI controller with Closed poles for gain $hK_I = 30$

Taking into account that the aim is moving the manipulator maintaining the image of the tracked object (its centre of gravity) coincident to the centre of the image plane, we can make $u=0$ and $v=0$, then $\mathbf{u}=0$ and (22) becomes

$$\mathbf{y}(k+2) = \mathbf{y}(k+1) - 0.15\mathbf{K}_I\mathbf{K}\mathbf{y}(k) \quad (23)$$

The Equations (23) can be represented in the state space as

$$\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.15\mathbf{K}_I\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix} \quad (24)$$

where

$$\Phi = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.15\mathbf{K}_I\mathbf{K} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.05hC\psi K_{I1} & -0.05hS\psi K_{I2} & 1 & 0 \\ 0.05hS\psi K_{I1} & -0.05hC\psi K_{I2} & 0 & 1 \end{bmatrix}$$

In this work a robot manipulator ASEA IRB6 with camera in hand is used as case study. This type of robot has the wrist mechanically decoupled from the arm's movements, this allows to maintain the orientation of the camera. It is possible to establish ψ constant and for simplicity we set $\psi = 0$.

With this consideration and according to Equation (4) Equation (24) can be transformed into two decoupled control systems, one in each axis. The representation of these systems in the space state is:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05hK_{I1} & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \quad (25)$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05hK_{I2} & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \quad (26)$$

The system's stability is determined by the roots of the characteristic polynomial of matrix Φ . In these conditions it is easy to design the regulator for each axis.

A better illustration is shown by the root-locus diagram of the systems (25) or (26), given in Fig. 4 a), where instability is clearly indicated as the loop gain increases. If we use a *PI* controller, including a zero in $z = -0.2$, the diagram of Fig. 4 a) is modified as is shown in Fig. 4 b). In this case the stability condition of the system has been increased.

With the *PI* controller Equations (25) and (26) are modified as:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.01hK_{I_1} & 1 - 0.05hK_{I_1} \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \quad (27)$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.01hK_{I_2} & 1 - 0.05hK_{I_2} \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \quad (28)$$

5. 3D Vision-based Control. Stability Analysis

Similar to the 2D vision-based control analysis, we use a very simple *I* controller in the control scheme, for that case the control law is given by:

$$\Delta' = \mathbf{K}'_I \int \tilde{\zeta}' \quad (29)$$

Where $\mathbf{K}'_I \in \mathbb{R}^{3 \times 3}$ is the symmetrical integral matrix:

$$\mathbf{K}'_I = \begin{bmatrix} -K'_{I_1} & 0 & 0 \\ 0 & -K'_{I_2} & 0 \\ 0 & 0 & K'_{I_3} \end{bmatrix} \quad (30)$$

In this case too, Δ' can be interpreted as the coordinates increment in the world space as a result of the image feature error $\tilde{\zeta}'$. Solving the inverse kinematics problem T^{-1} it is possible to obtain \mathbf{q}_d . An analysis of this control scheme in continuous time can be found in (Hernández et al.; 2008a), where also the control performance in following simple trajectories in 3D is presented.

The system works as regulator, because $\tilde{\zeta}'_d$ is constant and can be set = 0. In these conditions, following the analysis of Section 4, using Equations (15) and (29) the system's output $[u(k) \ v(k) \ r(k)]^T$, can be obtained as decoupled equations by axis, as:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05h'K'_{I_1} & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05h'K'_{I_2} & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \quad (32)$$

$$\begin{bmatrix} r(k+1) \\ r(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05\frac{h'r_o}{p'_{z_o}}K'_{I_3} & 1 \end{bmatrix} \begin{bmatrix} r(k) \\ r(k+1) \end{bmatrix} \quad (33)$$

We considered the disturbance $[\delta p'_{x_o} \ \delta p'_{y_o} \ \delta p'_{z_o}]^T$ as system initial conditions.

With $\psi \neq 0$ as operating point, for the three previous Equations, according Equation (16), only (33) is enabled. Its stability analysis can be done in a Root-locus similar to Fig. 4. For the analysis along the axes v and u Equation (16) can be simplified as:

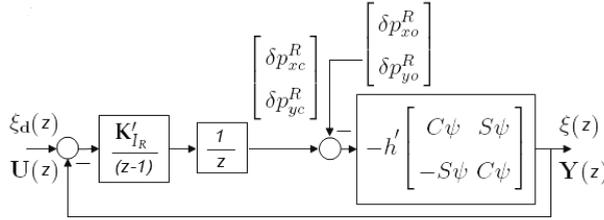


Fig. 5. 3D Vision-based simplified control scheme

$$\delta \xi'' = -\alpha \frac{\lambda}{p_{z0}^C} \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} \left(\begin{bmatrix} \delta p_{x0}^R \\ \delta p_{y0}^R \end{bmatrix} - \begin{bmatrix} \delta p_{xc}^R \\ \delta p_{yc}^R \end{bmatrix} \right) \tag{34}$$

For simplicity the δ has been avoided in Equations.

Taking into account Fig. 2 and according to Equation (17) a simplified diagram can be obtained as shown in Fig. 5.

Making

$$\mathbf{K}' = -\alpha \frac{\lambda}{p_{z0}^C} \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} = -h' \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix}$$

and

$$\mathbf{K}'_{IR} = \begin{bmatrix} -K'_{I1} & 0 \\ 0 & -K'_{I2} \end{bmatrix}$$

according to Fig. 5 and if we consider the disturbance $\begin{bmatrix} \delta p_{x0}^R & \delta p_{y0}^R \end{bmatrix}^T$ as the system's initial conditions, the closed loop transfer function, taking a sampling period of 50ms, can be written as:

$$\frac{0.05\mathbf{K}'_{IR} \mathbf{K}'}{(z^2 - z)} [\mathbf{U}(z) - \mathbf{Y}(z)] = \mathbf{Y}(z) \tag{35}$$

where

$$\mathbf{Y}(z) = \xi''(z) \tag{36}$$

solving and taking the inverse Z transform we obtain:

$$\mathbf{y}(k+2) - \mathbf{y}(k+1) = -0.15\mathbf{K}'_{IR} \mathbf{K}' \mathbf{y}(k) + 0.15\mathbf{K}'_{IR} \mathbf{K}' \mathbf{u}(k) \tag{37}$$

In the same way as in Section 4, we can make $u=0$ and $v=0$, then $\mathbf{u}=0$ and (37) becomes

$$\mathbf{y}(k+2) = \mathbf{y}(k+1) - 0.15\mathbf{K}'_{IR} \mathbf{K}' \mathbf{y}(k) \tag{38}$$

The Equations (38) can be represented in the state space as

$$\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.05\mathbf{K}'_{IR} \mathbf{K}' & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix}$$

and

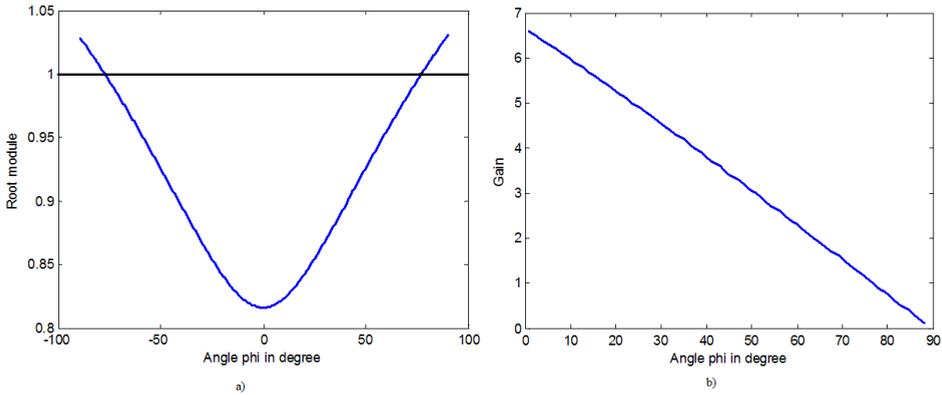


Fig. 6. a) Module of the two complex conjugated roots of the characteristic equation of I controller for $K'_{I_{1,2}} = 1$, in relation with ψ . b) Module of the angle limit for the stability of I controller in dependence of $K_{I_{1,2}}$

$$\Phi = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.05hC\psi K'_{I_1} & -0.05hS\psi K'_{I_2} & 1 & 0 \\ 0.05hS\psi K'_{I_1} & -0.05hC\psi K'_{I_2} & 0 & 1 \end{bmatrix}$$

The system’s stability is determined by the roots of the characteristic polynomial of matrix Φ . If the module of any root is bigger than 1, out of unit circle, the system is not stable. An analytical solution for the roots of characteristic equation of matrix Φ is very difficult in this case, but a graphic solution to illustrate the tendency of stability in relation with K'_{I_R} and ψ can be found. With $K'_{I_1} = K'_{I_2} = K'_{I_{1,2}} = 1$, in Fig. 6 a) the module of the two complex conjugated roots of the characteristic equation is plotted, in relation to ψ . It is clear that for $-77.4^\circ < \psi < 77.4^\circ$ the system is stable, these angle limits define the control system work space, while in Fig. 6 b) the module of the angle limit for the stability for differences values of $K'_{I_{1,2}}$ is shown.

If we use a PI controller, including a zero in $z = -0.2$ in each axis, Equation (35) is modified as:

$$\frac{0.05K'_{I_R} K'(z + 0.2)}{(z^2 - z)} [\mathbf{U}(z) - \mathbf{Y}(z)] = \mathbf{Y}(z) \tag{39}$$

and Equation (22) becomes:

$$\begin{aligned} \mathbf{y}(k + 2) - (\mathbf{I} - 0.05K'_{I_R} \mathbf{K}')\mathbf{y}(k + 1) = \\ - 0.01K'_{I_R} \mathbf{K}'\mathbf{y}(k) + 0.05K'_{I_R} \mathbf{K}'\mathbf{u}(k + 1) + 0.03K'_{I_R} \mathbf{K}'\mathbf{u}(k) \end{aligned} \tag{40}$$

The representation of Equation (40) in the state space, with $u=0$, is:

$$\begin{bmatrix} \mathbf{y}(k + 1) \\ \mathbf{y}(k + 2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.03K'_{I_R} \mathbf{K}' & \mathbf{I} - 0.15K'_{I_R} \mathbf{K}' \end{bmatrix} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k + 1) \end{bmatrix}$$

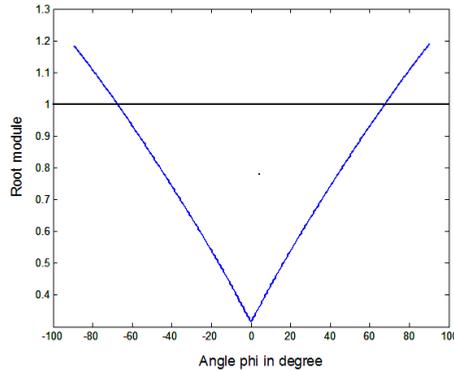


Fig. 7. Module of the two complex conjugated roots of the characteristic equation of *PI* controller for $K'_{I,2} = 10$, in relation with ψ

and

$$\Phi = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.01h' C \psi K'_{I_1} & -0.01h' S \psi K'_{I_2} & 1 - 0.05h' C \psi K'_{I_1} & 0.05h' S \psi K'_{I_2} \\ 0.01h' S \psi K'_{I_1} & -0.01h' C \psi K'_{I_2} & -0.05h' S \psi K'_{I_1} & 1 - 0.05h' C \psi K'_{I_2} \end{bmatrix}$$

With the inclusion of *PI* controller in each axis the stability system condition of the system is increased, as is shown in Fig. 7, where $K'_{I,2} = 10$, 10 times the gain of Fig. 6 the stability performance is similar.

6. Experimental Study

A robot manipulator ASEA IRB6 with open computer control architecture designed and built at the Departamento de Automática de la Universidad Central de Las Villas, Santa Clara, Cuba, is used as a study case. The control scheme described in Section 4, 2D vision-based control, has been implemented in the links two and three of the robot and the control scheme described in Section 5, 3D vision-based control, has been implemented in the links one, two and three of the robot. The inner loops are implemented in a PC Intel Pentium III 500 MHz connected to the robot through a Humusoft MF624 board which reads the encoder's joint position, executes the control algorithm and gives the control signal to the power unit with a sampling period of 1ms. The video signal is acquired via a frame grabber EZ-Capture with chipset BT878 mounted on a second Intel Celeron at 2.0 GHz computer which processes the images, extracts the object's centre of gravity and radius and solves the inverse kinematic problem. Data are sent back to the main host computer during robot operation through a RS232 serial communication link, the sampling period of the external loop is 50ms.

6.1 Practical Design Consideration

As shown in Fig. 2 the control system has two loops. The external loop calculates the image feature error at every measurement time. This control system allows the possibility of tracking

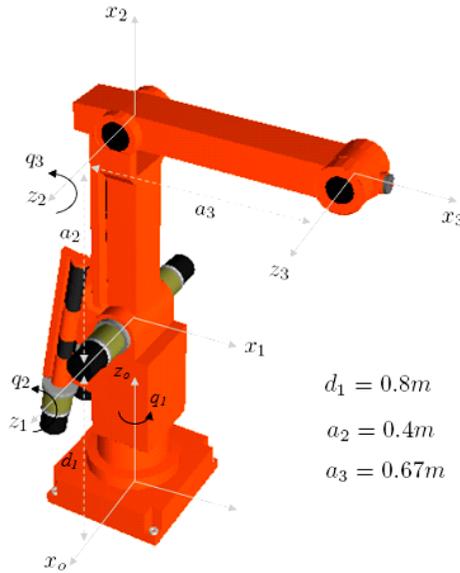


Fig. 8. Geometric description of the robotic system

Link	a_i	α_i	d_i	q_i
1	0	$\pi/2$	d_1	q_1
2	a_2	0	0	q_2
3	a_3	0	0	q_3

Table 1. Denavit-Hartenberg parameters

objects in the environment and hence using more complex vision tasks as e.g. in (Enescu et al.; 2006), for this reason a sampling period of $50ms$ has been taken for the external loop.

Åström (Astrom and Wittenmark; 1990) established that for a correct representation of time response of a continuous system, a reasonable sampling rate is 4 to 10 samples during the rise time. With this consideration $1.5s$ of settling time is adequate for our external loop.

The internal loop does not need the same sampling period, in this case we implemented a P - PI (Sciavicco and Sciliano; 1996) decoupled controller with $\varphi = 0.9$ and $\omega_n = 40$ as design characteristics and $1ms$ as sampling period for each robot joint.

With these conditions for the internal and external loop it is possible to avoid the dynamic effect of the internal loop in relation with the dynamic of the external loop (Lange and Hirzinger; 2003), (Hernández et al.; 2008), or make the approximation that the dynamic of the internal loop is equivalent to a time delay of the external loop, (Corke; 1996).

6.2 Simulation

Following the scheme of Fig. 2 a simulation process has been developed using MATLAB/Simulink. The Denavit-Hartenberg parameters of the robot's geometric configuration (Fig. 8) are specified in Table 1.

According to Fig. 8 the forward and inverse kinematic (1) and (2) are obtained.

For the forward kinematic:

$$p_{xc}^R = C_1(a_2C_2 + a_3C_{23})$$

$$p_{yc}^R = S_1(a_2C_2 + a_3C_{23})$$

$$p_{zc}^R = a_2S_2 + a_3S_{23} + d_1$$

Where S_{23} is $\sin(q_2 + q_3)$ and, C_{23} is $\cos(q_2 + q_3)$.

And for the inverse kinematic:

$$q_1 = \arctan\left(\frac{p_{yc}^R}{p_{xc}^R}\right)$$

$$q_2 = \arctan\left(\frac{p_{zc}^R - d_1}{\pm\sqrt{p_{xc}^R{}^2 + p_{yc}^R{}^2}}\right) + \arctan\left(\frac{a_3\sin(q_3)}{a_2 + a_3\cos(q_3)}\right)$$

$$q_3 = \arctan\left(\frac{\pm\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right)$$

where

$$\cos(q_3) = \frac{p_{xc}^R{}^2 + p_{yc}^R{}^2 + (p_{zc}^R - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3}$$

and: $d_1 = 0.8m$, $a_2 = 0.4m$ and $a_3 = 0.67m$

For our simulation the transfer function of the DC motors of each joint is:

$$G(s)_{motor} = \frac{K_m}{s(T_{ms} + 1)} = \frac{1550}{s(0.024s + 1)}$$

In the simulation, the robot dynamic model is avoided, taking into account the slow robot velocity and that the gear reducer is $\frac{1}{150}$.

The simulation of the 3D vision-based control is presented, in this case we used the vision system model as given by Equation (16) with a sampling period of 50ms.

We make the simulation over small variation about the operating point: $u_d = 0$, $v_d = 0$ and $r_d = 40pixels$; corresponding to object position in reference to the camera: $p_{x_0}^C = 0$, $p_{y_0}^C = 0$ and $p_{z_0}^C = 1.10m$.

In the simulation, after guaranteeing the correct position of the camera, in the second 2, a step displacement of 0.2m in the object position is produced in all the axes, as the following mathematical representation shows.

$$p_{x_0}^R = \begin{cases} 0.67m, & \text{if } t < 2sec \\ 0.87m, & \text{if } t \geq 2sec \end{cases}$$

$$p_{y_0}^R = \begin{cases} 0.00, & \text{if } t < 2sec \\ 0.20m, & \text{if } t \geq 2sec \end{cases}$$

$$p_{z_0}^R = \begin{cases} 2.30m, & \text{if } t < 2sec \\ 2.50m, & \text{if } t \geq 2sec \end{cases}$$

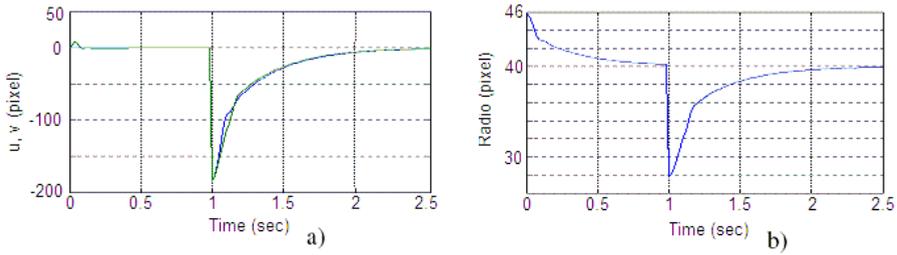


Fig. 9. Simulation of the step disturbance response in image coordinates: a) u (green) and v (blue); and b) radius

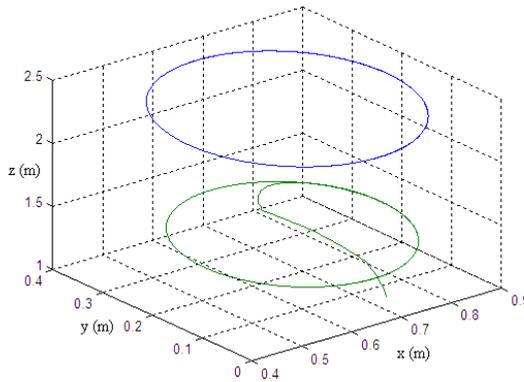


Fig. 10. Simulation of control to follow a desired trajectory

The result of the simulation for the image coordinates $\xi = [u \ v]^T$ is shown in Fig. 9 a) where it is clear that the condition $\tilde{\xi} = [\tilde{u} \ \tilde{v}]^T = [u_d \ v_d]^T - [u \ v]^T = 0$ is achieved near 1.5s after the disturbance and in Fig. 9 b) the same process for the image radius is shown.

As an additional test we simulated the movement of the object in a desired trajectory as an inclined ellipse in the space, as is shown in blue in Fig. 10. In green in the figure is shown the movement of the center of camera coordinate frame. The following is made with a permanent error.

6.3 Experimental Results

The control scheme proposed was implemented on the platform developed for a robot ASEA IRB6. The control algorithm of the inner loop has been implemented using MATLAB/Simulink with the *Real Time Workshop Toolbox* and *Real Time Windows Target*. For the vision loop a monochromatic camera JAI CV-252 has been used. A software component has been developed in Borland Delphi to capture the visual information and the Matrox Image Library is used to process the acquired images in real time. For our experiments, focal length $\lambda = 8mm$ and scaling factor $\alpha = 129 \frac{pixels}{m}$.



Fig. 11. 2D Vision-based control, experimental set-up

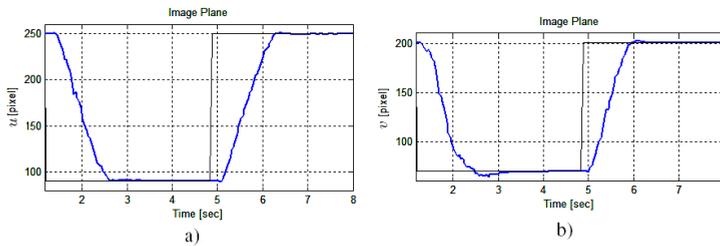


Fig. 12. Time response of image centre of gravity position in axis u a) and axis v b).

6.3.1 2D Vision-based Control.

Fig. 11 shows a view of the experimental set-up for the 2D vision-based control.

For the experiment the wanted image values changes in step for u_d between 250pixels to 90pixels ; and for v_d between 200pixels to 90pixels .

The time responses to the steps are showed in Fig. 12 for the parameters: Fig. 12 a) shows the time response of centre of gravity image u and Fig. 12 b) shows time response of centre of gravity image v . In all the cases the final values are obtained with a good settling time and without steady state error.

Also the control to follow a desired trajectory such as a circle has been implemented. In Fig. 13 the desired trajectory and the actual trajectory are shown, in the image plane. The trajectory control has a permanent error.

6.3.2 3D Vision-based Control.

For the 3D vision-based control, Fig. 14 shows a view of the experimental set-up. For easier physical implementation, the camera axis Z^C is parallel to Z^R axis, according to assumption A2.2, but with different direction. For that reason for the rotation matrix (10), $\phi = 0$ and $\theta = \pi$ and Equation (12) becomes:

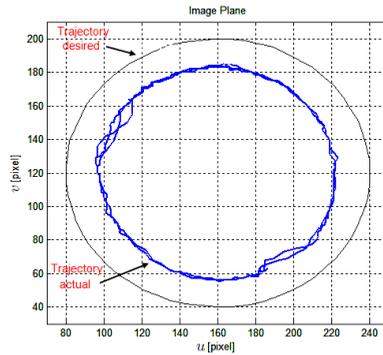


Fig. 13. Trajectory control in the image plane

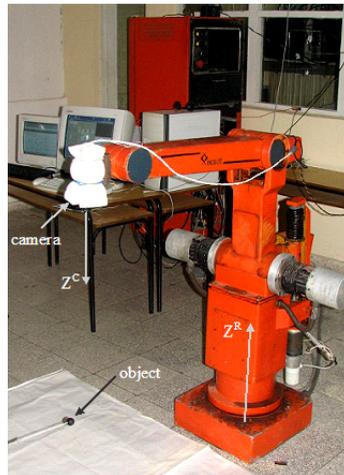


Fig. 14. 3D Vision-based control, experimental set-up

$$\mathbf{R}'_C^R(\psi) = \begin{bmatrix} -C\psi & S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{41}$$

The only consequence of this modification is the change in the theoretical sign in the regulator of axis u and z .

For the experiment the wanted image values are $u_d = 120\text{pixels}$, $v_d = 100\text{pixels}$ and $r_d = 30\text{pixels}$. In $t = 0$ the actual object image features, initial conditions, are $u_0 = 30\text{pixels}$, $v_0 = 240\text{pixels}$ and $r_0 = 13\text{pixels}$. The control system moves the robot camera (Tool Centre Point) as is presented in Fig. 15.

The time responses to the step in the initial conditions are shown in Fig. 16 for the three parameters: Fig. 16 a) and b) show respectively time response of centre of gravity image, u and v ; and c) show the time response of image radius. In all the cases the final values are obtained with a good settling time (around 1.5sec) and without steady state error.

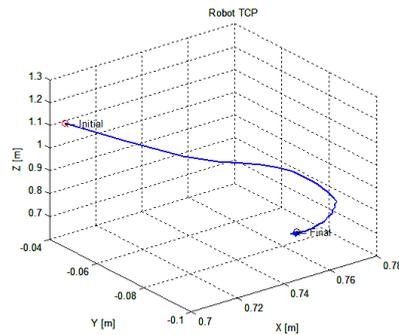


Fig. 15. Movement of the robot Tool Centre Point from the initial position to the wanted space position

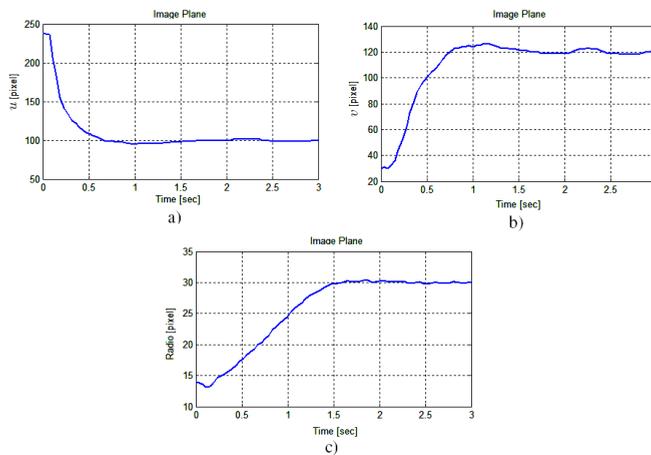


Fig. 16. Time response of image centre of gravity position in axis u a) and axis v b); and time response of image radius r c).

7. Conclusion

In this chapter we presented an image-based visual controller 2D and 3D control of camera-in-hand Robot Manipulators.

The controllers are structured in two loops, the internal loop solves the robot's joint control and in the external loop is implemented a visual controller. The dynamic effect of the internal loop is approximated as an external loop time delay. The robotic and vision systems are modeled for small variation around the operating point for position control. In these conditions the stability of the whole system in discrete time is balanced for I and PI controllers in the external loop in both cases. A particular study is made using an ASEA IRB6 robot manipulator which has its wrist mechanically decoupled. This allows maintaining the orientation of camera axis while the arm is moving. The experimental results presented illustrate the control system's stability and performance.

8. References

- Astrom, K. J. and Wittenmark, B. (1990). *Computer controlled systems: theory and design*, 2nd edn, Englewood Cliffs (NJ).
- Barrientos, A., Peñín, L. F., Balguer, C. and Aracil, R. (1997). *Fundamentos de Robótica*, McGraw Hill.
- Bonfe, M., Minardi, E. and Fantuzzi, C. (2002). Variable structure pid based visual servoing for robotic tracking and manipulation, in IEEE (ed.), *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland.
- Chaumette, F. and Hutchinson, S. (2006). Visual servo control. part i basic approaches, *IEEE Robotics and Automation Magazine*.
- Chaumette, F. and Hutchinson, S. (2007). Visual servo control. part ii advanced approaches, *IEEE Robotics and Automation Magazine*.
- Corke, P. I. (1996). *Visual control of robots : high-performance visual servoing*, Robotics and mechatronics series ; 2, Research Studies Press ; Wiley, Taunton, Somerset, England New York.
- Corke, P. I. (2005). The machine vision toolbox. a matlab toolbox for vision and vision-based control, *IEEE Robotics and Automation Magazine*.
- Enescu, V., De Cubber, G., Cauwerts, K., Sahli, H., Demeester, E., Vanhooydonck, D. and Nuttin, M. (2006). Active stereo vision-based mobile robot navigation for person tracking, *Integrated Computer-Aided Engineering* **13**: 203–222.
- Flandin, G., Chaumette, F. and Marchand, E. (2000). Eye-in-hand / eye-to-hand cooperation for visual servoing, *IEEE International Conference on Robotics and Automation, ICRA2000*, San Francisco.
- Hernández, L., González, R., Sahli, H., González, J. and Guerra, Y. (2008a). Simple solution for visual servoing of camera-in-hand robots in the 3d cartesian space, *10th Intl. Conf. on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, pp. 2020–2025.
- Hernández, L., González, R., Sahli, H., Rubio, E. and Guerra, Y. (2008). A decoupled control for visual servoing of camera-in-hand robot with 2d movement, in IEEE (ed.), *Electronics, Robotics and Automotive Mechanics Conference 2008 (CERMA 2008)*, Cuernavaca, Morelos, Mexico.
- Hutchinson, S., Hager, G. D. and Corke, P. I. (1996). A tutorial on visual servo control, *IEEE Transaction on Robotics and Automation* **12**(5): 651–670.
- Kelly, R., Bugarin, E., Cervantes, I. and Alvarez-Ramirez, J. (2006). Monocular direct visual servoing for regulation of manipulators moving in the 3d cartesian space, in IEEE (ed.), *Decision and Control*, pp. 1782–1787.
- Kelly, R., Carelli, R., Nasisi, O., Kuchen, B. and Reyes, F. (2000). Stable visual servoing of camera-in-hand robotic systems, *IEEE/ASME Transactions on Mechatronics* **5**(1).
- Kelly, R. and Santibáñez, V. (2003). *Control de Movimiento de Robots Manipuladores*, Pearson Education, Madrid.
- Lange, F. and Hirzinger, G. (2003). Predictive visual tracking of lines by industrial robots, *The International Journal of Robotics Research* **22**.
- Sciavicco, L. and Sciciliano, B. (1996). *Modeling and Control of Robot Manipulators*, McGraw Hill Co, New York.
- Sim, T. P., S., H. G. and Lim, K. B. (2002). A pragmatic 3d visual servoing system, *International Conference on Robotics and Automation, IEEE*, Washington, DC.

Xie, H., Sun, L., Rong, W. and Yuan, X. (2005). Visual servoing with modified smith predictor for micromanipulation tasks, in IEEE (ed.), *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Canada.

Using Object's Contour and Form to Embed Recognition Capability into Industrial Robots

I. Lopez-Juarez, M. Peña-Cabrera* and A.V. Reyes-Acosta

Centro de Investigacion y de Estudios Avanzados del IPN

**Universidad Nacional Autonoma de Mexico*

Mexico

1. Introduction

A great effort has been made towards the integration of object recognition capability in robotics especially in humanoids, mobile robots and advanced industrial manipulators. Industrial robots today are not equipped with this capability in its standard version, but as an option. Robot vision systems can differentiate parts by pattern matching irrespective of part orientation and location and even some manufacturers offer 3D guidance using robust vision and laser systems so that a 3D programmed point can be repeated even if the part is moved varying its rotation and orientation within the working space. Despite these developments, current industrial robots are still unable to recognise objects in a robust manner; that is, to distinguish among equally shaped objects unless and alternative method is used, for instance taking into account not only the object's contour but also its form, which is precisely the major contribution of this chapter.

How objects are recognized by humans is still an open research field. There are researchers that favour the theory of object recognition via object-models like Geons (Biederman, 1987), but other researchers agree on two types of image-based models: viewpoint dependent or viewpoint invariant. But, in general there is an agreement that humans recognise objects as established by the similarity principle –among others- of the Gestalt theory of visual perception, which states that things which share visual characteristics such as shape, size, colour, texture, value or orientation will be seen as belonging together. This principle applies to human operators; for instance, when an operator is given the task to pick up a specific object from a set of similar objects; the first approaching action will probably be guided solely by visual information clues such as shape similarity. But, if further information is given (i.e. type of surface), then a finer clustering is accomplished to identify the target object.

We believe that it is possible to integrate a robust invariant object recognition capability in industrial robots following the above assumptions by using image features from the object's contour (boundary object information) and its form (i.e. type of curvature or topographical surface information). Both features can be concatenated in order to form an invariant vector descriptor which is the input to an Artificial Neural Network (ANN) for learning and recognition purposes. In previous work, it was demonstrated the feasibility of the approach

to learn and recognise multiple 3D working pieces using its contour from 2D images using a vector descriptor called the Boundary Object Function (BOF) (Peña-Cabrera, et al., 2005). The BOF resulted invariant with different geometrical pieces, but did not considered surface topographical information. In order to overcome this condition and to have a more robust descriptor, a methodology that includes a shape index using the Shape From Shading (SFS) method (Horn, 1970) is presented. The main idea of our approach is to concatenate both vectors, (BOF+SFS) so that not only the contour but also the object's curvature information (form) is taken into account by the ANN.

The organisation of the chapter is as follows. In section 2, the related work is reviewing from the perspective of 2D-2.5D object recognition. In section 3, the original contribution is explained. Section 4, describes inspiring ideas that motivated the use of the FuzzyARTMAP ANN and a qualitative description of the network. Section 5 presents the algorithm for determining the object's contour of an object using the BOF algorithm while section 6 presents formally the SFS algorithm. Section 7 describes the robotic test bed as well as the workpieces that were used during experiments. Experimental results are provided in section 8 and finally, in section 9 conclusions and further work are given.

2. Related Work

Vision recognition systems must be capable of perceiving and detecting images and objects, as close as the human vision does; this fact has encouraged research activity to design artificial vision systems based on the neural morphology of the biological human vision system. Now scientists understand better about how computational neural structures and artificial vision systems must be designed following neural paradigms, mathematical models and computational architectures. When a system involves these aspects, it can be referred to as a "Neuro-Vision System" (Gupta & Knopf, 1993), which can be defined as an artificial machine with ability to see our environment and provide visual formatted information for real time applications. It has been shown by psychological and clinical studies that visual object recognition involves a large activity area on the cerebral cortex when objects are seen the first time and the region's activity is reduced when familiar objects are perceived (Gupta & Knopf, 1993). New objects can also be learned quickly if certain clues are given to the learner. Following this psychological evidence a novel architecture was designed that included information from its shape as well as its form.

Some authors have contributed with techniques for invariant pattern classification using classical methods as invariant moments (Hu, 1962), or artificial intelligence techniques, as used by (Cem Yüceer & Kemal Oflazer, 1993), which describes an hybrid pattern classification system based on a pattern pre-processor and an ANN invariant to rotation, scaling and translation, (Stavros J. & Paulo Lisboa, 1992), developed a method to reduce and control the number of weights of a third order network using moment classifiers and (Shingchern D. You and G. Ford, 1994), proposed a network for invariant object recognition of objects in binary images.

More recently, Montenegro uses the Hough transform to invariantly recognize rectangular objects (chocolates) including simple defects (Montenegro, 2006). This was achieved by using the polar properties of the Hough transform, which uses the Euclidian distance to classify the descriptive vector. This method showed to be robust with geometric figures, however for complex object it would require more information coming from other

techniques such as histogram information or information coming from images with different illumination sources and levels.

Another example is the use of the Fourier descriptor, which obtains image features through silhouettes from 3D objects (Gonzalez, 2004). Gonzalez's method is based on the extraction of silhouettes from 3D images obtained from laser scan, which increases recognition times.

Worthington studies topographical information from image intensity data in gray scale using the Shape from Shading (SFS) algorithm (Worthington, 2001). This information is used for object recognition. It is considered that the shape index information can be used for object recognition based on the surface curvature. Two attributes were used, one was based in low level using curvature histogram and the other is based on structural arrangement of the shape index maximal patches and its attributes in the associated region.

Lowe defines a descriptor vector named SIFT (Scale Invariant Feature Transform), which is an algorithm that detects distinctive image points and calculates its descriptor based on the histograms of the orientation of key points encountered (Lowe, 2004). The extracted points are invariants to scale, rotation as well as source and illumination level changes. These points are located within a maximum and minimum of a Gaussian difference applied to the space scale. This algorithm is very efficient, but the processing time is relatively high and furthermore the working pieces have to have a rich texture.

3. Original work

Moment invariants are the most popular descriptors for image regions and boundary segments, but computation of moments of a 2D image involves a significant amount of multiplications and additions in a direct method. In many real-time industry applications the speed of computation is very important, the 2D moment computation is intensive and involves parallel processing, which can become the bottleneck of the system when moments are used as major features. In addition to this limitation, observing only the piece's contour is not enough to recognise an object since object with the same contour can still be confused. In order to cope with this limitation a novel method that also includes its form (i.e. type of curvature or topographical surface information) is proposed. Both features (contour and form) are concatenated in order to form a more robust invariant vector descriptor which is the input to an Artificial Neural Network (ANN). The methodology includes a shape index using the Shape From Shading (SFS) method (Horn, 1970). The main idea of our approach is to concatenate both vectors, (BOF+SFS) so that not only the contour but also the object's curvature information (form) is taken into account by the ANN.

4. Inspiring ideas and ART models

Knowledge can be built either empirically or by hand as suggested by Towell and Shavlik (Towell & Shavlik, 1994). Empirical knowledge can be thought of as giving examples on how to react to certain stimuli without any explanation and hand-built knowledge, where the knowledge is acquired by only giving explanations but without examples. It was determined that in robotic systems, a suitable strategy should include a combination of both methods. Furthermore, this idea is supported by psychological evidence that suggests that theory and examples interact closely during human learning (Feldman, 1993).

Learning in natural cognitive systems, including our own, follows a sequential process as it is demonstrated in our daily life. Events are learnt incrementally, for instance, during childhood when we start making new friends, we also learn more faces and this process continues through life. This learning is also stable because the learning of new faces does not disrupt our previous knowledge. These premises are the core for the development of connectionist models of the human brain and are supported by Psychology, Biology and Computer Sciences. Psychological studies suggest the sequential learning of events at different stages or “storage levels” termed as sensory memory (SM), short term memory (STM) and long term memory (LTM). There are different types of ANN, for this research a Fuzzy ARTMAP network is used. This network was chosen because of its incremental knowledge capabilities and stability, but mostly because of the fast recognition and geometrical classification responses.

The adaptive resonance theory (ART) is a well established associative brain and competitive model introduced as a theory of the human cognitive processing developed by Stephen Grossberg at Boston University. Grossberg resumed the situations mentioned above in what he called the Stability- Plasticity Dilemma suggesting that connectionist models should be able to adaptively switch between its plastic and stable modes. That is, a system should exhibit plasticity to accommodate new information regarding unfamiliar events. But also, it should remain in a stable condition if familiar or irrelevant information is being presented. He identified the problem as due to basic properties of associative learning and lateral inhibition. An analysis of this instability, together with data of categorisation, conditioning, and attention led to the introduction of the ART model that stabilises the memory of self-organising feature maps in response to an arbitrary stream of input patterns (Carpenter & Grossberg, 1987). The core principles of this theory and how STM and LTM interact during network processes of activation, associative learning and recall were published in the scientific literature back in the 1960s.

The theory has evolved in a series of real-time architectures for unsupervised learning, the ART-1 algorithm for binary input patterns (Carpenter & Grossberg, 1987). Supervised learning is also possible through ARTMAP (Carpenter & Grossberg, 1991) that uses two ART-1 modules that can be trained to learn the correspondence between input patterns and desired output classes. Different model variations have been developed to date based on the original ART-1 algorithm, ART-2, ART-2a, ART-3, Gaussian ART, EMAP, ViewNET, Fusion ARTMAP, LaminART just to mention but a few.

4.1 FuzzyARTMAP

In the Fuzzy ARTMAP (FAM) network there are two modules ART_a and ART_b and an inter-ART module “Map field” that controls the learning of an associative map from ART_a recognition categories to ART_b categories. This is illustrated in Figure 1.

The Map field module also controls the match tracking of ART_a vigilance parameter. A mismatch between Map field and ART_a category activated by input I_a and ART_b category activated by input I_b increases ART_a vigilance by the minimum amount needed for the system to search for, and if necessary, learn a new ART_a category whose prediction matches the ART_b category. The search initiated by the inter-ART reset can shift attention to a novel cluster of features that can be incorporated through learning into a new ART_a recognition category, which can then be linked to a new ART prediction via associative learning at the Map-field.

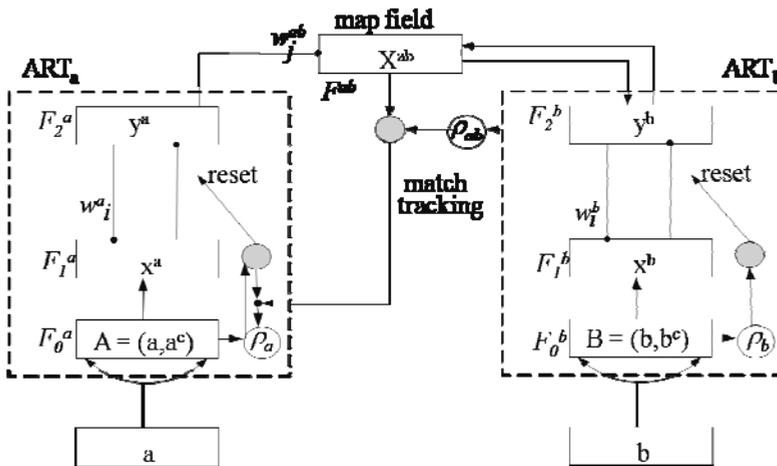


Fig. 1 Architecture FuzzyARTMAP

A vigilance parameter measures the difference allowed between the input data and the stored pattern. Therefore this parameter is determinant to affect the selectivity or granularity of the network prediction. For learning, the FuzzyARTMAP has 4 important factors: Vigilance in the input module (ρ_a), vigilance in the output module (ρ_b), vigilance in the Map field (ρ_{ab}) and learning rate (β).

5. Object's contour

As mentioned earlier, the Boundary Object Function (BOF) method considers only the object's contour to recognize different objects. It is very important to obtain as accurately as possible, metric properties such as area, perimeter, centroid, and distance from the centroid to the points of the contour of the object, to obtain better results and therefore a better analysis of visual data. In this section, a detailed description of the BOF method is presented.

5.1 Metric properties

The metric properties for the algorithm used are based on distance $\delta(P_1, P_2)$ between two points in the plane of the image. For this measure is used the Euclidean distance.

As first step, the object in the image is located, performing a pixel-level scan from left to right and top to bottom, so that if an object is higher than the others, in the image, this will be the first object found. So the first point found inside an object is the highest pixel (first criterion) and more to the left (as the second criterion).

5.1.1 Perimeter

The definition of perimeter is the set of points that make up the shape of the object, in discrete form is the sum of all pixels that lie on the contour, which can be expressed as:

$$P = \sum_i \sum_j pixels(i, j) \in contour \quad (1)$$

The equation (1) shows how to calculate the perimeter, the problem is to know which are the pixels in the images that belong to the perimeter. For searching purposes, the system calculates the perimeter obtaining:

- number of points around a piece
- group of points coordinates X&Y, corresponding to the perimeter of the piece measured clockwise
- boundaries of the piece 2D Bounding Box (2D-BB)

The perimeter calculation for every piece in the ROI is performed after the binarization. Search is always accomplished from left to right and from top to bottom. Once a white pixel is found, all the perimeter is calculated with a search function (figure 2).

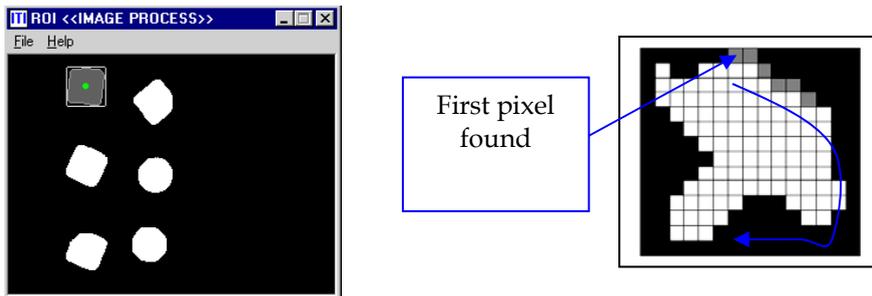


Fig. 2. Perimeter calculation of a workpiece

The next definitions are useful to understand the algorithm:

- A *nearer pixel to the boundary* is any pixel surrounded mostly by black pixels in connectivity eight.
- A *farther pixel to the boundary* is any pixel that is not surrounded by black pixels in connectivity eight.
- The *highest and lowest* coordinates are the ones that create a rectangle (Boundary Box).

The search algorithm executes the following procedures once it has found a white pixel:

1. Searches for the nearer pixel to the boundary that has not been already located.
2. Assigns the label of actual pixel to the nearer pixel to the boundary recently found.
3. Paints the last pixel as a visited pixel.
4. If the new coordinates are higher than the last higher coordinates, it is assigned the new values to the higher coordinates.
5. If the new coordinates are lower than the last lower coordinates, it is assigned the new values to the lower coordinates.
6. Steps 1 to 5 are repeated until the procedure begins to the initial point, or no other nearer pixel to the boundary is found.

This technique will surround any irregular shape, and will not process useless pixels of the image, therefore this is a fast algorithm that can perform on-line classification, and can be classified as linear:

$$O(N * 8*4)$$

where N is the size of the perimeter, and 8 & 4 are the number of comparisons the algorithm needs to find the pixel farther to the boundary, the main difference with the traditional algorithm consist of making the sweep in an uncertain area which is always larger than the figure, this turns the algorithm into:

$$O(N*M)$$

$N*M$, is the size of the Boundary Box in use, and it does not obtain the coordinates of the perimeter in the desired order.

5.1.2 Area

The area of the objects is defined as the space between certain limits, in other words, the sum of all pixels that make up the object, which you can be defined by:

$$A = \sum_i \sum_j pixels(i, j) \in form \quad (2)$$

5.1.3 Centroid

The centre of mass of an arbitrary shape is a pair of coordinates (X_c, Y_c) in which all its mass is considered concentrated and also on which the resultant of all forces is acting. In other words is the point where a single support can balance the object. Mathematically, for the discrete domain of any form they are defined as:

$$X_c = \frac{1}{A} \sum_i \sum_j i \quad Y_c = \frac{1}{A} \sum_i \sum_j j \quad (3)$$

5.1.4 Distance from centroid to the contour

This phase provides valuable information for the invariant recognition of objects by the BOF, finding the distance from the centroid to the perimeter or boundary pixels. If assuming that $P_1(X_1, Y_1)$ are the coordinates of the centroid (X_c, Y_c) and $P_2(X_2, Y_2)$ a point on the perimeter, then, that distance is determined by the following equation:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

5.2 Generation of descriptive vector

The part of the descriptor vector in 2D contains 180 elements, which are obtained every two degrees around the object and is normalized by dividing all vectors by the maximum value found in the same vector, so as shown in Figure 3, where the beginning or first value of the descriptor vector is at the top of the piece however it can start at any point for the case of a circle.

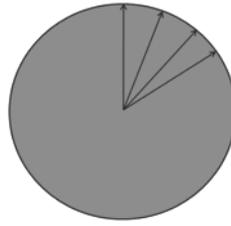


Fig. 3. Obtaining BOF for a circle

In more complicated figures, the starting point is crucial, so the following rules apply: the first step is to find the longest line passing through the centre of the piece, as shown in Figure 4(a), where there are several lines. The longest line is taken and divided by two, taking the centre of the object as reference. Thus, the longest middle part of the line is as shown in Figure 4(b) and this is taken as starting point for the vector descriptor.

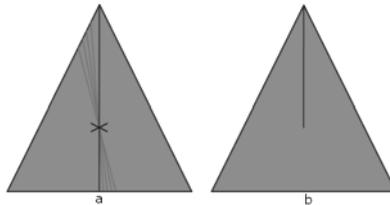


Fig. 4. Example of begin of BOF vector.

Figure 5 shows an ideal example of the BOF vector descriptor of a triangle.

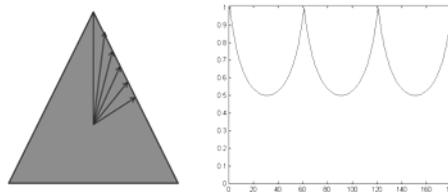


Fig. 5. Ideal example of the BOF vector descriptor

6. Object's form

The Shape From Shading (SFS) consists primarily of obtaining the orientation of the surface due to local variations in brightness that is reflected by the object, in other words, the intensities of the greyscale image is taken as a topographic surface.

SFS is the process of obtaining three-dimensional surface shape from reflection of light from a greyscale image. In the 70's, Horn formulated the problem of Shape From Shading finding the solution of the equation of brightness or reflectance trying to find a single solution (Horn, 1970). Today, the issue of Shape from Shading is known as an ill-posed problem, as mentioned by Brooks, causing ambiguity between what has a concave and convex surface, which is due to changes in lighting parameters (Brooks, 1983).

To solve the SFS problem, it is important to study how the image is formed, as mentioned by Zhang (Zhang, et al., 1999). A simple model of the formation of an image is Lambertian model, where the gray value in the pixels of the image depends on the direction of light and surface normal. So if we assume a Lambertian reflection, we know the direction of light and brightness can be described as a function of the object surface and the direction of light, the problem becomes a little simpler.

The algorithm consist to find the gradient of the surface and determine the normals, since they are perpendicular to the normals and appear in the reflectance cone whose centre is given by the direction of light, to calculate the normal of the entire surface of the object to be recognised, then smoothing is performed so that the normal direction of the local regions are not very uneven. When smoothing is performed, some lie outside of the normal cone reflectance, then it is necessary to make them to rotate and to re-enter these normals within the cone, smoothing and rotation using iterations. Finally getting the kind of local curvature of the surface generates a histogram.

The greyscale image is taken as a topographic surface and is known the vector of reflected light, so, the reflectance equation is calculated, as shown below:

$$E(i, j) = n_{i,j}^k \cdot s \quad (5)$$

Where: s is a unit vector of the direction of light, E is the reflectance of the light in (i, j) , and $n_{i,j}^k \cdot s$ are the estimated normals in the k^{th} iteration

The reflectance equation of the image is defined by a cone of possible directions normal to the surface as shown in Figure 6 where the reflectance cone has an angle of $\cos^{-1}(E(i, j))$

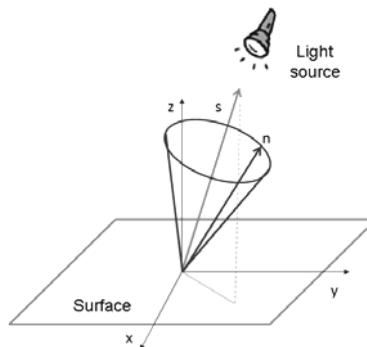


Fig. 6. Possible normal directions to the surface over the reflectance cone

If the normals satisfy recovered reflectance equation of the image, then the normals must fall on their respective reflectance cones.

6.1 Image's Gradient

The first step is to calculate the surface normals which are calculated using the gradient of the image (I), as shown in equation (6).

$$\nabla I = [p \quad q]^T = \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right]^T \quad (6)$$

Where $[p \quad q]$ are used to obtain the gradient and they are known as Sobel operators.

6.2 Normals

As the normals are perpendicular to the tangents, the tangents can be found by the cross product, which is parallel to $(-p, -q, 1)^T$. Thus we can write the normal like:

$$n = \frac{1}{\sqrt{p^2 + q^2 + 1}} (-p, -q, 1)^T \quad (7)$$

Assuming that z component of the normal to the surface is positive.

6.3 Smoothness and rotation

The smoothing, in few words can be described as avoiding abrupt changes between normal and adjacent. The Sigmoidal Smoothness Constraint makes the restriction of smoothness or regularization forcing the error of brightness to satisfy the matrix rotation θ , deterring sudden changes in direction of the normal through the surface.

With the normal smoothed, proceed to rotate these so that they are in the reflectance cone as shown in Figure 7.

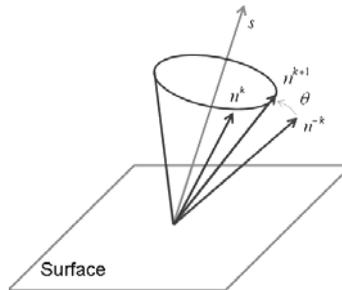


Fig. 7. Rotation of the normals in the reflectance cone

Where $n_{i,j}^k$ are the normals smoothed. $n_{i,j}^{-k}$ are the normals after the smoothness and before the rotation. $n_{i,j}^{k+1}$ are the normals after the rotation of θ grades. With the normals smoothed and rotated with the smoothness constraints, this can result in having several iterations, which is represented by the letter k .

6.4 Shape index

Koenderink (Koenderink, & Van Doorn, 1992) separated the shape index in different regions depending on the type of curvature, which is obtained through the eigenvalues of the Hessian matrix, which will be represented by k_1 and k_2 as showing the equation 7.

$$\phi = \frac{2}{\pi} \arctan \frac{k_2 + k_1}{k_2 - k_1} \quad k_2 \geq k_1 \tag{8}$$

The result of the shape index ϕ has values between [-1, 1] which can be classified, according to Koenderink it depends on its local topography, as shown in Table 1.

Cup	Rut	Saddle rut	Saddle Point	Plane	Saddle Ridge	Ridge	Dome
$\left[-1, -\frac{5}{8}\right)$	$\left[-\frac{5}{8}, -\frac{3}{8}\right)$	$\left[-\frac{3}{8}, -\frac{1}{8}\right)$	$\left[-\frac{1}{8}, -\frac{1}{8}\right)$	---	$\left[\frac{1}{8}, \frac{3}{8}\right)$	$\left[\frac{3}{8}, \frac{5}{8}\right)$	$\left[\frac{5}{8}, 1\right]$

Table 1. Classification of the Shape Index

Figure 8 shows the image of the local form of the surface depending on the value of the Shape Index, and in the Figure 9 an example of the SFS vector is showed.

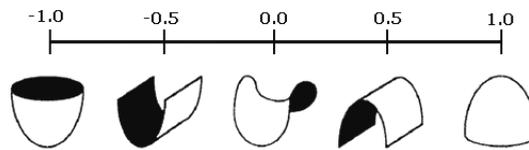


Fig. 8. Representation of local forms of the classification of Shape Index.

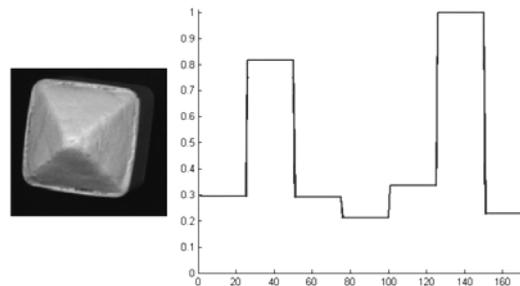


Fig. 9. Example of SFS Vector

7. Robotic Test Bed

The robotic test bed is integrated by a KUKA KR16 industrial robot as it is shown in figure 10. It also comprises a visual servo system with a ceiling mounted Basler A602fc CCD camera (not shown).



Fig. 10. Robotc test bed

The work domain is comprised by the pieces to be recognised and that are also illustrated in figure 10. These workpieces are geometric pieces with different curvature surface. These figures are showed in detail in figure 11.

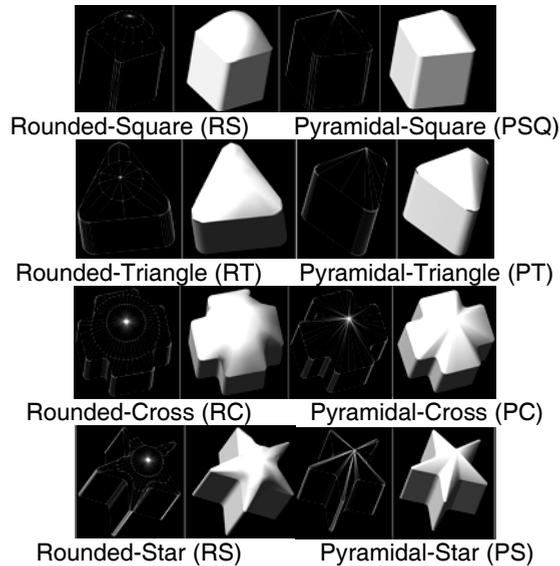


Fig. 11. Objects to be recognised

8. Experimental results

The object recognition experiments by the FuzzyARTMAP (FAM) neural network were carried out using the above working pieces. The network parameters were set for fast learning ($\beta=1$) and high vigilance parameter ($\rho_{ab} = 0.9$). There were carried out three. The first experiment considered only the BOF taking data from the contour of the piece, the second experiment considered information from the SFS algorithm taking into account the

reflectance of the light on the surface and finally, the third experiment was performed using a fusion of both methods (BOF+SFS).

8.1 First Experiment (BOF)

For this experiment, all pieces were placed within the workplace with controlled light illumination at different orientation and this data was taken to train the FAM neural network. Once the neural network was trained with the patterns, then the network was tested placing the different pieces at different orientation and location within the work space.

The figure 12 shows some examples of the object's contour.

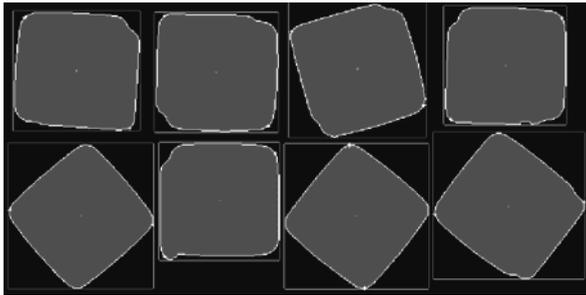


Fig. 12. Different orientation and position of the square object.

The object's were recognised in all cases having only failures between Rounded shaped objects and Square shaped ones. In these cases, there was always confusion due to the fact that the network learned only contours and in both cases having only the difference in the type of surface the contour is very similar.

8.2 Second Experiment (SFS)

For the second experiment and using the reflectance of the light over the surface of the objects (SFS method), the neural network could recognise and differentiate between rounded and pyramidal objects. It was determined during training that for the rounded objects to be recognised, it was just needed one vector from the rounded objects because the change in the surface was smooth. For the pyramidal objects it was required three different patterns during training to recognise the objects, from which it was used one for the square and triangle, one for the cross and other for the star. It was noticed that the reason was that the surface was different enough between the pyramidal objects.

8.3 Third Experiment (BOF+SFS)

For the last experiment, data from the BOF was concatenated with data from the SFS. The data was processed in order to meet the requirement of the network to have inputs within the $[0, 1]$ range. The results showed a 100% recognition rate, placing the objects at different locations and orientations within the viewable workplace area.

To verify the robustness of our method to scaling, the distance between the camera and the pieces was modified. The 100% size was considered the original size and a 10% reduction

for instance, meant that the piece size was reduced by 10% of its original image. Different values with increment of 5 degrees were considered up to an angle $\theta = 30$ degrees (see figure 13 for reference).

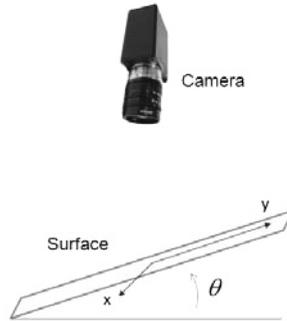


Fig. 13. Plane modifies.

The obtained results with increments of 5 degrees step are shown in Table 2.

Grades	R.S.	P.SQ	R.T.	P.T	R.C.	P.C.	R.S	P.S.
5	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	87	100
15	98	100	96	100	82	100	72	100
20	91	100	<u>81*</u>	97*	58	100	<u>51*</u>	82*
25	53	100	<u>73*</u>	93*	<u>37*</u>	91*	<u>44*</u>	59*
30	43	100	<u>54*</u>	90*	<u>4*</u>	83*	<u>20*</u>	26*

Table 2. Recognition results

The “*numbers*” are errors due to the BOF algorithm, the “*numbers**” are errors due to SFS algorithm, and the “*numbers***” are errors due to both, the BOF and SFS algorithm. The first letter is the capital letter of the curvature of the objects and the second one is the form of the object, for instance, RS (Rounded Square) or PT (Pyramidal Triangle). Figure 14 shows the behaviour of the ANN recognition rate at different angles.

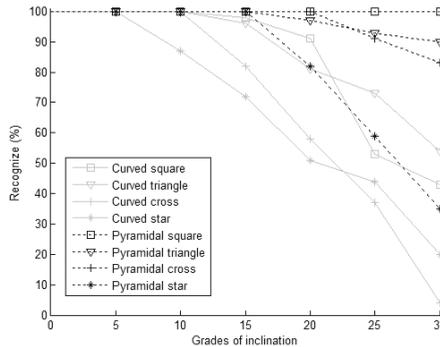


Fig. 14. Recognition graph

The Figure 14 shows that the pyramidal objects have fewer problems to be recognized in comparison with the rounded objects.

9. Conclusions and future work

The research presented in this chapter presents an alternative methodology to integrate a robust invariant object recognition capability into industrial robots using image features from the object's contour (boundary object information) and its form (i.e. type of curvature or topographical surface information). Both features can be concatenated in order to form an invariant vector descriptor which is the input to an Artificial Neural Network (ANN) for learning and recognition purposes.

Experimental results were obtained using two sets of four 3D working pieces of different cross-section: square, triangle, cross and star. One set had its surface curvature rounded and the other had a flat surface curvature so that these object were named of pyramidal type.

Using the BOF information and training the neural network with this vector it was demonstrated that all pieces were recognised irrespective from its location an orientation within the viewable area since the contour was only taken into consideration. With this option it is not possible to differentiate the same type of object with different surface like the rounded and pyramidal shaped objects.

When both information was concatenated (BOF + SFS), the robustness of the vision system improved recognising all the pieces at different location and orientation and even with 5 degrees inclination, in all cases we obtained 100% recognition rate.

Current results were obtained in a light controlled environment; future work is envisaged to look at variable lighting which may impose some consideration for the SFS algorithm. It is also intended to work with on-line retraining so that recognition rates are improved and also to look at the autonomous grasping of the parts by the industrial robot.

10. Acknowledgements

The authors wish to thank The Consejo Nacional de Ciencia y Tecnologia (CONACyT) through Project Research Grant No. 61373, and for sponsoring Mr. Reyes-Acosta during his MSc studies.

11. References

- Biederman I. (1987). Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94, pp. 115-147.
- Peña-Cabrera, M; Lopez-Juarez, I; Rios-Cabrera, R; Corona-Castuera, J (2005). Machine Vision Approach for Robotic Assembly. *Assembly Automation*. Vol. 25 No. 3, August, 2005. pp 204-216.
- Horn, B.K.P. (1970). Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. *PhD thesis, MIT*.
- Brooks, M. (1983). Two results concerning ambiguity in shape from shading. *In AAAI-83*, pp 36-39.

- Zhang, R; Tsai, P; Cryer, J. E.; Shah, M. (1999). Shape from Shading: A Survey. *IEEE Transaction on pattern analysis and machine intelligence*, vol. 21, No. 8, pp 690-706, Agosto 1999.
- Koenderink, J &. Van Doorn, A (1992). Surface shape and curvature scale. *Image and Vision Computing*, Vol. 10, pp. 557-565.
- Gupta, Madan M.; Knopf, G, (1993). Neuro-Vision Systems: a tutorial. A selected reprint Volume *IEEE Neural Networks Council Sponsor*, IEEE Press, New York.
- Worthington, P.L. and Hancock, E.R. (2001) Object recognition using shape-fromshading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (5). pp. 535-542.
- Cem Yüceer adn Kema Oflazer, (1993). A rotation, scaling and translation invariant pattern classification system. *Pattern Recognition*, vol 26, No. 5 pp. 687-710.
- Stavros J. and Paulo Lisboa, (1992). Translton, Rotation , and Scale Invariant Pattern Recognition by High-Order Neural networks and Moment Classifiers., *IEEE Transactions on Neural Networks*, vol 3, No. 2 , March 1992.
- Shingchern D. You , Gary E. Ford, (1994). Network model for invariant object recognition. *Pattern Recognition Letters* 15, 761-767.
- Gonzalez Elizabeth, Feliu Vicente, (2004). Descriptores de Fourier para identificacion y posicionamiento de objetos en entornos 3D. *XXV Jornadas de Automatica*. Ciudad Real. Septiembre 2004
- Worthington, P.L. and Hancock, E.R. (2001) Object recognition using shape-fromshading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (5). pp. 535-542.
- David G. Lowe, (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Computer Science Department*. University of British Columbia. Vancouver, B.C., Canada. January 2004.
- Hu, M.K., (1962). Visual pattern recognition by moment invariants, *IRE Trans Inform Theory*. IT-8, 179-187.
- Cem Yüceer and Kema Oflazer, (1993). A rotation, scaling and translation invariant pattern classification system. *Pattern Recognition*, vol 26, No. 5 pp. 687-710.
- Montenegro Javier, (2006). Hough-transform based algorithm for the automatic invariant recognition of rectangular chocolates. Detection of defective pieces. Universidad Nacional de San Marcos. *Industrial Data*, vol. 9, num 2.
- Geoffrey G. Towell; Jude W. Shavlik, (1994). Knowledge based artificial neural networks. *Artificial Intelligence*. Vol. 70, Issue 1-2, pp. 119-166.
- Robert S. Feldman, (1993). *Understanding Psychology*, 3rd edition. Mc Graw-Hill, Inc.
- Carpenter, G.A. and Grossberg, S., (1987). A massively parallel architecture for a selforganizing. *Neural pattern recognition machine*, *Computer Vision, Graphics, and Image Processing*, 37:54-115.
- Gail A. Carpenter, Stephen Grossberg, John H Reynolds, (1991). ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by Self-Organizing Neural Network. *Neural Networks*. Pp 565-588.

Autonomous 3D Shape Modeling and Grasp Planning for Handling Unknown Objects

Yamazaki Kimitoshi (*1), Masahiro Tomono (*2)
and Takashi Tsubouchi (*3)

**1 The University of Tokyo*

**2 Chiba Institute University*

**3 University of Tsukuba*

1. Introduction

To handle a hand-size object is one of fundamental abilities for a robot which works on home and office environments. Such abilities have capable of doing various tasks by the robot, for instance, carrying an object from one place to another. Conventionally, researches which coped well with such challenging tasks have taken several approaches. The one is that detail object models were defined in advance (Miura et al., 2003) , (Nagatani & Yuta, 1997) and (Okada et al., 2006). 3D geometrical models or photometric models were utilized to recognize target objects by vision sensors, and their robots grasped its target objects based on the handling point given by manual. Other researchers took an approach to give information to their target objects by means of ID tags (Chong & Tanie, 2003) or QR codes (Katsuki et al., 2003). In these challenges, what kind of information of the object should be defined was mainly focused on.

These researches had an essential problem that a new target object cannot be added without a heavy programming or a special tools. Because there are plenty of objects in real world, robots should have abilities to extract the information for picking up the objects autonomously. We are motivated above way of thinking so that this chapter describes different approach from conventional researches. Our approach has two special policies for autonomous working. The one is to create dense 3D shape model from image streams (Yamazaki et. al., 2004). Another is to plan various grasp poses from the dense shape of the target object (Yamazaki et. al., 2006). By combining the two approaches, it is expected that the robot will be capable of handling in daily environment even if it targets an unknown object.

In order to put all the characteristics, following conditions are allowed in our framework:

- The position of a target object is given
- No additional information on the object and environment is given
- No information about the shape of the object is given
- No information how to grasp it is given

According to our framework, robots will be able to add its handling target without giving shape and additional marks by manual, except one constraint that the object has some texture on its surface for object modeling.

The major purpose of this article is to present whole framework of autonomous modeling and grasp planning. Moreover, we try to illustrate our approach by implementing a robot system which can handle small objects in office environment. In experiments, we show that the robot could find various ways of grasp autonomously and could select the best grasping way on the spot. Object models and its grasping ways had enough feasibility to be easily reused after they acquired at once.

2. Issues and approach

2.1 Issues on combination with modeling and grasp planning

Our challenge can roughly be divided two phases, (1)the robot creates an object model autonomously, and (2)the robot detects a grasp pose autonomously. An important thing is that these two processes should be connected by a proper data representation. In order to achieve it, we apply a model representation named "oriented points". An object model is represented as 3D dense points that each point has normal information against object surface. Because this representation is pretty simple, it has an advantage to autonomous modeling.

In addition, the oriented points representation has another advantage can in grasp planning because the normal information enables to plan grasp poses effectively. One of the issues in the planning is to prepare sufficient countermeasures against the shape error of the object model which is obtained from a series of images. We take an approach to search good contacts area which is sufficient to cancel the difference.

The object modeling method is described in section 3, and the grasp planning method is described in section 4.

2.2 Approach

In order to generate whole 3D shape of an object, sensors have to be able to observe the object from various viewpoint. So we take an approach to mount a camera on a robotic arm. That is, multiple viewpoint sensing can be achieved by moving the arm around the object. From the viewpoint of shape reconstruction, there is a worry that a reconstruction process tends to unstable comparing with a stereo camera or a laser range finder. However, a single camera is suitable to mount a robotic arm because of its simple hardware and light weight.

A hand we utilize for object grasping is a parallel jaw gripper. Because one of the purposes of the authors is to develop a mobile robot which can pick up an object in real world, such compact hand has an advantage. In grasp planning, we think grasping stability is more important than dexterous manipulation which takes rigorous contact between fingers and an object into account. So we assume that fingers of the robot equip soft cover which has a role of conforming to irregular surfaces to the object. The important challenge is to find stable grasping pose from a model which includes shape error. Effective grasp searching is also important because the model has relatively large data.

3. Object Modeling

3.1 Approach to modeling

When a robot arranges an object information for grasping it, main information is 3D shape. Conventionally, many researchers focused on grasping strategy to pick up objects, the representation of object model has been assumed to be formed simple predefined shape primitives such as box, cylinder and so on. One of the issues of these approaches is that such model is difficult to acquire by the robot autonomously.

In contrast, we take an approach to reconstruct an object shape on the spot. This means that the robot can grasp any object if an object model is able to be acquired by using sensors mounted on the robot. Our method only needs image streams which are captured by a movable single camera. 3D model is reconstructed based on SFM (structure from motion) which provides an object sparse model from image streams. In addition, by using motion stereo and 3D triangle patch based reconstruction, the sparse shape improved into 3D dense points. Because this representation consists of simple data structure, the model can be autonomously acquired by the robot relatively easily. Moreover, unlike primitive shape approach, it can represent the various shapes of the objects.

One of the issues is that the object model can have shape errors accumulated through the SFM process. In order to reduce the influence to grasp planning, each 3D point on reconstructed dense shape is given a normal vector standing on the object surface. Oriented points is similar to the "needle diagram" proposed by Ikeuchi (Ikeuchi et al., 1986). This representation is used as data registration or detection of object orientation.

Another issue is data redundancy. Because SFM based reconstruction uses multiple images, the reconstructed result can have plenty of points that are too much to plan grasp poses. In order to cope with this redundancy, we apply voxelization and its hierarchy representation to reduce the data. The method described in chapter 5 improves planning time significantly.

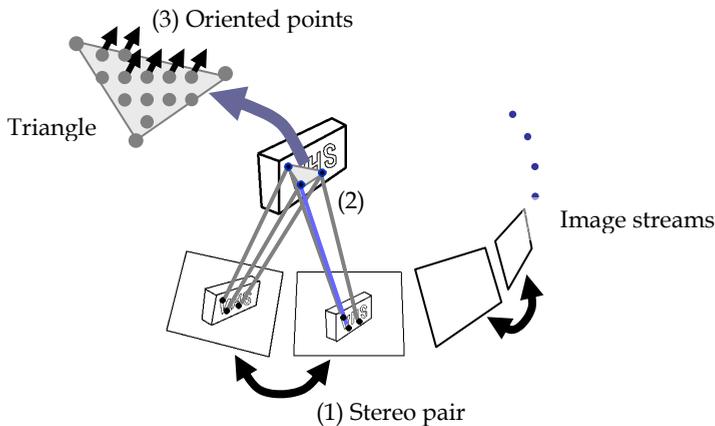


Fig. 1. Surface model reconstruction

3.2 Modeling Outline

Fig.1 shows modeling outline. An object model is acquired according to following procedure: first, image feature points are extracted and tracked from a small area which has

strong intensity by using KLT-tracker (Lucas & Kanade, 2000). From these points, object sparse shape and camera poses are reconstructed by means of SFM (we call this process "sparse model reconstruction" in the rest of this paper). Next, dense shape is acquired from a close pair of images ("dense shape reconstruction" in the rest of this paper). As a result, quite a number of points are reconstructed in online. Details of these two phases are described in next subsection.

3.3 Sparse Shape Reconstruction

In our assumption, because there are almost no given information about an object when the robot tries to grasp it, what the robot has firstly to do is to acquire its shape by using sensors mounted on. We especially focus on SFM by means of a single camera because of its small and light system. This means that the robot can have an ability to acquire whole shape of an object with observing from various viewpoints by moving its manipulator. In this approach, it is hoped that we should also consider a viewpoint planning which decide manipulator motion on the spot, so that sequential reconstruction should be applied.

Factorization method (Tomasi & Kanade, 1992) is a major approach to SFM. 3D shape can be acquired only from image feature correspondences. However, because it is basically batch process, this property prevents our purpose which demands sequential reconstruction. So we apply the factorization only initial process, and use the result as input to sequential reconstruction process. The process consist of motion stereo and bundle adjustment.

Moreover, there are other issues to utilize the result to object grasping, that is, (1) the reconstruction result includes the error of camera model linearization, (2) the scale of reconstructed object is not considered, (3) the shape is basically sparse. We cope with the item (1) by compensating the result of factorization method by means of bundle adjustment. The item (2) will be solved by using odometry or other sensors such as LRF before reconstruction. The item (3) will be solved by an approach described in next subsection.

3.3.1 Initial Reconstruction

In our assumption, the position of a target object is roughly given in advance. What the robot should firstly do is to specify the position of the object. In this process, the robot finds the target object and measures the distance between itself and the object. Next, image streams which observe the object from various viewpoints are captured, and feature points are extracted from the first image and tracked to other images. By using feature correspondences in several images which are captured from the beginning, a matrix \mathbf{W} is generated. A factorization method is suitable in this condition because it is able to calculate camera poses and 3D position of feature points simultaneously. The \mathbf{W} is decomposed as follows :

$$\mathbf{W} = \mathbf{MS}$$

where the matrix \mathbf{M} includes camera poses and the matrix \mathbf{S} is a group of 3D feature points. We use the factorization based on weak perspective camera model (Poalman & Kanade, 1997) whose calculation time is very short but its reconstruction result includes linear approximation error. In order to eliminate the linearization error, bundle adjustment is applied. Basically the adjustment needs the initial state of camera poses and 3D feature points, the result of factorization applies it with good input. After the robot acquired the

distance between itself and a target object, nonlinear minimization is performed obeying the following equation:

$$C = \sum_{i=0}^n \sum_{j=0}^2 \left\{ \left(\frac{\mathbf{r}_x^T \cdot \mathbf{m}_i}{\mathbf{r}_z^T \cdot \mathbf{m}_i} - f_1 \frac{X_i + t_{xj}}{Z_i + t_{zj}} \right)^2 + \left(\frac{\mathbf{r}_y^T \cdot \mathbf{m}_i}{\mathbf{r}_z^T \cdot \mathbf{m}_i} - f_2 \frac{Y_i + t_{yj}}{Z_i + t_{zj}} \right)^2 \right\}$$

where \mathbf{m}_i denotes i th coordinates of a feature point in j th image. P is number of observable feature points. \mathbf{r} is a column vector of a rotation matrix \mathbf{R} , t_x , t_y and t_z are the elements of translation vector from world coordinates to camera coordinates. X , Y and Z indicate 3D position of the feature point.

Through this process, despite the factorization includes linear approximation error, finally obtained result has good values for the next step.

3.3.2 Sequential Reconstruction

The initial reconstruction result provides next process with a part of 3D shape and camera poses, remained object shape is reconstructed sequentially in the next step. One of the issues on this phase is the influence of occlusion, that is, image feature points disappear or arise according to viewpoint changes. In such condition, the reconstruction should be performed whenever a new image is captured.

As often as new image is obtained, following processes are applied:

- A. A camera pose is estimated by means of bundle adjustment by using feature points which are well tracked and their 3-D position has already obtained in the former processes.
- B. 3D position of newly extracted feature points are calculated by means of motion stereo.

Feature point extraction will have frequent changes obeying the viewpoint of the camera. In this situation, motion stereo is effective because it can calculate the 3-D position of a point in each. However this method needs a pair of pre-estimated camera poses, the position of a new camera pose is firstly calculated by means of bundle adjustment. Several feature points whose 3D position is known is utilized to this process. The evaluation equation is as follows:

$$C = \sum_{i=0}^p \left\{ \left(\frac{\mathbf{r}_x^T \cdot \mathbf{m}_i}{\mathbf{r}_z^T \cdot \mathbf{m}_i} - f_1 \frac{X_i + t_x}{Z_i + t_z} \right)^2 + \left(\frac{\mathbf{r}_y^T \cdot \mathbf{m}_i}{\mathbf{r}_z^T \cdot \mathbf{m}_i} - f_2 \frac{Y_i + t_y}{Z_i + t_z} \right)^2 \right\}$$

where \mathbf{m}_i denotes i th coordinates of a feature point, P is number of observable points.

By using this equation, back projection error is evaluated and adjusted by means of Newton method. On the other hand, the equation of motion stereo is as follows:

$$C = \|\mathbf{X} - s_1 \tilde{\mathbf{m}}_1\|^2 + \|\mathbf{X} - s_2 \mathbf{R} \tilde{\mathbf{m}}_2 + \mathbf{T}\|^2$$

where $\tilde{\mathbf{m}}_1$ and $\tilde{\mathbf{m}}_2$ denotes extended vectors about corresponded feature point between two images. $\mathbf{X} = (X, Y, Z)$ indicates 3D position of the feature point, \mathbf{R} and \mathbf{T} denotes relative

rotation matrix and relative translation vector between two images, respectively. From this equation, 3D feature position is calculated by means of least squares.

In this step, each process is fast and reconstruction of the target object can be performed sequentially when an image is captured. This enables a robot to plan next camera viewpoint to acquire better shape model from the reconstructed shape in realtime.

3.3.3 Dense Reconstruction

3D dense shape is approximately calculated by using triangle patches (Fig.1, (2)). By using three vertices which are selected from neighboring features in an image, 3D patches are generated by means of motion stereo. In addition, pixels existing inner the triangle are also reconstructed by means of affine transformation based interpolation.

The reconstruction procedure is as follows: first, three feature correspondences in a pair of images are prepared, and a triangle patch is composed. Next, image pixels are densely sampled on the triangle. At this time, normal information of the patch is also added to each point (Fig.1 (3)). These process is applied to mutiple pairs of images, and all the results of 3D points are integrated as a 3D shape of the target object.

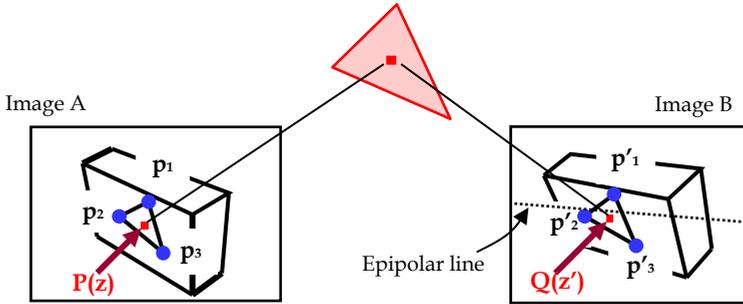


Fig. 2. Feature correspondense by using affine invariance

Fundamentally, dense 3D shape reconstruction is achieved by a correlation base stereo, all the correspondence of pixels in two images must be established and camera poses of them are known. However, making correlation is computational power consuming process and takes long time. So this section describes a smart and faster algorithm for dense 3-D reconstruction, where sparse correspondence of the feature points which is already obtained in the sequential phase is fully utilized. The crucial point of the proposed approach is to make use of affine invariance in finding a presumed pixel Q in Image B in Fig.2 when a pixel P in Image A in Fig.2 is assigned in a triangle that is formed by the neighbor three feature points. The affine invariance parameter α and β is defined as follows:

$$P(\mathbf{z}) = \alpha(\mathbf{p}_2 - \mathbf{p}_1) + \beta(\mathbf{p}_3 - \mathbf{p}_1) + \mathbf{p}_1$$

where \mathbf{z} is a coordinate vector of pixel P, and \mathbf{p}_n ($n = 1, 2, 3$) is a feature point in image A in Fig.2. α and β are invariant parameters which enable to correspond a pixel P in image A with a pixel Q in image B by following equation:

$$Q(\mathbf{z}') = \alpha(\mathbf{p}'_2 - \mathbf{p}'_1) + \beta(\mathbf{p}'_3 - \mathbf{p}'_1) + \mathbf{p}'_1$$

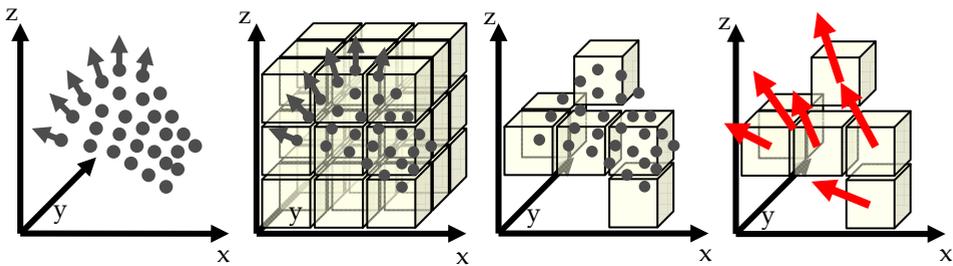
where \mathbf{p}'_n ($n = 1, 2, 3$) in image B in Fig.2 is a corresponding feature point to \mathbf{p}_n ($n = 1, 2, 3$) in image A respectively.

In this approach, we must take notice that the proposed approach employs 2 dimensional affine transformation, and the presumed point Q contains an error in the coordinate vector \mathbf{z}' . Therefore, it is necessary to verify the point \mathbf{z}' with the criteria as follows:

- Distance between presumed pixel \mathbf{z}' of Q to epipolar line in image B in Fig.2 from image A is within a certain threshold.
- A radiance of the pixel Q in image B in Fig.2 is same with the pixel P in image A.

After making the pixel to presumed pixel correspondence in the two images, conventional motion stereo method yields dense 3-D object shape reconstruction. Avoiding conventional correlation matching of the pixels in the two images provides computation time merit in the reconstruction process.

In the next step, 3-D points which are obtained by above stereo reconstruction are voted and integrated into a voxel space. Because the reconstruction method by affine invariance includes 2-D affine approximation, reconstruction error will become larger at a scene which has long depth or a target object which has rough feature points. There will be phantom particles in shape from the reconstruction by two images. Therefore, voting is effective method to scrape redundant or phantom particles off and to extract a real shape. Fig.3 shows the voxelization outline. The generated model (oriented points) becomes a group of voxels with giving normal information in each voxel.



(1) Original oriented points (2) Superimpose voxel space on the points (3) Delete voxels which include few points (4) Replace the points with one voxel in each

Fig. 3. Model voxelization

In addition to above voxelization process to cope with 3-D error originated from Affine transformation, not only the voxel just on the surface of the reconstructed 3-D shape but also the adjacent voxels are also voted into the voxel space. After finishing the vote from all the reconstructed shapes originated from the image stream around the target object, voxels that has the large voted number exceeding the threshold are saved and other voxels are discarded. The result of reconstruction is presented by a group of voxels which has thickness in its shape.

We also propose hierarchy data representation for effective grasp planning. It is described in section 5 in detail.

4. Grasp Planning

The purpose of our grasp planning is to find reasonable grasp pose based on automatically created model.

4.1 Approach to Grasp Planning

Grasp planning in this research has two major issues:

- how to plan a grasp pose efficiently from the 3D dense points,
- how to ensure a grasp stability under the condition that the model may have shape error.

It is assumed that fingers will touch the object by contacting with some area not at a point. Because the object model obtained from a series of images in this paper is not perfectly accurate, the area contact will save the planning algorithm from the difference of the model shape and real shape of the object.

In order to decide the best grasp pose to pick up the object, planned poses are evaluated by three criteria. First criterion is the size of contact area between the hand and the object model, second criterion is a gravity balance depending on grasp position on the object, and third criterion is manipulability when a mobile robot reaches it hand and grasps the object.

4.2 Evaluation method

The input of our grasp planning is 3D object model which is built autonomously. The method should allow the model data redundancy and the shape error. The authors propose to judge grasp stability by the lowest sum total of three functions as follows:

$$F = w_1 F_1(\mathbf{P}_1, \mathbf{x}_h^o) + w_2 F_2(\mathbf{P}_1, \mathbf{x}_h^o) + w_3 F_3(\mathbf{P}_1, \mathbf{x}_h^o, \eta)$$

where \mathbf{P}_1 is a center point of finger plane on the hand. This point is a point to contact with object. \mathbf{x}_h^o is a hand pose (6-DOF), η is a position of a robot. w_i is a weight.

$F_1(\cdot)$ represents the function of contact area between the hand and the object. The evaluation value becomes smaller if the hand pose has more contact area. $F_2(\cdot)$ represents the function of a gravity balance. The evaluation value becomes small if a moment of the object is small. $F_3(\cdot)$ represents the function of the grasping pose. The evaluation value becomes small if the amount of robot motion to reach to the object is small. The policy of grasp planning is to find \mathbf{P}_1 , \mathbf{x}_h^o and η which minimize the function of F .

As it is necessary to yield the moment of inertia of the object, the model must be volumetric. For this purpose, voxelized model are extended to everywhere dense model through following procedure: a voxel space including all the part of the model is defined, then the voxels of outside of the object are pruned away. Finally, the reminder voxels is a volumetric model.

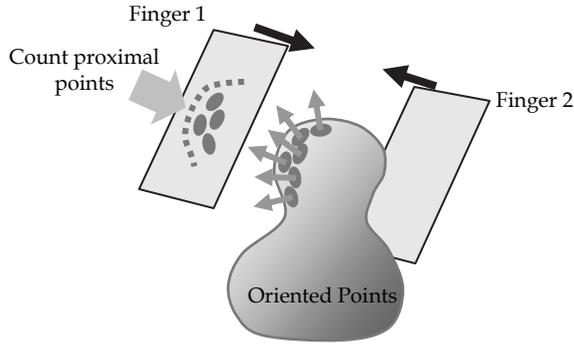


Fig.4 Grasp evaluation based on contact area

4.2.1 Grasp Evaluation based on Contact Area

In order to calculate the function $F_1(\mathbf{P}_1, \mathbf{x}_h^o)$, contact area between the hand and the object is considered. Detail equation can be represented as follows :

$$F_1(\mathbf{P}_1, \eta) = \begin{cases} c & (\text{if } S(\mathbf{P}_1, \eta) \geq S_0) \\ \exp(S_0 / S(\mathbf{P}_1, \eta)) & (\text{if } S(\mathbf{P}_1, \eta) < S_0) \\ \infty & (\text{if } S(\mathbf{P}_1, \eta) = 0) \end{cases}$$

where S_0 is a threshold. $S(\mathbf{P}_1, \mathbf{x}_h^o)$ is the size of contact area. c is a positive constant.

The size of contact area is approximately estimated by counting the voxels in the vicinity of the fingers. The advantage of this approach is that the estimation can merely be accomplished in spite of complexity of the object shape. As shown in Fig.4, the steps to evaluate the contact area are as follows: (i) assume that the hand is maximally opened, (ii) choose one contact point \mathbf{P}_1 which is a voxel on the surface of the model, (iii) consider the condition that the center of the one finger touches at \mathbf{P}_1 and the contact direction is perpendicular to the normal at \mathbf{P}_1 , (iv) calculate contact area as the number of voxels which are adjacent \mathbf{P}_1 with the finger tips. (v) Assume that the other finger is touched with the counter side of the object and count the number of voxels which are touched with the finger plane.

The grasping does not possible if any of following contact conditions applies.

- contact area is too small for either one or both of fingers,
- the width between the finger exceeds the limit,
- the normal with the contacting voxel is not perpendicular to the finger plane.

Change the posture \mathbf{P}_1 by rotating the hand around the normal with certain step angles, above evaluation (i) to (v) is repeated.

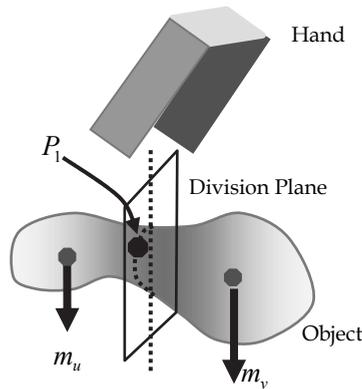


Fig. 5. Grasping evaluation based on gravity balance

4.2.2 Grasp Evaluation of Gravity Balance at Gradient

In order to calculate the function $F_2(P_1, \mathbf{x}_h^o)$, a moment caused by a gravity is considered. The moment is easily calculated by investigating voxels which occupies in the volume of the object model. As shown in Fig.5, the model is divided into two volumes by a plane which is parallel to the direction of gravitation. If the two volumes give equivalent moment, good evaluation is obtained:

$$F_2(P_1, \xi) = \begin{cases} c & (\text{if } M < M_0) \\ \exp(M) & (\text{if } M \geq M_0) \\ \infty & (\text{if } m_u = 0 \text{ or } m_v = 0) \end{cases}$$

where

$$M = K \frac{|m_u - m_v|}{m_u + m_v}$$

The m_u is a moment to u derived from gravitation. K is a positive constant. The equation to calculate M has a role of normalization which prevent a difference of the moment at volume u, v relying on the size of the object.

Although it is naturally strict to consider another balance requirement such as force-closure, the authors rather take $F_2(\cdot)$ for moment balance criterion according to the following reasons. The one reason is that it is difficult to evaluate the amount of the friction force between the hand and grasped object, because there are no knowledge about the material or mass of the object. The second reason is that a grasping pose which is finally fixed on the basis of this evaluation can be expected to maintain the gravity balance of the object. Our approach assumes that the grasping can be successfully achieved unless the grasp position is shifted in very wrong balance, because a jaw gripper hand is assumed to have enough grasping force. This means that the finally obtained grasp pose by the method proposed here roughly maintains force-closure grasp.

4.2.3 Grasp pose evaluation based on robot poses

Although evaluation criteria described above are a closed solution between an object and a hand, other criteria should be considered when we aim to develop an object grasping by a real robot. Even if good evaluation is acquired from the functions $F_1(\cdot)$ and $F_2(\cdot)$, it may be worthless that the robot cannot have grasping pose due to kinematic constraint of its manipulator.

In order to judge the reachability to planned poses, we adopt two-stage evaluation. At first, whether or not inverse kinematics can be solved is tried to a given grasp pose. If manipulator pose exist, the $F_3(\cdot)$ is set to 0. In other case, second phase planning is performed. Robot poses including standing position of the wheelbase are also planned. In this phase grasping pose is decided by generating both wheelbase motion and joint angles of the manipulator (Yamazaki et al, 2008).

4.3 Efficient grasp pose searching

In the pose searching process, oriented point which is touched to P_1 is selected from the model in order. Because such monotonous searching is inefficient, it is important to reduce vain contact between finger and the object model. In order to implement fast planner, oriented points which can have good evaluation are firstly selected. This can be achieved to restrict the direction of the contact by utilizing normal information of each point. In addition, another approach to reduce the searching is also proposed in next section.

5. Model Representation for Efficient Implementation

As described in section 3, the model represented by oriented points has redundant data for grasp planning. By transforming these points to voxelized model, redundant data can be reduced. This section describes some issues on the voxelization and its solution.

5.1 Pruning voxels away to generate thin model

From a viewpoint of ensuring grasping success rate, it is expected that the size of voxel is set 2mm to 5mm because of allowable shape error. One of the issues of voxelization under the setting is that the voting based model tends to grow in thickness on its surface. This phenomenon should be eliminated for effective grasp planning.

An algorithm to acquire a "thin" model is as follows: (1) select a certain voxel from voxelized model, (2) define cylindrical region whose center is the voxel and its direction is parallel to the normal of the voxel. (3) Search 26 neighbor voxels and find voxels which are included the cylindrical region. This process is performed recursively. (4) calculate an average position and normal from the listed voxels, and decide a voxel which can be ascribed to object surface.

Through this thinning, number of reconstructed points reduces from several hundred thousands to several hundreds. Moreover, this averaging has effect of diminishing shape error of the model.

As described in section 4.2, volumetric model is also needed. Such model is generated from the model created through above procedure. Because the process consumes few time, this is one of the advantage of voxelized model.

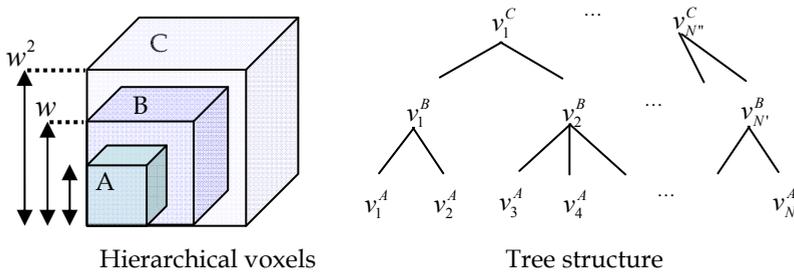


Fig. 6. hierarchical representation

5.2 Hierarchical Data Representation

The method mentioned in 5.1 can reduce the number of pose searching. However, the searching has potential to be still capable of improving. For instance, there are somewhat points which obviously need not to be checked. From this reason, hierarchical data representation is adopted to exclude needless points before judging the quality of grasp pose. Using the new formed model, the searching can be performed at some parts on object model where will have rich contact area with fingers.

The hierarchical representation is similar to octree. Octree is often used for judging collision in the field of computer graphics. The transformation procedure is as follows: at first, initial voxels which construct original voxelized model are set hierarchical A. Next, other voxel space which is constructed w times larger voxels than hierarchical A is superimposed on the voxels of hierarchical A. A new model is represented by the larger voxels which are set hierarchical B. In this processing, only voxels belonging to hierarchical B are adopted when these voxels include much number of voxels which has similar orientation at hierarchical A. The same hierarchy construction is performed from hierarchical B to hierarchical C, too. As a result, one voxel of hierarchical C includes several voxels of hierarchical A. Because these voxels of hierarchical A are grouped and has similar orientation, the area can be expected that it supplies rich contact area with finger.

In the grasp pose searching, voxels of hierarchical C are selected in order. The evaluation is performed about inner voxels which belong to hierarchical A. This approach can achieve efficient searching with selecting only voxels which are guaranteed to provide good evaluation result about contact area.

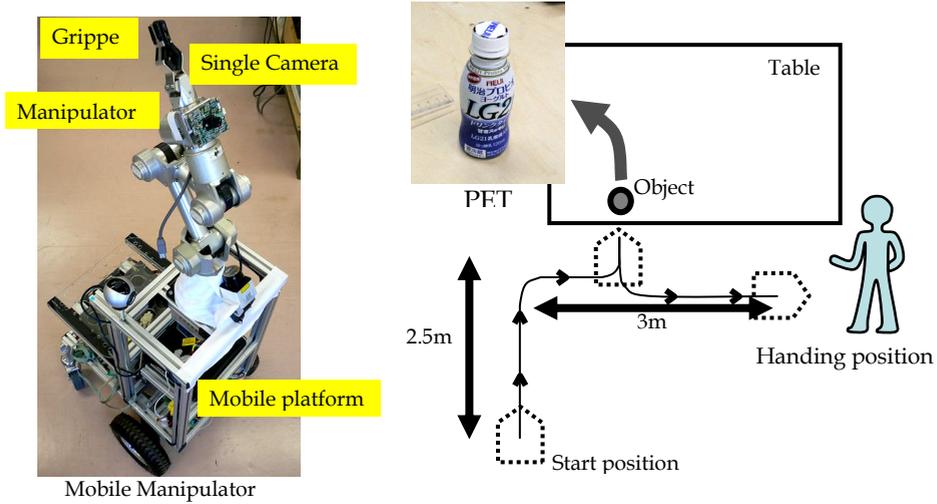


Fig. 7. A robot system and an assumed task

6. Experiments

6.1 System setup

Fig7. shows a robot system in our use. A 5DOF manipulator made by Nuronics Inc. was mounted on the mobile base "Yamabico" which was developed by Intelligent robot laboratory, Univ. of Tsukuba. A camera mounted on the wrist of the manipulator was used to capture image streams with observing a target object while the manipulator moving. A LRF sensor, URG04-LX made by Hokuyo Inc. was mounted on the wheelbase. Two portable computers were also equipped. The One (Celeron 1.1GHz) was to controll the wheelbase and the manipulator from the result of planning. Another (Pentium M 2.0GHz) was to manage reconstruction and planning process.

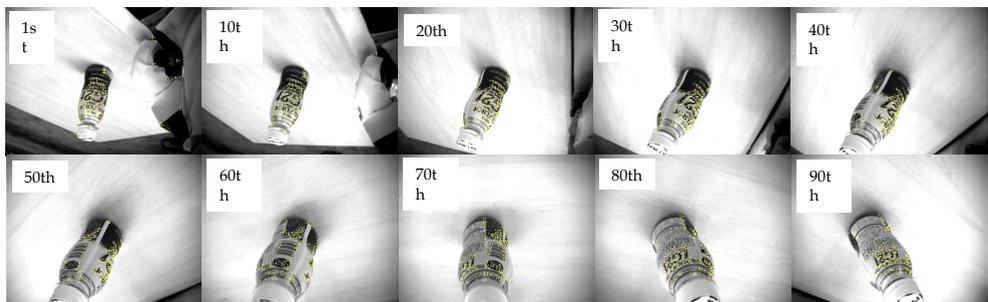


Fig. 8. Image streams in case of a plastic bottle

6.2 Proof experiments of automatic 3D modeling and grasp planning

Firstly, several small objects having commonly texture and shape were selected and they were tried to reconstruct the shape and to plan grasp poses.

The procedure of proof experiments was as follows : (i) the robot moved to the front of instructed position by manual, and (ii) observed to detect a target object position by using LRF. (iii) From the result, camera trajectory (several via points and their interpolation for the end-effector of the manipulator) was calculated to capture image streams related to the target object.

Fig.8 shows an example when a target object was a plastic bottle which had 120mm height. Number of captured images were 134, and 150 feature points were extracted and tracked in each image. These image features were used to reconstruct its 3D shape by means of an algorithm described in section 3.

Factorization method and bundle adjustment took 300 msec for firstly captured 10 images. After that, one time of sequential reconstruction of a camera pose and object shape took 30 msec in each image. Dense shape reconstruction was performed when all images were finished to capture. It used 55 image pairs to make oriented points and the result was integrated into voxel space. The processing time was 2.4 sec.

Types of objects					
Results	Rectangular box	Cup	Toy	Stapler	Ornament
Points	807	986	966	934	1531
Grasp pose candidates	64	37	49	48	58
Processing time (A) [sec]	0.96	1.01	0.94	0.86	1.36
Processing time (B) [sec]	9.3	6.3	6.7	7.2	16.3

Fig. 9. Experimental results of object modeling

Fig.9 shows several results of object modeling and grasp planning. There were five objects which had relatively rich texture on its surface. Numbers in 'Points' row shows number of reconstructed points after voxelization, and numbers in 'Grasp pose candidates' shows number of grasping poses through the algorithm described in section 4. Other 2 rows shows processing times of the planning.

Notice that the planning times were not related to object shape complexity. In these experiments, dozens of grasp poses could be found from each created models about 1 second (Pentium M, 2.0 GHz) as shown in processing times (A). On the other hand, the results in (B) as shown in Fig.9 indicate the processing time without utilizing hierarchy data representation describe in section 5.2. The representation succeeded 7 to 10 times speeding up the planning.

On the other hand, some problems were cleared up through this experiments. For instance, an area where had no texture cannot reconstructed by our modeling method. This means grasp poses which touch to inner of the cup could not be selected in grasp planning.

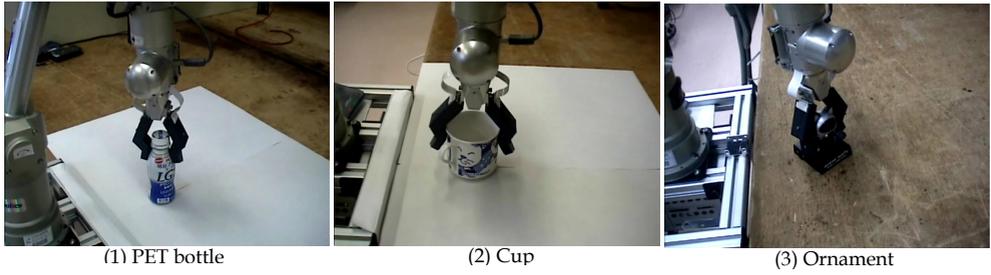


Fig. 10. Examples of object grasping by using planning result

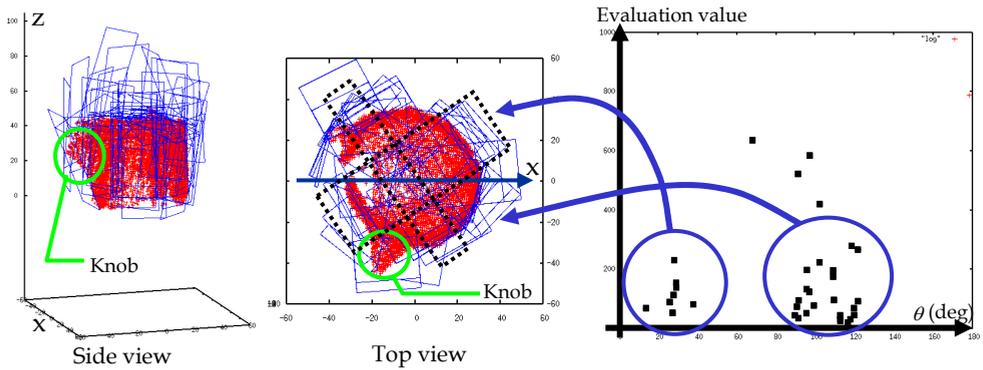


Fig. 11. Planned grasping poses in case of a cup

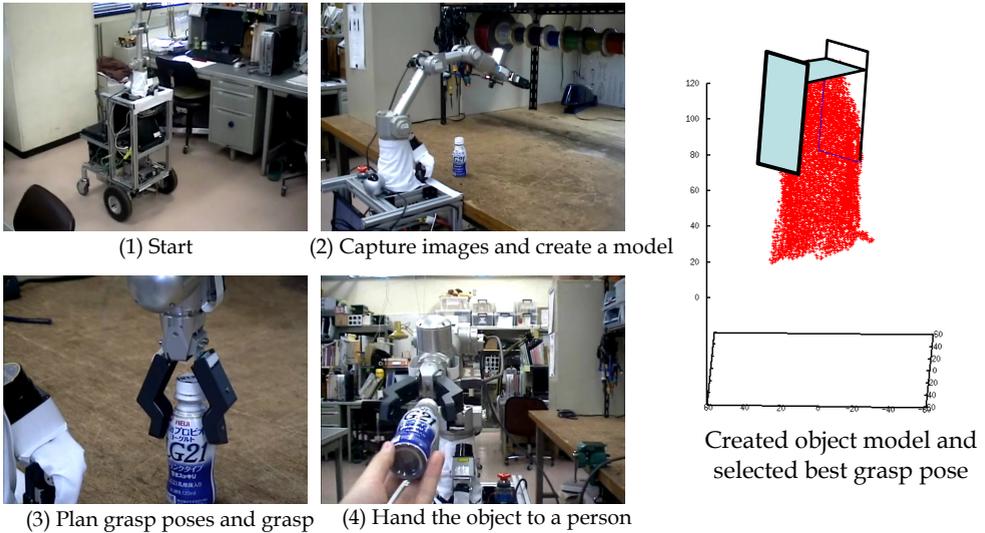


Fig. 12. Implementation to delivery task

Fig.11 shows grasp planning results in case of a cup as shown second example in Fig.9. Red points indicate 3D reconstructed points, and blue rectangles show pseudo finger planes. The most right graph shows a distribution of evaluation of grasp poses. Because low value indicates a good evaluation, grasp poses which touched to the side of the cup without a knob were judged to stable area to contact.

6.4 Integration to delivery task

An object carrying task was tried by using a mobile manipulator. The goal of this experiment was to hand an object to a person who stood another place from the object position. As described in section 2, we assumed that (i)there were no constraint on the object shape and no tags or marks on the object, (ii)relatively much natural texture could be found on the object surface, and (iii)the object has equivalent size that human could grasp it by one hand. Jaw Gripper hand which was a compact and light weight were used for grasping the object. Thin sponges were pasted up to the fingers to ensure area-based contact with the object.

Environment map which included the position of the object was given in advance. Moreover, the initial position of the robot and the position of the person were given in advance, too. In this condition, the robot planned its motion trajectory automatically by using artificial potential method (Connolly et al., 1990). As shown in Fig.12, the robot succeeded to picking up the object based on our automatic 3D modeling and grasp planning, and handed the object to a person who sit down on the side of a table.

7. Conclusion

In this chapter, a 3D modeling and grasp planning methods were discribed. Because the two methods were densely combined with the model representation 'oriented points', autonomous mobile manipulator implemented these methods can handle objects which are placed on real world without giving their shape and grasp information in advance. The authors showed the effectiveness of our approach through experiments by using a real robot.

8. References

- N. Y. Chong & K. Tanie, *Object Directive Manipulation Through RFID*, Proc. of Int. Conf. on Control, Automation, and Systems, pp.22–25, 2003.
- C. Connolly, J. Burns and R. Weiss, *Path Planning using Laplace's Equation*, Proc. of the IEEE Intl. Conf. on Robotics and Automation, pp.2102–2106, 1990.
- R. Katsuki et al., *Design of Artificial Marks to Determine 3D Pose By Monocular Vision*, Proc. 2003 IEEE Int. Conf. Robotics and Automation, pp.995–1000, 2003.
- J. Miura et al., *Development of a Personal Service Robot with User-Friendly Interfaces*, 4th Int. Conf. on Field and Service Robotics, pp.293– 298, 2003.
- K. Nagatani & S. Yuta, *Autonomous Mobile Robot Navigation Including Door Opening Behavior-System Integration of Mobile Manipulator to Adapt Real Environment-*, Intl. Conf. on FSR, pp.208–215, 1997.
- K. Ikeuchi et al., *Determining Grasp Configurations using Photometric Stereo and the PRISM Binocular Stereo System*, The Int. Journal of Robotics Research, Vol. 5, No. 1, pp.46.65, 1986.

- K. Okada et al., *Vision based behavior verification system of humanoid robot for daily environment tasks*, Proc. of 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006), pp 7-12, 2006.
- L. Petersson et al., *Systems Integration for Real-World Manipulation Tasks*, Proc. of IEEE Int. Conf. Robotics and Automation, pp.2500.2505, 2002.
- C. J. Poalman and T. Kanade. *A paraperspective factorization method for shape and motion recovery*, IEEE Trans. Pattern And Machine Intelligence, Vol.17, No.3, pp.206–217, 1997.
- J. Shi et al., *Good Features to Track*, IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp.593–600, 1994.
- C. Tomasi & T. Kanade. *The factorization method for the recovery of shape and motion from image streams*, Proc. Image Understanding Workshop, pp.459–472, 1992
- K. Yamazaki et al., *3D Object Modeling by a Camera mounted on a Mobile Robot*, Proc. of the 2004 IEEE Int. Conf. Robotics and Automation, 2004.
- K.Yamazaki et al., *A Grasp Planning for Picking up an Unknown Object for a Mobile Manipulator*, Proc. of the 2006 IEEE Int. Conf. Robotics and Automation, 2006.
- K.Yamazaki et al., *Motion Planning for a Mobile Manipulator Based on Joint Motions for Error Recovery*, Proc. of the 2006 IEEE Int. Conf. Intelligent Robots and Systems, pp. 7–12, 2006.

Open Software Structure for Controlling Industrial Robot Manipulators

Flavio Roberti, Carlos Soria, Emanuel Slawiński,
Vicente Mut and Ricardo Carelli
*Universidad Nacional de San Juan
Argentina*

1. Introduction

Automatic control has become an important part of the modern industrial processes. Progress both in basic research as applied to automatic control, provide a way to obtain the optimum performance of the dynamical systems, improve the quality and reduce the costs. Robotics, as a part of automatics, represents nowadays an important research area, and it has an essential role in the productive modernization (UNECE and IFR, 2005). The inclusion of industrial manipulators in the manufacturing process allows obtaining better and cheaper products. Therefore, the development of an open software structure for the industrial robots controlling is a very important objective to be achieved (William, 1994), (Frederick and Albus, 1997).

The main characteristic of an open software structure for robotics applications is the interface that relates the components of the robot with the basic internal structure. In industrial area, one of the most important works was developed in the framework of the European project OSACA (Open System Architecture for Control within Automation Systems). Similarly, significant contributions were reached in Japan through OSEC (Open System Environment for Controllers) under IROFA (International Robotics and Factory Automation Center), (Sawada and Akira, 1997), and in the United States of America through OMAC (Open Modular Architecture Control). The objective of all these research projects is to develop an open control system including the reference model of the components, the general application interface and the structure so that all the components work together. So far manufacturers do not work together to develop standard control software that could be applied to any industrial robot.

On the other hand, several commercial software packages, that run under Windows, for mobile robots can be found. Among the best known ones, Advanced Robotics Interface for Applications (ARIA) is used in the robots manufactured by Mobile Robots Inc., BotController software were developed by MobotSoft and it is used for the well known Khepera and Koala robots. Even when these software packages are powerful and have many benefits, they can be applied only to the robots that were developed.

The main objectives of this chapter are the development and the implementation of an open software structure with reusable components, which works as a link between the hardware of an industrial robot manipulator and its control algorithm in order to implement these control algorithms with minimum efforts. Having this kind of software structure is very useful for researching and teaching in robotics as well as for industrial applications. The software structure runs under QNX Real Time operating system (Krtén, 1999), and can be used for a large number of industrial robots.

With the aim of achieving the raised objectives, the developed system is compound by two different programs. First one is the responsible for the sensors' data acquisition and sending the control action to the servos. This program uses a shared memory block to save the data obtained from the sensors and to get the control action to be sent to the servos. In the second one runs the control algorithm. This program, similar to the first one, uses the same shared memory block to get the sensors' data and to save the control action to be sent to the servos. This way, the control algorithm execution is isolated from the signals transmission between the software and the robot's hardware, allowing a time and efforts reduction in the implementation of different control algorithms.

Then, two different controllers have been implemented in order to evaluate the performance of the proposed open software structure, applied to the SCARA robot manipulator Bosch SR-800. First, a classical PD (proportional-derivative) controller is used to allow the robot to achieve a desired position on the workspace. This controller uses the position information from the encoders of the robot. Finally, an advanced passivity based visual servoing with "eye-in-hand" camera configuration (Weiss et al., 1987) is implemented to allow the robot to reach a position relative to some static target. Additionally, finite L_2 -gain for the passivity based control system is proven when a moving object is considered, allowing the robot to track the moving target with L_2 -gain performance. Experimental results for both, the classical PD controller and the passivity based visual controller are presented in order to show the good performance of the proposed open software structure when it is applied to industrial robot manipulators.

This chapter is organized as follows. Section 2 describes the used industrial robot manipulator. Section 3 presents the open control software developed. Section 4 comments the control strategies used to evaluate the software structure and shows the experimental results. Finally, Section 5 presents same conclusions of the work.

2. Industrial robot Bosch SR-800

The robot manipulator Bosch SR-800 is 4 dof SCARA like industrial robotic arm. This kind of manipulator is useful for smooth and fast movements, especially for assembly tasks. First, second and fourth joints are rotational and they move on the horizontal plane; and third joint is linear and it moves on the vertical plane. Figure 1 shows the robot's configuration and its physical dimensions. It is important to remark that the third joint is uncoupled by a mechanical system based on toothed belts. This way, the end effector is always in the same orientation when no control action is applied to the third joint.

The manipulator Bosch SR-800 has a Riho control unit, provided by the manufacturer, consisting of four servo-amplifiers and a CPU. The servo-amplifiers command the joints of the robot and the CPU is used to compute a position control algorithm with internal velocity loop for each joint.

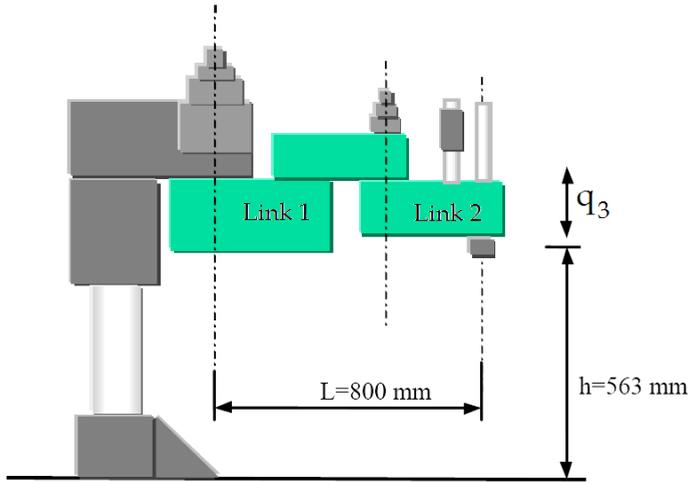


Fig. 1. Industrial robot Bosch SR-800

In order to reach the proposed objectives, the closed control system, i.e. the CPU provided by the manufacturer, was replaced by an open control system, i.e. a PC based control system. This new control system has input-output data boards AD/DA-Q12 from Microaxial®, to make the data interchange between the control system and the robot's hardware. A block diagram of the described system is shown in Fig. 2. The robot was also equipped with a force sensor FS6-120 and a vision camera Sony XC77, both located at the end effector of the robot.

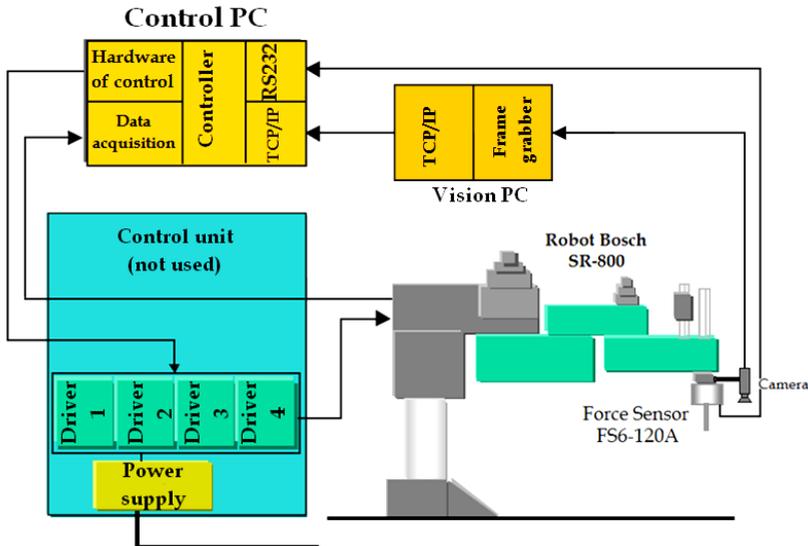


Fig. 2. Control diagram of the robot Bosch SR-800

2.1 Robot kinematic model

Let's consider the industrial manipulator briefly described above, with a global coordinate system whose origin is located at the intersection between the rotation axis of the first joint and the horizontal plane $\langle x, y \rangle$, as Fig. 3 shows.

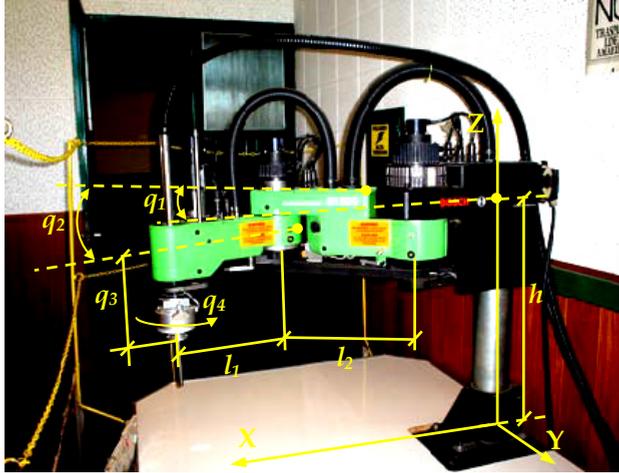


Fig. 3. Geometric description of the robot Bosch SR-800

In Fig. 3, l_1 and l_2 are the length of the first and second links respectively, q_i are the joint positions of each link, and h is the distance between the first link and the base of the robot. Then, the kinematic model that relates the position of the end effector with the joints variables are represented by the following set of equations,

$$\begin{aligned} x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ y &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ z &= h - q_3 \end{aligned} \quad (1)$$

2.2 Dynamic model

In the absence of friction or other disturbances, the dynamics of a n -link rigid SCARA robot manipulator can be written as (Spong and Vidyasagar, 1989),

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} = \boldsymbol{\tau} \quad (2)$$

where:

$\mathbf{q} \in \mathbb{R}^{n \times 1}$ is the vector of joint displacements;

$\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$ is the vector of applied joint torques;

$\mathbf{M} \in \mathbb{R}^{n \times n}$ is the symmetric positive definite manipulator inertia matrix;

$\mathbf{C}\dot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$ is the vector of centripetal and Coriolis torques.

Some important properties of the robot dynamics are the following.

Property 1 – The time derivative of the inertia matrix, and the centripetal and Coriolis matrix satisfy

$$\mathbf{x}^T \left[\frac{d}{dt} \mathbf{M} - 2\mathbf{C} \right] \mathbf{x} = 0 \quad \forall \mathbf{x} \in \mathbb{R}^n$$

that is, $[\dot{\mathbf{M}} - 2\mathbf{C}]$ is an antisymmetric matrix.

Property 2 – The dynamic structure of the manipulator can be written as,

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} = \phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta}$$

where $\phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n \times m}$; and $\boldsymbol{\theta} \in \mathbb{R}^m$ is a vector of parameters.

Property 3 – Matrix \mathbf{M} has the following properties,

- $\mathbf{M} = \mathbf{M}^T > 0$
- $\exists \inf \|\mathbf{M}\|$

For the considered robot manipulator Bosch SR-800,

$$\mathbf{M} = \begin{bmatrix} 1.7277 + 0.1908 \cos(q_2) & 0.0918 + 0.0954 \cos(q_2) \\ 0.0918 + 0.0954 \cos(q_2) & 0.0918 \end{bmatrix};$$

$$\mathbf{C} = \begin{bmatrix} 31.8192 - 0.0954 \sin(q_2)\dot{q}_2 & -0.0954 \sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ 0.3418 \sin(q_2)\dot{q}_1 & 12.5783 \end{bmatrix}$$

2.3 Camera model

A vision camera transforms a 3D space into a 2D projection on the image plane, where the vision sensor is located. This projection causes the loss of the depth perception, which means that each point on the image plane corresponds to a ray in the 3D space.

Several projection models for the representation of the image formation process have been proposed. The most used is the perspective projection model or “pin-hole” model. In this model, a coordinate system $\langle O_C, {}^C X, {}^C Y, {}^C Z \rangle$ attached to the camera is defined in such a way that the X and Y axes define a base for the image plane and the Z axis is parallel to the optic axis. The origin of the framework $\langle O_C, {}^C X, {}^C Y, {}^C Z \rangle$ is located at the focus of the camera lens.

From Fig. 4, a fixed point \mathbf{P} in the 3D space with coordinates $\mathbf{P} = [X_C \ Y_C \ Z_C]^T$ on the framework attached to the perspective camera will be projected on the image plane as a point with coordinates $\boldsymbol{\xi} = [u \ v]^T$ given by (Hutchinson et al., 1996),

$$\boldsymbol{\xi} = -\frac{\lambda}{Z_C} \begin{bmatrix} X_C \\ Y_C \end{bmatrix} \quad (3)$$

where λ is the focal length of the camera expressed in pixels.

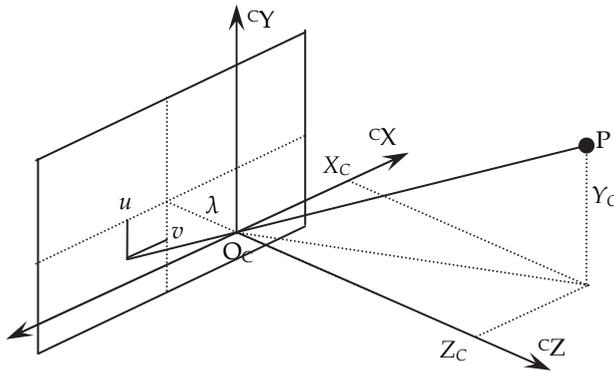


Fig. 4. Perspective projection camera model

2.3.1 Punctual image feature

An image feature is usually defined as a quantifiable relation on the image plane. In (Jang et al., 1991), a formal definition for image features is given,

$$f = \iint \mathfrak{I}(u, v, I(u, v)) du dv \quad (4)$$

where $I(u, v)$ is the intensity of the pixel at the position (u, v) . Function \mathfrak{I} can be a linear or a non linear mapping, depending on the considered image feature. It may even be a delta function.

Some common examples of image features are:

- Cross-correlation correspondence or sum of squares' difference to determine the coordinates of a known pattern of pixels in the image.
- Spatial or central moments of the image.
- Length or orientation of objects' borders.
- Length or orientation of the segments that connect different objects in the scene.

In (Kelly et al., 2000), it is presented the relation between the time variation of the image feature vector and the movement velocity of an object relative to the vision system placed at the end effector of the robot, when a punctual image feature is considered.

$$\dot{\xi} = \mathbf{J}_{img} \begin{bmatrix} {}^c \mathbf{R}_W & 0 \\ 0 & {}^c \mathbf{R}_W \end{bmatrix} \mathbf{J}_G \dot{\mathbf{q}} + \mathbf{J}_O \dot{\mathbf{P}} \quad (5)$$

where ${}^c \mathbf{R}_W$ is the rotation matrix of the coordinate system attached to the robot's base relative to the coordinate system attached to the vision camera; \mathbf{J}_G is the geometric Jacobian of the robot (Sciavicco and Siciliano, 2001); and \mathbf{J}_{img} and \mathbf{J}_O are the image and the object Jacobians respectively, with:

$$\mathbf{J}_{img} = \begin{bmatrix} \frac{\lambda}{Z_c} & 0 & \frac{u}{Z_c} & -\frac{uv}{f} & \frac{\lambda^2 + u^2}{\lambda} & v \\ 0 & \frac{\lambda}{Z_c} & \frac{v}{Z_c} & -\frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} & -u \end{bmatrix}; \mathbf{J}_O = -\frac{\lambda}{Z_c} \begin{bmatrix} 1 & 0 & -\frac{X_c}{Z_c} \\ 0 & 1 & -\frac{Y_c}{Z_c} \end{bmatrix} {}^C \mathbf{R}_W$$

3. Open software design

3.1 Users

The software system developed in this chapter is expected to be useful in control system teaching, human resources training, research, as well as in industrial area. Users in all these areas can be classified in four different levels, depending on how they would use the software system.

- *Level 1:* those users who do not need to make any changes in the software system, for example: undergraduate students.
- *Level 2:* those users who need to evaluate the performance of new control algorithms. They would need to modify just the implemented control law, using the rest of the system without any change. Those users have to have minimum knowledge about data structures and the system operation in order to make appropriate modifications. For example: postgraduate students, researchers.
- *Level 3:* those users who want to make their own control software implementation, using only the sensors' data acquisition program.
- *Level 4:* those users who need to add one or more sensors or actuators in the system. Those users have to be knowledgeable about data structures and the system operation.

3.2 Operational requirements

Based on the main objectives of this development, the operational requirements of the software system are:

- The control software for the industrial robot manipulator Bosch SR-800 must allow implementing and evaluating different control algorithms, using the information from the force sensor, position sensors, and visual sensor. All relevant data of the experiments have to be saved for later analysis.
- The software developed must be flexible and with an open architecture in order to facilitate the incorporation of new components, such as sensors, actuators, teleoperation devices, etc.

3.3 Reuse-based design

In many engineering disciplines, like mechanical or electrical engineering, the design process is based on the reuse of the components. In the last decades, software engineering

has directed its efforts to imitate these techniques by encapsulating software units for its later reuse (Sommerville, 2000). With this aim, object oriented architecture is developed to handle different devices and hardware components, such as sensors, actuators, teleoperation devices. Therefore, data and inner tasks of each device are encapsulated, running in independent threads. This way, the software modules designed for each device can be reused for the inclusion of some new hardware component.

3.4 Operating system and programming environment

All the software development was made under platform QNX (Krtén, 1999). This operating system has been selected because it is one of the best real time operating systems with high stability and robustness of operation. Additionally, QNX supports multi-processors systems and several benefits can be obtained from the memory management unit (MMU) protection. The programming language chosen is C++, and the user interface has been implemented by using the Photon microGui.

Different objects in the software are implemented in classes, which are initialized at the beginning of the program but they do not start working until their activation function is called. In the particular cases of objects related to the sensors and the actuators, each one of them has an associated function that runs in a different thread, with a suitable sample time for each device.

3.5 Design of the software structure

The software structure is designed with independent modules for the user interface, the hardware devices, and the control algorithm. Figure 5 shows a block diagram of the software structure. The different tasks are divided into two processes or programs that communicate each other and work cooperatively. Communication tasks between the software and the hardware devices, and the synchronization of the control sample time are carried out by the so called *Critic Time Program*; whereas the control algorithm runs in the so called *Control Program*.

The function that implements the control algorithm can be easily modified to allow evaluating different control strategies with a minimum effort. This function is called at each control sample instant, which is defined by the user through the user interface.

In the following sections, main characteristics of both the *Critic Time Program* and the *Control Program* of the software are briefly described.

3.5.1 Critic Time Program

The *Critic Time Program* is responsible for communicating with the sensors and the actuators through the data acquisition and control hardware, updating the sensors' data in the shared memory block, and it is also responsible for synchronizing the *Control Program* for the correct running of the control algorithm at each sample instant.

This program has four different classes,

- **Motor:** this object is responsible for applying the control actions obtained by the control algorithm to the motors of the industrial manipulator through the D/A converter of the data acquisition and control hardware.

- Vision system: this object uses the TCP/IP connection functions to receive the visual information from the vision PC. This vision PC process the image obtained through the camera and sends the image features to the Critic Time Program via the TCP/IP connection.
- Position: this object is responsible for obtaining the position data from the internal encoders of each joint of the industrial robot. The data acquisition hardware is used to carry out this task.
- Force: this object is responsible for obtaining the force data from the force sensor FS6-120. The serial port RS232 of the control PC is used.

Additionally, a timer is used in this program to determine the sample instant of the control algorithm; and a graphic user interface is also implemented. Through this interface, users have a set of graphic controls that allow them to select the desired sensors and set their parameters, set the sample period, and start or stop the experiment.

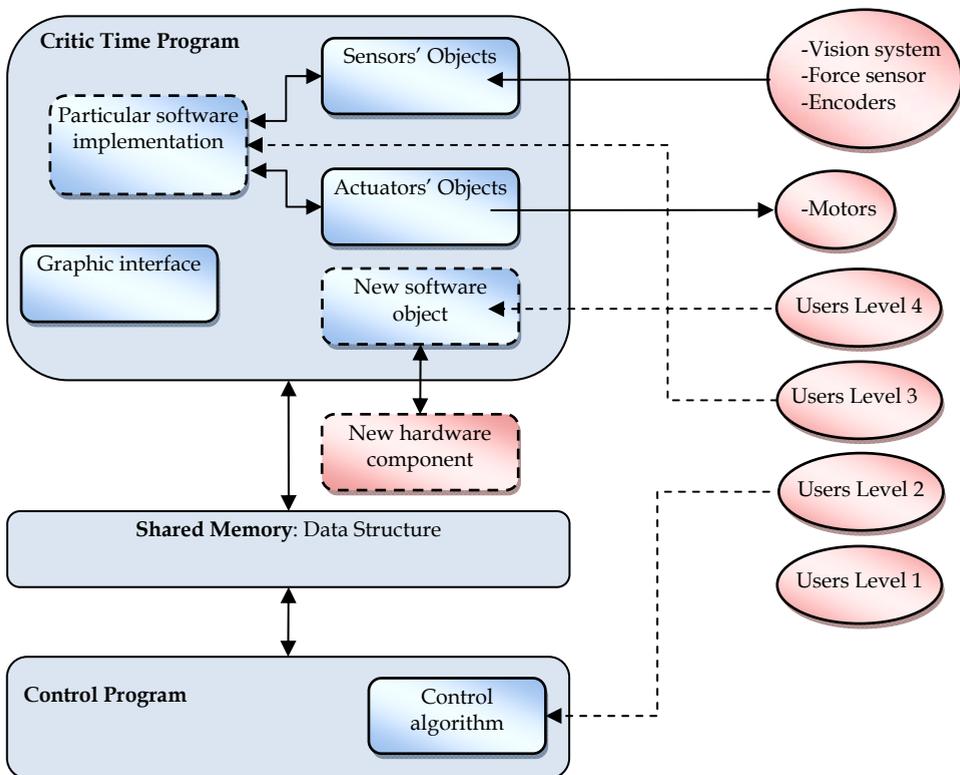


Fig. 5. Block diagram of the developed software system

3.5.2 Control Program

As explained above, the control algorithm runs in this program. Since the *Control Program* may be modified by users of *Level 2*, who may have not a large experience in software

development, some issues are commented. These issues should be taken into account to develop a program that efficiently uses the available hardware resources.

- Determine the correct number of threads of the program, according to available PC hardware.
- Avoid high time demanding operations. Perform I/O operations on files and communication devices asynchronously.
- Do not use global variables. The reuse-based design using object oriented programming is desirable.
- Use shared memory blocks for the data interchange between different processes.
- Use events for the system synchronization.
- If an on-line data writing to a hard disk device is needed, use a double buffer structure and an asynchronous writing.
- Determine and set the correct priority of each thread according to its tasks.

4. Implemented control laws

The open software system developed has been tested by the implementation of two different control strategies. First, a classical PD position controller was implemented, based in the robot position information obtained from the internal encoders of the robot. Then, a passivity based visual controller was implemented. This way, the performance of the software system is evaluated not only when internal sensors are used, but also when a vision camera placed at the end effector of the robot is used as sensor of the control system. In addition, it allows showing the possibility of a fast and easy control law interchange. Throughout this Section, a brief description of the control laws and some experimental results will be presented.

4.1 PD controller

The PD controller is a typical control algorithm used in robotics teaching. With the proposed open software structure, teaching duties relative to the laboratory experimentation can be fast and easy, bringing more time to the theoretical classes. Next, a brief explanation of the PD controller is presented.

The PD position controller is defined as,

$$\boldsymbol{\tau} = \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}} \quad (6)$$

where $\mathbf{K}_p = \text{diag}\{k_{pi}\}$ and $\mathbf{K}_v = \text{diag}\{k_{vi}\}$ are positive definite gain matrices; $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ is the joint position error; and $\dot{\tilde{\mathbf{q}}} = -\dot{\mathbf{q}}$ since a position problem is considered. A block diagram of the control system is shown in Fig. 6.

By equating the control law (6) with the robot's dynamic model (2), the close loop equation is obtained,

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} = \mathbf{K}_p \tilde{\mathbf{q}} - \mathbf{K}_v \dot{\mathbf{q}} \quad (7)$$

Considering the following Lyapunov candidate function and its time derivative (Slotine and Li, 2001; Khalil, 2001),

$$\begin{aligned}
 V &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} + \frac{1}{2} \tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}} \\
 \dot{V} &= -\dot{\mathbf{q}}^T \mathbf{K}_v \dot{\mathbf{q}} \leq 0
 \end{aligned}
 \tag{8}$$

and recalling La Salle theorem (Slotine and Li, 2001; Khalil, 2001), the asymptotic stability of the control system can be proven.

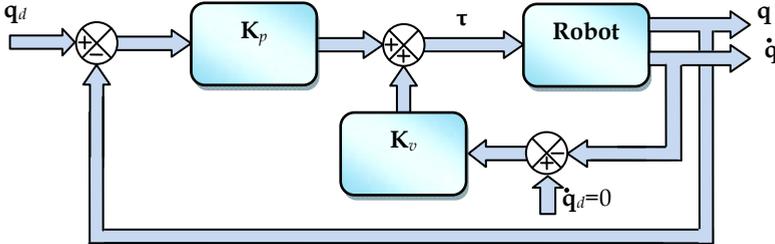


Fig. 6. PD controller block diagram

4.2 Passivity based visual controller

The proposed open software structure can also be used for the experimentation of new advanced control algorithms, such as passivity based visual servoing. This way, researchers can find in the proposed software system a useful experimentation platform, saving time in the implementation, focusing their efforts on the controllers design. Next, a brief explanation of the passivity based visual controller is presented.

Passivity is an important property between input and output of a system that has been widely used in the stability analysis of non-linear systems (Hill and Moylan, 1976; Lin, 1995; Willems, 1972a; Willems, 1972b) and the stability analysis of interconnected systems, especially in cascade structures (Vidyasagar, 1979; Byrnes et al., 1991; Ortega et al., 1995). The concept of passivity shows, in an intuitive way, that a passive system cannot provide more energy than the energy received, and it allows to prove that a non linear passive system can be stabilized by a simple negative output feedback $v = -k y$, with $k > 0$ (see Fig. 7). Therefore, passivity is a useful property for the non linear systems analysis and design, representing a good alternative to the Lyapunov method.

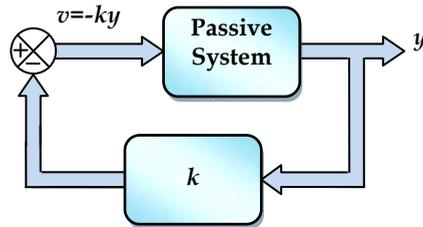


Fig. 7. Stabilized passive system

Some important definitions about the passive systems theory are (Ortega et al., 1998; van der Schaft, 2000),

Definition 1 - The mapping $H : L_{2e} \rightarrow L_{2e}$ is passive if there exist some constant β such that,

$$\langle Hx, x \rangle_T \geq \beta \quad \forall x \in L_2; \quad \forall T \in [0, \infty]$$

Definition 2 - The mapping $H : L_{2e} \rightarrow L_{2e}$ is strictly passive if there exist some constants $\delta > 0$ and β such that,

$$\langle Hx, x \rangle_T \geq \beta + \delta \|x\|_{2,T}^2 \quad \forall x \in L_2; \quad \forall T \in [0, \infty]$$

Definition 3 - The mapping $H : L_{2e} \rightarrow L_{2e}$ is strictly input passive if there exist some constants δ and β such that,

$$\langle Hx, x \rangle_T \geq \beta + \delta \|x\|_{2,T}^2 \quad \forall x \in L_2; \quad \forall T \in [0, \infty]$$

Definition 4 - The mapping $H : L_{2e} \rightarrow L_{2e}$ is strictly output passive if there exist some constants δ and β such that,

$$\langle Hx, x \rangle_T \geq \beta + \delta \|Hx\|_{2,T}^2 \quad \forall x \in L_2; \quad \forall T \in [0, \infty]$$

4.2.1 Passivity property of the vision system

Considering a static object $\dot{\mathbf{P}} = \mathbf{0}$, equation (5) can be written as,

$$\dot{\xi} = \mathbf{J}\dot{\mathbf{q}} \quad (9)$$

where $\mathbf{J} = \mathbf{J}_{img} \begin{bmatrix} {}^C \mathbf{R}_W & 0 \\ 0 & {}^C \mathbf{R}_W \end{bmatrix} \mathbf{J}_G$; being \mathbf{J}_{img} the image Jacobian matrix, and \mathbf{J}_G the robot geometric Jacobian (Kelly et al., 2000).

Taking the energy function $V_\xi = \frac{1}{2} \xi^T \xi$ and making its time derivative (Fujita et al., 2007),

$$\dot{V}_\xi = \xi^T \dot{\xi} = \xi^T \mathbf{J}\dot{\mathbf{q}} \quad (10)$$

and integrating in $[0, T]$

$$\int_0^T \dot{V}_\xi dt = \int_0^T \xi^T \mathbf{J}\dot{\mathbf{q}} dt = \int_0^T \mathbf{v}_\xi^T \dot{\mathbf{q}} dt = V_\xi(T) - V_\xi(0) \geq -V_\xi(0) \quad (11)$$

where $\mathbf{v}_\xi = \mathbf{J}^T \xi$.

Therefore, it can be concluded that the mapping $\mathbf{v}_\xi \rightarrow \dot{\mathbf{q}}$ is passive.

4.2.2 Control system design

Considering now the variable $\tilde{\xi}(t) = \xi(t) - \xi_d$ instead of $\xi(t)$ in order to contemplate the regulation problem, and also considering perfect velocity tracking ($\dot{\mathbf{q}} \equiv \mathbf{u}$), it is possible to prove that the passivity property of the vision system is preserved, that is,

$$\int_0^T \mathbf{u}^T \mathbf{v}_{\tilde{\xi}} dt \geq -\beta \quad \forall T; \text{ then } \mathbf{u} \rightarrow \mathbf{v}_{\tilde{\xi}} \text{ is passive.} \tag{12}$$

where $\mathbf{v}_{\tilde{\xi}} = \mathbf{J}^T \tilde{\xi}$ and $\beta = V_{\tilde{\xi}}(0)$ with $V_{\tilde{\xi}} = \frac{1}{2} \tilde{\xi}^T \tilde{\xi}$.

Then, the following control law is proposed, according to the general structure of Fig. 7,

$$\begin{aligned} \mathbf{u} &= -\mathbf{K} \mathbf{v}_{\tilde{\xi}} \\ \mathbf{u} &= -\mathbf{K} \mathbf{J}^T \tilde{\xi} \end{aligned} \tag{13}$$

where \mathbf{K} is a symmetric and positive definite gain matrix. The control structure is shown in Fig. 8.

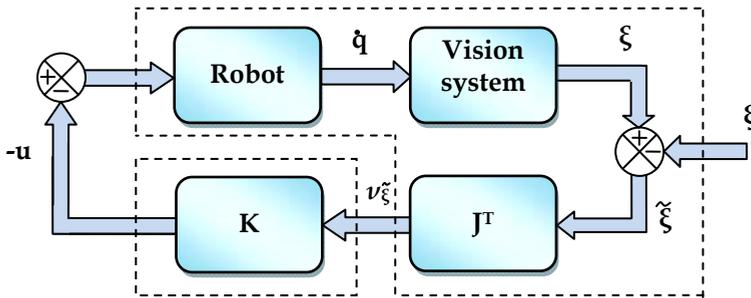


Fig. 8. Block diagram of the passivity based control approach

4.2.2 Control system analysis

From (12), and replacing the control law expression (13),

$$\int_0^T \mathbf{v}_{\tilde{\xi}}^T \mathbf{u} dt = - \int_0^T \mathbf{v}_{\tilde{\xi}}^T \mathbf{K} \mathbf{v}_{\tilde{\xi}} dt \leq -\lambda_{\min}(\mathbf{K}) \int_0^T \|\mathbf{v}_{\tilde{\xi}}\|^2 dt \tag{14}$$

Or,

$$\int_0^T \mathbf{v}_{\tilde{\xi}}^T (-\mathbf{u}) dt \geq \lambda_{\min}(\mathbf{K}) \int_0^T \|\mathbf{v}_{\tilde{\xi}}\|^2 dt \tag{15}$$

where $\lambda_{\min}(\mathbf{K})$ represents the minimum eigenvalue of matrix \mathbf{K} . Therefore, from Definition 3, the controller is strictly input passive from $\mathbf{v}_{\tilde{\xi}} \rightarrow -\mathbf{u}$. This way, the closed loop system of Fig. 8 is built by the interconnections of passive subsystems.

By adding equations (12) and (15), the following inequality can be obtained

$$\begin{aligned} \lambda_{\min}(\mathbf{K}) \int_0^T \|\mathbf{v}_{\tilde{\xi}}\|^2 dt - \beta &\leq 0 \\ \int_0^T \|\mathbf{v}_{\tilde{\xi}}\|^2 dt &\leq \frac{\beta}{\lambda_{\min}(\mathbf{K})} \end{aligned} \tag{16}$$

which implies that $\dot{\mathbf{v}}_{\tilde{\xi}} \in L_2$. Then, for $\dot{\mathbf{v}}_{\tilde{\xi}} \in L_\infty$, the Barbalat's lemma allows concluding that $\mathbf{v}_{\tilde{\xi}}(t) \rightarrow 0$, and therefore $\tilde{\xi} \rightarrow \mathbf{0}$ with $t \rightarrow \infty$, achieving the control objective.

4.2.3 Robustness to the object movement: L_2 -gain performance design

In this section, the possibility of moving objects existence is considered and the control system's performance for tracking tasks is evaluated. With this aim, the object's velocity is considered as an external disturbance of the control system and a robust controller with L_2 -gain performance criteria is designed (Fujita et al., 2007).

The system $\mathbf{w} \rightarrow \tilde{\xi}$ would have finite L_2 -gain if (van der Schaft, 2000),

$$\int_0^T \|\tilde{\xi}\|^2 dt \leq \gamma^2 \int_0^T \|\mathbf{w}\|^2 dt + \delta \quad ; \quad \forall T > 0 \quad (17)$$

being $\mathbf{w} = \mathbf{J}_o \dot{\mathbf{P}}$ the object's velocity on the image plane, considered as an external disturbance; $\gamma > 0$; and $\delta > 0$. In this context, γ represents an indicator of the system's tracking performance. The proposed control system will have finite L_2 -gain if

$$\dot{V}_{\tilde{\xi}} \leq \frac{1}{2} \left(\gamma^2 \|\mathbf{w}\|^2 - \|\tilde{\xi}\|^2 \right) \quad (18)$$

As can be easily verified by integrating (18) in $[0, T]$. In order to find a gain matrix \mathbf{K} that fulfils the L_2 -gain performance criteria (18), it is considered again the positive definite function $V_{\tilde{\xi}}$ and its time derivative,

$$\begin{aligned} V_{\tilde{\xi}} &= \frac{1}{2} \tilde{\xi}^T \tilde{\xi} \\ \dot{V}_{\tilde{\xi}} &= \tilde{\xi}^T \dot{\tilde{\xi}} = \tilde{\xi}^T (\mathbf{J} \dot{\mathbf{q}} + \mathbf{J}_o \dot{\mathbf{P}}) \end{aligned} \quad (19)$$

Considering again perfect velocity tracking ($\dot{\mathbf{q}} \equiv \mathbf{u}$), the control law (13) is introduced in (19),

$$\dot{V}_{\tilde{\xi}} = \tilde{\xi}^T \dot{\tilde{\xi}} = \tilde{\xi}^T \mathbf{J} \mathbf{K} \mathbf{J}^T \tilde{\xi} + \tilde{\xi}^T \mathbf{w} \quad (20)$$

and imposing L_2 -gain performance condition (18) to (20), the following inequality is obtained,

$$\dot{V}_{\tilde{\xi}} = \tilde{\xi}^T \dot{\tilde{\xi}} = \tilde{\xi}^T \mathbf{J} \mathbf{K} \mathbf{J}^T \tilde{\xi} + \tilde{\xi}^T \mathbf{w} \leq \frac{1}{2} \left(\gamma^2 \|\mathbf{w}\|^2 - \|\tilde{\xi}\|^2 \right) \quad (21)$$

Reorganizing (21), the following matrix inequality is obtained,

$$\begin{bmatrix} \tilde{\xi}^T & \mathbf{w}^T \end{bmatrix} \begin{bmatrix} -\mathbf{J} \mathbf{K} \mathbf{J}^T + \frac{\mathbf{I}}{2} & \frac{\mathbf{I}}{2} \\ \frac{\mathbf{I}}{2} & -\frac{\mathbf{I}}{2} \gamma^2 \end{bmatrix} \begin{bmatrix} \tilde{\xi} \\ \mathbf{w} \end{bmatrix} \leq 0 \quad (22)$$

The problem now is to find a symmetric and positive definite matrix \mathbf{K} and a value for γ , such that the matrix inequality (22) is fulfilled. With this aim, the LMI technique (Boyd et al.,

1994) is used by restricting the Jacobian matrix \mathbf{J} to a convex set. The only restriction imposed to the gain matrix \mathbf{K} is that it must be symmetric and positive definite (in order to fulfill the passivity property of the controller (15)), and the condition of being diagonal is not imposed allowing the controller to incorporate dynamics coupling, obtaining better performances.

Now, the problem that immediately rises in the selection of the gain matrix \mathbf{K} is that, if a small value for γ is adopted for a good performance in moving objects tracking, actuators could be saturated in presence of large image features errors. On the other hand, if a large value for γ is adopted, the saturation of the actuators would be avoided to the detriment of the tracking performance. The proposed solution to this problem lies in the use of a variable gain matrix, as a function of the image features error. With this aim, two different gain matrices \mathbf{K}_1 and \mathbf{K}_2 are found by solving the matrix inequality (22) (\mathbf{K}_1 for small features errors and \mathbf{K}_2 for large features errors), and the gain matrix \mathbf{K} is obtained as,

$$\mathbf{K} = (1 - \alpha)\mathbf{K}_1 + \alpha\mathbf{K}_2 \quad (23)$$

where $\alpha = \frac{\|\tilde{\mathbf{s}}\|}{\|\tilde{\mathbf{s}}\|_{\max}}$; and being $\|\tilde{\mathbf{s}}\|_{\max}$ the maximum image features error. This way, matrix

\mathbf{K} always fulfils the performance condition γ , accepting a large value for large image features errors and adopting a smaller value for small image features errors, according to the design specifications.

4.3 Experimental results

Both the PD controller and the passivity based visual controller explained above were implemented in the industrial robot manipulator Bosch SR-800 shown in Fig. 9, with the open software developed. For the first experiment, as well as for the second one, it could be confirmed the fast implementation of the control algorithms.



Fig. 9. Industrial robot manipulator Bosch SR-800, at the National University of San Juan, Argentina

4.3.1 Experimental results for the PD controller

Some experiments are carried out with the classical PD controller, considering only first and second joint of the robot. The controller is implemented in the Control Program and runs with a sample time of 1 msec. In the first experiment, the end effector of the robots must achieve the desired position (50,30)(expressed in centimetres) on the Cartesian space, which means that the desired joint positions are $q_1 = 1.194$ rad and $q_2 = -1.52$ rad . On the other hand, in the second experiment, the end effector of the robots must achieve the desired position (50,-30)(expressed in centimetres) on the Cartesian space, which means that the desired joint positions are $q_1 = 0.113$ rad and $q_2 = -1.52$ rad . In both experiments, the following gain matrices were used,

$$\mathbf{K}_p = \begin{bmatrix} 40.85 & 0 \\ 0 & 46.39 \end{bmatrix}$$

$$\mathbf{K}_v = \begin{bmatrix} 12.74 & 0 \\ 0 & 13.62 \end{bmatrix}$$

Figures 10, 11 and 12 show the results for the first experiment. Figures 10 and 11 show the time evolution of the joint positions; and Fig. 12 shows the trajectory described by the end effector on the Cartesian space. Figures 13, 14 and 15 show the results for the second experiment. Figures 13 and 14 show the time evolution of the joint positions; and Fig. 15 shows the trajectory described by the end effector on the Cartesian space.

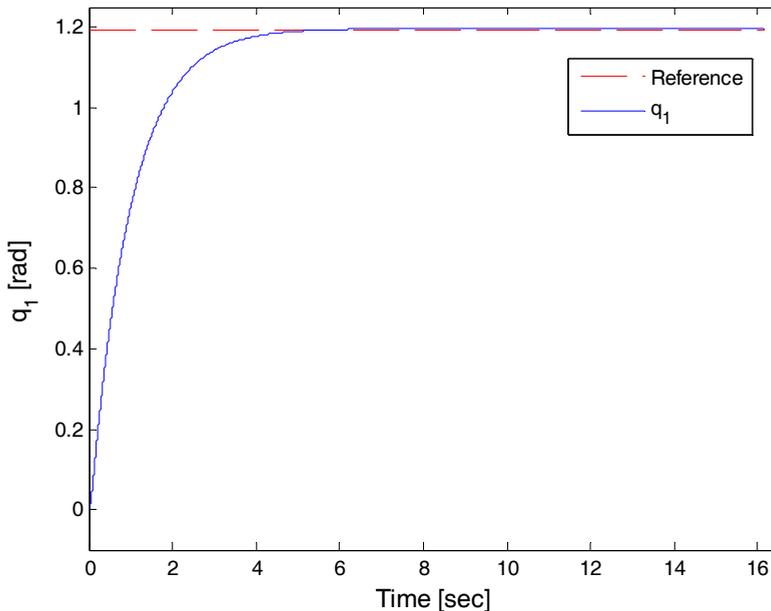


Fig. 10. Time evolution of q_1

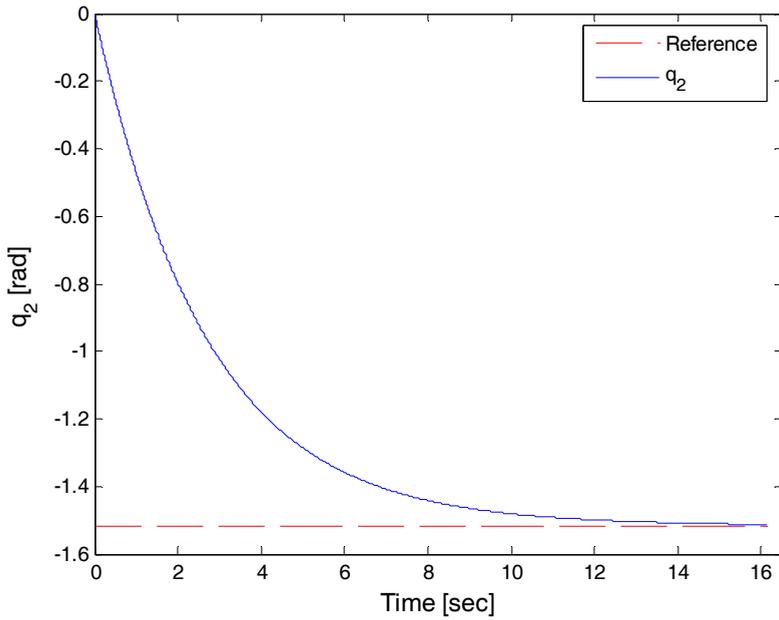


Fig. 11. Time evolution of q_2

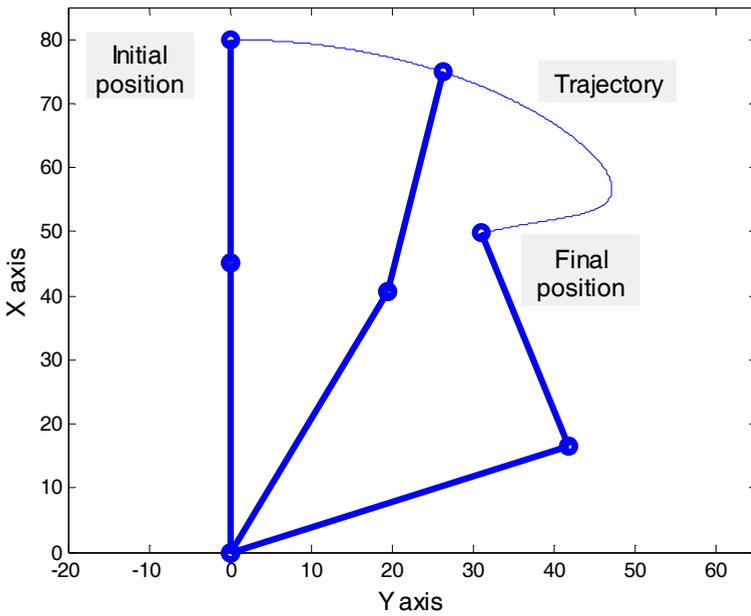
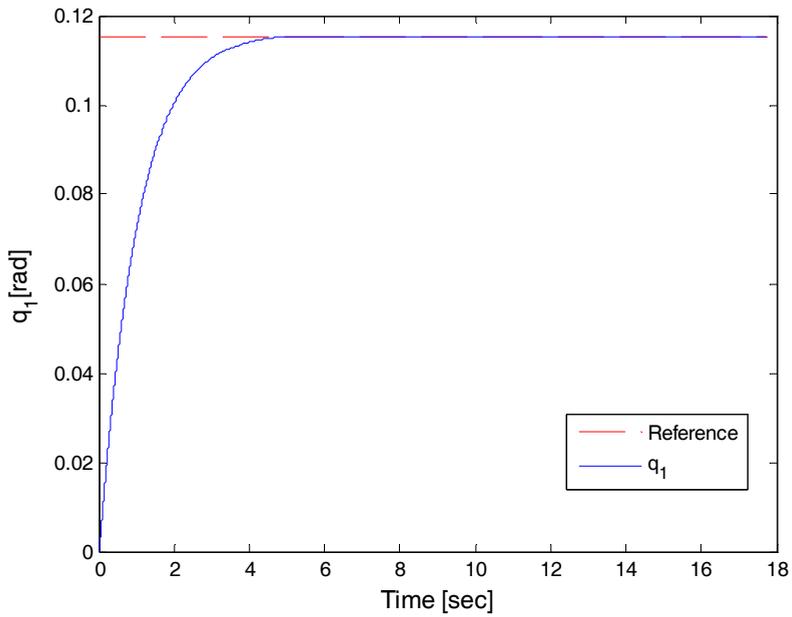
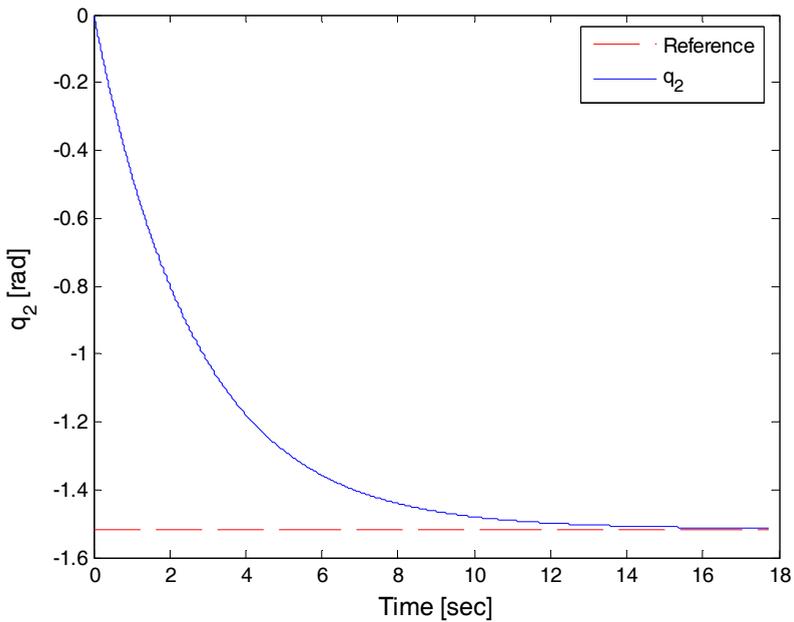


Fig. 12. Trajectory described on the Cartesian space

Fig. 13. Time evolution of q_1 Fig. 14. Time evolution of q_2

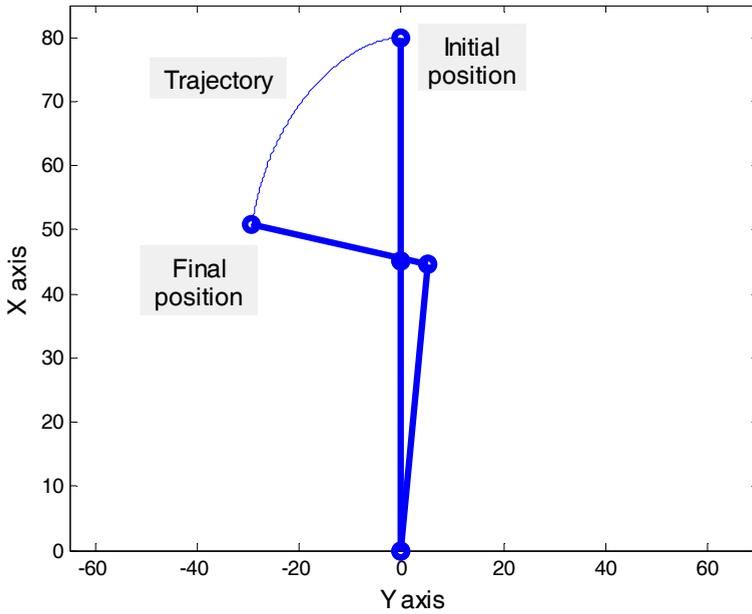


Fig. 15. Trajectory described on the Cartesian space

4.3.2 Experimental results for the visual controller

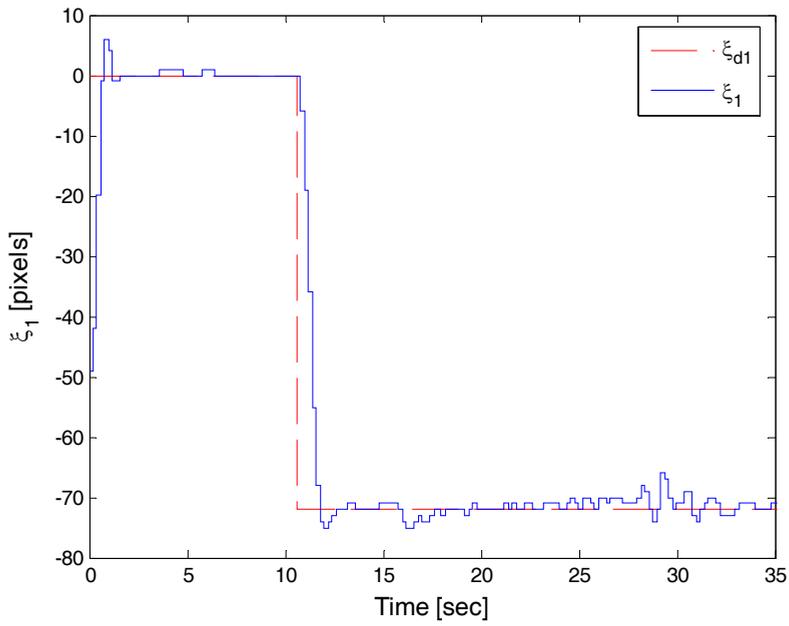
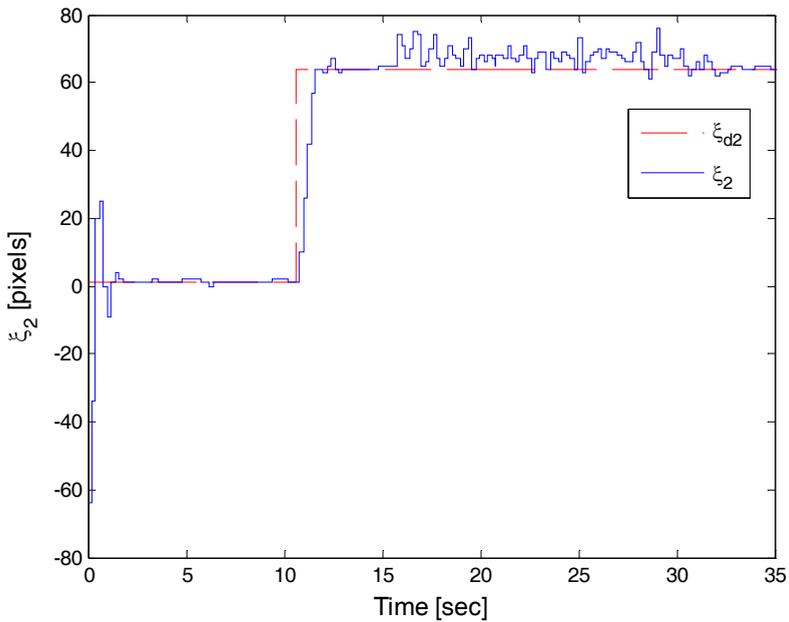
Third experiment is carried out with the passivity based visual controller, considering only first and second joint of the robot. The controller is implemented in the Control Program and runs with a sample time of 1 msec. for the controller and 33 msec. for the image processing. The gain matrices, obtained with the LMI-tool (El Ghaoui et al., 1995) are,

$$\mathbf{K}_1 = 10^{-5} \begin{bmatrix} 0.1443 & -0.1443 \\ -0.1443 & 0.4096 \end{bmatrix} \text{ with } \gamma = 3.9$$

$$\mathbf{K}_2 = 10^{-4} \begin{bmatrix} 0.0496 & -0.0496 \\ -0.0496 & 0.1399 \end{bmatrix} \text{ with } \gamma = 0.9$$

The experiment starts with an initial vector of image features $\xi(0) = [-48 \quad -65]$ pixels and the first reference on the image plane is chosen as $\xi_{d1} = [0 \quad 2]$ pixels, and then the reference changes to $\xi_{d2} = [-72 \quad 64]$ pixels. At instant $t = 15$ sec. the object starts moving.

Figures 16 and 17 show the time evolution of the image features ζ_1 and ζ_2 respectively, being ζ_1 and ζ_2 the components of the vector ξ . The time evolution of the features error norm can be seen in Fig. 18. In this last plot, it can be seen that the image error is below 2 pixels when the object is not moving ($t < 15$ sec); and with a moving object, the features error is below 10 pixels. Figure 19 shows the control actions for q_1 and q_2 . Finally, Fig. 20 shows the evolution of the image features on the image plane.

Fig. 16. Time evolution of the image feature ξ_1 Fig. 17. Time evolution of the image feature ξ_2

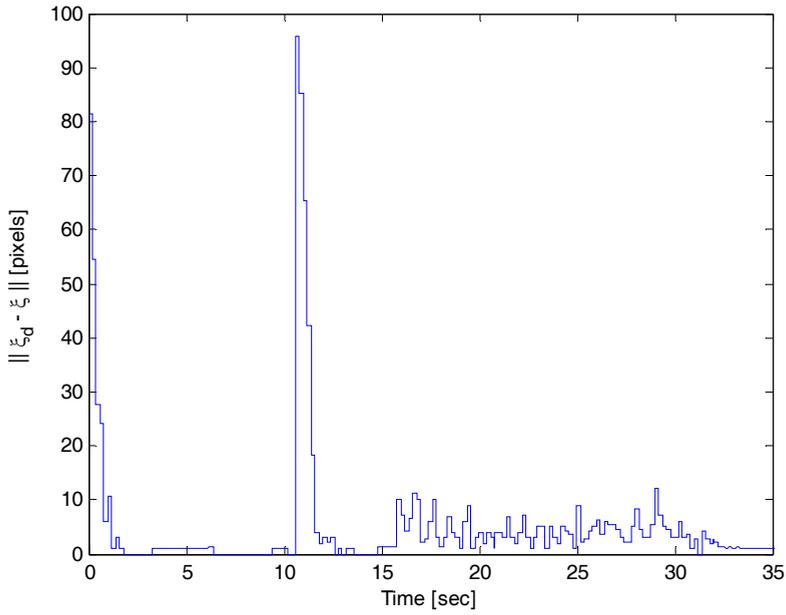


Fig. 18. Time evolution of the image features error norm $\|\xi_d - \xi\|$

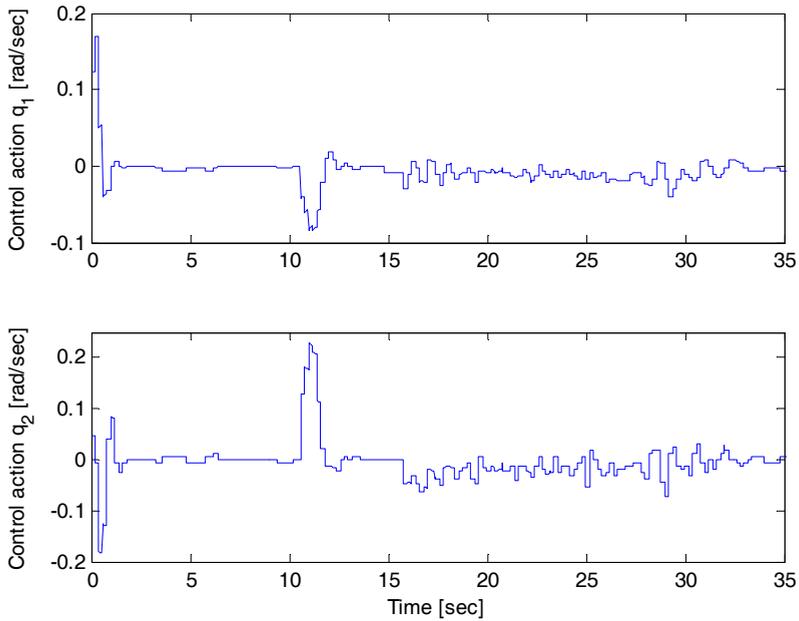


Fig. 19. Control actions for q_1 and q_2

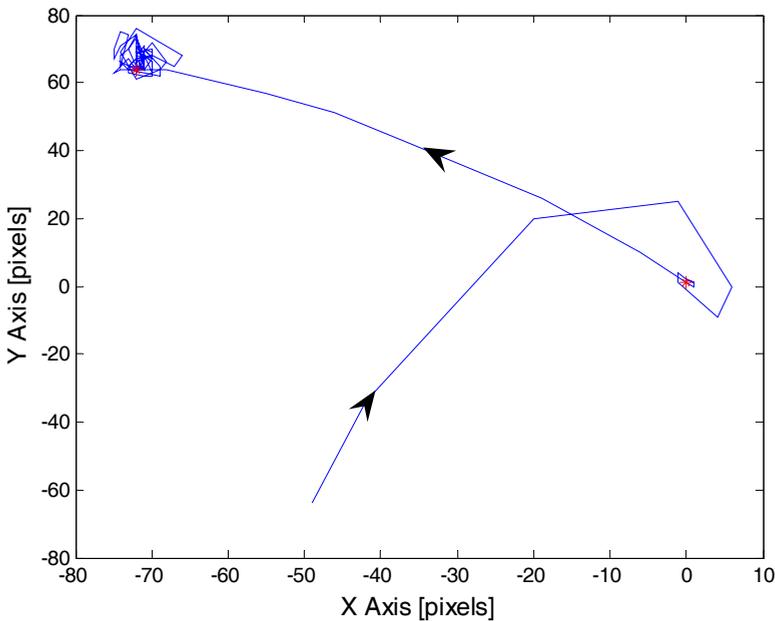


Fig. 20. Image features trajectory on the image plane

5. Conclusions

In this chapter, the design, implementation and experimentation of an open software structure for industrial robot manipulators have been presented. The developed software allows the users to save time and efforts in the implementation and performance evaluation of new control algorithms, as well as in the addition of new hardware components, i.e. sensors or actuators. Therefore, the developed software is useful for research in the field of robotics and human resource training, with potential impact in industry.

The software system has been split into two different programs that communicate each other, clearly dividing different tasks of the control system. This way, a modular reuse based system is obtained. First program (*Critic Time Program*) is responsible for communicating with the sensors and the actuators through the data acquisition and control hardware, updating the sensors' data in the shared memory block, and it is also responsible for synchronization of the two programs. Each one of the hardware devices is handled with a different object, obtaining the desirable encapsulation for the data and methods associated to each device. Second program (*Control Program*) is responsible for running the control algorithm and updating the control actions in the shared memory block.

Additionally, the proposed open software structure has been evaluated with two different control algorithms: first, a classical PD controller using the internal position sensors of the robot; and second, a passivity based visual controller using a vision system placed at the end effector of the robot. Both, the classical PD controller and the visual controller were successfully implemented in the proposed software structure, showing that the main objectives of the work presented in this chapter have been achieved.

6. Acknowledgment

Authors thank to the National Council of Scientific and Technical Research of Argentina (CONICET) for partially supporting this research.

7. References

- Boyd, S.; El Ghaoui, L.; Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*, Society for Industrial Mathematics, ISBN: 0-89871-334-X, Philadelphia, PA, USA.
- El Ghaoui, L.; Nikoukhah, R. and Delebecque, F. (1995). LMITOOL: a Package for LMI Optimization, *Proceedings IEEE Conference on Decision and Control*, pp. 3096-3101, ISBN: 0-7803-2685-7, New Orleans, LA, USA, December 1995.
- Frederick, M. P. and Albus, J. S. (1997). Open architecture controllers, *IEEE Spectrum*, Vol. 34, N° 6, (June, 1997) 60-64, ISSN: 0018-9235.
- Fujita, M.; Kawai, H. and Spong, M. W. (2007). Passivity-based Dynamic Visual Feedback Control for Three Dimensional Target Tracking: Stability and L_2 -gain Performance Analysis. *IEEE Transactions on Control Systems Technology*, Vol. 15, N° 1, (January 2007) 40-52, ISSN: 1063-6536.
- Hill, D. and Moylan, P. (1976). Stability results for nonlinear feedback systems. *Automatica*, Vol. 13, N° 4, (July 1976) 377-382. ISSN: 0005-1098.
- Hutchinson, S.; Hager, G. and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, Vol. 12, N° 5, (October 1996) 651-670, ISSN: 1042-296X.
- Jang, W. and Bien, Z. (1991). Feature-based visual servoing of an eye-in-hand robot with improved tracking performance, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2254-2260, ISBN: 0-8186-2163-X, Sacramento, USA, April 1991.
- Kelly, R.; Carelli, R.; Nasisi, O.; Kuchen, B. and Reyes, F. (2000). Stable Visual Servoing of Camera-in-Hand Robotic Systems. *IEEE Transactions on Mechatronics*, Vol. 5, N° 1, (March 2000) 39-48, ISSN: 1083-4435.
- Khalil, H. K. (2001). *Non-linear Systems*, Prentice-Hall, ISBN: 978-0130673893, New Jersey, USA.
- Krten, R. (1999), *Getting Started with QNX Neutrino 2: A Guide for Realtime Programmers*, PARSE Software Devices, ISBN: 978-0968250112, Ottawa, Canada.
- Lin, W. (1995). Feedback Stabilization of General Nonlinear Control System: A Passive System Approach. *Systems & Control Letter*, Vol. 25, N° 1, (May 1995) 41-52, ISSN: 0167-6911.
- Ortega, R.; Loria, A.; Kelly, R. and Praly, L. (1995). On passivity based output feedback global stabilization of Euler-Lagrange systems. *International Journal of Robust and Nonlinear Control*, Vol. 5, N° 4, 313-323, ISSN: 1049-8923.
- Ortega, R.; Loria, A.; Nicklasson, P. J. and Sira-Ramirez, H. (1998). *Passivity-based control of Euler-Lagrange systems: Mechanical, Electrical and Electromechanical Applications*, Springer-Verlag, ISBN: 978-1852330163, Berlin.

- Sawada, C. and Akira, O. (1997). Open controller architecture OSEC-II: architecture overview and prototype system, *Proceedings of International Conference of Emerging Technologies and Factory Automation*, pp. 543-550, ISBN: 0-7803-4192-9, Los Angeles, CA, USA, September 1997.
- Sciavicco, L. and Siciliano, B. (2001). *Modelling and Control of Robot Manipulators*, Springer-Verlag, ISBN: 978-1852332211, London, Great Britain.
- Slotine, J and Li, W. (1991). *Applied non linear control*, Prentice-Hall, ISBN: 978-0130408907, New Jersey, USA.
- Sommerville, I. (2000). *Software Engineering*, Pearson Education, ISBN: 978-0201398151, USA.
- Spong, M. and Vidyasagar, M. (1989). *Robot dynamics and control*, John Wiley & Sons, ISBN: 978-0471612438.
- United Nations Economic Commission for Europe (UNECE) and International Federation of Robotics (IFR). (2005). *World Robotics - Statistics, Market Analysis, Forecasts, Case Studies and Probability of Robot Investment*, International Federation of Robotics and United Nations Publication, ISBN: 92-1-1011000-05, Geneva, Switzerland.
- van der Schaft, A. (2000). *L₂-Gain and Passivity Techniques in Nonlinear Control*, Springer-Verlag, ISBN: 978-1852330736, London, Great Britain.
- Vidyasagar M. (1979). New passivity-type criteria for large-scale interconnected systems. *IEEE Transactions on Automatic Control*, Vol. 24, N° 4, (August 1979) 575-579, ISSN: 0018-9286.
- Weiss, L. E.; Sanderson, A. and Neuman, P. (1987). Dynamic Sensor-based Control of Robots With Visual Feedback. *IEEE Journal of Robotics and Automation*, Vol. 3, N° 9, (October 1987) 404-417, ISSN: 0882-4967.
- Willems J. C. (1972a). Dissipative dynamical systems part I: General theory. *Archive for Rational Mechanics and Analysis*, Vol. 45, N° 5, (January 1972) 325-351, ISSN 0003-9527.
- Willems J. C. (1972b). Dissipative dynamical systems part II: Linear systems with quadratic supply rates. *Archive for Rational Mechanics and Analysis*, Vol. 45, N° 5, (January 1972) 352-393, ISSN 0003-9527.
- William, E. F. (1994). What is an open architecture robot controller?, *Proceedings of IEEE International Symposium on Intelligent Control*, pp. 27-32, ISBN: 0-7803-1990-7, Columbus, Ohio, USA, August, 1994.

Miniature Modular Manufacturing Systems and Efficiency Analysis of the Systems

Nozomu Mishima, Kondoh Shinsuke,
Kiwamu Ashida and Shizuka Nakano
*Advanced Manufacturing Research Institute, AIST
Japam*

1. Introduction

In recent world, there are many small mechanical parts and products are used for mobile phones, medical devices, home appliances, and so on. However, manufacturing systems for those devices are large and complex. Manufacturing systems are not goals. So, manufacturing systems should be small as possible within satisfying requirements in the production. In addition, every activity in manufacturing industry is required to be environmentally benign, these days. Being environmental consciousness a big trend in manufacturing technology, space occupied and energy used by conventional manufacturing systems became considered as big wastes. Among all the energy usage of a manufacturing system, just a small portion is used for cutting and the rest for moving heavy structures of machines or generating heat. So, a large machine represents considerable waste. As a countermeasure for the situation, AIST (National Institute of Advanced Industrial Science and Technology) proposed a concept of a microfactory that consists of tiny machine tools and robots. However, for the first decade, the concept had been only a figure indicating a future application after micro-machine technology has been developed. Miniaturization of machine tools to size compatible to the target products without compromising the machining tolerances leads to enormous savings in energy, space, and resources. It also makes it easy to change the production layout of the factory. In 1996, AIST developed the first prototype of the miniaturized machine tool; a micro-lathe [1], with considerable metal cut capability and substantial energy saving effects. The machining capability of the lathe was far better than we expected in advance. This success of the micro lathe was the driving force to prototype a whole factory that performs a series of fabrication and assembly on a desktop. In 1999, AIST designed and established a machining microfactory, which consisted of afore-mentioned micro-lathe, other small size machine tools and manipulators for parts handling and assembly. Ttest results showed that a downsized manufacturing system could be a feasible option for micro mechanical fabrication. Some other miniature manufacturing systems [4-6] have been proposed since then and the concept has now become quite common.

Downsizing of manufacturing systems could potentially reduce environmental impacts and manufacturing costs, especially for diverse-types-and-small-quantity production. However, since no studies have been carried out to evaluate the effect of downsizing quantitatively, the actual potential of such systems to reduce environmental impacts in micro mechanical fabrication is still unknown. In addition, it was found that aforementioned miniature systems had some problems in the aspect of productivity and flexibility, since they mainly focused on reducing the size. In 2007, AIST developed the new concept of downsized manufacturing system called “on-demand MEMS factory.” And a simple method for evaluating the environmental consciousness and productivity to help system configuration design was also proposed. Then AIST compared the evaluation results with that of a conventional manufacturing facility, in order to say that the concept is feasible and hopeful.

2. The First Prototype; Microfactory

2.1 Micro/Meso Mechanical Fabrication

“Microfactory” was a concept of a future manufacturing system as an answer to the situation. It was proposed in the Japanese national R&D project named “Micro Machine Project [1].” The concept of the microfactory was very simple. The development team including one of the authors thought if it is possible to build “a super-miniature factory” for micro mechanical fabrication, environmental impact of manufacturing can be decreased greatly. In 1999, AIST developed the first prototype of a microfactory that consists of miniature machine tools and miniature manipulators. (Fig.1) The microfactory was able to perform a series of fabrication and assembly within a small desktop [2,3]. The result of the test production led us to conclude that the microfactory had considerable capability of micro mechanical fabrication.

The development team insisted that the microfactory would reduce environmental impact and costs of “diverse-types-and-small-quantity production”, “one-off production” or “variety-and-variant production”. Since the smallness of the machines enables flexible layout changes, it can control the increase of the costs when the product designs have been modified. However, since there have been no effort to evaluate efficiency of the microfactory comparing with conventional factory quantitatively, the advantage to reduce environmental impact is still uncertain. The purpose of this report is to explain briefly about the microfactory and propose a simple and useful efficiency index to support system configuration design of microfactory-like system.

2.2 Design of the Microfactory

The features of the microfactory due to extreme compactness are shown below.

- a) Significant reduction of energy consumptions for machine drive and atmosphere control.
- b) Increase of flexibility in the system layout.
- c) Improvement of machine robustness against external error sources due to low heat generation and high resonance frequency.
- d) Increase of speed and positioning accuracy due to decrease of inertial forces.

These features can be implemented to systemize various type of future manufacturing systems, which were extracted from an investigation [2]. Those are on-site manufacturing, mobile manufacturing, manufacturing under extreme condition, and so forth. In 1998, the authors proposed a conceptual drawing shown in Fig.1. The figure shows the microfactory

under microscopic vision and master-slave control by an operator to assemble small parts to a product. We tried to prototype the practical microfactory according to the figure.

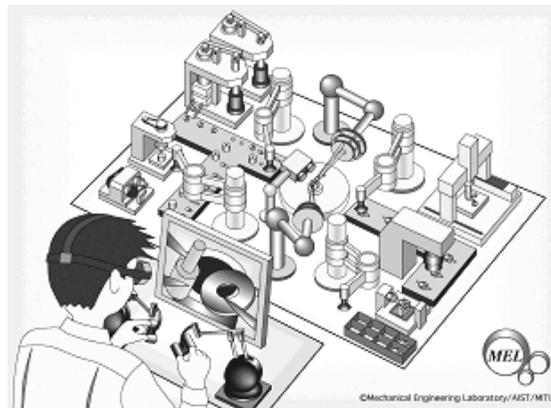


Fig. 1. Concept of the microfactory

Although the original concept contains not only machine tools and manipulators but also measurement instruments, for the first systemizing effort, we tried to prototype the left upper half of the drawing, except inspection devices. The actual factory shown in Fig. 2 is the first prototype of the microfactory developed in 1999, integrating of three machine tools and two manipulators. The components of the factory were set on a desktop, which is approximately 50cm deep and 70cm wide. Controllers, amplifiers and measurement systems were placed under the table. Using the desktop microfactory, test production experiments were conducted to confirm the capability of the system for machining and assembly to manufacture miniature mechanical products. The concept of the microfactory is to fabricate small products using small amount of energy and space. Production rate has not been a critical issue at this time. On the other hand, the configuration change of the factory corresponding to the product will be flexible.

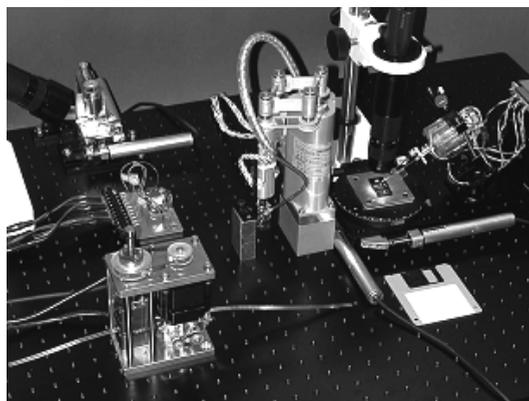


Fig. 2. Overview of the desktop microfactory

From the beginning of the project, new manufacturing systems enabled by the microfactory technology have been always focused on. For example, microfactories will enable on-site and on-demand manufacturing by transferring complete set-ups of factories to places where small products are necessary. In 2000, the second prototype of the microfactory was packaged in a suitcase having the same components as the first desktop prototype had, to demonstrate its portability and the above-mentioned concept will have a reality in future. The portable microfactory shown in Fig. 3 is driven by single AC100V power source and its power consumption during operation is 60W. The dimensions of the external case are 625mm long, 490mm wide and 380mm high and it weighs approximately 34kg. The target device is selected with a rotary switch and controlled manually by using two levers. An operator can observe the machining sections via an LCD monitor and three CCD cameras mounted on three machine tools.

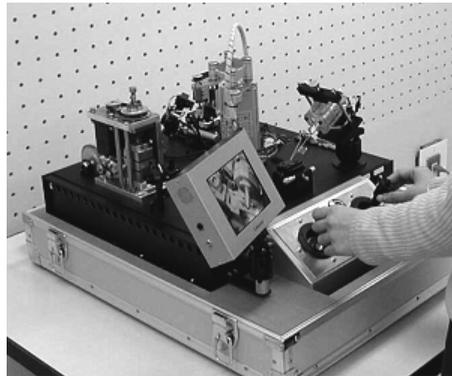


Fig. 3. Portable microfactory

Development of a miniature manufacturing system as the afore-mentioned microfactory is becoming a trend in precision manufacturing area. In the US [3], Germany [4], Switzerland, Singapore and in some other countries, many microfactories have been developed or proposed. In Japan, the other interesting trial is an EDM microfactory prototyped through collaborative research work by several private companies. (Fig. 4)

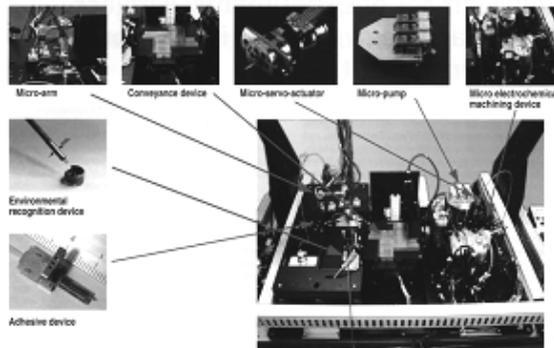


Fig. 4. EDM microfactory

2.3 Component Design of the Microfactory

As it was mentioned in the former section, the microfactory consists of three miniature machine tools and two small manipulators. The components are the micro lathe, a micro press machine, a micro mill, a micro transfer arm and a micro two-fingered hand.

Micro-lathe:

The first performable component of the factory, micro-lathe developed in 1996. Fig.5 shows a photograph of the micro-lathe. The major parts of the micro-lathe are a main spindle unit, two linear feed units, and a tool holder. The dimensions of the micro-lathe are 32 mm in length, 25 mm in depth, and 30.5 mm in height. It weighs 100g and the motor to drive the main spindle is only 1.5 W DC motor. The dimensions, weight and the rated power of the micro-lathe are respectively about 1/50, 1/10000, and 1/500 of a normal lathe. By attaching cemented carbide or diamond tool as the cutting tool, the micro-lathe could machine brass, stainless steel and other materials. Surface roughness and roundness error were measured to evaluate the machining performance of the lathe. In the case of brass, the surface roughness (R_{max}) in the feed direction was approximately $1.5 \mu\text{m}$ and the roundness error was $2.5 \mu\text{m}$ in average. Roughly speaking, these results indicate that the micro lathe is more accurate than a conventional lathe.

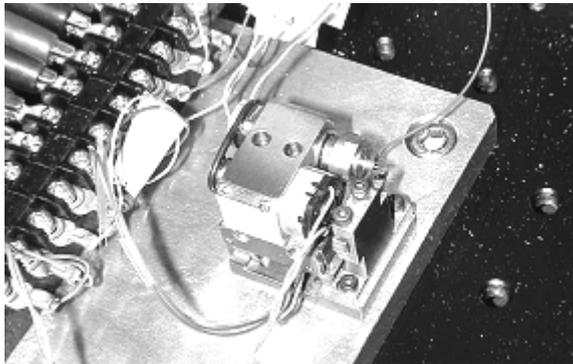


Fig. 5. Micro-lathe

Micro press:

The next component of the microfactory is the micro press machine indicated in fig. 6. Plasticity processing seems to be a hopeful area to replace conventional large machines by miniature machines. Although the press machine is only 170mm in height, it implements six stages forward-feed process including four punching and two bending processes shown in Fig. 7. in a single small die-set.(Fig.8) The small die-set enabled a high accuracy and high speed stroking performance reaches nearly 500 strokes/min. The rate is no less than that of a big so-called "high speed press machine". As the study indicated, the micro press machine has a high possibility for practical use for micro mechanical fabrication.

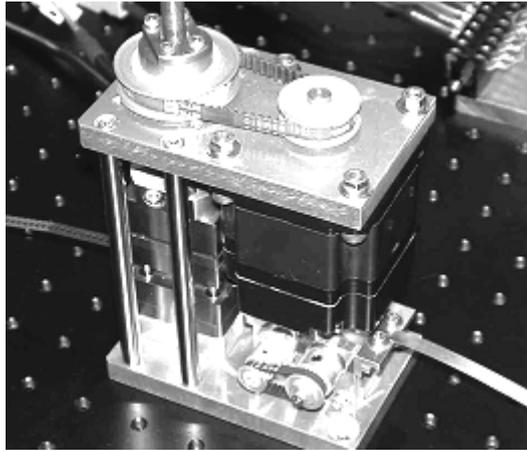


Fig. 6. Micro press machine



Fig. 7. Integrated production processes

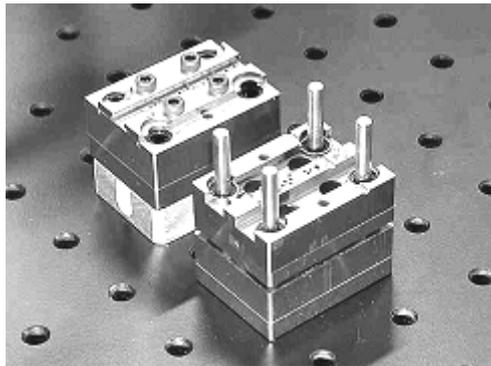


Fig. 8. Miniature die-set

Micro mill:

For the machine tools of the microfactory, the existing experience of machine tool design may not be applicable. Machine tool designers will need a general design guideline to appropriately reduce the size of machine tools. In designing the third component of the microfactory; the micro mill, we proposed a new design assisting tool. The tool combines an analytical procedure representing the machining motions known as form-shaping theory [6] with a well-known robust design procedure; the "Taguchi method" [7]. The effort identifies critical design parameters that have significant influence on the machining tolerance [8]. In this paper we applied the method to analyze the effect that the machine tool structure has

on its machining performance. To obtain a design guideline, we compared two different designs having same machine parts, same dimensions and different distribution of degrees of freedom (DOF). Following the machine tool elements from the product towards the cutting tool, one has two axes before the static part (type 1), whereas the other has all three axes concentrated (type 2). Fig. 9 shows the two designs. Machine structure like type 1 is the most common design for mills. A significant question is which of the two typical types has better theoretical performance than the other. Design evaluation method proposed by one of the authors [9] clarified type 1 is better. According to the evaluation result, actual micro mill used in the microfactory was designed as Fig.10. The machine has three feed axes and the main spindle, being approximately 12 X 12 X 10 cm. It can perform drilling up to 2mm in depth and face milling up to 3 mm X 3 mm in area. For the feed motions and the rotational motion, DC servo motors were used.

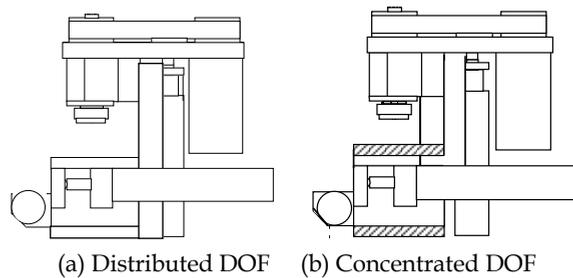


Fig. 9. Two design candidates of the micro mill

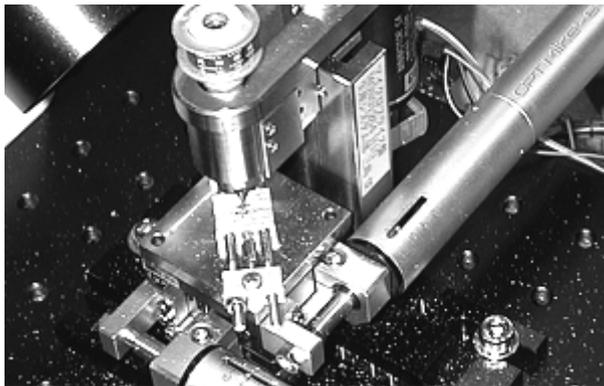


Fig. 10. Micro mill

Micro transfer arm:

In a factory, assembly is also an important process. As for the parts handling of the microfactory, the micro transfer arm was designed. (Fig. 11) One of the primary functions needed to the arm was a pick-and-place capability of the parts machined by micro machine tools. According to the purpose, multiple requirements such as compact mechanism for less space occupation, wide work-area for transfer capability and fine positioning accuracy were necessary. To meet these requirements, the arm features parallel mechanism shown in the figure and has 4 degrees of freedom (DOF) for the end-effector, 3-DOF for transitional

motions, and 1 for a wrist rotational motion. AC servo motors to drive two transitional motions and one rotational motion are concentrated in the cylindrical body to achieve a compact size and flexible movement at the same time. Vertical motion of the gripper is done by a linear ultrasonic motor. Using vacuum suction, the arm can transfer all the parts handled in the microfactory, from a parts stocker to an assembly yard. Locations of the stocker and the assembly yard should be studied by the controller in advance to the operation. However, to detach the parts from the machine tools and to place them to the stocker, must be done by human operation. The basic configuration of the transfer arm is shown in Fig. 12. The parallel mechanism in the top consists of one active joint A and three passive joints B, C and D. The links AB and AC are driven at joint A by DC servo motors. By moving AB and AC to the same directions, the end-effector rotates around joint A. The end-effector moves towards or against joint A, by moving AB and AC to opposite directions. Through experimental evaluations, the repetitive positioning error of the gripper edge along the X-Y plane was less than $10 \mu\text{m}$.

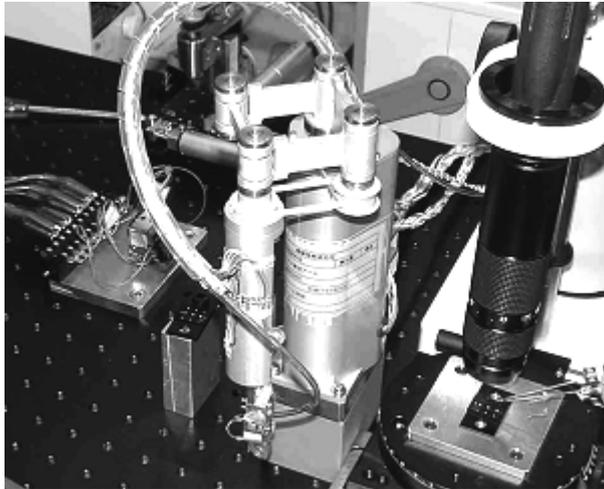


Fig. 11. Micro transfer arm [9]

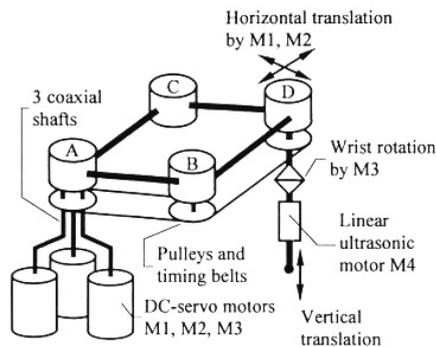


Fig. 12. Configuration of the micro transfer arm

Micro two-fingered hand:

The last component of the microfactory is shown in fig. 13. The two-fingered micro hand with two finger modules was designed for parts assembly in the microfactory. It was designed based on the chopsticks manipulation [10]. The chopsticks manipulation can be performed with two fingers having 3-DOF parallel mechanism each. Each finger module has a thin glass rods, is driven by three PZT actuators and works collaboratively to achieve high positioning accuracy for micro assembly. Using two fingers, the micro-hand can grasp, move or rotate the small objects from about 50 to 200 microns. In this size, since surface force is more significant than gravity, it is relatively difficult to release the object, rather than to grasp it. The main reason to use glass for the fingers is that it is relatively easy to obtain sharp edges by heating and stretching the glass rods. Being the positioning accuracy of the edge of the glass finger within 0.5 microns, this micro hand was originally developed for cell handling, and contributed in the microfactory project by assembling the tiny parts placed by the transfer arm. Parts to be handled in the micorfactory have a few hundreds microns in size, and it is necessary that a working area of the hand is more than the parts sizes at least. However, a PZT device as an actuator has only about 1 % elongation capability of its length. Layer type PZT actuators were used to satisfy the requirements. Because a micro hand with compact size was demanded for the micractory, the hand was arranged as turning back an inner finger module into inside of an outer finger module.

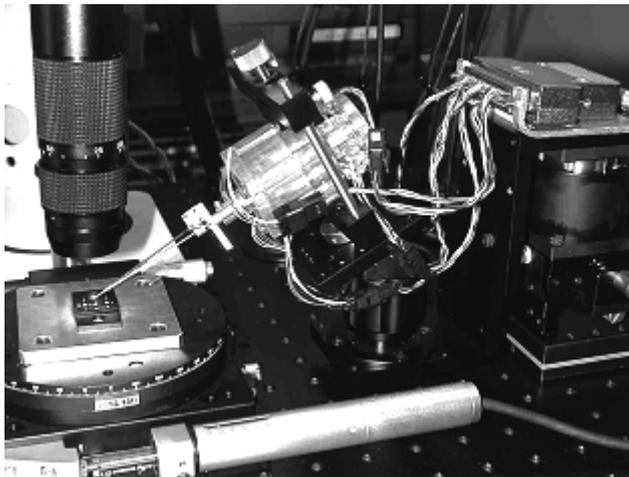


Fig. 13. Micro two-fingered hand

2.4 Test Production

To prove that the microfactory is capable to machine and assemble a “product”, we selected an extra small ball bearing as the test product. Fig. 14 shows the target product, which is a ball bearing having 100 μm rotary shaft diameter and 900 μm outer diameter. The next Fig. 15 shows the results of the test production. The shape appears in the top of the photograph is the bearing assembly that 7 steel balls each side.

Following procedure was the anufacturing process for the test product shown in Fig.15.

- 1) The **micro press machine** punched the **top cover** of the bearing.

- 2) The **micro lathe** turn-cut the **rotary shaft**.
- 3) **Micro mill** machined the top and bottom surfaces of the cup type bearing **housing** and drilled the inner cavity.
- 4) The **micro transfer arm** transferred the parts from the parts stocker to the assemble yard by vacuum suction type gripper.
- 5) The **micro two-fingered hand** assembled all the machined parts and steel balls.
- 6) To ensure the assembly, top cover and the bearing housing were fixed by liquid bond.

As the result of the fabrication of the ball bearing, the microfactory was capable to assemble the test product within a single desktop, which is approximately 50 by 70cm. In addition, because of its extremely small size, it would be easy to reconfigure the system corresponding to the large variety of the products. Therefore, the microfactory has a future possibility as a manufacturing system to produce many varieties of extra small machine parts. However, it still has some problems, such as the low production rate or the difficulty of the fixture of the product. To apply the microfactory or similar small manufacturing systems to actual productions, those problems have to be solved.

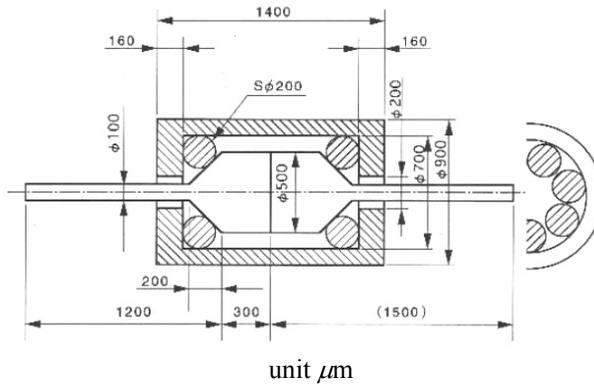


Fig. 14. Target product

3. Proposal of Total Performance Analysis

3.1 Total Performance Analysis of Eco-Products

This paper tries to define an index that can be used to evaluate the efficiency of manufacturing processes. Based on existing research [11], the authors have proposed an index that can be used to evaluate the real environmental performances of products by considering each product's utility value, cost and environmental impact, throughout the lifecycle of the product. The efficiency index is defined by (1).

$$TPI = \frac{UV}{\sqrt{LCC} \sqrt{LCE}} \quad (1)$$

TPI: total performance indicator

UV: utility value of the product

LCC: lifecycle cost of the product

LCE: lifecycle environmental impact of the product

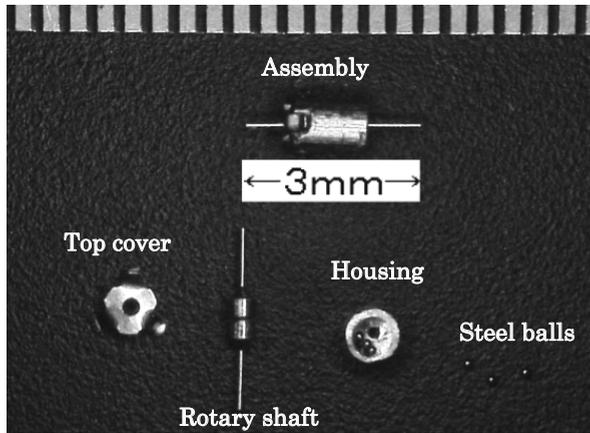


Fig. 15. Bearing parts and the assembly

Value per cost is often used to evaluate product performance in the field of quality engineering, and additional value per environmental impact (so-called eco-efficiency [12]) is also a common index used in design for the environment [13] when evaluating other aspects of product performance. However, there are three major reasons why we consider that these existing evaluation indexes cannot be applied to practical design improvements for individual products.

- 1) Existing indexes cannot evaluate the environmental and economic aspects simultaneously.
- 2) Since the “value” is a fixed amount, existing indexes cannot accommodate any change in value throughout the product lifecycle.
- 3) Since existing indexes often consider a product as an inseparable object, they are not helpful in identifying bottleneck components.

In order to address the point 1) of the abovementioned problems, the proposed index comprises the simplest possible combination of the value/economic efficiency and the value/environmental efficiency. In our proposal, because the utility value of the product can be expressed as an integral of its occasional value throughout its lifecycle, it can simulate value deviation. About the point 2), Fig. 16 shows the assumed value decrease curve, due to two reasons shown in the figure.

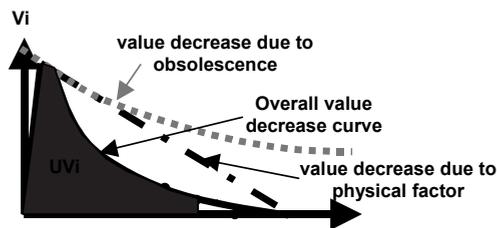


Fig. 16. Changenof product value due to two causes

In addition to solve the point 3), when estimating the value in the equation, it is possible to allocate value to each component by using a QFD [14] approach. By this approach, it will be possible to calculate the defined TPI for each component. Then, by comparing the component TPI with that of the product average, we can answer which component is the bottleneck in enhancing the total environmental efficiency of the product. Therefore, our proposed TPI can provide an answer to the problems encountered with existing eco-performance indicators.

3.2 Extension to Manufacturing Systems

The same approach can be used to evaluate the efficiencies of manufacturing systems. The extended calculation shown in (2) was proposed in the existing paper [15] and is used to evaluate system efficiency of the on-demand MEMS factory on the same basis. In this equation, the efficiency of the system is defined by the sum total of product values produced by the manufacturing system within a set period of time. Cost and environmental impact during the corresponding period are also considered. The cost and environmental impact involved in building the manufacturing system itself (the so-called initial cost and initial environmental impact) should be divided by the lifetime of the manufacturing system, and assigned to C and E in the equation.

$$\text{System TPI} = \frac{\int_{t_0}^{t_0+T} \sum_{i=1}^{i=m} \Delta UV_i dt}{\sqrt{\int_{t_0}^{t_0+T} \sum_{i=k}^{i=l} E_i C_i dt}} \quad (2)$$

T: period of estimation

m: kinds of product

k: number of the first process in the system

l: number of the last process in the system

However, it is not easy to quantitatively calculate (2) when the system produces a range of different products because it is necessary to quantify all the values of these different products. However, when the product of the manufacturing system is always the same, the system TPI equation can be simplified to (3).

$$\text{System TPI} = \frac{T_p}{\sqrt{\left(\frac{C_m}{L} + C_L + C_E\right) \cdot \left(E_e + \frac{E_m}{L \cdot 1600}\right)}} \quad (3)$$

T_p: throughput of the system

(number of products produced in an hour)

C_m: initial total cost of the machines (10⁴ yen)

C_L: labor cost (10⁴ yen/year)

C_E: energy cost (10⁴ yen/year)

- L: lifetime of the system (years)
- Ee: environmental impact caused by electricity use
- Em: environmental impact caused by the machines

Instead of using the utility value of a product, as defined in the original index, the throughput of the manufacturing system, “Tp”, can be adopted. By defining the throughput as the number of products fabricated within an hour, the total performance indicator of the manufacturing system can be calculated. “C” can be calculated by using the sum total of machine costs, labor costs and energy costs during the corresponding time period. However, other costs such as the cost of electricity (about 20 yen/kWh) are negligible. The cost of labor is assumed to be 5.0 (million yen) per person per year. For “E,” equivalent CO2 emission is the simplest index that can be used to evaluate the environmental impact. Therefore, “E” can be expressed by using the sum of CO2 emission values caused by power consumption and the materials used for machines and products. When considering electricity usage, 1kWh of electricity consumption is equivalent to 0.38kg-CO2.

4. Analysis of the Microfactory

4.1 Analysis of the Manufacturing Process

To fabricate the test product; miniature ball bearing shown in Fig.15, manufacturing process in Fig.17 was applied. Every part starts from the material shown in the upper side, passes through some sub-processes shown in the block and reaches the assembly processes written in the lower side. From the figure it is easily imaginable that the assembly processes are very time-consuming, because the processes should be done sequentially under a microscopic vision using the micro-hand. Table 1 indicates the average process time of the corresponding processes in Fig.17, after operators had been skilled enough. Number of operators required for each process is also shown in the table 1.

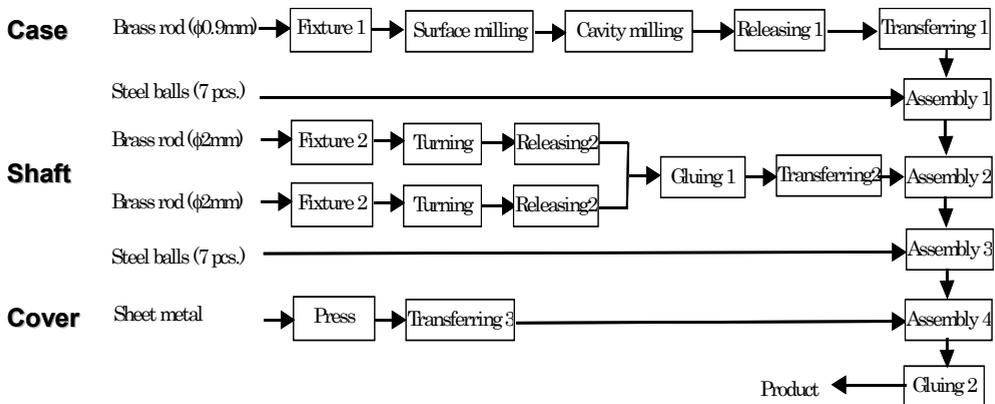


Fig. 17. Manufacturing process used in the test production of the microfactory

Process name	Process time	Number of operators
Fixture 1	10 sec.	1
Fixture 2	5 sec.	1
Surface milling	1 min.	1
Cavity milling	2 min.	1
Turning	2 min.	1
Press	0.2 sec.	0
Releasing 1	10 sec.	1
Releasing 2	5 sec.	1
Transferring 1	1 sec.	0
Transferring 2	1 sec.	0
Transferring 3	1 sec.	0
Assembly 1	3 min. (per ball)	1
Assembly 2	3 min.	1
Assembly 3	3 min. (per ball)	1
Assembly 4	3 min.	1
Gluing 1	1 min.	1
Gluing 2	2 min.	1

Table 1. Required time for each process per unit

4.2 Total Performance Analysis of the Microfactory

According to Fig.17 and Table 1, it is evident that the assembly operations done by the micro hand were the bottlenecks for the throughput. However, it is necessary to be aware that the flexibility of the process was assured by the function of the micro hand. Machine and labour costs are also important for manufacturing system designs. Table 2 shows the rough estimation for the cost of the machines used in the microfactory. And also the energy consumption of each machine is an important factor to consider system efficiencies. Table 3 shows the average power consumption of the machines during the operation. Both tables show that the micro hand was the most critical components for cost and energy, as well.

Machine	Milling	Turning	Press	Arm	Hand
Cost (millionYen)	0.7	1.2	2.0	3.0	5.0

Table 2. Machine costs

Machine	Milling	Turning	Press	Arm	Hand
Average power consumption (kw)	0.25	0.3	0.05	0.2	0.4

Table 3. Energy consumption

4.3 Configuration Design Based on the Analysis

Analysis of the manufacturing process mentioned in the former section showed that the assembly processes performed by “micro-hand” is critical both for throughput and environmental impact. When the number of components or operators is not limited to be 1, a simple strategy to enhance system efficiency will be to increase the number of the “hands.” By assuming the annual operation time of the system is 1600 hours, the efficiency is calculated by aforementioned equation (3). By changing the system configuration,

bottleneck will not always be the micro hand. Case study is necessary. Fig.18 shows the behaviour of the system efficiency calculated by the equation.

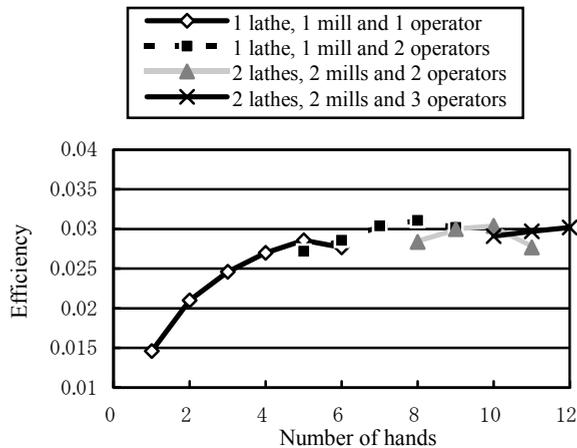


Fig. 18. Behaviour of the system efficiency (TPI)

In the figure, since press and transferring processes are not significant for the overall throughput, the figure shows the behaviour of the system efficiency according to the change of the number of hands, lathes and mills. According to Fig.18, it can be said that there are some local maximums. The result suggests that there are some suitable system configurations to satisfy low environmental impact and high efficiency simultaneously. By using the proposed efficiency indicator, it is possible to optimize system configuration of a microfactory-like miniature and modular manufacturing system.

5. Proposal of On-Demand Factory

5.1 Overview of the New System

To apply the idea of the first “microfactory” to practical manufacturing systems, a new development was necessary. The next-generation microfactory should be a system which enhances the original concepts of “miniature,” “flexible,” “environmentally benign” and “lean” to practical levels. Fig.19 shows a new prototype for a downsized factory (developed by AIST) which is called the “on-demand MEMS factory” [16]. As shown in the figure, there are four modularized units connected in-line. Each unit is 500mm wide, 800mm deep and 1200mm high. In this system, each unit corresponds to one process.

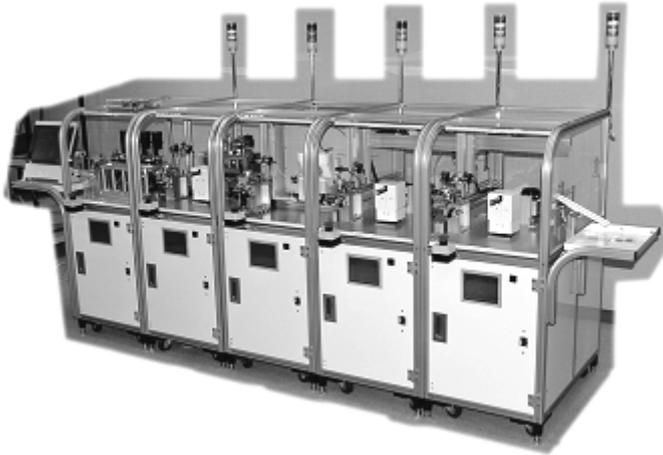


Fig. 19. Overview of the On-demand factory

5.2 What's New?; Metal-Based MEMS

The set-up shown in the figure includes a mechanical press unit, an aerosol deposition (AD) [17] unit, an annealing unit and a wiring unit. AD deposition (Fig.20) is an innovative and very productive manufacturing process for fabricating thin functional layers such as PZT layer. The system was designed to produce an optical scanner [18]. Micro optical scanners are a typical MEMS device and are usually fabricated on a silicon base structure. Silicon processing technologies, such as chemical vapour deposition, photolithography, reactive ion etching, etc. are used for the fabrication. Silicon is a good option for the base material of optical scanners, but other options are available. If there is an alternative material that can support the functional layers of the scanner, such as the electrode and PZT-resistor, and if a micro machining process can be applied, then "silicon" is not irreplaceable. In the on-demand MEMS factory, the use of a metal-base structure was proposed. It meant that the structure of the optical scanner is made of metal and that the electrode, PZT-resistor, etc. are directly deposited onto the base structure. For the fabrication of the base structure shown in Fig.21, a progressive micro mechanical press (Fig.22) can be used. Mechanical press is a low-cost, environmentally benign and highly efficient manufacturing process. Since the micro press unit composes the base structure of the device, it is not necessary to use silicon dry etching process which is a rather time-consuming process, requires a large facility, also. In addition, when product design changes, in the system can correspond by exchanging the die-set of the press unit. When there is a change in the process requirement, it is also possible to comply by switching the modularized units. Therefore, when product design changes occur frequently, high flexibility may compensate for low throughput.

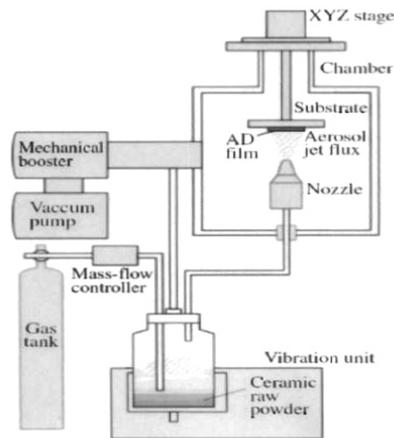


Fig. 20. Schematic view of the AD unit

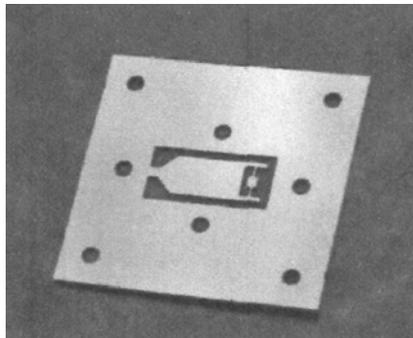


Fig. 21. Base structure of the optical scanner



Fig. 22. Progressive mechanical press unit

5.3 Comparison with Conventional MEMS Fabrication

To test whether the on-demand MEMS factory is a feasible system for diverse-types-and-small-quantity production, system efficiency was compared with that of a conventional MEMS facility. AS for the set-up of the on-demand MEMS factory shown in Fig.19, task times for each unit are shown in Table 4. It is also necessary to estimate machine costs, power consumptions and environmental impacts. However, since the system was developed by a package, it is difficult to determine the initial cost of each unit. Basically, the budget to prototype the units varied from 4 to 6 million JPY. Therefore, the average cost of each unit is assumed to have been 5 million JPY. Power consumption for each unit is shown in Table 5. Finally, the environmental impact (based on machine fabrication) should also be estimated. The weight of each unit varies from about 50 to 100kg, and each unit has one or two computers inside a structure made from aluminum, stainless steel and various plastics. A rough approximation of environmental impact was therefore based on LCA examples for computers and a typical mixed material machine (automobiles [19]). Survey data showed that, for a laptop computer, CO₂ emissions from the production stage totaled 109kg, while the equivalent figure for a car was about 2,800kg (estimated to be 2.8kg-CO₂/kg). The results are shown in Table 6.

Name of the process	Average task time (seconds)
Mechanical press	26
Aerosol deposition	28
Annealing	60
Wiring	60

Table 4. Task time for on-demand factory units

Name of the unit	Average power consumption (kW)
Mechanical press	0.2
Aerosol deposition	0.5
Annealing	1.5
Wiring	0.2
Other (pump, compressor etc.)	1.6

Table 5. Power consumptions for on-demand factory units

Name of the unit	Weight (kg)	Environmental impact (kg-CO ₂)
Mechanical press	100	519
Aerosol deposition	70	319
Annealing	70	319
Wiring	50	259

Table 6. Environmental impact caused by the fabrication of on-demand factory units

Based on these data, the system TPI of the on-demand factory, in various configurations, can be calculated. Typical manufacturing processes for MEMS devices are based on silicon processing technologies. The process requires expensive machines for photolithography and etching, high levels of power consumption, solvents, and so on. Fig.23 shows a typical

manufacturing process to fabricate silicon-made MEMS devices using semi-conductor-fabrication based technologies.

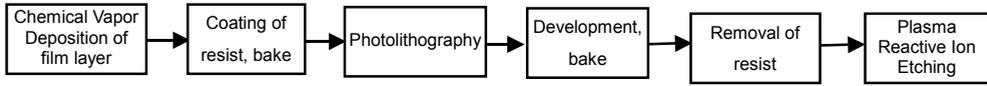


Fig. 23. Block diagram of a typical MEMS device fabrication process

However, focusing on diverse-types-and-small-quantity production of MEMS devices, an important requirement for manufacturing systems is the ability to change processes and system configurations flexibly in response to the frequent design changes made to the product. The on-demand factory has the ability to cope with such changes. In this system, one unit corresponds to one process. Since the micro press unit makes the base structure of the device, the system can respond to product design changes by exchanging the die-set of the press unit. Therefore, when frequent product design changes occur, high flexibility may compensate for low throughput. To simulate this point, the system efficiency of the on-demand factory was compared with that of a conventional MEMS line. Table 7 shows the comparison between an on-demand MEMS factory and a conventional silicon-based MEMS fabrication facility. Assuming that etching is the bottleneck process, the system throughput of the silicon-based MEMS can be calculated from the etch rate of silicon. In a technical paper [20], it was stated that the maximum Si etch rate was 15.6mm/min. Using the most recent dry etching equipment [21], an etch rate of 56mm/min has been reported. Assuming that the thickness of the sacrificial layer is about 500nm and that the equipment can have 6 parallel lines, the maximum throughput of the total system can be calculated. The comparison shown assumes that 12 devices can be made from one 4-inch wafer. Other data have been derived from surveys [22] As for the machine weight, since this is difficult to estimate quantitatively, we have assumed that it is roughly proportional to the machine set-up area. Fig.24 shows the comparison between the system TPI of the on-demand MEMS factory and that of a conventional MEMS production facility. In this figure, the maximum throughput of the conventional MEMS facility is shown along the horizontal axis. As shown in the figure, the maximum throughput of a single optimized on-demand factory is 120 units/hour. Therefore, as long as the required throughput is lower than this, one on-demand factory is sufficient. In order to respond to higher throughput, however, a second on-demand factory system is needed.

Type	Silicon-based MEMS fabrication	On-demand MEMS factory
Power Consumption (kWh/year)	320000	6400-9120
System throughput	1.5 min/sheet max. [21] (12 units/sheet)	0.5-1 min/unit
Other environmental impacts	Process gas, solvent etc.	None
Machine cost	540 million JPY[22]	20-30 million JPY
Machine set-up area	50m ²	3m ²
Machine weight	5,000kg (total)	290-430kg (total)

Table 4. Comparison of power consumption and throughput

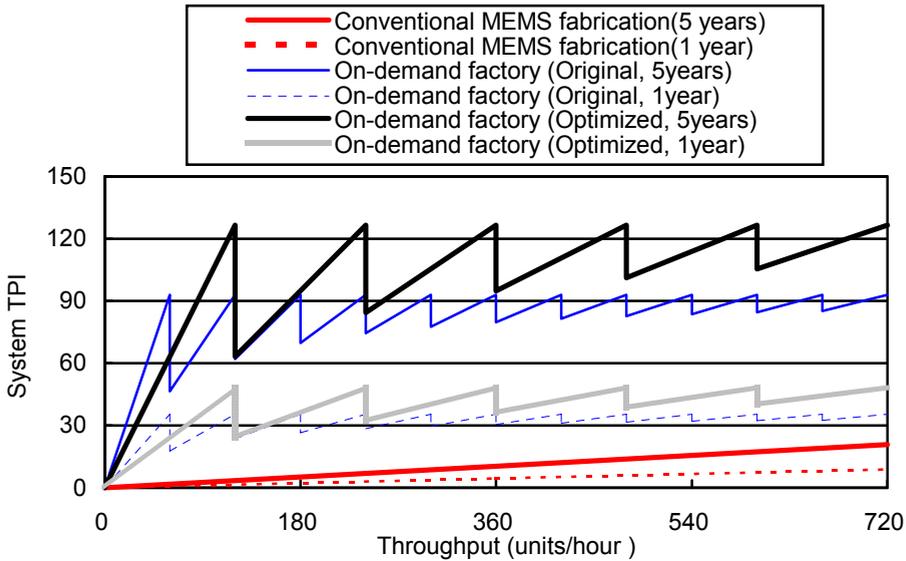


Fig. 24. Comparison of system efficiency(TPI)

When the required throughput is just over 120 units/hour, the TPI of two on-demand factory systems is almost half the maximum TPI of the single system, and when the second system approaches its full operation stage, the throughput of the total system approaches 240 units/hour and the TPI of the total system reaches its maximum.

As shown in the figure, it is evident that the system TPI of the on-demand factory is always higher than that of the conventional MEMS facility. Of course, this is only a simulation and it contains many assumptions. For example, it is not clear whether the maximum throughput can be always achieved by the on-demand MEMS factory, and it is not evident whether the value of a silicon-based MEMS device and a metal-based MEMS device is the same. Even though there are still some problems to be resolved, the comparison indicates that the newly developed on-demand MEMS factory has a bright future as a practical diverse-types-and-small-quantity production system of MEMS devices.

6. Summary

In this chapter, the microfactory developed by AIST in 1999 was introduced. The prototyped desktop microfactory showed the capability as a space-saving and energy-saving manufacturing system for small products. The portable microfactory demonstrated a concept of a future on-site manufacturing. The capability was proved through the test production. The development team explained that the advantages of microfactory-like miniature manufacturing systems are low environmental impact and high production flexibility. However, the actual effect has not been estimated quantitatively until now.

In order to evaluate system efficiencies of those manufacturing systems and to prove that microfactory-like systems are reasonable options for diverse-types-and-small-volume

production, a simple index to evaluate the system efficiency of manufacturing systems was proposed, based on the Total Performance Indicator proposed by the authors group. The indicator can evaluate the balance of functionality, cost and environmental impact of various products, considering value change throughout the product lifecycle. This time, the same idea was applied to manufacturing system evaluation. Using the modified equation, system efficiency of the first microfactory was calculated. As the results of the analysis, it was suggested that it would be able to re-design system configuration, by using the evaluation result. It led us to conclude that the proposed efficiency index is suitable for a modular manufacturing system like the microfactory. However, the actual efficiency of the microfactory was not high comparing to a typical mass production system for the same product. The fact suggested that a newer development would be necessary.

Then, a new concept of a miniature manufacturing system called on-demand MEMS factory was introduced. And the same index to evaluate the system efficiency of a downsized manufacturing system was applied by considering system throughput, machine costs, labor cost, and CO₂ emission caused by machine material and electricity. The evaluation result was compared with a rough estimation of the efficiency of a conventional MEMS fabrication facility based on semi-conductor fabrication technologies. The comparison indicated that the efficiency index of the on-demand MEMS factory was basically higher than that of the conventional facility. Especially when the required throughput is low, the advantage of the on-demand MEMS factory was evident. Thus, it was concluded that the concept is suitable for diverse-types-and-small-quantity productions of MEMS devices. In addition, manufacturing processes implemented in the on-demand MEMS factory such as "aerosol deposition" and "progressive mechanical press" were highly productive and environmentally benign processes comparing to conventional semiconductor fabrication technologies. Therefore, it was suggested that by applying new manufacturing technologies, an alternative material and a well-designed system, MEMS fabrication can be greatly improved.

As the future work, more precise comparisons with conventional manufacturing systems are required in order to prove the effectiveness of the concept. And quantification of product value when the design of the product is changed is also necessary. About the evaluation method, modification of efficiency index to consider production flexibility along with frequent changes of product design will become necessary to estimate the real advantage of modularized manufacturing systems. And comparison with other integrated eco-efficiency evaluation methods will be necessary to insist the advantages of the proposing method.

7. References

- [1] T. Kitahara, Y. Ishikawa, K. Terada, N. Nakajima, and K. Furuta, Development of Micro-lathe, *Journal of Mechanical Engineering Laboratory*, Vol. 50, No. 5, pp. 117-123, 1996.
- [2] N. Kawahara, T. Suto, T. Hirano, Y. Ishikawa, N. Ooyama and T. Ataka, Microfactories; New Applications of micro machine technology to the manufacture of small products, *Microsystem Technologies*, 3-2, pp37-41, 1997.
- [3] R. Hollis and A. Quaid, An architecture for agile assembly, *Proceedings of ASPE 10th Annual Meeting*, 1995.

- [4] T. Gaugel et. al., Advanced Modular Production Concept for Miniaturized Products, Proceeding of 2nd International workshop on Microfactories, Fribourg, Switzerland, pp. 35-38, 2000.10.
- [5] K. Furuta, Experimental Processing and Assembling System (Microfactory), Proceedings of the 5th International Micromachine Symposium, pp. 173-177, 1999.
- [6] D. N. Reshtov and V. T. Portman, Accuracy of Machine Tools, ASME Press, New York, 1988.
- [7] G. Taguchi and S. Konishi, Quality Engineering Series Vol.1-4, ASI Press, 1994.
- [8] N. Mishima and K. Ishii, Robustness Evaluation of a Miniaturized Machine Tool, Proceedings of ASME/DETC99, Las Vegas, NV, DETC/DAC-8579, 1999.9.
- [9] T. Tanikawa, H. Maekawa, K. Kaneko and M. Tanaka, Micro Arm for Transfer and Micro Hand for Assembly on Machining Microfactory, Proceedings of 2nd Int'l Workshop on Microfactories, pp. 155-158, 2000.10.
- [10] T. Tanikawa and T. Arai, Development of a Micro-Manipulation System Having a Two-Fingered Micro-Hand, IEEE Transaction on robotics and Automation, Vol. 15, No. 1, pp.152-162, 1999.
- [11] S. Kondoh, K. Masui, M. Hattori, N. Mishima and M. Matsumoto, "Total Performance Analysis of Product Life Cycle Considering the Deterioration and Obsolescence of Product Value," Proc. of CARE INNOVATION 2006, Vienna, Austria, 2006, 2.10.1.
- [12] Y. Kudoh, K. Tahara and A. Inaba, Environmental Efficiency of Passenger Vehicles: How Can the Value of a Vehicle be Determined?, Proc. of 7th International Conference on Ecobalance, Tsukuba, Japan, 2006, B3-7.
- [13] M. Ernzer, M. Lindahl, K. Masui and T. Sakao, An International Study on Utilizing of Design Environmental Methods (DfE) -a pre-study-, Proc. of Third International Symposium on Environmentally Conscious Design and Inverse Manufacturing, IEEE Computer Society, 2003, pp.124-131.
- [14] K. Akao, Quality Function Deployment, Productivity Process, Cambridge, MA, 1990.
- [15] N. Mishima, S. Kondoh and K. Masui, Proposal of An Efficiency Evaluation Index for Micro Manufacturing Systems, Proc. of IEEE ICMA 2007, Harbin, China, Aug. 2007, pp.51-56.
- [16] S. Nakano, K. Ashida, J.Park and J.Akedo, On-demand MEMS factory system consists of metal forming and aerosol deposition, Proc. of the 24th International Japan-Korea Seminar on Ceramics, pp.735-738.
- [17] J.Akedo and M.Lebedev, Japan Journal of Applied Physics, 38, 9B (1990), 5397-5401.
- [18] Y.Takahashi, Y. Takeuchi and H. Fujita, Micromachined Optical Scanner for Optical Measurement System, IEEE Trans. SM, Vol.123, No.10, 2003, pp403-408. (In Japanese)
- [19] Japan Automobile Manufacturers Association Inc., Web page of Japan Automobile Manufacturers Association, <http://www.jama.or.jp/lib/jamagazine/199806/02.html> (In Japanese)
- [20] S. Frederico et. al., Silicon Layer Dry Etching for free-standing RF MEMS architectures, Proc. Of IEEE 16th Annual International Conference on MEMS, Kyoto, Japan, Jan. 2003, pp570-573.
- [21]http://industrial.panasonic.com/ww/products_e/product_cat2/ACAH000_e/ACAH000_e/select_e.html
- [22] TRONIC'S Microsystems - Micro Electro Mechanical Systems (MEMS) Facility, France, News, views and contacts from the semiconductor industry, <http://www.semiconductor-technology.com/projects/tronic/>

Implementation of an Intelligent Robotized GMAW Welding Cell, Part 1: Design and Simulation

I. Davila-Rios, I. Lopez-Juarez¹, Luis Martinez-Martinez
and L. M. Torres-Treviño²

Corporacion Mexicana de Investigacion en Materiales (COMIMSA)

¹*Centro de Investigacion y de Estudios Avanzados del IPN (CINVESTAV)*

²*Universidad Autonoma de Nuevo Leon (UANL)*

Mexico

1. Introduction

Today, welding tasks can be accomplished in two ways: 1) Manual or conventional, which is made by humans with a welding gun; and 2) Automated welding that is carried out by robots, this robots are called "Welding Robots". Current welding robots are able to perform welding tasks continuously under different working conditions in low-scale production such as shipbuilding or in high-scale production such as in the automotive industry. Even in well defined and structured environments such as in the automotive industry robot reprogramming is still necessary in order to cope with uncertainties. This additional task involves hiring specialized personnel, lost of production time, quality assessment, destructive testing, etc., which necessarily increases the production costs.

The design proposal considers first to simulate the whole welding process considering issues like floor plant space, robot configuration, trajectory planning, welding equipment and supplies, etc. and secondly, the utilization of novel teaching tools for welding trajectories. The contribution has been divided in two chapters as follows: In Part I, the robotic cell set up (including off-line and on-line programming) using current 3D software simulation, equipment commissioning and testing, distributed workcell communication and the voice-command program design are presented; while in Part II, the description of an error recovery strategy in conjunction with a novel teaching trajectory using machine vision is presented.

In this chapter, we describe the design to integrate a robotic welding cell within a 3D simulation environment. The design was used as a reference for the actual construction of the robotized GMAW welding cell. We report a methodology to set up a welding robot cell from scratch using a 3D software robot simulation from Delmia named Robotics V5®. The simulation included welding accessories that were designed first in CAD software and imported by the Delmia® software. These accessories were gas tanks, wire coils, Lincoln 455

Power wave station, 10R automated Lincoln wire feeder, Magnum 400 robotic torch, welding part conveyor and an industrial KUKA KR16 industrial robot.

The methodology included layout definition, welding part design, robot and welding station commissioning, off-line programming including collision avoidance reports during simulations. The whole setting was completed by simulation and different layout schemes were tested. The design also considered a voice-command driven environment, so that the robot cell could be commanded via voice. The voice-command software was developed in C++ using the Microsoft Speech Application Interface (SAPI V5.0). This interface was also simulated off-line using two computers first with wireless communications and later, including the robot.

The organisation of the chapter is as follows. In section 2, the GMAW welding process is reviewed. In section 3, different aspects of the robot welding system are considered including the design stages of the welding workcell from simulation to implementation. Section 4, describes the robot simulation as well as an assessment of collision failures. Section 5 presents a robot program example using off-line programming. In section 6, the design of the distributed robotic workcell is explained as well as some issues regarding distributed systems using CORBA. The voice-command application and grammar description are presented in section 7. Finally, section 8 provides the conclusions and further work.

2. Gas Metal Arc Welding Process

In Gas Metal Arc Welding (GMAW) process an electric arc is established between a consumable electrode fed continuously to the weld pool and the work-piece. When this process starts, the weld pool is shielded by an inert gas, giving the process the popular designation of Metal Inert Gas (MIG). Nowadays active gases such as carbon dioxide or mixtures of inert and active gases are also used and the designation GMAW includes all these cases. This process is widely used in industrial application due to its numerous benefits. It can weld almost all metallic materials, in a large range of thicknesses (above 1 mm and up to 30 mm or more) and is effective in all positions. GMAW is a very economic process because it has higher deposition rates than for example the manual metal arc process, and does not require frequent stops to change electrodes, as is the case of this former process. Less operator skill is required than for other conventional processes because electrode wire is fed automatically (semi-automatic process) and a self-adjustment mechanism maintains the arc length approximately constant even when the distance weld torch to work-piece varies within certain limits. These advantage make the process very well adapted to be automated and particularly to robotic welding applications. The process is sensitive to the effects of wind, which can disperse the shielding gas, and it is difficult to use in narrow spaces due to the torch size (Holliday, D B 2005).

2.1 Welding equipment

Basic equipment for conventional GMAW consist of the power source, the electrode feed unit, the welding torch and the shielding gas regulator, as represented schematically in Figure 1.

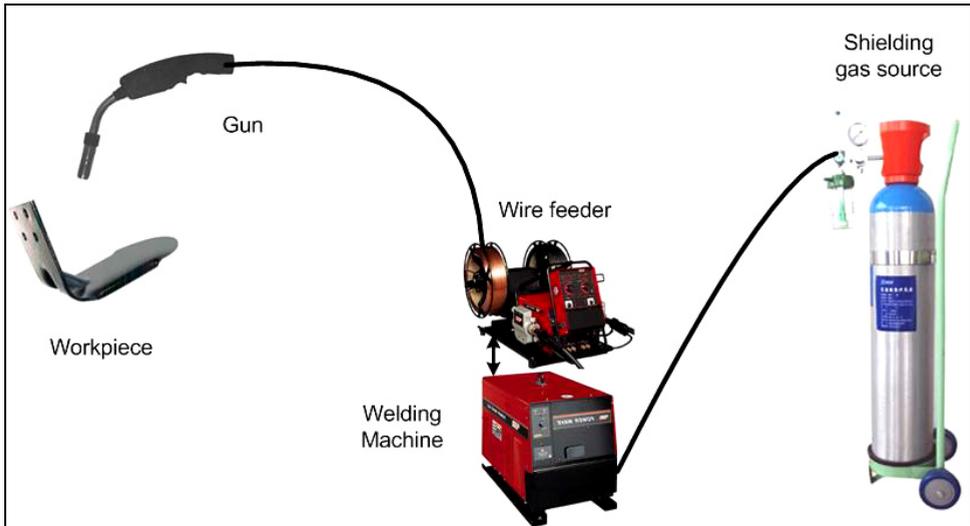


Fig. 1. Schematic representation of gas metal arc welding process (GMAW)

2.1.1 Power Source

Most common GMAW power sources are of the inverter type providing a constant-voltage output. A constant-voltage power source used in conjunction with a constant speed wire feeder can provide self-adjustment and stabilization of the arc length, in order to compensate for the variation in the torch to work-piece distance that occur mainly during manual welding operation. In addition these machines provide slope control of the power source characteristics and of the inductance in order to control spatter in short-circuiting transfer (Norrish, J. 1992).

2.1.2 Electrode Feed Unit

The electrode feed unit and the welding control mechanism are generally furnished in one integrate package. The electrode feed unit pulls the electrode from the reel and pushes it through a conduit to the welding torch (gun). This unit is composed of a direct-current motor, that varies the motor speed over a large range, a gear box and two pairs of rolls with a pressure adjusting screw and wire guides, that transmit mechanical energy, straighten and guide the electrode. The welding control mechanism regulates not only the electrode feed speed and the start and stop of the electrode but also the delivery of shielding gas, current and cooling water (when necessary) to the torch. When the torch cable is externally attached to the robot arm it is exposed to work-piece interface and to premature wear. Modern robotic systems can include special arms with internal cabling, in order to prevent interference, increasing cable life.

2.1.3 Welding Torch

Main functions of the welding torch are to furnish the electrode with electrical current and direct the electrode and gas flow to the work-piece. Main components of the welding torch

are the contact tube, where the current is transmitted to the electrode, the nozzle, which provides a laminar gas flow to the weld pool, the torch switch, which sends signals to the feed unit, and the handle. The handle supports the gas and water (if necessary) tubes, the electrode guide tube and cables for current and signals. MIG torches for low current and light duty cycle (up to 60%) are gas cooled. Robotic torches are in general water cooled, but if gas cooled torches are used they must be larger than manual torches. Robotic torches usually have emergency-stop capability to prevent damage to the robot arm and the welding torch is in the event of a collision. They are also provided with automatic cleaning, that may include a pressurized air system for blowing spatter out of the nozzle (Hancock & Johnsen, 2004).

2.2 Process Parameters

Welding parameters affect the way the electrode is transferred to the work-piece, the arc stability, spatter generation, weld bead geometry and overall weld quality. The main parameters of the process are current, voltage, travel speed, electrode extension and electrode diameter, though others, such as electrode orientation, electrode composition and shielding gas, also have direct influence on the metal transfer mechanisms. These parameters are not independent. The current and voltage, for example, are correlated by the arc characteristic curves. Voltage depends not only of the arc length but also on the electrode extension and on the shielding gas.

2.2.1 Current

Direct current electrode positive (DCEP) is the most used current in GMAW because it gives stable electric arc, low spatter, good weld bead geometry and the greatest penetration depth. For low currents and voltages in combination with active shielding gases or mixture containing active gases, dip or short-circuiting transfer is obtained. The utilization of relatively low current can give insufficient penetration and excessive weld reinforcement, occasioned by poor wetting action of the weld metal. Globular repelled transfer can be found when electrode negative polarity is used with solid wire, but this mode of transfer has no industrial application due to poor stability and high spatter levels.

2.2.2 Voltage

Arc voltage is directly related to current, as indicated above, and with arc length, increasing with it. Voltage also depends on the shielding gas and electrode extension. The increase of arc voltage widens and flattens the weld bead. Low voltages increase the weld reinforcement and excessively high voltages can cause arc instability, spatter, and porosity and even undercut.

2.2.3 Welding Speed

Increase in the welding speed gives a decrease in the linear heat input to the work-piece and the filler metal deposition rate per unit of length. The initial increase in welding speed can cause some increase in penetration depth, because the arc acts more directly in the parent material, but further increase in speed decreases penetration and can cause undercut, due to insufficient material to fill the cavity produced by the arc.

2.2.4 Electrode Extension

The electrode extension is the electrode length that is out of the contact tube. The increase of electrode extension, produced by the increase of the torch distance to the work-piece for a specific parameters set, increases electrode melting rate because of the Joule effect. Electrode extension ranges from 5 to 15 mm for dip transfer, being higher (up to 25 mm) for the other transfer modes.

2.2.5 Shielding Gas

Shielding gases have an effect on arc stability, metal transfer mode, weld bead shape and melting rate. Gases used in GMAW can be pure gases, binary, ternary and exceptionally quaternary mixtures. Common pure gases are argon, helium and carbon dioxide. The first two are inert gases and are used principally in welding of light alloys, nickel, copper and reactive materials. Helium has a higher ionization potential than argon, providing larger weld pools, but is more expensive. Carbon dioxide is an active gas and is used in welding of carbon steel. It produces high levels of spatter but provides high penetration depth. Binary mixture are commonly argon/carbon dioxide (up to 20% CO₂), argon/oxygen (up to 5% O₂) and argon/helium (up to 75% He). The first is used in the welding of carbon and low alloy steels, the second of stainless steels and the third of nonferrous materials. The addition of oxygen or carbon dioxide to argon stabilizes the welding arc and changes the bead shape. The objective of adding helium to argon is to increase heat input and consequently welding speed, but also to reduce the incidence of weld porosity (Lyttle, K. A. 2002).

2.2.6 Health and Safety

The major potential hazards of arc welding processes are the high-voltage electricity, which can injure and kill personnel, the fumes and gases, which can be dangerous to health, the electric arc radiation, which can injure eyes and burn skin and the noise that can damage hearing. The exposure to the high open - circuit voltage of power supplies can cause dangerous electric shocks, which can be prevented by connecting all the electrical equipment and work - pieces to a suitable electrical ground. All electric cables should be suited to the maximum current and must remain insulated and dry.

Fumes and gases are generated in all arc welding processes, being particularly intense in the flux cored arc welding process. Metal fumes of nickel, chromium, zinc, lead or cadmium, for example, and gases such as carbon monoxide, ozone and nitrogen oxides formed in the arc are very harmful to the health. Enough ventilation or exhaust at the arc, or both must be used in order to keep fumes and gases from the personnel breathing zone.

3. Robotic Welding: System Issues

Robotic welding research deals with the relevant technical and scientific aspects involved in the task of reproducing the work of the experienced and skilled human welder. Welding was for a long time a task performed only by humans, being a craft that combines skills with art and science. Automating welding is therefore a very difficult and demanding objective, because of the required adaptive behaviour of the automatic system. Therefore, for the design of the automated system, a simulator tool is highly desirable in order to consider all

the aspects before the actual construction of the cell. In this section, we present the simulation design using Delmia Robotics V5 and wherever needed the design of the required parts using Solid Works® is explained.

3.1 Environment in 3D Simulator

The simulation is used for the identification of design problems in an early phase, for an iterative design process from design to the finished product. The use of dynamic 3D simulations constitutes an advanced methodology for the analysis of the risks associated to the installation of the systems and solutions in robotics. The advantages of these techniques become evident when facilitating, among other things, the determination of impact risks (collisions) of the robot's part in the tasks during assembly, ergonomics of the position, demarcation of the environment of the robot's work, etc.

Delmia Robotics V5 allows to be carried out simulations of automated manufacturing processes, specifically with robots; this software has a great variety of robots models, as well as of different brands. Delmia Robotics V5 offers an easy and flexible use of solutions for the definition of tools and simulation cells. It provides all the necessary tools to define and to analyze the behaviours and the necessary resources to apply the process plan. The software helps to describe exactly how parts are loaded and unloaded, fixtures, and selection of welding tools. The analysis and validation of the process can be carried out easily and accurately through simulations based on the resources available and have the option to incorporate additional tooling or fixtures (Caie, J. 2008).

The Robotics V5® environment is composed for a great variety of tools called “toolbars”, which are presented as icons around the main screen, inside it; they are two very important functions, the compass and the tree PPR (Product, Process and Resource). The compass is used to change position and to move the elements to use, and the tree PPR is used to show the products, resources and how it is carrying out the process inside the simulator, as well as to take a relation among them because they work together. Figure 2 shows the main screenshot when selecting the robot type to be used.



Fig. 2. Main screenshot of the Robotics V5

For the design of specific components, there are many types of software, in our case Solid Works® was used, due to its compatibility with Robotics V5 when exporting the parts.

3.2 Virtual Development of a Robotic Workcell

The first step in our design was to identify the physical location of the welding cell as well as its area dimensions in the shop floor. An important aspect to consider is the floor and environment; the floor requires withstanding the weight of the various devices and machines and the environment have to be dust-free to guarantee appropriate working condition for the electronic devices.

For the workcell simulation, some of the components that are not included in the libraries had to be designed separately. For instance, the welding table, the conveyor belt, the KRC2 robot controller, the physical structure of the cell (walls, windows), etc. These components were designed using Solid Works® and exported to Robotics V5®. A very important issue at this stage is that the design of the components must be carefully made to scale. Finally, but not the least, it is recommended to consider the security of the cell considering emergency exits, placement of fire extinguishers, fume vents and emergency kit location. A screenshot of the cell layout is shown in Figure 3.

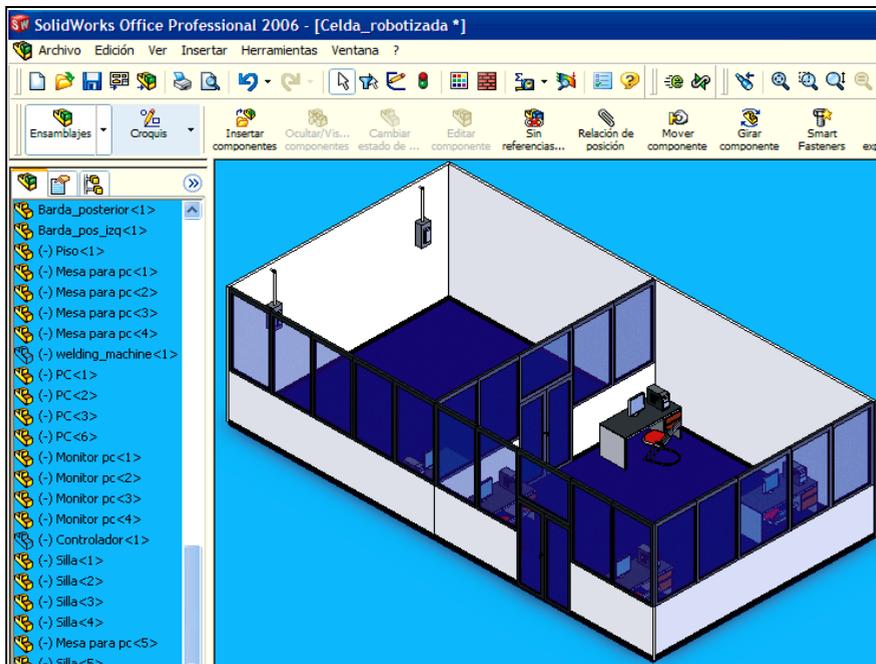


Fig. 3. Workcell in Solid Works® environment

It is also possible to design parts using CATIA® software. This software can be used directly in the simulator Robotics 3D without having to export the designs. They only have to be saved using a CATPart extension. The first step to design the part is to open an application

in Delmia called “Mechanical Design”, and then in the option of part design choice open a new project in CatProduct mode. An example of a piece to be welded it is shown in Figure 4.

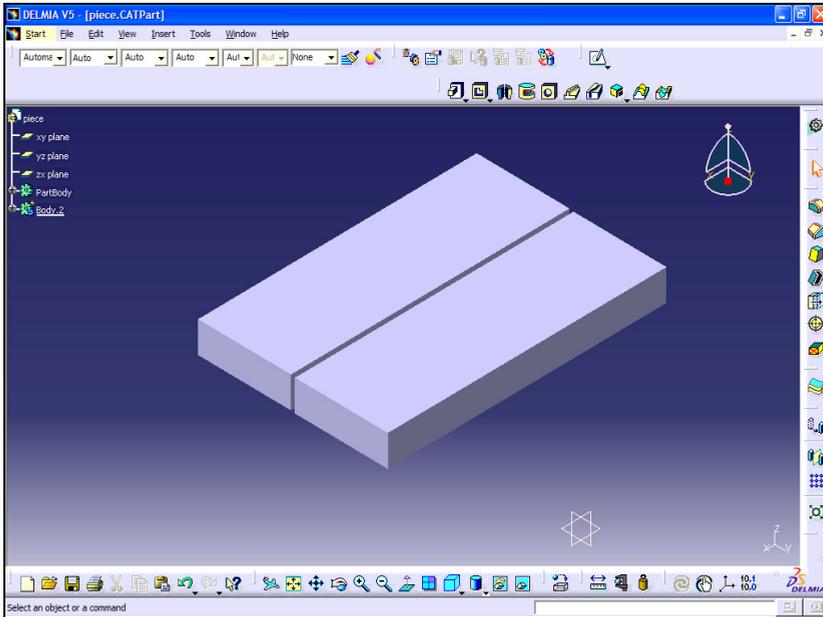


Fig. 4. Piece to be welded in CATIA environment

3.3 Layout Construction in the Robotics V5

Industrial robots generally work with others devices such as conveyor belts, production machines, holding fixtures and tools. The robot and the equipment form a workcell. Also you can use the term work station, but this term is generally used to identify 1) One workcell with only one robot or 2) a work position along of a line production of some station of robots work. Two problems in applications of robots engineering are, the physical design of the work cell and the design of the control system that it will coordinate the activities among the diverse components of the workcell (Groover M. et al, 1995).

For the construction of the layout in the simulation software Robotics V5, is it necessary to open a document in “process” mode in order to be able to work in the simulation environment and to be able to open the “PPR tree”. It is necessary to program in the beginning mode and simulation resource the option of “Device task definition”. To create the layout in the first screen, it is necessary to insert products and resources. Inserting products and resources will depend on the way they are used in the process. Products such as *Body Side Assembly Part*, *Robot Kuka KR16*, *welding table*, *torch*, *workcell* and a *riser* (platform for the Robot) need to be selected from the PPR screen and then attached together. To mount the welding torch on the robot, the toolbar “Robot management>Robot dress up” needs to be used as it is illustrated in Figure 5. DELMIA can import files with .cgr extension from Solid Works (EES 2006).

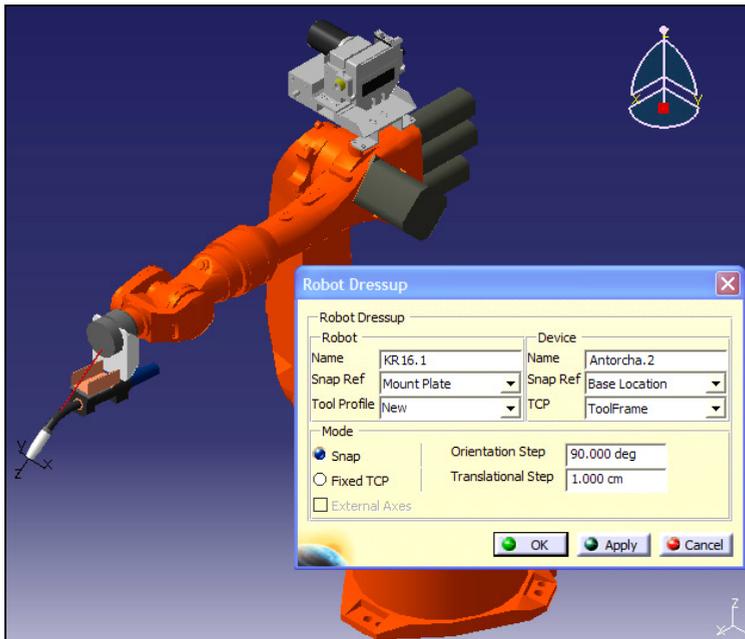


Fig. 5. Mounted torch and power feed to the Robot Kuka KR16

Finally, after considering all accessories and equipment needed in the welding cell, the final layout was completed and built as shown in Figure 6(a) and 6(b), respectively. The lay out construction is presented in Figure 6.

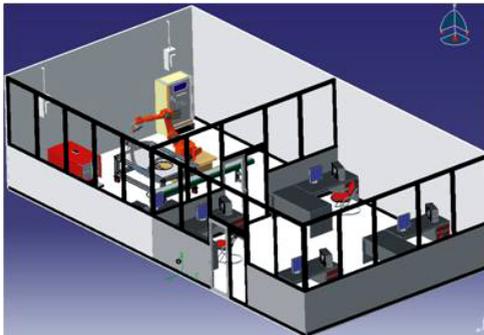


Fig. 6(a). Lay out of the workcell.



Fig. 6(b). Finished workcell.

4. Robot Welding Simulation

The second part of the research involved the simulation of the robot welding in order to analyse the robot's working envelope to prevent any crash or collision with the environment. This is an important off-line stage since the user can learn how to move the

robot and program diverse tasks and later, if needed, the robot program can be translated into another robot's manufacturer language. This is also a useful feature that could be used for learning how to program certain robot.

In the simulator, it is necessary to create *tags* and robot tasks associated to them. *Tags* are those points for which the robot will carry out the welding operations. To complete the programming process, the robot's motion has to be adjusted with the help of the Teach pendant and moving the compass to determine the robot movement. Alternatively, a function inside the Teach Pendant called *Jog*, that manipulates the robot's movement through each one of the DOF, can be used (EES 2006).

4.1 Simulation Example

First of all, it is important to maintain intact the positions of the components within the workcell at the beginning of each simulation, an initial state has to be selected and a robot type. In our case we worked with the KUKA KR16 Industrial Robot. A snapshot of the simulation is shown in figure 7(a) and in figure 7(b) the actual workcell is shown.

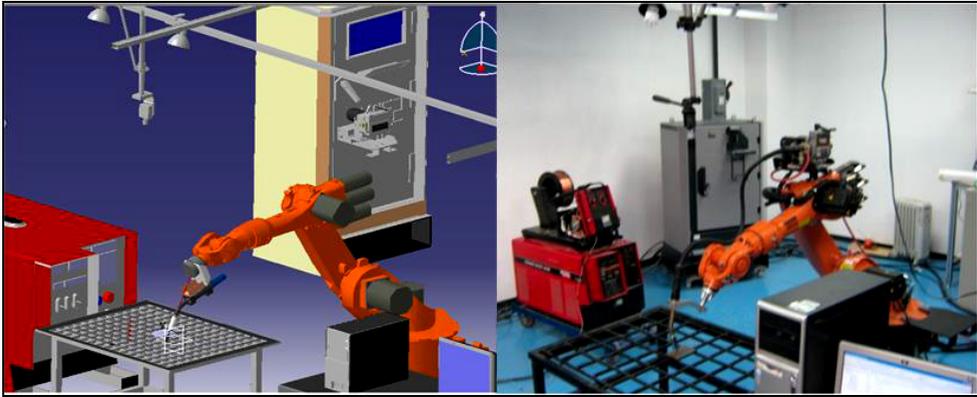


Fig. 7(a). Simulation welding process.

Fig. 7(b). Real workcell

Despite the cell distribution and robot location and configuration, it is possible to make contact and collision with other components. The simulator provides a Matrix of contacts and clashes in order to reprogram the trajectories if necessary. Figure 8 shows the results of the clashes with high relevance due to detected collisions between the torch and the work table and also small contacts between robot articulations.

In Figure 8, it is also shown on the left hand side a graph of the area affected by the collision between the robot and the torch with one product named "piece to be welded". On the right hand side, it shows the relation matrix of collisions and contacts, among all the devices. In this example, the simulation resulted in a total of 19 interferences, from which 9 were collisions and 10 contacts. In these results, the collisions are important since they can damage some devices and even the robot.

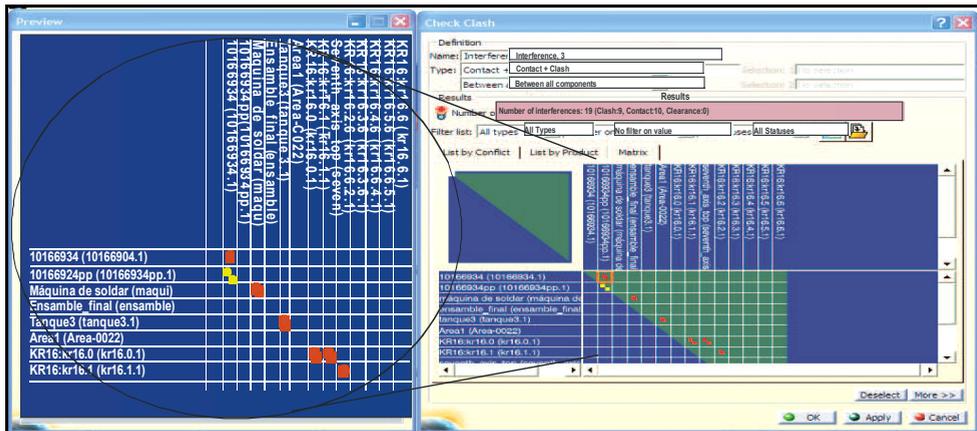


Fig. 8. Matrix simulation results

A human welder must be able to interpret all the results of this simulator. The examination of the Standard CRAW-OT (Certified Robotic Arc Welding, Operator and Technician) by the AWS is designed to test the knowledge of welding fundamentals and robotic welding (EES 2006). Table 1 presents the CRAW-OT subjects that can be evaluated in the simulator in comparison with a real workcell.

Subject	Real Workcell	Virtual Simulator
Weld Equipment Setup	✓	
Welding Processes	✓	✓
Weld Examination	✓	
Definitions and Terminology	✓	✓
Symbols	✓	
Safety	✓	✓
Destructive Testing	✓	
Conversion and Calculation	✓	✓
Robot Programming	✓	✓
Welding Procedures	✓	✓
Programming Logic	✓	✓
Kinematics Concepts	✓	✓
Robot Arc Weld Cell Components	✓	✓

Table 1. Comparison between Real Workcell and Virtual Simulator

5. Robot Program Using Off-Line Programming

One of the most important tasks is programming the robot, which is considered very difficult to evaluate because each brand has its own robot programming code. This programming task is easier to evaluate since the simulator provides a conversion utility

between different commercial robots. A code example produced by the simulator is provided in Table 2.

Robot KUKA	Robot FANUC	Robot RAPID
<pre> &ACCESS RVP &REL 1 &PARAM EDITMASK = * DEF Welding_Task() ;FOLD PTP ViaPoint2 Vel= 50 % PDAT1 Tool[1]:cell12- torch.1. ToolFrame Base[0];%{PE} %R 4.1.5,%MKUKATPBASIS, %CMOVE,%VPTP,%P 1:PTP, 2:ViaPoint2, 3:, 5:50, 7:PDAT1 \$BWDSTART=FALSE PDAT_ACT=PPDAT1 BAS(#PTP_DAT) ;ENDFOLD END </pre>	<pre> /PROG Welding_Task /ATTR OWNER = MNEDITOR; PROG_SIZE = 0; FILE_NAME = ; VERSION = 0; LINE_COUNT = 0; MEMORY_SIZE = 0; PROTECT = READ_WRITE; TCD: STACK_SIZE = 0, TASK_PRIORITY = 50, TIME_SLICE = 0, BUSY_LAMP_OFF = 0, ABORT_REQUEST = 0, PAUSE_REQUEST = 0; DEFAULT_GROUP = 1,1,1,*,*; /POS P[1]{ GP1: UF : 1, UT : 2, CONFIG: 'S 2 , 0, 0, 0', X = 1157.208 mm, Y = 212.886 mm, Z = 1002.855 mm, W = 136.239 deg, P = 11.951 deg, R = 103.725 deg }; </pre>	<pre> %% VERSION:1 LANGUAGE:ENGLISH %% MODULE Welding_Task_mod PERS robtarget ViaPoint2:= [[1157.208,212.886,1002.8551, 0,0,0 PERS robtarget Tag1:=[[938.236,285.366,460.4 29], [0.111893,0.51089,0.851944,0. 025754] ,[,0,0,0],[9E+09,9E+09,9E+09, 9E+09,9E+09,9E+09]]; PERS robtarget PROC Welding_Task() MoveJ ViaPoint2,Default,Default,cel l2-torch_1_ToolFrame; MoveJ Tag1,Default,Default,cell12- torch_1_ToolFrame; ENDPROC ENDMODULE </pre>

Table 2. Program code for three different robots

The simulation process produces a robot program like the one shown in Table 2. This program can be loaded to the specific robot controller to perform a real welding cycle using the specified robot. This program is ready to start a welding task which can be programmed either by using a predefined routine (off-line) or via a teaching device (on-line) (Lopez-Juarez, I et al. 2009). In both cases a friendly user interface via voice has been designed in order to facilitate robot programming which is described in section 7.

6. Robotized Welding System

The welding system used for experimentation is integrated by a KUKA KR16 industrial robot. It also comprises a visual servo system with a ceiling mounted Basler A602fc CCD camera as it is shown in figure 9.

Two computers are used, the Master Controller and the Speech Recognition. The Master Controller is in charge of low-level serial communication with the robot controller using the 3964a protocol. It also connects to the Lincoln 455M power source and 10R wire feeder using an I/O Data Acquisition Card so that the welding process can be switched on-off and the current and voltage can be controlled by this computer. Additionally, it also handles the programming user-interface through a wireless gamepad. On the other hand, the Speech Recognition computer is in charge of giving voice commands to the robot in order to carry out the welding tasks.

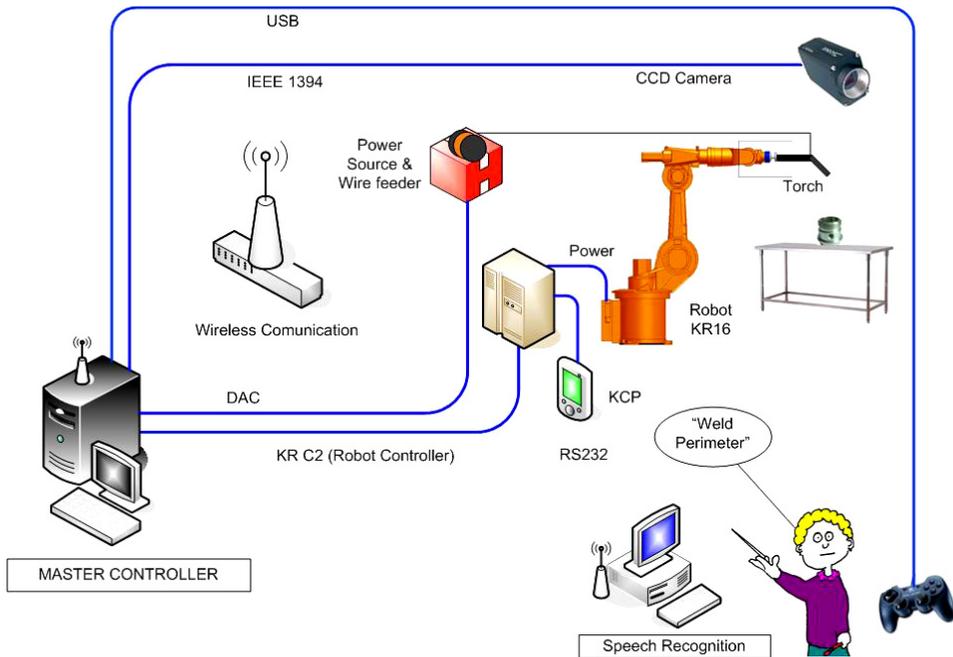


Fig. 9. Robotized Welding System

6.1 Distributed Robotic System

The concept of distributed systems and other technologies recently have made possible the creation of new application called "Networked Robot Systems". The main idea is to solve the heterogeneity problem found in robotic systems due to the multiple component vendors and computational platforms.

The development of robot systems based on distributed components is well supported by different researchers. In (Amoretti et al., 2003), Michael Amoretti et al., present an analysis of three techniques for sensor data distribution through the network. In (Amoretti, 2004) it is proposed a robotic system using CORBA as communication architecture and it is determined several new classes of telerobotic applications, such as virtual laboratories, remote maintenance, etc. which leads to the distributed computation and the increase of new developments like teleoperation of robots. In (Bottazzi et al., 2002), it is described a software development of a distributed robotic system, using CORBA as middleware. The system permits the development of Client-Server application with multi thread supporting concurrent actions. The system is implemented in a laboratory using a manipulator robot and two cameras, commanded by several users. In (Dalton et al., 2002), several middleware are analyzed, CORBA, RMI (Remote Method Invocation) and MOM (Message Oriented Middleware). But they created their own protocol based on MOM for controlling a robot using Internet. In (Lopez-Juarez & Rios-Cabrera, 2006) a CORBA-based architecture for robotic assembly using Artificial Neural Networks was introduced.

In the current investigation, though the system only includes two computers using the same OS, the master controller and the speech recognition. It is important in this early stage to

consider the overall layout considering that additional components are being included in the network.

6.1.1 CORBA specification and terminology

The CORBA specification (Henning, 2002), (OMG, 2000) is developed by the OMG (Object Management Group), where it is specified a set of flexible abstractions and specific necessary services to give a solution to a problem associated to a distributed environment. The independence of CORBA for the programming language, the operating system and the network protocols, makes it suitable for the development of new application and for its integration into distributed systems already developed.

It is necessary to understand the CORBA terminology, which is listed below:

- A *CORBA object* is a virtual entity, found by an ORB (Object Request Broker, which is an ID string for each server) and it accepts petitions from the clients.
- A *destine object* in the context of a CORBA petition, it is the CORBA object to which the petition is made.
- A *client* is an entity which makes a petition to a CORBA object.
- A *server* is an application in which one or more CORBA objects run.
- A *petition* is an operation invocation to a CORBA object, made by a client.
- An *object reference* is a program used for identification, localization and direction assignment of a CORBA object.
- A *server* is an entity of the programming language that implements one or more CORBA objects.

The petitions are showed in the figure 10: it is created by the client, goes through the ORB and arrives to the server application.

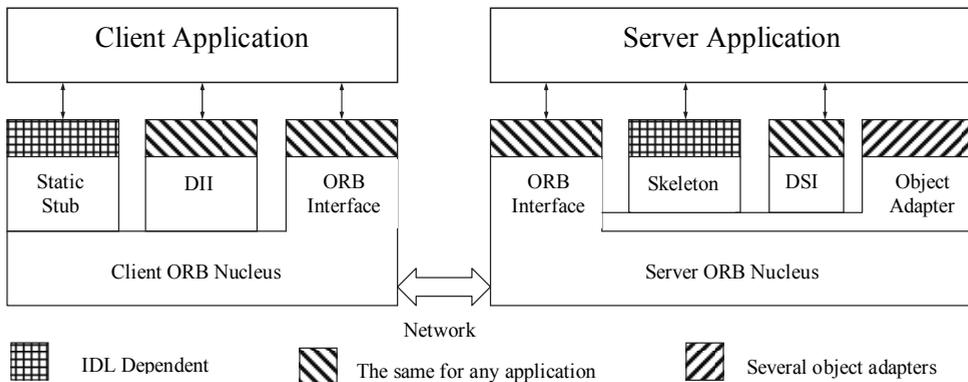


Fig. 10. Common Object Request Broker Architecture (CORBA)

The client makes the petitions using static stub or using DII (Dynamic Invocation Interface). In any case the client sends its petitions to the ORB nucleus linked with its processes. The ORB of the client transmits its petitions to the ORB linked with a server application. The ORB of the server redirect the petition to the object adapter just created, to the final object.

The object adapter directs its petition to the server which is implemented in the final object. Both the client and the server can use static skeletons or the DSI (Dynamic Skeleton Interface). The server sends the answer to the client application.

In order to make a petition and to get an answer, it is necessary to have the next CORBA components:

Interface Definition Language (IDL): It defines the interfaces among the programs and is independent of the programming language.

Language Mapping: it specifies how to translate the IDL to the different programming languages.

Object Adapter: it is an object that makes transparent calling to other objects.

Protocol Inter-ORB: it is an architecture used for the interoperability among different ORBs. The characteristics of the petitions invocation are: transparency in localization, transparency of the server, language independence, implementation, architecture, operating system, protocol and transport protocol. (Henning, 2002).

The aim of having a Master Controller, is to generate a high level central task controller which uses its available senses (vision and voice commands) to make decisions, acquiring the data on real-time and distributing the tasks for the welding task operation.

The architecture of the distributed system uses a Client/Server in each module. Figure 11 shows the relationship client-server in the Master Controller and Speech Recognition. With the current configuration, it is possible a relationship from any other future server to any client, since they share the same network. It is only necessary to know the name of the server and obtain the IOR (Interoperable Object Reference). The interfaces or IDL components would need to establish the relations among the modules.

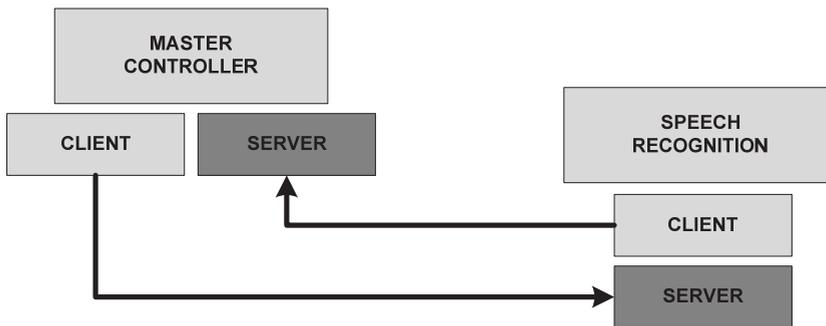


Fig. 11. Client/server architecture of the distributed cell

7. Robot Controller Using Voice-Command Software

The system provides a user interface to receive directions in natural language using natural language processing and Context Free Grammars (CFG). After the instruction is given, a code is generated to execute ordered sentences to the welding system. To effectively communicate the robot controller, it was needed to work in *speech recognition (speech-to-text)* as well as in *speech synthesis* to acknowledge the command (*text-to-speech*). Using these features it is possible to instruct the robot via voice-command and to receive an

acknowledgement when tasks such as the weld perimeter, weld trajectory, stop, start, go-home, etc. are accomplished.

The Voice Interface was based on Windows XP SP3 operating system using a Speech Recognition PC (§ see section 6). The implementation of the voice command software for the robotic welding system was developed in C++ using the Microsoft Software Development Kit (SDK) and the Speech Application Programming Interface (SAPI) 5.0, which is a programming standard that provides tools and components to speech recognition and speech synthesis.

The SAPI is a high-level interface between the application and the speech engine that implements low-level details to control and to manipulate the real-time operation in several speech engines. There are two basic SAPI engines as it is shown in figure 12. One is the text-to-speech system that synthesis strings and files into spoken audio signals using predefined voices. On the other hand, the speech recognition engine or speech-to text converts the human spoken voice into text strings and readable files. The SAPI is middleware that provides an API and a device driver interface (DDI) for speech engines to implement. The managed System.Speech namespace communicates to these engines both directly and indirectly by calling through the SAPI.DLL. Native

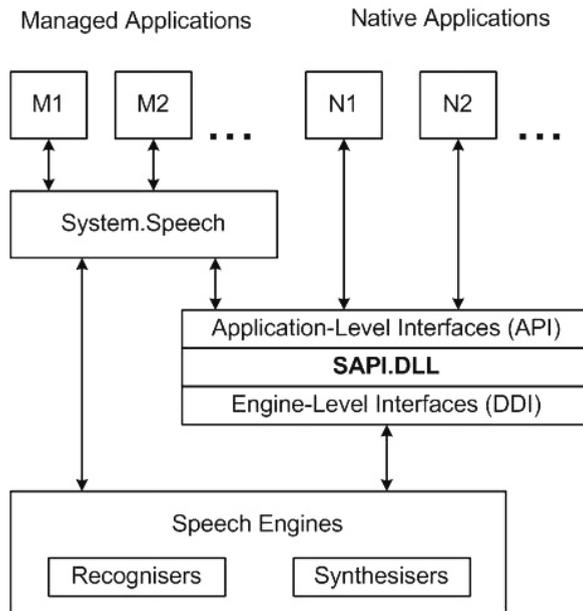


Fig. 13. SAPI engines

There are several category interfaces apart from the speech engines that were used in the application:

Audio

Used to control and customize real-time audio streams compatible with speech synthesis.

Grammar Compiler

Used to dynamically define Context-Free Grammars (CFGs) and compile them into the binary form used by the speech recognition engine.

Lexicon

Provides a uniform way for applications and engines to access the user lexicon, application lexicon, and engine private lexicons. Lexicons provide custom word pronunciations for speech synthesis.

7.1 Grammar

The Context Free Grammar (CFG) format in SAPI 5 defines the structure of grammars and grammar rules using Extensible Markup Language (XML). The CFG/Grammar compiler transforms the XML tags defining the grammar elements into a binary format used by SAPI 5-compliant SR engines. This compiling process can be performed either before or during application run time.

The Speech SDK includes a grammar compiler, which can be used to author text grammars, compile text grammars into the SAPI 5 binary format, and perform basic testing before integration into an application. An example of the developed code is as follows:

```
<GRAMMAR LANGID="409">
<RULE NAME="RobotTask" ID="139" TOPLEVEL="ACTIVE">
<OPT>
<L>
<P>robot</P>
<P>stop</P>
<P>hello</P>
</L>
</OPT>
<OPT>
<L>
<P>weld</P>
<P>go</P>
<P>kuka</P>
</L>
</OPT>
<OPT>
<L>
<P>weld</P>
<P>objects</P>
<P>home</P>
</L>
</OPT>
</RULE>
</GRAMMAR>
<GRAMMAR LANGID="409">
</GRAMMAR>
```

In the program, <GRAMMAR> has a numeric attribute LANGID. We can observe at the beginning of the program, there is also a RULE NAME where "RobotTask" is the

grammatical rule; ID, the language identification and TOPLEVEL is declared ACTIVE, but it can be dynamically configured in real-time. The user has to talk only TOPLEVEL rules for the robot to recognise the words. For instance, in the program the words *robot*, *stop*, *hello*, can be recognized by the engine. Note that these words are enclosed by <OPT> and </OPT> directives.

Several words were included in the Lexicon being the more important: *weld perimeter*, *weld trajectory*, *stop*, *start*, *go-home*.

8. Conclusions and Ongoing Work

In this chapter, we described the design and integration of a robotic welding cell using a 3D simulation environment. The design was useful for building the CORBA-based distributed robotized welding cell in this research project. Issues such as layout definition, communication design, welding part design, robot and welding station commissioning were considered. The design also included a voice-command driven environment using the Microsoft Speech Application Interface V5.0. Definition of Context Free Grammars were used so that it was possible to start a typical robot task using a human operator's voice using verbal commands such as "weld perimeter" or "weld trajectory".

The design and simulation previous to the implementation of an automated welding cell is useful, because possible errors can be prevented such as problems of area distribution, security, dimensions, etc. In addition to its great utility to save costs and avoid unnecessary damage to machinery and equipment.

The design of complex robot systems using multisensorial inputs, high-level machine interfaces and distributed networked systems will be elements of primary importance for advance robot manipulators in the near future so that the work reported in this chapter intends to demonstrate alternative guidelines to design such complex systems.

9. Acknowledgements

The authors wish to thank the following organizations who made possible this research: The Consejo Nacional de Ciencia y Tecnologia (CONACyT) through Project Research Grant No. 61373, and for sponsoring Mr. Davila-Rios during his doctoral studies and to the Corporacion Mexicana de Investigacion en Materiales for its support through Research Grant Project No. GDH - IE - 2007.

10. References

- Amoretti Michele; Stefano Bottazzi; Monica Reggiani & Stefano Caselli. (2003). "Evaluation of Data Distribution Techniques in a CORBA-based Telerobotic System" *Proceedings of the 2003 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2003)*, October, Las Vegas, NV.
- Amoretti, Michele, Stefano Bottazzi, Stefano Caselli, Monica Reggiani, (2004), "Telerobotic Systems Design based on Real-Time CORBA", *Journal of Robotic Systems* Volume 22, Issue 4, PP. 183 - 201.

- Barney Dalton, Ken Taylor, (2000). "Distributed Robotics over the Internet", *IEEE Robotics and Automation*. 7(2): 22-27.
- Bottazzi S., S. Caselli M. Reggiani & M. Amoretti, (2002). "A Software Framework based on Real-Time CORBA for Telerobotic Systems", *Proceedings of the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland, October, 2002.
- Holliday D. B., Gas-metal arc welding, (2005) *ASM Handbook*, Vol 6, Welding, Brazing and Soldering, 2005 pp (180-185).
- Henning, Michi, Steve Vinoski. (2002) "Programación Avanzada en CORBA con C++", *Addison Wesley*, ISBN 84-7829-048-6.
- I. Lopez-Juarez, R. Rios-Cabrera, & I. Davila-Rios (2009). Implementation of an Intelligent Robotized GMAW Welding Cell, Part 2: Intuitive visual programming tool for trajectory learning". In *Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria (In press).
- Norrish, J. (1992) Advanced welding processes, *Proceedings of the Institute of Physics Publishing*, 1992.
- Hancock, R. & Johnsen, M. (2004) Development in guns and torches, *Welding J* 2004, 83(5), pp29-32.
- Lyttle, K A, Shielding gases, *ASM Handbook*, Vol 6, Welding, Brazing and Soldering, pp. 64-69.
- Anibal Ollero Baturone, (2001). Robotic, manipulators and Mobile robots. *Alfaomega*.
- AWS: American Welding Society (2004). Certified Robotic Arc Welding Operator and Technician. Approved *American National Standard* ANSI.
- AWS: American Welding Society (2005). Specification for the Qualification of Robotic Arc Welding Personnel. Approved *American National Standard* ANSI.
- EES: Enterprise Engineering Solutions (2006). V5 Robotics Training Manual. Delmia Education Services Enterprise.
- Caie, Jim (2008). Discrete Manufacturers Driving Results with DELMIA V5 Automation Platform. ARC Advisory Group.
- Ericsson, Mikael, (2003). "Simulation of robotic TIG-welding". PhD Thesis, Division of Robotics Department of Mechanical Engineering *Lund Institute of Technology Lund University*, P.O. Box 118, SE-221 00 Lund, Sweden.
- Groover Mikell P., Weiss Mitchell, Nagel Roger & Odrey Nicholas. (1995). *Industrial Robotics*. McGraw-Hill, Inc., USA pp. 375-376.
- I. Lopez-Juarez, R Rios Cabrera. (2006) Distributed Architecture for Intelligent Robotic Assembly, Part I: Design and Multimodal Learning. In *Manufacturing the Future: Concepts, Technologies & Visions*. Edited by Vedran Kordic, Aleksandar Lazinica, Munir Medran. *Advanced Robotics Systems International*. Pro Literatur Verlag, Mammendorf, Germany. Pp. 337-366.

Implementation of an Intelligent Robotized GMAW Welding Cell, Part 2: Intuitive visual programming tool for trajectory learning

I. Lopez-Juarez¹, R. Rios-Cabrera¹ and I. Davila-Rios²

¹*Centro de Investigacion y de Estudios Avanzados del IPN*

²*Corporacion Mexicana de Investigacion en Materiales SA de CV
Mexico*

1. Introduction

Robotized GMAW welding is a demanding process. Current robots are able to perform welding tasks continuously under different working conditions in low-scale production such as shipbuilding or in high-scale production such as in the automotive industry. In well defined and structured environments such as in the automotive industry robot reprogramming is still necessary in order to cope with uncertainties. This additional task involves hiring specialized personnel, lost of production time, quality assessment, destructive testing, etc., which necessarily increases the production costs.

During the welding task, the joint part specification has to be met in order to meet the desired quality and productivity in industry. However, there are several factors that affect the process accuracy such as welding part positioning; motion errors in the production line, mechanical errors, backlash, ageing of mechanisms, etc. which are error sources that make robots to operate in uncertain conditions, i.e. unstructured environments. The scope of this work is focused on the compensation of these stochastic errors generated during the process and that the robot system needs to cope with in order to meet the required quality specification. To reach this goal, it is required to have an appropriate test-bed integrated with the process parameters sensing capacity (laser system, camera, proximity sensors, etc.) to follow the welding bead and to provide robust information to the robot controller. The use of multiple sensors and different computers make a centralised control very complex, hence it is preferred the use of the CORBA specification to implement a Distributed System. In this chapter we present the machine vision system and the distributed control for the welding cell as well as the Human-Machine Interface (HMI) developed to “teach” the manipulator any welding trajectory.

¹ This work was carried out during Dr. Lopez-Juarez research visit at Corporacion Mexicana de Investigacion en Materiales SA de CV (COMIMSA) under General and Specific CINVESTAV-COMIMSA Collaboration Agreements.

The robust design solution as proposed in this research is a two-fold issue. First, it is necessary to minimize design errors by simulating the whole welding process considering issues like floor plant space, robot configuration, welding equipment and supplies, etc. and second, the utilization of a novel teaching tool for welding trajectory. The contribution of this research has been split in two parts: In part I, the robotic cell set up (including off-line and on-line programming) using current 3D software simulation, voice command simulation design, equipment commissioning and testing was presented; whereas in this Part II, a novel robot programming tool for teaching welding trajectories and a built-in error compensation during production of welded parts are presented.

The programmed tool for trajectory learning is implemented in a Visual C++ application named *StickWeld V1.0* that involves the use of a friendly Graphical User Interface (GUI) for trajectory *compensation* and *teaching*. The software runs in a PC-based computer and uses a top mounted fast Firewire Colour Camera, a wireless gamepad and a pointing stick. The purpose of the software is to *compensate* on-line any error misalignment during perimeter welding of flat metal parts. The system compensates any offset error in the robot's welding torch due to conveyor or line production transport errors. The misalignment is captured by the camera and the image is processed in the server computer to find the new perimeter information, which is translated into a new robot trajectory and sent to the robot controller for execution. The *teaching* option can also be accessed via the GUI; by selecting this option the teaching/learning mode is activated. While in this mode, the user can define any welding trajectory using a stick as a pointer to define the trajectory. The trajectory input data, parameters selection and the robot motion control is made through the wireless gamepad controller so that the user has always full motion control on the robot assuring the safety within the cell. Once the new trajectory is entered, the robot can repeat the operation in another metal part. The system uses a three layer communication structure. The lowest layer is the serial standard communication RS232, followed by the SIEMENS 3964r protocol and at the top are the *StickWeld* commands that communicate the host PC master controller with the KUKA KRC2 robot controller. The defined trajectory path is stored continuously in a processing FIFO allocation in order to have a continuous interpolated motion at execution time.

The organisation of the chapter is as follows. This introduction belongs to section 1, which also includes the description of the distributed system and related work. In section 2, the GMAW welding process and different subsystems of the workcell are explained. In section 3, issues concerning with the server-robot communication protocol are provided. In section 4, the program *StickWeld V1.0*, GUI and the use of the peripherals and programming modes are described in detail. Finally, in section 5 conclusions are provided and the envisaged future work is highlighted.

1.1 Distributed System and Related Work

The CORBA specification (Henning, 2002), is developed by the OMG (Object Management Group), where it is specified a set of flexible abstractions and specific necessary services to give a solution to a problem associated to a distributed environment. The independence of CORBA for the programming language, the operating system and the network protocols, makes it suitable for the development of new application and for its integration into distributed systems already developed. In this investigation, it was decided to implement CORBA due to previous experience in Robotic Assembly (Lopez-Juarez & Rios-Cabrera,

2006). For a comprehensive description of the specification as well as its integration in the workcell, the reader is referred to (Davila-Rios, et al., 2009), where the distributed system is described in detail.

Jia proposes robotized systems using CORBA as the communication architecture in telerobotic applications like in virtual laboratories, remote maintenance, etc. (Jia, et al., 2002). Other authors look more at new paradigms rather than interoperability at the object level, but at the service level to facilitate the interoperability of industrial robots in the service environment and what has been called: Service Oriented Architectures (SOA) (Veiga, et al., 2007). Other authors use simple I/O devices like digitising pens to facilitate robot programming (Pires, et al., 2007). In our case, we have taken ideas from the mentioned authors and we have implemented a novel distributed programming tool to teach a robot random welding trajectories.

2. Welding process and Robotic System

Gas Metal Arc Welding (GMAW) is a welding process which joins metals by heating the metals to their melting point with an electric arc. The arc is between a continuous, consumable electrode wire and the metal being welded. The arc is shielded from contaminants in the atmosphere by a shielding gas.

GMAW can be done automatic by using an industrial robot manipulator as it is the case in this research and without the constant adjusting of controls by a welder or operator.

Basic equipment for a typical GMAW automatic setup are:

- Robot manipulator
- Welding power source: provides welding power.
- Wire feeders (constant speed and voltage-sensing): controls the supply of wire to welding gun.
- Constant speed feeder: used only with a constant voltage (CV) power source. This type of feeder has a control cable that will connect to the power source. The control cable supplies power to the feeder and allows the capability of remote voltage control with certain power source/feeder combinations. The wire feed speed (WFS) is set on the feeder and will always be constant for a given preset value.
- Voltage-sensing feeder: can be used with either a constant voltage (CV) or constant current (CC) - direct current (DC) power source. This type of feeder is powered by the arc voltage and does not have a control cord. When set to (CV), the feeder is similar to a constant speed feeder. When set to (CC), the wire feed speed depends on the voltage present. The feeder changes the wire feed speed as the voltage changes. A voltage sensing feeder does not have the capability of remote voltage control.
- Supply of electrode wire.
- Welding gun: delivers electrode wire and shielding gas to the weld puddle.
- Shielding gas cylinder: provides a supply of shielding gas to the arc.

When this process starts, the weld pool is shielded by an inert gas, giving the process the popular designation of Metal Inert Gas (MIG). Nowadays active gases such as carbon dioxide or mixtures of inert and active gases are also used and the designation GMAW includes all these cases. This process is widely used in industrial application due to its numerous benefits. It can weld almost all metallic materials, in a large range of thicknesses

(above 1 mm up to 30 mm or more) and is effective in all positions. GMAW is a very economic process because it has higher deposition rates than for example the manual metal arc process, and does not require frequent stops to change electrodes, as is the case of this former process. Less operator skill is required than for other conventional processes because electrode wire is fed automatically (semi-automatic process) and a self-adjustment mechanism maintains the arc length approximately constant even when the distance weld torch to work-piece varies within certain limits. These advantage make the process very well adapted to be automated and particularly to robotic welding applications. The process is sensitive to the effects of wind, which can disperse the shielding gas, and it is difficult to use in narrow spaces due to the torch size (Holliday, D B 2005).

2.1. Robotized Welding System

The welding system used for experimentation is integrated by a KUKA KR16 industrial robot. It also comprises a visual servo system with a ceiling mounted Basler A602fc CCD camera as it is shown in figure 1.

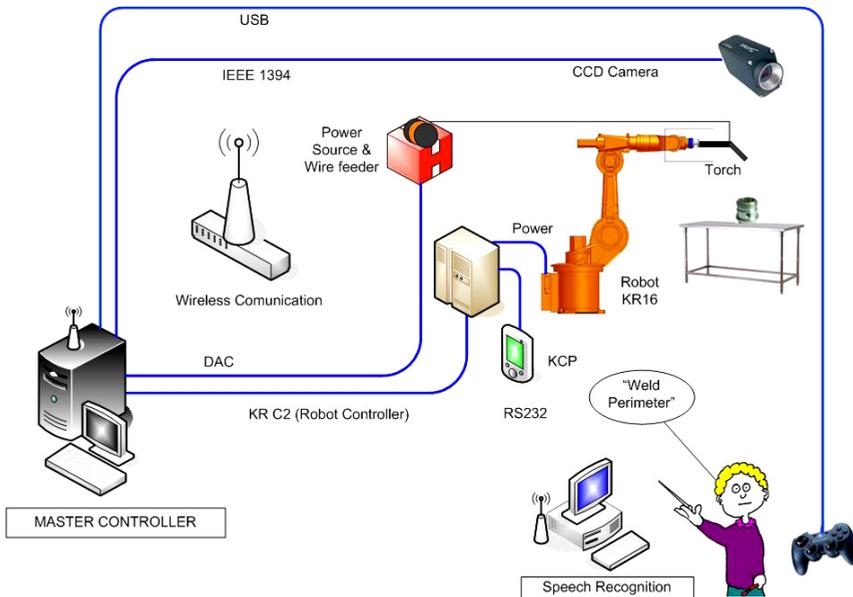


Fig. 1. Robotized Welding System

Two computers are used, the Master Controller and the Speech Recognition computer. The Master Controller is a PC Intel Xeon a 1.86GHz with 3GB RAM in charge of low-level serial communication with the robot controller using the 3964a protocol. It also connects to the Lincoln 455M power source and 10R wire feeder using an I/O Data Acquisition Card so that the welding process can be switched on-off and the current and voltage can be controlled by this computer. Additionally, it also handles the programming user-interface through a wireless gamepad. On the other hand, the Speech Recognition computer is in charge of giving voice commands to the robot in order to carry out the welding tasks.

The cell is based on the CORBA omniORB 4.1 Open Source GNU. The distributed system is designed to work within a wireless network. The elements of the wireless distributed system are described in the following sections.

2.1.1 Master Controller

The master controller connects to four subsystems as it could be seen in figure 1, namely:

- KUKA KRC2 Robot Controller
- GMAW Power source and wire feeder
- Vision System
- Programming Tool for Trajectory Learning

Each of these subsystems is explained below.

KUKA KRC2 Robot Controller

The KUKA KR16 robot is used in slave mode. Its motion is directed in low-level using the 3964a protocol and the RS232 serial communication. During operations a slave motion program in KPL runs in the KRC2 controller. This motion program is in charge, among other options, of the arm incremental motions, selection of tool/world coordinates, motion distance and speed. The other communication program resides in the Master Controller and forms part of the Stick Weld application.

GMAW Power source and wire feeder

The control and communication between the master controller and the power source and wire feeder is established using a general purpose DAC Sensoray 626. This is illustrated in figure 2. This card connects to the power feed to switch ON/OFF the power. It is also possible to modify the voltage and current. This was achieved by replacing the encoders from the voltage and current controls of the 455M Power Feed and to emulate its operation using a PIC microcontroller. The microcontroller receives a voltage or current data from the welding application and it translates in Gray code and send its data to the power feed, which in turn controls the wire feeder to the welding gun.

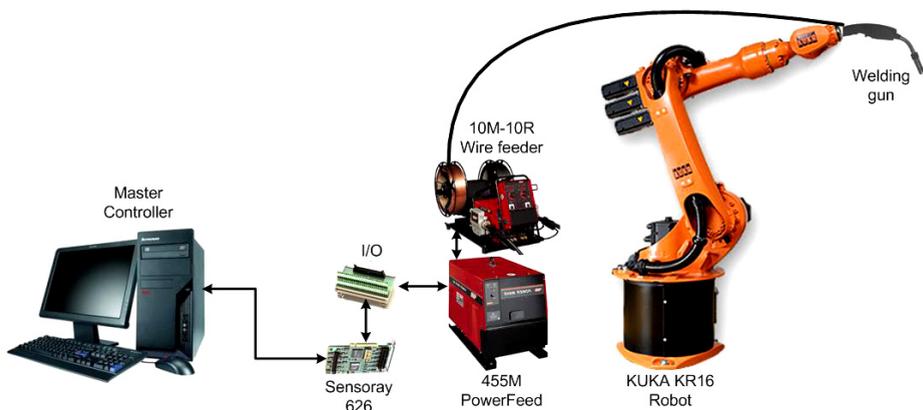


Fig. 2. Welding System

Vision System

The vision system was implemented using a fast Firewire Camera Basler A602fc with a 1/2" CCD sensor with 656 x 490 pixel resolution and using a IEEE 1394 bus. The application program for the vision system was developed in C++ using Visual Studio 2005 and using the SDK from Basler. This application is embedded into the Master Controller.

During welding tasks, the vision system is used to recognise welding trajectories. The distributed system defines several methods to be used that are available for other clients/server elements. The IDL functions that are available from other elements are (See (Davila-Rios, et al. 2009) for further details):

- Functions for robot motion
- Robot status
- Workcell status
- Force/Torque sensing
- Available welding trajectories for clients
- Modification functions for the welding power source
- Welding machine status
- General parameters
- Others.

The camera system is used to input data to program the robot new trajectories and to correct any misalignment during part welding but also it is being used as an input to the object recognition application. This object recognition is a developed application for 2.5D object recognition, complete details of the algorithm and development is explained further in this book (Lopez-Juarez, et al., 2009).

The server share methods for other object recognition through the following IDL methods:

- Object recognition
- Data from the recognised object
- Robot training execution
- Data from the training task
- Others

Programming Tool for Trajectory Learning

The master controller computer also uses the vision system, a wireless gamepad and a teaching tool (pointing stick) to train the robot new welding trajectories. In terms of the IDL functions there are some elements that are used for teaching purposes.

Industrial Robot KUKA KR16

The robot only works as a service provider. The master controller has access to its general parameters such as speed; world/tool coordinates selection, motion axis, etc. The communication is serial and there are no IDL functions, since it is not a CORBA client or server, but it is used by a server (the master controller).

Speech recognition

This system is based on the Microsoft SAPI 5.0 and works with Context Free Grammar. It generally works as a client making requests to the other systems. The IDL functions mostly interact with the robot using the following services:

- Speech recognition initialisation
- Stop the robot
- General parameters

The speech recognition system can be used directly to control the workcell or to initialise a specific process within the cell. The reader is referred to (Davila-Rios, et al., 2009) for further details on the implementation of the speech recognition system.

Additionally, the robot controller also can use another system for part location purposes. The workcell includes a Hytrol TA model belt conveyor, which is speed/position controlled by a Micromaster 420. The belt conveyor IDL functions are:

- Belt motion with X speed
- Status from the current location
- General parameters configuration

The belt conveyor only accepts directions and provides current status. This is considered a server only since it never acts as a client.

The issues concerning with the server-robot communication protocol are described in detail in section 3. In section 4, the program StickWeld V1.0, GUI and the use of the peripherals and programming modes are explained.

3. Programming and communication

The welding software is connected through CORBA to the other servers (the other modules of the manufacturing cell) as it was showed in figure 1. But a complete real-time communication is hold as well with the robot controller in order to access the robot movements.

In order to obtain continuous robot movements it was necessary to implement a stack of communication protocols. As it can be seen in figure 3, the welding SERVER (Master Controller) holds both, a CORBA module, and a set of functions to control the robot. The CORBA module embedded in the Stick Weld system gives access to other modules of the cell to manipulate some robotic tailored functions. This allows for example to receive direct commands from the speech module and to execute them.

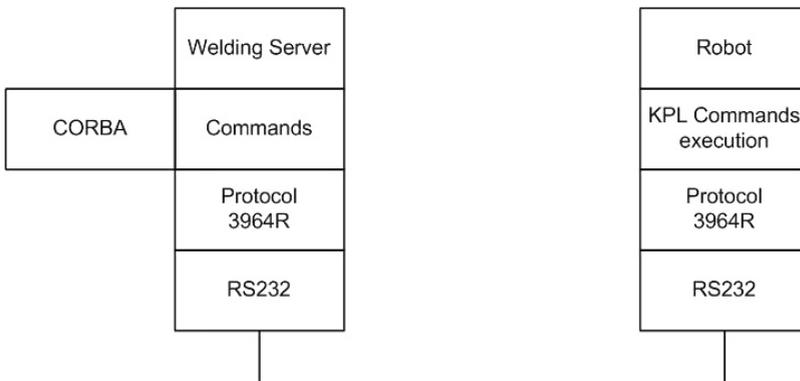


Fig. 3. Stack of communication protocols Master Controller PC - KUKA KR16

In the lowest layer, the RS232 standard is used for communication protocol. The second layer uses the SIEMENS 3964R protocol. The third layer contains the basic commands definition to be executed in the robot.

Some of these commands are presented in table 1.

Functions
Go Home
Coord World
Restart FIFO
AddMovement (x,y,z,speed)
Start FIFO processing
Stop FIFO processing
From current+1 delete all FIFO
Emergency Stop
DefineApproximation
SimpleMovement(x,y,z,speed)
SimpleRotation(x,y,z,speed)
Exit program

Table 1. Basic functions available in the communication.

In order to create a continuous movement of the robot, interpolation is carried out using a FIFO structure in the robot controller programming. This structure maintains the movements until a buffer is empty, or until a stop command is received. This could be an emergency stop for example or other stop function. The internal programming uses different interruptions to achieve continuous movements while receiving other commands at the same time, and to generate soft movements of discrete coordinates.

4. Stick Weld 1.0

Stick Weld 1.0 is a beta software project, presented as a prototype to create a functional powerful tool to teach easily different trajectories to an industrial robot. A user interface was developed containing two main functions:

1. It allows the robot to follow the contour of irregular or regular metal objects to be welded.
2. It allows a real-time robot programming tool to follow and weld random paths on flat surfaces.

To execute these tasks, an industrial robot is used, as well as different basic programming elements to teach the robot the desired trajectories. It includes basically: Fast colour camera, pointing stick, a gamepad, and a welding table. These elements are showed in figure 4.

In order to keep the programming tools simple, the robot is taught using an interface consisting of a pointing stick and a wireless gamepad. With the pointing stick, the user only has to move the stick on the metal part using the desired trajectory. The camera captures the 2D trajectories that the robot must follow later, executing the welding task. When the

teaching is being done, the vision system tracks the movements and records the Cartesian coordinates of the pointing stick as illustrated in figure 5.

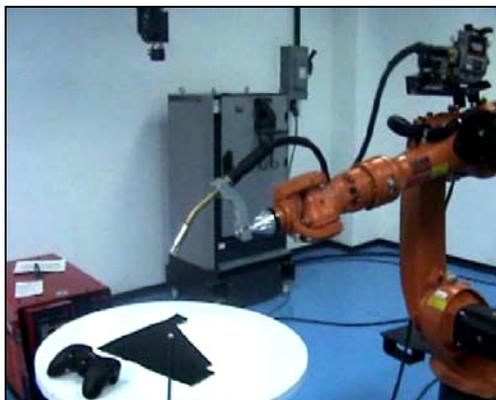


Fig. 4. Basic elements for trajectory programming

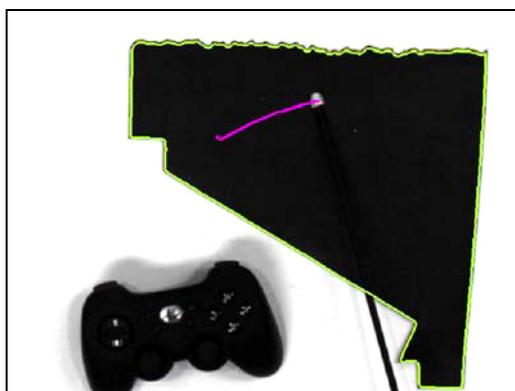


Fig. 5. Tracking of the pointing stick

All main functions are included in the buttons of the gamepad, to provide a complete useful interface. Among these functions, the start-finish robot movements are included as well as an emergency stop. Once the programming is finished, it is intended that the vision system, must recognize a specific piece, and also recall the already programmed objects (including all trajectories and configurations) and apply that in the task execution.

While scanning the object to be welded, all the contour, holes and special forms founded in the piece are recorded and converted to coordinates that the robot could follow if necessary. An example is provided in figure 6 where two embedded contours were identifies and recorded. In the upper part of the piece desired trajectories are computed. The vision system also has the option to work with different background colours. As showed in figure 6, the system works with a white metal surface and in this case a white pointing stick with a black tip was employed.

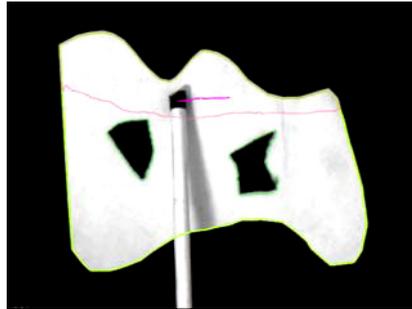


Fig. 6. Automatic detection of contours

Figure 7 shows a main screen of the software showing the designed GUI. There is a direct access to different configurations of the visual programming. The button: *Mover Robot*, execute different manual movements such as Go Home Position, Rotation in Z direction, etc.

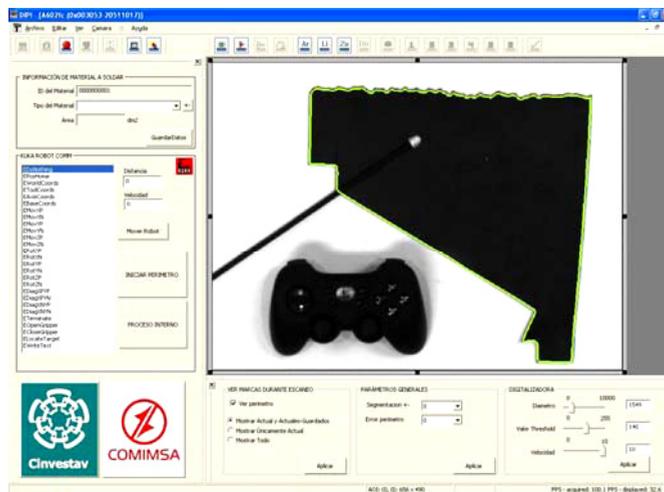


Fig. 7. Main Program Interface

It has two inputs, *Distancia* given in millimetres and *Velocidad* (Speed), which is given in m/s. The interface has also two buttons for automatic processing of the already recorded trajectories. The first one is *Iniciar Perimetro* (Start Perimeter), which starts the communication with the robot to follow the trajectory of a workpiece perimeter for welding purposes. The second button *Proceso Interno* (Internal Process), gives the instructions to the robot of welding all internal pre-recorded trajectories in the current object. Those options can be triggered also from the wireless gamepad.

The general configuration includes, diameter of pointing stick, threshold values, and scanning speed for the movements of the stick. For the programming of trajectories, some geometrical predefined figures have been programmed, such as closed areas, open lines, and zones. Also a visual configuration is available to help introducing new trajectories of a

specific type (i.e. closed area, line). These are identified in the upper toolbar as Ar, Li and Zo.

There are other features that can be accessed via the menu bar or the upper left toolbar. They are intended to modify the camera settings such as brightness and shutter speed and other options such as taking a single snapshot or taking image data continuously.

5. Conclusions and further work

During welding tasks there are quality specifications to be met. However, there are several factors that affect the process accuracy such as welding part positioning; motion errors in the production line, mechanical errors, backlash, ageing of mechanisms, etc. which are error sources that make robots to operate in uncertain conditions, i.e. unstructured environments. The scope of this work has been focused on the compensation of these errors generated during the welding process and that the robot system needs to compensate automatically.

The proposed solution includes the development of the *StickWeld V1.0* application which is a windows based solution programmed in Visual C++ that uses a CCD camera, Basler SDK, wireless gamepad and pointing stick as a teaching tools. The system also uses IDL functions so that the manipulator can receive verbal instructions such as motion commands or start/stop the task.

The developed user interface contains two main functions. One operation is to follow the contour of irregular or regular metal objects to be welded and the other is to follow and weld random paths on flat surfaces.

Simple welding trajectories were tested using the KUKA manipulator as shown in figure 8. Several issues rose from the accomplished welding tasks such as starting point synchronization, best torch angle, setting of the correct parameters (voltage and wire speed). It was detected that the geometric parameters such as width and high of the seam weld were not uniform along the paths, though the trajectory was correctly followed.



Fig. 8. Robot welding

On going work is looking at improving the actual welding stage by improving the selection of the welding parameters namely, voltage, current and wire feeding speed. At present, the

robot is able to work only on flat surfaces but future work has been envisaged to work on 3D surfaces using depth information as well as monitoring the pool weld that would require implementing an effective seam tracking mechanism including a robust vision system to be used on-line during welding tasks.

Acknowledgements

The authors wish to thank the following organizations who made possible this research: The Consejo Nacional de Ciencia y Tecnología (CONACyT) through Project Research Grant No. 61373, and for sponsoring Mr. Davila-Rios during his doctoral studies and to the Corporación Mexicana de Investigación en Materiales for its support through the Research Grant Project No. GDH - IE - 2007.

6. References

- Davila-Rios, I.; Lopez-Juarez, I.; Martinez-Martinez, L; and Torres-Treviño, L.M.(2009) Implementation of an Intelligent Robotized GMAW Welding Cell, Part I: Design and Simulation. In *Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria.
- Henning, Michi, Steve Vinoski. (2002) "Programación Avanzada en CORBA con C++", Addison Wesley, ISBN 84-7829-048-6.
- Holliday, D B, Gas-metal arc welding, (2005) ASM Handbook, Vol 6, *Welding, Brazing and Soldering*, 2005 pp (180-185).
- Jia, Songmin; Hada, Yoshiro; Ye, Gang; Takase, Kunikatsu (2002) "Distributed Telecare Robotic Systems Using CORBA as a Communication Architecture" *International Conference on Robotics & Automation Washington, DC*. May 2002.
- I. Lopez-Juarez, R Rios Cabrera. (2006) Distributed Architecture for Intelligent Robotic Assembly, Part I: Design and Multimodal Learning. In *Manufacturing the Future: Concepts, Technologies & Visions*. Edited by Vedran Kordic, Aleksandar Lazinica, Munir Medran. *Advanced Robotics Systems International*. Pro Literatur Verlag, Mammendorf, Germany. Pp. 337-366.
- I. Lopez-Juarez, M. Peña-Cabrera*, A.V. Reyes-Acosta (2009). Using Object's Contour and Form to Embed Recognition Capability into Industrial Robots. In *Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria.
- Peña-Cabrera, M & Lopez-Juarez, I. (2006). Distributed Architecture for Intelligent Robotic Assembly, Part III: Design of Invariant Recognition Vision Systems. In *Manufacturing the Future: Concepts, Technologies & Visions*. Edited by Vedran Kordic, Aleksandar Lazinica, Munir Medran. *Advanced Robotics Systems International*. Pro Literatur Verlag, Mammendorf, Germany. 2006. Pp. 400-436.
- Pires, J.N., Godinho, T. Nilsson, K., Haage M., Meyer, C.(2007). Programming industrial robots using advanced input-output devices: test-case example using a CAD package and a digital pen based on the Anoto technology. *International Journal of Online Engineering (iJOE)*, Vol 3, No 3.
- Veiga, G., Pires, J.N., Nilsson, K., (2007). On the use of SOA platforms for industrial robotic cells. In: *Proceedings of, Proceedings of Intelligent Manufacturing Systems, IMS2007*, Spain, 2007.

Dynamic Behavior of a Pneumatic Manipulator with Two Degrees of Freedom

Juan Manuel Ramos-Arreguin¹, Efren Gorrostieta-Hurtado¹, Jesus Carlos Pedraza-Ortega¹, Rene de Jesus Romero-Troncoso², Marco-Antonio Aceves¹ and Sandra Canchola¹
¹*CIDIT-Facultad de Informática*
²*Facultad de Ingeniería*
Universidad Autónoma de Querétaro, Querétaro, México

1. Introduction

The manipulator robots have many applications, such as industrial process, objects translation, process automation, medicine process, etc. Therefore, these kind of robots are studied in many ways.

However, most of the reported works use electrical or hydraulic actuators. These actuators have a linear behaviour and the control is easier than pneumatic actuators. The main disadvantages of electrical actuators are the low power-weight rate, the high current related with its load and its weight. The hydraulic actuators are not ecological, needs hydraulic oil which is feedback to the pump. On the other hand, the pneumatic actuators are lighter, faster, having a greater power-weight rate and air feedback is not needed. However, pneumatic actuators are not used into flexible manipulators developing, due to their highly non linear behaviour.

In the case of robot dynamic analysis, other researchers have presented the following works. An equations and algorithms are presented in an easy way to understand for all practical robots (Featherstone, 1987), (Featherstone & Orin, 2000), (Khalil & Dombre, 2002). Also, a dynamic simulation is developed to investigate the effects of different profiles into the contact force and the joint torques of a rigid-flexible manipulator (Ata & Johar, 2004). A study of dynamic equations is developed for a robot system holding a rigid object without friction (Gudiño-Lau & Arteaga, 2006). Furthermore, a dynamic modeling analysis is developed for parallel robots (Zhaocai & Yueqing, 2008). The link flexibility is considered for system performance and stability. Moreover, an innovative method for simulation is developed (Celentano, 2008) to allow students and researchers, to easily model planar and spatial robots with practical links. However, the reported works are developed using only electrical actuators, not pneumatic actuators. Therefore, this chapter presents how the pneumatic model is used with a two-link flexible robot and its dynamic analysis for the simplified system.

Previous researches by the authors include: a simplified thermo-mechanical model for pneumatic actuator has been obtained (Ramos et al., 2006). For animation purposes, the

Matlab - C++ - OpenGL Multilanguage platform, is used to solve the thermo-mechanical model (Gamiño et al., 2006), to get advantages of each language. Next, several algorithms are implemented to control the position (Ramos et al., 2006), including fuzzy control systems (Ramos-Arreguin et al., 2008). The flexible robot with pneumatic actuator is implemented, and practical results are compared with theoretical results (Ramos-Arreguin et al., 2008); where the PD control algorithm has been implemented into a FPGA device.

2. Pneumatic robot structure with one degree of freedom

A one degree of freedom (dof) pneumatic flexible robot is developed, for applications where a high risk for personal injury is present. This is shown in figure 1 (Ramos-Arreguin et al., 2008). This robot is light, cheap and ecological. Also, it uses a pneumatic actuator with high power-weight rate.

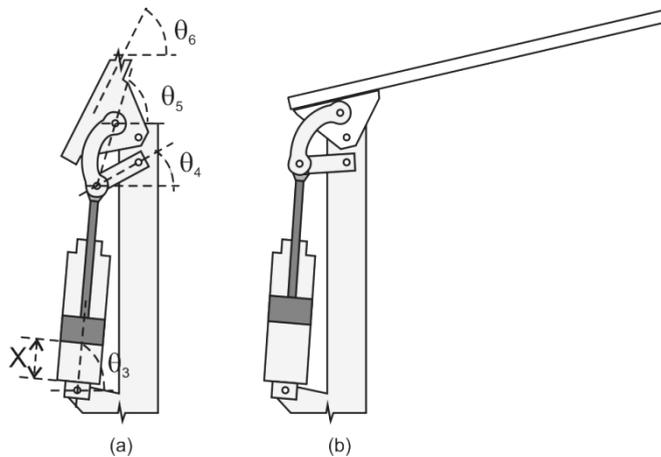


Fig. 1. One degree of freedom pneumatic robot. (a) Variables of the system. (b) General view.

In figure 1, the actuator used has damping at the boundaries of the cylinder. The control variable to take into account is θ_6 , which defines the slope of the arm. The value of θ_3 defines the angle of the pneumatic force to be applied. A diagram of the actuator is shown in figure 2, including the parameters involved for the mathematical model.

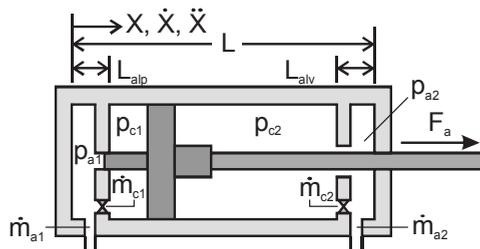


Fig. 2. Pneumatic actuator diagram, showing two damping zones.

Table 1 shows a description of each variable of pneumatic actuator in figure 2.

Variable	Description	Variable	Description
X	Rod position relative to the cylinder	\dot{X}	Rod speed relative to the cylinder
\ddot{X}	Rod acceleration relative to the cylinder	F_a	Actuator force
P_{a1}	Piston damping pressure	P_{c1}	Piston chamber pressure
P_{a2}	Rod damping pressure	P_{c2}	Rod chamber pressure
\dot{m}_{a1}	Mass flow from compressed air tank to the piston pad	\dot{m}_{c1}	Mass flow between the pad and chamber 1
\dot{m}_{a2}	Mass flow between the atmosphere and the piston pad	\dot{m}_{c2}	Mass flow between the pad and chamber 2

Table 1. Variables descriptions for pneumatic actuator.

The thermo-mechanical model (Kiyama & Vargas, 2005) use the variables defined in table 1. This original model takes considerable time to compute it. Therefore, a simplified model has been performed. Section 3 presents the simplified thermo-mechanical model.

3. Pneumatic actuator modelling

The rod displacement is controlled with a 5/2 electrovalve, as presented in (Ramos-Arreguin et al., 2008). The simplified thermo-mechanical model for pneumatic actuator of figure 2, is shown in equation (1) (Ramos et al. 2006). This model includes the parameters values and the space states are expressed as polynomial functions.

For the interval $0 \leq X \leq L$:

$$\dot{X} = \frac{d}{dt} X \tag{1a}$$

$$DX = \frac{d^2}{dt^2} X \tag{1b}$$

For the interval $0 \leq X \leq Lalp$:

$$P_{a1} = (-4.2316X + 8.0741)(\dot{m}_{a1} - \dot{m}_{c1} - 9.176 \times 10^{-10} P_{a1}DX) \times 10^{16} \tag{1c}$$

$$\dot{P}_{c1} = 38.8 \times 10^6 X^{-1} (\dot{m}_{c1} - 3.608 \times 10^{-8} P_{c1}DX) \times 10^6 \tag{1d}$$

For the interval $Lalp < X \leq L$

$$\dot{P}_{a1} = (-2.0952X^3 + 0.6233X^2 - 0.0777X + 0.0060)(\dot{m}_{a1} - 3.7 \times 10^{-8} P_{a1}DX) \times 10^{22} \tag{1e}$$

$$\dot{P}_{c1} = (-2.0952X^3 + 0.6233X^2 - 0.0777X + 0.0060)(\dot{m}_{c1} - 3.7 \times 10^{-8} P_{c1}DX) \times 10^{22} \tag{1f}$$

For the interval $0 \leq X < L-Lalp$

$$\dot{P}_{c2} = (1.1549X^3 + 0.0900X^2 + 0.0152X + 0.0025)(\dot{m}_{c2} + 3.469 \times 10^{-8} P_{c2}DX) \times 10^{22} \tag{1g}$$

$$\dot{P}_{a2} = (1.1549X^3 + 0.0900X^2 + 0.0152X + 0.0025)(\dot{m}_{a2} + 3.469 \times 10^{-8} P_{a2}DX) \times 10^{22} \tag{1h}$$

For the interval $L-Lalp \leq X \leq L$

$$P_{c2} = (1.3895X^2 - 0.2189X + 0.0088)(m_{c2} + 3.352 \times 10^{-8} X_4 X_6) \times 10^{26} \tag{1i}$$

$$P_{a2} = (8.3664X + 9.2571)[9.983 \times 10^3(m_{a2} - m_{c2}) + 1.168 \times 10^{-5} X_5 X_6] \times 10^8 \tag{1j}$$

The mass flow is given by

$$\dot{m} = \sqrt{\frac{2}{k-1}} \left(\frac{p_t}{p_0}\right)^{\frac{1+k}{2k}} \left[\left(\frac{p_t}{p_0}\right)^{\frac{1-k}{k}} - 1 \right]^{\frac{1}{2}} \sqrt{\frac{k}{RT_o}} p_o A_t \tag{2}$$

This mathematical model let us to compute the rod force over the mechanism, and is useful for simulation process. The actuator force (F_a) is expressed as:

$$F_a = P_{c1}A_{c1} - P_{c2}A_{c2} + P_{a1}A_{a1} - P_{a2}A_{a2} - P_{atm}A_v \tag{3}$$

Section 4 presents the methodology used for simulation purposes.

4. Simulation methodology

The simulation methodology is based in a multilanguage interface, developed with Matlab - C++ - OpenGL (Gamiño et al. 2006), (Ramos et al. 2009). Figure 3 shows a block diagram of the simulation process.

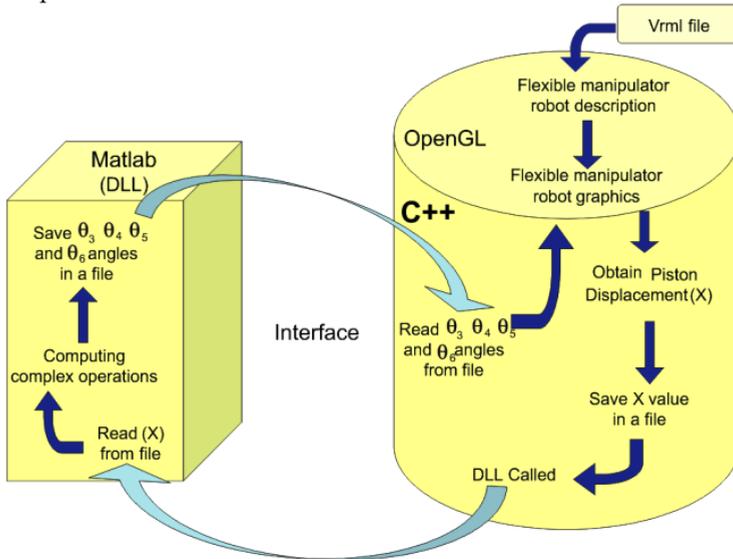


Fig. 3. Block diagram for Multilanguage interface development.

Figure 4 shows the result of this methodology. Matlab is used to compute the thermo-mechanical model; C++ is used for data interchange and to execute the matlab process, and OpenGL libraries are used for graphical view.

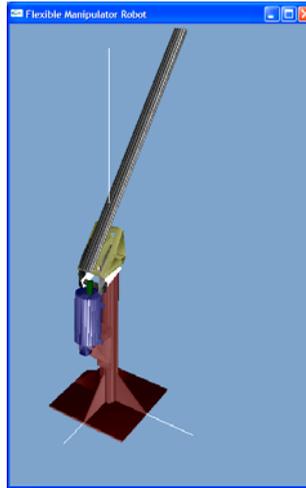


Fig. 4. Simulation of one degree of freedom flexible arm with pneumatic actuator.

Using this methodology, the control simulation with shorter computing time has been achieved. Some control algorithms used are PID, Discrete PID and Fuzzy (Ramos et al. 2006). The simulation results are used to implement a digital control, using FPGA technology (Ramos et al. 2008).

5. Pneumatic Flexible Robot with two degrees of freedom

5.1 Dynamic equations

The previous works developed are based in one degree of freedom for a flexible manipulator robot. To take this project further, a second degree of freedom system is considered.

The new contribution presented in this chapter, is a dynamic analysis of pneumatic flexible robot with two degrees of freedom. This analysis considers the pneumatic force for each of the joints. This first approach considers only a simplified structure, as shown in figure 5.

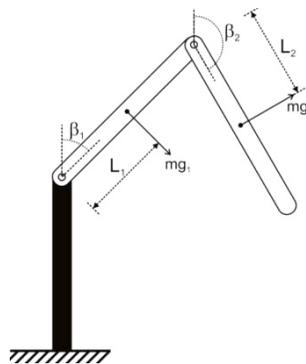


Fig. 5. Structure proposed for a manipulator robot with two degrees of freedom.

For the propose system on figure 5, the generalized variables are β_1 y β_2 . The speed equation for the gravity center of each link is

$$V_1^2 = L_1^2 \dot{\beta}_1^2 \quad (4)$$

$$V_2^2 = 4L_1^2 \dot{\beta}_1^2 + L_2^2 \dot{\beta}_2^2 + 4L_1L_2 \cos(\beta_1 - \beta_2) \dot{\beta}_1 \dot{\beta}_2 \quad (5)$$

The kinematic energy of the system is given by

$$V_1^2 = L_1^2 \dot{\beta}_1^2 \quad (6)$$

$$K = \left(\frac{1}{2}m_1L_1^2 + 2m_2L_1^2 + \frac{1}{2}J_1\right)\dot{\beta}_1^2 + \frac{1}{2}(m_2L_2^2 + J_2)\dot{\beta}_2^2 + 2m_2L_1L_2 \cos(\beta_2 - \beta_1) \dot{\beta}_1 \dot{\beta}_2 \quad (7)$$

The potential energy is

$$U = gL_1\left(\frac{1}{2}m_1 + m_2\right) \cos \beta_1 + \frac{1}{2}m_2gL_2 \cos \beta_2 \quad (8)$$

Finally, the Lagrangian equation, $L=K-U$, is given by:

$$L = \left(\frac{1}{2}m_1L_1^2 + 2m_2L_1^2 + \frac{1}{2}J_1\right)\dot{\beta}_1^2 + \frac{1}{2}(m_2L_2^2 + J_2)\dot{\beta}_2^2 - gL_1\left(\frac{1}{2}m_1 + m_2\right) \cos \beta_1 - \frac{1}{2}m_2gL_2 \cos \beta_2 + 2m_2L_1L_2 \cos(\beta_2 - \beta_1) \dot{\beta}_1 \dot{\beta}_2 \quad (9)$$

The equations of movement, are derived as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) - \frac{\partial L}{\partial \beta_1} = 0 \quad \text{and} \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) - \frac{\partial L}{\partial \beta_2} = 0 \quad (10)$$

$$2\left(\frac{1}{2}m_1L_1^2 + 2m_2L_1^2 + \frac{1}{2}J_1\right)\ddot{\beta}_1 + 2m_2L_1L_2 \cos(\beta_2 - \beta_1)\ddot{\beta}_2 - 2m_2L_1L_2 \sin(\beta_2 - \beta_1)\dot{\beta}_2^2 - gL_1\left(\frac{1}{2}m_1 + m_2\right) \text{sen} \beta_1 = 0 \quad (11)$$

$$(m_2L_2^2 + J_2)\ddot{\beta}_2 + 2m_2L_1L_2 \cos(\beta_2 - \beta_1)\ddot{\beta}_1 + 2m_2L_1L_2 \sin(\beta_2 - \beta_1)\dot{\beta}_1^2 - \frac{1}{2}m_2gL_2 \text{sen} \beta_2 = 0 \quad (12)$$

To convert the equations (11) and (12) into space states, the equation (13) are used.

$$\begin{aligned} X_1 &= \dot{\beta}_1; & \dot{X}_1 &= \ddot{\beta}_1 \\ X_2 &= \beta_1; & \dot{X}_2 &= \dot{\beta}_1 = X_1 \\ X_3 &= \dot{\beta}_2; & \dot{X}_3 &= \ddot{\beta}_2 \\ X_4 &= \beta_2; & \dot{X}_4 &= \dot{\beta}_2 = X_3 \end{aligned} \quad (13)$$

The space state for dynamic equations are:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 \cos(X_4 - X_2) & 0 & 2C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 2C_2 & 0 & C_3 \cos(X_4 - X_2) \end{bmatrix} \begin{bmatrix} \dot{X}_4 \\ \dot{X}_3 \\ \dot{X}_2 \\ \dot{X}_1 \end{bmatrix} = \begin{bmatrix} X_3 \\ C_4 \sin X_2 + C_3 \sin(X_4 - X_2) X_4^2 \\ X_1 \\ C_5 \sin X_4 - C_3 \sin(X_4 - X_2) X_1^2 \end{bmatrix} \tag{14}$$

Where:

$$\begin{aligned} C_1 &= \frac{1}{2}m_1L_1^2 + 2m_2L_1^2 + \frac{1}{2}J_1 \\ C_2 &= \frac{1}{2}(m_2L_2^2 + J_2) \\ C_3 &= 2m_2L_1L_2 \\ C_4 &= gL_1 \left(\frac{1}{2}m_1 + m_2 \right) \\ C_5 &= \frac{1}{2}m_2gL_2 \end{aligned} \tag{15}$$

Considering the friction force, on each of joint, the equation (16) is given by:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 \cos(X_4 - X_2) & 0 & 2C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 2C_2 & 0 & C_3 \cos(X_4 - X_2) \end{bmatrix} \begin{bmatrix} \dot{X}_4 \\ \dot{X}_3 \\ \dot{X}_2 \\ \dot{X}_1 \end{bmatrix} = \begin{bmatrix} X_3 \\ C_4 \sin X_2 + C_3 \sin(X_4 - X_2) X_4^2 - K_2X_2 \\ X_1 \\ C_5 \sin X_4 - C_3 \sin(X_4 - X_2) X_1^2 - K_1X_1 \end{bmatrix} \tag{16}$$

Where K is the constant friction of each joint.

5.2 Test of state space

In order to test equation (16), a Matlab program is used, considering the information given on table 2.

Parameter	Description	Value
m ₁	Link 1 weight	1.1 Kg
m ₂	Link 2 weight	1.5 Kg
L ₁	Length of link 1	0.5 m
L ₂	Length of link 2	0.5 m
J ₁	Inertia of link 1	0.015193 Kgm ²
J ₂	Inertia of link 2	0.03192 Kgm ²
K	Friction constant	0.25
g	Gravity force	9.81 m/s ²

Table 2. Values used for evaluation of equation (16).

Figure 6 shows the results of test of equation (16), without external force applied.

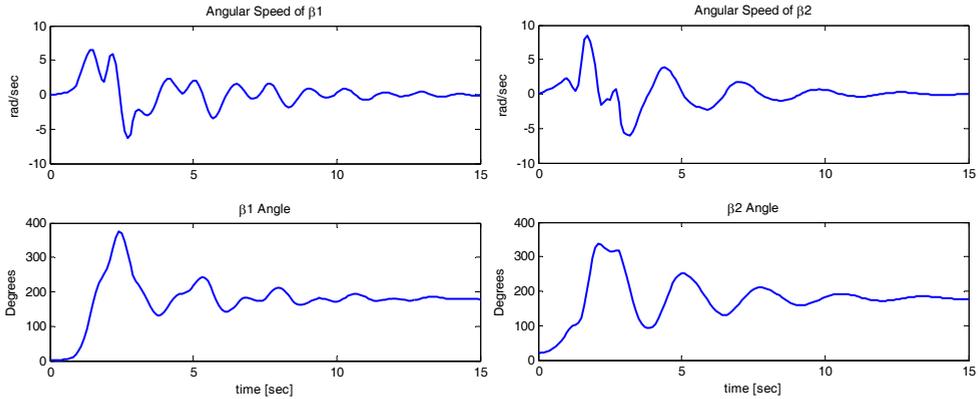


Fig. 6. Proposal structure behavior with pneumatic force equal to zero.

Figure 6 shows a steady behavior. The conditions initials are: $\beta_1 = 1^\circ$, $\beta_2 = 20^\circ$, $\dot{\beta}_1 = 0$ and $\dot{\beta}_2 = 0$. The movement is limited from 0° to 180° for both β_1 and β_2 . Since a force is not applied to the arms, from the initial conditions, the arms fall and their final position are 180° .

5.3 Dynamic behavior with pneumatic actuator

The figure 7 shows how the force of the pneumatic actuator is applied to the flexible manipulator robot, combining both pneumatic and actuator dynamic models. The torque for each joint link is generated by the pneumatic actuator, according with equation (17).

$$\tau_1 = Fa_1 d_1; \quad \tau_2 = Fa_2 d_2 \tag{17}$$

Equation (18) shows the state equations with torque considerations.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_3 \cos(X_4 - X_2) & 0 & 2C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 2C_2 & 0 & C_3 \cos(X_4 - X_2) \end{bmatrix} \begin{bmatrix} \dot{X}_4 \\ \dot{X}_3 \\ \dot{X}_2 \\ \dot{X}_1 \end{bmatrix} = \begin{bmatrix} X_3 \\ C_4 \text{sen } X_2 + C_3 \text{sin}(X_4 - X_2) X_4^2 - K_2 X_2 + \tau_2 \\ X_1 \\ C_5 \text{sen } X_4 - C_3 \text{sen}(X_4 - X_2) X_1^2 - K_1 X_1 + \tau_1 \end{bmatrix} \tag{18}$$

Figure 8, shows the sinoidal forces applied to links 1 and 2. Figure 9 shows the dynamic structure response. Given the initials conditions, the behaviour of both arms show a complete turn before stabilizing. The force applied to link 1, doubles the force on link 2. Figure 9 shows the pneumatic force on each joint.

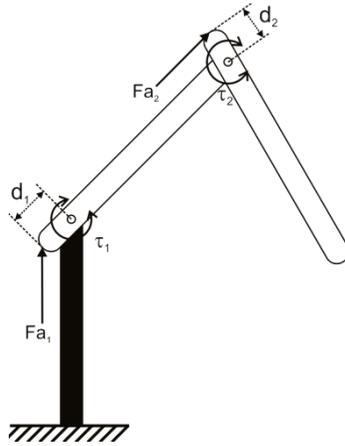


Fig. 7. Diagram of pneumatic force applied into each joint of flexible robot manipulator.

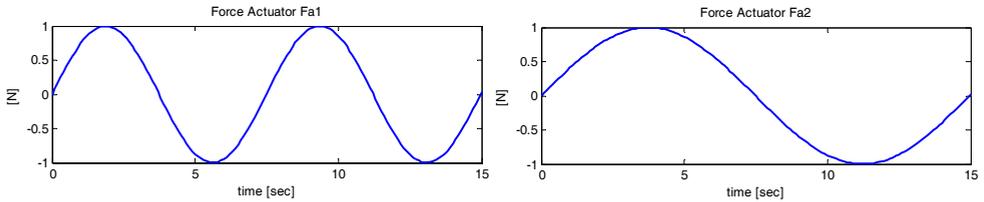


Fig. 8. Sinoidal pneumatic force applied into both link robots.

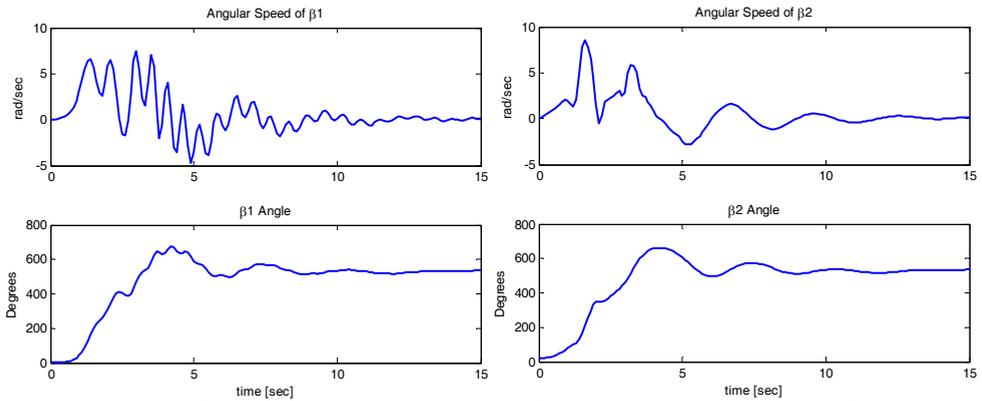


Fig. 9. Behaviour of the structure with pneumatic force applied into both link robots.

Figure 10 shows a block diagram where the simplified thermo-mechanical model and the equation (16) are evaluated to learn about the dynamic behavior of the structure. The valves apertures are the input of the thermo-mechanical model, and thermo-mechanical output is the pneumatic actuator force (F_a). The force F_a is the input for the state spaces and the output is β , which is the arm angle of the links 1 and 2. The diagram on figure 10 shows the proposed simplified model for each link.

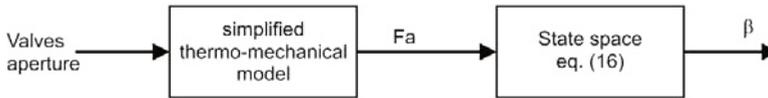


Fig. 10. Block diagram for simplified thermo-mechanical model and space states integration.

The result of the integration of the simplified thermo-mechanical model and state space are shown in figure 11.

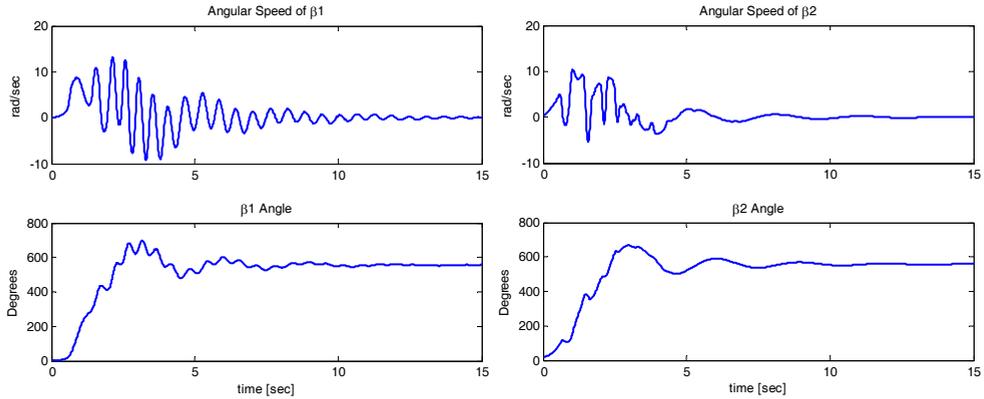


Fig. 11. Behavior of the robot, with both simplified thermo-mechanical and state space model.

With constant force applied on link 1 and 2, an initial oscillation on both links is observed, before reaching a steady state. That behavior can be observed in figure 11. To control the position of the angles β_1 and β_2 , is necessary to control the pneumatic force, and the aperture of the air flow through the pneumatic cylinder. Figure 12 shows the pneumatic forces F_{a1} and F_{a2} , generated by thermo-mechanical model.

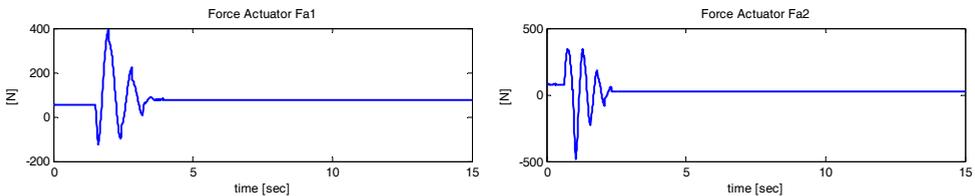


Fig. 12. Pneumatic forces generated using the simplified thermo-mechanical model.

With this work, a base of knowledge is generated about a pneumatic flexible manipulator robot with two degrees of freedom.

6. Future work

For future work, the simulation methodology for mechatronic applications (Ramos et al. 2009) will be applied to get a visual simulation of the behavior of flexible manipulator robot

with two degrees of freedom, with pneumatic actuator. Figure 13 shows a proposed flexible structure of the two-link robot, with pneumatic actuator on each arm. Therefore, including the effect of both four bar mechanism effects on the analysis is required. Also, a finite element analysis of both flexible links would be applied.

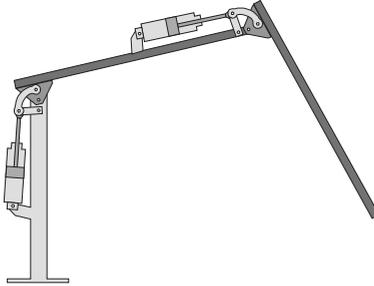


Fig. 13. Proposed structure of two degrees of freedom for flexible manipulator robot, with two pneumatic actuators.

7. Conclusions

The majority of the work previously developed use electrical actuators. The pneumatic actuators are not considered due to the highly non linear behavior. Previous work only considered one link and dynamic behavior of the pneumatic actuator. Also, graphical simulations have been developed under the same considerations. The innovation of this work is addition of the second degree of freedom for the flexible manipulator robot, together with dynamic analysis to study the robot behavior. In simulation of dynamic models, an oscillation can be observed; however, the oscillations decrease until the structure achieves the stability. The mechanical limitations are not considered into dynamic simulation. With this dynamic analysis, we can conclude that the proposed structure of two degrees of freedom for flexible robot with pneumatic actuators is feasible and can be controlled.

8. References

- Ata, A. A. & Johar, H. (2004). Dynamic simulation of task constrained of a rigid-flexible manipulator, pp. 61 - 66, *International Journal of Advanced Robotic Systems, Volume 1 Number 2*, ISSN 1729-8806
- Celentano, L. An Innovative Method for Robots Modeling and Simulation. *New Approaches in Automation and Robotics*, Book edited by: Harald Aschemann, ISBN 978-3-902613-26-4, pp. 392, I-Tech Education and Publishing, Vienna, Austria.
- Featherstone, R. & Orin, D.E. (2000). Robot Dynamics: Equations and Algorithms. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 826-834
- Featherstone, R. (1987). *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Boston/Dordrecht/ Lancaster.

- Gamiño, M.; Pedraza, J. C.; Ramos, J. M. & Gorrostieta, E. (2006). Matlab-C++ Interface for a Flexible Arm Manipulator Simulation using Multi-Language Techniques. *5th Mexican International Conference on Artificial Intelligence, MICAI 2006*. Tlaxcala, México.
- Gudiño-Lau, J. & Arteaga, M. A. (2006). Dynamic Model, Control and Simulation of Cooperative Robots: A Case Study. *Mobile Robots, Moving Intelligence*, Edited by Jonas Buchli, pp. 241-270, ISBN: 3-86611-284-X, Germany.
- Khalil, W. and Dombre, E. (2002). *Modelling, Identification and Control of Robots*, Hermes Penton Science, London, UK
- Kiyama, F. F. & Vargas, J. E. (2005). Dynamic Model Analysis of a Pneumatically Operated Flexible Arm. *WSEAS Transactions on Systems*, vol. 4, pp. 49-54.
- Ramos-Arreguin, J. M.; Pedraza-Ortega, J. C.; Gamiño-Galindo, M.; Gorrostieta-Hurtado, E.; Romero-Troncoso, R. J.; Aceves-Fernandez, M. A. & Hernandez-Valerio J. S. (2009). Simulation Methodology for Mechatronic Applications using Multilanguage Techniques. *14th IFAC International Conference on Methods and Models in Automation and Robotics*. August, Miedzyzdroje, Poland.
- Ramos-Arreguin, J. M.; Pedraza-Ortega, J. C.; Gorrostieta-Hurtado, E.; Romero-Troncoso, R. J.; Vargas-Soto, J. E. & Hernandez-Hernandez, F. (2008). Pneumatic Fuzzy Controller Simulation vs Practical Results for Flexible Manipulator. *Automation and Robotics*. Edited by Juan Manuel Ramos-Arreguin, Ed. I-Tech Education and Publishing, pp. 191-200, ISBN 978-3-902613-41-7, Austria
- Ramos-Arreguin, J. M.; Gorrostieta-Hurtado, E.; Vargas, J. E.; Pedraza-Ortega, J. C.; Romero-Troncoso, R. J. & Ramírez B. (2006). Pneumatic cylinder control for a flexible manipulator robot. *12th IFAC International Conference on Methods and Models in Automation and Robotics*. August, Miedzyzdroje, Poland.
- Ramos, J. M.; Vargas, J. E.; Gorrostieta, E. & Pedraza, J. C. (2006). New Polynomial Model of Pneumatic Cylinder Model. *Revista Internacional Información Tecnológica*, vol. 17, no. 3, Edited by José O. Valderrama, ISSN 0716-8756, La Serena, Chile.
- Zhaocai, D. & Yueqing, Y. (2008). Dynamic Modeling and Inverse Dynamic Analysis of Flexible Parallel Robots. *International Journal of Advanced Robotic Systems*, Vol. 5, No. 1, pp. 115-122, ISSN 1729-8806.

Dexterous Robotic Manipulation of Deformable Objects with Multi-Sensory Feedback - a Review

Fouad F. Khalil and Pierre Payeur

School of Information Technology and Engineering

University of Ottawa

Ottawa, ON, Canada, K1N 6N5

[fkhalil, ppayeur]@site.uottawa.ca

1. Introduction

Designing autonomous robotic systems able to manipulate deformable objects without human intervention constitutes a challenging area of research. The complexity of interactions between a robot manipulator and a deformable object originates from a wide range of deformation characteristics that have an impact on varying degrees of freedom. Such sophisticated interaction can only take place with the assistance of intelligent multi-sensory systems that combine vision data with force and tactile measurements. Hence, several issues must be considered at the robotic and sensory levels to develop genuine dexterous robotic manipulators for deformable objects. This chapter presents a thorough examination of the modern concepts developed by the robotic community related to deformable objects grasping and manipulation. Since the convention widely adopted in the literature is often to extend algorithms originally proposed for rigid objects, a comprehensive coverage on the new trends on rigid objects manipulation is initially proposed. State-of-the-art techniques on robotic interaction with deformable objects are then examined and discussed. The chapter proposes a critical evaluation of the manipulation algorithms, the instrumentation systems adopted and the examination of end-effector technologies, including dexterous robotic hands. The motivation for this review is to provide an extensive appreciation of state-of-the-art solutions to help researchers and developers determine the best possible options when designing autonomous robotic systems to interact with deformable objects.

Typically in a robotic setup, when robot manipulators are programmed to perform their tasks, they must have a complete knowledge about the exact structure of the manipulated object (shape, surface texture, rigidity) and about its location in the environment (pose). For some of these tasks, the manipulator becomes in contact with the object. Hence, interaction forces and moments are developed and consequently these interaction forces and moments, as well as the position of the end-effector, must be controlled, which leads to the concept of “force controlled manipulation” (Natale, 2003). There are different control strategies used in

general to solve the force controlled manipulation problem and to develop the required contact task control with the environment. These are respectively the hybrid position/force control scheme (Raibert & Craig, 1981) and the impedance control scheme (Hogan, 1987), the hybrid impedance control (Anderson & Spong, 1988), and the parallel force/position control (Siciliano & Villani, 1999).

Alternatively, to enhance interaction with the object, the process of measuring the variables resulting from the application of the forces on the object must be considered. Another important aspect of interaction with the object is the derivation of a contact model and the selection of proper grasp points (Mason & Salisbury, 1986; Salisbury & Roth, 1983; Cutkosky, 1989; Bicchi & Kumar, 2000; Mason, 2001). These have a crucial role in performing the grasping process where the objective is usually to mimic the human hand (Kaneko *et al.*, 2007). To achieve the resemblance with human arm/hand in robotics, force/tactile sensors (Javad & Najarian, 2005; Tegin & Wikander, 2005) can be mounted on robotic hands. These are usually comprised of two or more fingers. These types of sensors give crucial information such as the occurrence of a contact with the object, its size and shape, the exchanged forces between the object and the robot hand, the mechanical properties of the object in contact, and the detection of slippage of the body in contact. A smart combination of all this information opens the door to more sophisticated manipulation known as dexterous manipulation (Bicchi, 2000; Okamura *et al.*, 2000). Hand dexterity refers to the capability of changing the position and orientation of the manipulated object from a given reference configuration to a different one arbitrarily chosen within the hand workspace. It is a rather broad concept that involves aspects of, and usually a compromise between, ability and stability in performing motions of the manipulated object by hand palm and fingers.

2. Robotic Interaction

Handling rigid objects has been a dominant subject of study, and the literature reports numerous works on this aspect. Modeling and control of robots whose tasks include interaction with their environment is still a very active area in the robotic community. When robot manipulators are performing such tasks, motion planning is carried out during the unconstrained phase of the task where the robot moves toward the object. During this phase, sub-goals over the motion trajectory are calculated to control the robot until completion of the task. In this case position controllers are adequate as the robot is required to follow a desired motion trajectory. During the contact phase which follows, the interaction forces and moments between the robot manipulator and the environment, as well as the position of the end-effector, must be controlled. Modeling and control of robots based on contact dynamics is a challenging research field which attracts the interest of many researchers in the robotics community. A good coverage on the robotic manipulators foundations and different aspects of kinematics and dynamics modeling are presented in (Siciliano & Khatib, 2008; Pires, 2006), as well as on the related force and motion control strategies. The book (Vukobratovic *et al.*, 2003) reviews the different effects of robot dynamics while the manipulator is in contact with the environment. It provides an interesting overview of the research efforts carried out by the robotics community to tackle the problem of contact dynamics with the environment.

After the contact is established the required manipulation process can be performed. Manipulation processes are carried out by controlling both the interaction forces and the

corresponding position at the contact points with the object. The different control schemes that were introduced for the purpose of interaction control when dealing with rigid objects are reviewed in the following sections.

2.1 Robotic Interaction Modeling and Control

The robot manipulator system, that is the arm and gripper, or the finger manipulator in case of a multi-fingered dexterous hand, is a complex nonlinear dynamical system. Moreover, its subsystems, the links and joints, can be coupled. This eventually leads to sophisticated modeling and control approaches. However, when a robot manipulator becomes in contact with its environment (Vukobratovic *et al.*, 2003) its motion becomes constrained, and a deformation process occurs. The amount of deformation is depending on the stiffness characteristic of the environment or object, whether it is deformable or rigid, as well as depending on the end-effector type and shape. Consequently, due to this interaction, some reaction forces at the end-effector will be generated and felt at each joint.

The research efforts reported in the literature to solve the interaction problem, which involve force control and contact dynamics with the object, are designed with the aim of creating efficient control schemes for contact task control. For this reason different control schemes were developed to achieve effective force control during the interaction with the object. According to (Siciliano & Villani, 1999), these schemes belong to two main categories. The first category is the “direct force control” which achieves force control by means of explicit closure of the force feedback loop. The second category is the “indirect force control” which achieves force control through motion control. However, two classical control strategies are used in general to solve the force controlled manipulation problem and the required contact task control with the environment. They are namely: the hybrid position/force control scheme, which belongs to the first category; and the impedance control scheme, which belongs to the second category.

The hybrid position/force control scheme which was originally proposed in (Raibert & Craig, 1981) tries to decouple the directions in which the force is controlled, e.g. the force normal to the surface, from those in which the position is to be tracked, e.g. forces along the surface. This scheme is developed originally in response to the simultaneous presence of constrained and unconstrained directions for a robot manipulator in contact with the environment. The unconstrained direction is treated as a position control problem, while the constrained direction is explicitly force controlled. Therefore, the scheme structure consists of two parallel feedback loops, one for the position and another for the force. Each of these loops uses separate sensors and separate control laws. In fact, this separation results in two perpendicular subspaces: one for position and another for force. Due to this orthogonal structure, switching from force control to position control and vice versa might not be smooth and fast enough to cope with an interaction carried out in unstructured environment under real-time constraints. The hybrid position/force control scheme neglects the dynamic coupling effects that exist among each of the robot joints. This problem was subsequently investigated in (Khatib & Burdick, 1986). Exact decoupling of motion and force equations and linearization of the resulting system via nonlinear feedback has been accomplished in the joint space in (Yoshikawa, 1986) and in the task space in (McClamroch, 1986).

In contrast to hybrid position/force control, the impedance control scheme (Hogan, 1987) combines the position and force control rather than separating them. This approach aims at

softening the rigidity of robotic manipulators by assigning desired impedance to the end-effector and therefore represents a strategy suitable for constrained motion. The objective of this control approach is to achieve target impedance by having the end-effector perform a certain mechanical behavior. In other words, this method aims at controlling the position and force at the same time by translating a task into some desired impedance. The actual achieved position and corresponding forces will then be a function of the robot impedance, the environment admittance, and the desired position and force. The design concept adopted here is that the controller should be used to regulate the dynamic behavior between the robot manipulator motion and the force exerted on the environment. Therefore, impedance control has been considered as one of the most suitable control schemes to solve the interaction problem in unstructured environments. However, errors in the manipulator kinematics or due to unmodeled dynamics could cause excessive control action. Using hybrid position/force control for controlling dexterous hands was investigated in (Yin *et al.*, 2003) and impedance control in (Biagiotti *et al.*, 2003).

The method developed in (Anderson & Spong, 1988) benefited from the concepts of these two approaches and developed an alternative unified strategy under the name of hybrid impedance control which combines the two classical control techniques described above. The structure of the hybrid impedance control consists of inner and outer control loops. The inner loop provides the inverse dynamics control while the role of the outer loop is to achieve the desired characteristics like set-point tracking, disturbance rejection, and to cope with the robustness issue. Hybrid impedance control in general has been applied over the past two decades to enable robot end-effector to smoothly move between contact and non-contact phases of motion. The idea of this type of control has in fact emerged from the examination of how humans interact with their environment. Impedance control defines the relationship between the manipulator end-effector and the external forces generated when the end-effector is in contact with the environment. Depending on what is required to be controlled, that is force or position, the hybrid impedance control can use either position-based impedance (sometimes termed as admittance control) or a force-based impedance. Consequently this requires different control structures to be applied orthogonally to satisfy the nature of the hybrid position/force control. Position-based impedance control can be applied in the direction of the manipulated object to ensure that the contact point does not shift during the manipulation process; while the force-based impedance control can be applied in the direction perpendicular to the surface of the object. Applying these two structures in two directions can compensate for the change in object location because of its motion due to the manipulation process.

At the controller level, many control algorithms from both classical and modern control theory are found in the robotic literature for the purpose of controlling the manipulator motion. These controllers range from the traditional three term PID controller to more sophisticated nonlinear ones, like variable structure, adaptive and robust controllers. The book (Lewis *et al.*, 2004) reviews the application of these controllers and provides a critical evaluation of each stating the purpose of use, as well as their respective advantages and disadvantages. However, recently the attention of the robotic community has drifted more toward artificial intelligence (AI) concepts like experts systems, fuzzy logic or neural networks where more research efforts are reported since the mid 1990s (Katic & Vukobratovic, 2003). Still the industry remains favorable to the analytical solutions and

hence simple analytical learning algorithms are always potential candidates for industrial applications, especially at the task level.

2.2 Robotic Interaction Feedback Instrumentation

Alternatively, interaction control has also been considered as a sensor-based problem, in which two categories of research efforts can be distinguished. Force sensing is considered in the first category such that the manipulator can sense the interaction with the object during the interaction execution phase. In this case force sensors can be used effectively to implement hybrid position/force control or impedance control strategies. However, force sensing can usually provide 3D information only about the local contact points with the grasped object. On the other hand, vision systems, which are considered in the second category, can produce global information about the 3D environment. In the latter case the interaction control comes in form of visual feedback, to enable the robot to see the object and refine accordingly the interaction process. Unfortunately, vision approaches are generally not suitable to establish and maintain contact with the object surface if precise position and orientation are unknown.

Another trend emerged following some attempts to combine the two complementary sensory systems, that is vision and force. Different sensor fusion strategies have been proposed to merge force/torque-based and vision-based measurements where combined vision/force control scheme are developed.

2.2.1 Visual Servoing

One of the early control methods used with robot manipulators is visual feedback, which is often referred to as visual servoing. Visual servoing has proven to be a way of performing accurate movement in free space of the robot work cell without the need for accurate a priori models. The early work in visual servoing was initiated in the late 1970's with the pioneering work presented in (Hill & Park, 1979; Weiss *et al.*, 1987). More recently, various visual servoing systems have been reported in the literature where different approaches have been developed for robot task planning and to identify the geometry of unknown objects. In visual servoing two camera configurations are typically used: fixed and eye-in-hand, where a camera is mounted on the end-effector. Lately, these configurations became common in industrial settings to guide robots to perform manufacturing tasks on unknown objects. An extensive tutorial and survey on visual servoing can be found in (Hutchinson *et al.*, 1996; Hager *et al.*, 1996; Kragic & Christensen, 2002; Hashimoto, 2003).

For the visual servoing approaches used in practice, the depth information of an object cannot be measured directly. Therefore, different methods have been developed to obtain 3D coordinates of the manipulated object. One method is to use the images from multiple perspectives, either through stereovision or by moving the camera to multiple locations. In general, the approaches for visual servoing can be classified into two categories: position-based and image-based (Hutchinson *et al.*, 1996). In the position-based approach, a set of images are utilized together with a known camera model to extract the 3D pose of an object. The measured variables to be controlled are the Cartesian position and orientation of the object. In the fixed camera case, where the pose of the end-effector is to be controlled, pose has to be reconstructed from the available image data. Consequently, object tracking can also be performed by computing the error in the 3D space, and the position of the object is

extracted using the image information and a calibrated camera model. Therefore, a series of calibrations are necessary, such as between the robot base and the camera, between the tool and the camera, and for the camera itself.

Alternatively, in the image-based approach, the variables to be controlled are defined directly as features in the image space and hence it is not necessary to perform a complete 3D reconstruction of the scene. Tracking objects with the image-based approach is performed by computing the error on the image plane and asymptotically reducing this error to zero such that the robot is controlled to track a target, based on the errors in the image frames. For the fixed camera configuration, the image Jacobian can be calculated using the camera model. Because there are distortions of the targets in the image frame for the fixed camera configuration, the identification of features is not accurate. On the other hand, for the eye-in-hand configuration, the image Jacobian is more difficult to compute (Hutchinson *et al.*, 1996). However, the feature identification errors can be greatly reduced if the end-effector is perpendicular to the features on a surface.

However, due to the lack of precise position and orientation, none of the above two approaches is suitable to establish and maintain contact with the object surface. Many of the early research in visual servoing also ignored the dynamics of the robot and focused on estimating motion or recovering the image Jacobian. The paper (Papanikolopoulos *et al.* 1993) proposed an adaptive control scheme for an eye-in-hand system in which the depth of each individual feature is estimated at each sampling time during execution. Another method introduced in (Castano & Hutchinson, 1994) called visual compliance, which is a vision-based control scheme, was achieved through a hybrid vision/position control structure. In (Smits *et al.*, 2008) the possible visual feedback control transformations are studied among different spaces, including image space, Cartesian space, joint space or any other task space defined in a general task specification framework. In (Moreno *et al.*, 2001) a 3D visual servoing system is proposed based on stability analysis. They used Lyapunov's theorem to ensure that the transformation from the image frame to the world frame for 3D visual servoing system is carried out with less uncertainty. Several design issues for 3D servoing controllers in eye-in-hand setups were discussed by (Bachiller *et al.*, 2007). Especially they proposed a benchmark for evaluating the performance of such systems.

2.2.2 Feedback Control Based on Vision and Force Sensing

More recently modern robotic systems have been developed to enhance robot autonomy such that robots behave as artificially intelligent devices and act according to what they can perceive from their environment, either by seeing or touching the objects they manipulate. Thus, an important trend emerged to combine different sensory information, mainly vision and force feedback. In these dual sensory schemes, force sensing may result in full 3D information about the local contact with the grasped object, and hence enables the control of all possible six degrees of freedom in the task space. On the other hand, the vision system produces the global information about the 3D environment from 2D or 3D images to enable task planning and obstacle avoidance. Even if the exact shape and texture of the object remain unknown, the vision system can adequately measure feature characteristics related to the object position and orientation. Therefore, the levels of such vision/force integrated controller are classified into different categories (Lippiello *et al.*, 2007b): shared and traded, hybrid visual/force and visual impedance control. In shared control scheme, both sensors control the same direction simultaneously while in traded control, a given direction is

alternately controlled by vision or by force. The Hybrid control scheme involves the simultaneous control of separate directions by vision and force, while the impedance scheme rather combines the two control variables.

In an integrated vision/force control scheme, however, defining how to divide the joint subspace in vision or force controlled directions, or assigning which direction to share and how to share among others, is not always a clear problem. A review and comparison of the different algorithms that combine both visual perception and force sensing is presented in (Deng *et al.*, 2005). A critical evaluation of the two main schemes for visual/force control, namely the hybrid and impedance control is also presented in (Mezouar *et al.*, 2007).

Combining force with vision, which are in fact highly complementary to each other, was reported earlier in (Nelson & Khosla, 1996). Their implementation proposed to switch between vision-based and force-based control during different stages of execution. The paper (Hosoda *et al.*, 1998) introduced an integrated hybrid visual/force control scheme. Another hybrid visual/force control algorithm was proposed for uncalibrated manipulation in (Pichler & Jagersand, 2000). In these hybrid control methods the transform between the two sensory systems, force and vision, can be learned and refined during contact manipulations. Alternative visual impedance control schemes are introduced in (Morel *et al.*, 1998; Olsson *et al.*, 2004). Damping and stability issues of the interaction control at contact point in combined vision/force control schemes were investigated also in (Olsson *et al.*, 2004). Interaction control under visual impedance control using the two sensors was studied in (Lippiello *et al.*, 2007a), proposing a framework that allows to update in real time the constraint equations of the end-effector. In a hybrid force/position control scheme, the same authors also proposed in (Lippiello *et al.*, 2007b) a time varying pose estimation algorithm based on visual, force and joint positions data. A stereoscopic vision is used in (Garg & Kumar, 2003) to build a 3D model for the manipulated object and with a learning algorithm they map the object pose from camera frame to world frame. In (Kawai *et al.*, 2008) the hybrid visual/force control is extended to accommodate 3D vision information analysis taken from fixed camera based on a passivity dynamic approach.

Based on such integrated sensory systems, research efforts were reported on using fixed camera configuration and hybrid position/force control (Xiao *et al.*, 2000). In contrast to these efforts, others privileged an end-effector mounted camera, rather than a fixed one. Such a combined vision/force control scheme was reported by (Baeten & De Schutter, 2003) who use both force and vision sensors mounted on the end-effector at the same time. Using this eye-in-hand camera configuration, a common global 3D framework for both force and vision control was proposed to model, implement and execute robotic tasks in an uncalibrated workspace. The method to control the orientation of the end-effector using the force/torque sensor in this framework was investigated later by (Zhang *et al.*, 2006) and it was found that the torque measurement is not accurate enough for a free-form surface, which could cause orientation control errors. To overcome this problem an automated robot path generation method was developed based on vision, force and position sensor fusion in an eye-in-hand camera configuration. The combined sensor is used to identify the line or edge features on a free form surface. A robot is then controlled to follow the feature more accurately.

In integrated multi-sensory robotic setups it is important to accurately and coherently fuse measurements of complementary sensors. Therefore, sensor fusion becomes a crucial research topic. Sensor fusion as has been investigated in several ways to increase the

reliability of the observed sensor data by performing some statistical analysis, e.g. averaging sensors readings over redundant sensory measurements. A sensor fusion strategy has been proposed by (Ishikawa *et al.*, 1996) to fuse complementary information to obtain inferences that an individual sensor is not able to handle. In (Xiao *et al.*, 2000), they proposed a complementary sensor fusion strategy to fuse force/torque based and vision-based sensors, while in (Zhang *et al.*, 2006), they integrated sensor fusion with an automated robot program generation method for the vision, force and position sensors. In (Pomares *et al.*, 2007), researchers were able to plan the manipulator motion in 3D by fusing data from force and vision sensors in an eye-in-hand setup. Other sensor fusion techniques were introduced by (Smits *et al.*, 2006) using Bayesian filter, and by (Thomas *et al.*, 2007) using particle filters.

2.2.3 Integrating Vision, Force and Tactile Sensing

To better achieve autonomy in the robotic manipulation, robots should ultimately produce similar adaptive sensorial coordinations as human beings do (i.e. vision, servo and touch capabilities) in order to be effective to work in unknown and uncalibrated environments and therefore be able to adapt their behavior to unpredictable modifications. To achieve the resemblance with human arm/hand in robotics, tactile sensors along with force sensors can be used. Tactile sensors give crucial information such as the presence of a contact with the object, its physical size and shape, the exchanged forces/torques between the object and the robot hand, the mechanical properties of the object in contact (e.g. friction, rigidity, roughness, etc), as well as the detection of slippage of the body in contact. Hence, the robot hand can be used in a variety of ways. In particular, an important function that mimics human hand, other than grasping, is the ability to explore and to probe objects with fingers. Adding such type of interactions over the ability of grasping leads to the concept of dexterity of manipulation.

While vision can guide the manipulator toward the object during the pre-grasping phase, force and tactile sensors are used to provide real-time sensory feedback to complete and refine the grasping and manipulation tasks. The measurements obtained from force and tactile sensors are used to perform grasp control strategies aimed at minimizing the grasp forces or optimizing the end-effector's posture, as well as to perform force control strategies necessary for dexterous manipulation. Based on the provided measurements about the object in contact, the corresponding control strategies can then be performed in an autonomous manner during the task execution phase.

Force sensors commercially available are devices installed mostly at the robot manipulator wrist or at hand tendons. They usually measure the forces and moments experienced by the robot hand in its interaction with the environment. In fact, the major part of these sensors is composed of transducers which measure forces and torques by means of the induced mechanical strains on flexible parts of their mechanical structure. These strains are generally measured using strain gauges which in turn change their resistance according to local deformation during the interaction with the object. This way, these sensors provide the equivalent force/torque measurements.

On the other hand, tactile sensors are mounted on the contact surface of the fingertips of a robot hand, and eventually on the inner fingers and the palm, to measure the amount of contact pressure that is exerted. They consist of a matrix or array of sensing elements. Their function is to measure the map of pressure over the sensing area. A number of force and tactile sensors have been proposed for robotic applications with different realisations. The

work of (Javad & Najarian, 2005; Tegin & Wikander, 2005) give good overviews on the technologies and implementations used for such type of sensors.

The integration of vision, force and tactile sensors for the control of robotic manipulation can be found for example, in the work of (Payeur *et al.*, 2005) using industrial manipulator setup. There are also some other research efforts reported in the literature on using haptic systems to handle robotic manipulation at the dexterous hand level in (Barbagli *et al.*, 2003; Schiele & De Bartolomei, 2006; Peer *et al.*, 2006). In such systems, where the focus is on virtual control prototyping, users interact with virtual manipulated objects in the exact same way they would interact with the physical objects. The limitation in interacting with these objects in virtual manipulation rests the same that is faced by robotic systems working in the real world. These systems also assume that an in-depth knowledge of the object characteristics is available for inclusion into the simulated environment.

2.3 Robotic Grasping and Contact Modeling

In order to perform robotic grasping, contact points should be established first between the end-effector and the object. Contact points are of different types and physically differ in the shape of the contact area, and in the magnitude and direction of friction forces. Several types of such possible contacts are identified and examined thoroughly in (Mason and Salisbury, 1986). Grasping can be seen as the resultant of the interaction with an object at these contact points, while the location of the contact points can determine the quality and stability of the grasp.

There exists a substantial research effort carried out on robotic grasping and contact modeling of rigid objects where deriving the contact and grasping model is one of the essential operations in the manipulation process. A robot end-effector or hand is usually comprised of two or more fingers that restrain object (fixturing) or act on manipulated objects through multiple contacts at the same time. A standard classification of such interaction contacts according to specific models was introduced in (Salisbury & Roth, 1983; Cutkosky, 1989; Bicchi & Kumar, 2000; Mason, 2001). These contact models, which affect the analysis of the manipulation process, can be classified mainly into hard-finger (point contact with friction or without friction) and soft-finger (constraint contacts). In (Li & Kao, 2001) the review focuses specifically on the recent developments in the areas of soft-contact modeling and stiffness control for dexterous manipulation. Other important aspects of contact modeling consider also the viscoelastic behavior during rolling and slippage conditions. Under such circumstances the static and kinetic coefficients of friction play an important role in the grasp analysis, as well as whether the contact point moves on the contacting surfaces as they rotate with respect to each other or not.

In grasp analysis, the corresponding contact ways between hand fingers and objects to perform the desired grasp are also analyzed extensively in the literature. Extensive surveys on robot grasping of rigid objects reviewing the concepts and methodologies used can be found in (Bicchi & Kumar, 2000; Mason, 2001). Form closure and force closure are the most widely covered topics on grasp modeling that concern the conditions under which a grasp can restrain an object. These two concepts have been originally proposed for evaluating stable grasping of rigid objects. Form closure grasp (Bicchi, 1995), which was motivated by solving fixturing problems in assembly lines, considers the placement of frictionless contact points so as to fully restrain an object and thus can resist arbitrary disturbance wrenches due to object motion. Alternatively, force closure grasp (Nguyen, 1988) is more related to the

ability of a grasp to reject disturbance forces and usually considers frictional forces. The latter can resist all object motions provided that the end-effector can apply sufficiently large forces. A survey about force closure grasp methods was presented by (Shimoga, 1996). In this survey, different algorithms are reviewed for the computation of contact forces in order to achieve equilibrium and force closure grasps. Criteria for grasping dexterity are also presented. On the other hand, power grasps (Mirza & Orin, 1990) are characterized by multiple points of contact between the grasped object and the surfaces of the fingers and palm and hence increase grasp stability and maximize the load capability. The paper (Vassura & Bicchi, 1989) proposed a dexterous hand using inner link elements to achieve robust power grasps and high manipulability. Later on, in (Melchiorri & Vassura, 1992) mechanical and control issues are discussed for realizing such dexterous hand.

In another category, the research on multi-fingered robot grasping modeling can be classified as fingertip grasp and enveloping grasp (Trinkle *et al.*, 1988) respectively. In fingertip grasp the manipulation of an object is expected to be dexterous since the finger can exert an arbitrary contact force onto the object. Alternatively, when an object is grasped using the enveloping grasp model, the grasping process is expected to be stable and robust against external disturbance since the fingers contact with the object at many points.

There has been significant work as well towards recovering good grasp point candidates on the object. In this case the focus is not only on the contact forces, but also on investigating the optimal grasp points on the manipulated object. A comprehensive review is presented in (Watanabe & Yoshikawa, 2007) where different classifications are proposed for the methods used to choose such grasp points. In their work, choosing optimal grasp points was investigated on an arbitrary shaped object in 3D space using the concept of required external force set. A graphical method is presented in (Chen *et al.*, 1993) for investigating optimal contact positions for grasping 3D objects while identifying some grasp measures. Some researchers aimed at investigating optimal grasp points or regions for balancing forces to achieve equilibrium grasp. A breakthrough in the study of grasping-force optimization was made by (Buss *et al.*, 1996), while in (Liu *et al.*, 2004) the researchers presented an algorithm to compute 3D force closure grasps on objects represented by discrete points. The proposed algorithm combines a local search process with a recursive problem decomposition strategy. In (Ding *et al.*, 2001) they proposed a simple and efficient algorithm for computing a form closure grasp on a 3D polyhedral object using local search strategy. A mathematical approach is presented in (Cornellà *et al.*, 2008) to efficiently obtain the optimal solution of the grasping problem using the dual theorem of nonlinear programming. However, these methods yield optimal solutions at the expense of extensive computation. In (Saut *et al.*, 2005) an alternative on-line solution is introduced to solve the grasping force optimization problem in multi-fingered dexterous hand by minimizing a cost function. Another real-time grasping force optimization algorithm for multi-fingered hand was introduced in (Liu & Li, 2004) by incorporating appropriate initial points.

3. Manipulation of Deformable Objects

The main challenge in developing autonomous robotic systems to manipulate deformable objects comes from the fact that there are several generic interconnected problems to be resolved. Mainly it involves the collection of deformation characteristics, the modeling and simulation of the deformable object from these estimates, and the definition and tuning of

an efficient control scheme to handle the manipulation process based on multi-sensory feedback. A recent trend aims at merging measurements taken from vision, force and tactile sensors to accelerate the development of autonomous robotic systems capable of executing intelligent exploratory actions and to perform dexterous grasping and manipulation.

3.1 Deformable Objects Modeling and Simulation

Automatic handling of deformable objects usually requires that the evaluation of the deformation characteristics is carried out using simulated environments before conducting the physical experiment. Hence, the manipulation process can be successfully performed by analyzing the manipulative tasks and deriving their control strategies using deformable object models.

3.1.1 Computer Simulation of the Object Elasticity

A wide variety of approaches have been presented in the literature dealing with computer simulation of deformable objects (Gibson & Mirtich, 1997; Lang *et al.*, 2002; Terzopoulos *et al.*, 1987). These approaches are mainly derived from physically-based models that emulate physical laws to produce physically valid behaviors. Using these models to provide interactive simulation of deformable objects dynamics has been a major goal of the computer graphics community since the 1980s (Pentland & Williams, 1989; Pentland & Sclaroff, 1991). Mass-spring system simulations and finite-elements methods (FEM) are the major physically-based modeling techniques considered. Under these frameworks, it can be considered that a deformable object has infinite degrees of freedom and therefore an attempt to simplify the problem is to discretize the structure, reducing the number of its degrees of freedom to a finite countable set.

Mass-spring system techniques have widely and effectively been used for modeling deformable objects. These objects are described by a set of mass particles dispersed throughout the object and interconnected with each other through a network of springs in 3D. This configuration constitutes a mathematical representation of an object with its behavior represented according to Newton's laws which incorporates calculating forces, torques, and energies. This model is faster and easier to implement as it is based on well understood physics, than finite-elements methods. It is also well suited for parallel computation, making it possible to run complex environments in real-time for interactive simulations. On the other hand, mass-spring systems have some drawbacks. Incompressible volumetric objects and high stiffness materials, which have poor stability, require small time integration step during the simulation process. This considerably slows down the simulation. Another weakness is that most of the materials found in nature maintain a constant or quasi-constant volume during deformations; unfortunately, mass-spring models do not have this property.

In finite-elements methods, unlike mass-spring methods where the equilibrium equation is discretized and solved at each finite mass point, objects are divided into unitary 2D surfaces, or volumetric 3D elements, joined at discrete node points. The relationship between the nodal displacements and the force applied follows Hooke's law where a continuous equilibrium equation is approximated over each element. Therefore, finite-elements methods offer an approach with much higher accuracy. However, while finite-elements methods generate a more physically realistic behavior, at the same time they require much

more numerical computation and therefore are difficult to use for real-time simulations. This is due to the fact that the object discretization and calculation of a stiffness matrix are computationally expensive.

In practice the physically motivated deformable models are mostly limited to surface modeling, mainly due to overwhelming computational requirements. Therefore, for simulation of robot interaction with deformable objects, mass-spring models prove to be very efficient. On the other hand, the deformable materials are considered to be either elastic, viscous, or viscoelastic. Objects with elastic behavior have the ability to recover from deformation caused by an externally applied force. Objects with viscosity resist such applied force due to their internal forces which act as damping force. The viscoelastic objects combine the elastic and viscous behaviors together. Such objects can also be deformed to the required shape according to applied force. Therefore automating and controlling the process of casting the raw viscoelastic material is crucial in some industrial applications (Tokumoto *et al.*, 1999).

As mentioned above the mass-spring model normally describes a deformable object as a set of particles constructed from a discretized sampling of its volume using a lattice configuration where a network of interconnected particles and springs is formed. These particles are the mass points in which the body mass is concentrated and are related to each other by forces acting on the object. Springs connecting these mass points exert forces on neighboring points when the object mass is displaced from its rest positions due to interaction. Therefore, the deformation of the object can be characterized by the relationship between the applied force and the corresponding particle displacement reflecting the deformation taking place. This means that this displacement describes the movement of the particle during the process of deformation.

Deformable materials can be described by models that are essentially made of different configurations of mass-spring-damper. The basic models are determined by the Kelvin model (or Voigt model) and the Maxwell model. The Kelvin model consists of a spring and a damper which connect two mass points in parallel. The Maxwell model is a series of a spring and a damper connecting two mass points. Other models can also be derived from the combination of the basic models or elements. For example, the Standard Linear model is a combination of the Maxwell model in parallel with a spring. (Byars *et al.*, 1983) give further details on the models mentioned above and discuss further issues on deformable objects modeling and analysis from a mechanical engineering perspective. A new approach is presented in (Tokumoto *et al.*, 1999) for the deformation modeling of viscoelastic objects for their shape control. In this work, the deformable object is modeled as a combination in series of Kelvin and Maxwell models. In a later step of their experiment they introduced a nonlinear damper into the model to solve a discrepancy between an actual object and its linear model. The drawbacks of Kelvin-Voigt modeling were investigated by (Diolaiti *et al.*, 2005) proposing an alternative solution for estimating the contact impedance using nonlinear modeling.

3.1.2 Modeling and Simulating the Physical Interaction

In addition to computer modeling and simulation of deformable objects, other research efforts in robotics were dedicated to the problem of modeling the physical process of manipulation. In order to implement and evaluate the manipulative operations on deformable objects by a robotic system, an object model is indispensable to represent the

elasticity and deformation characteristics during the physical interaction. The corresponding modeling problem for 1D and 2D deformable objects was studied extensively for specific applications in (Henrich & Worn, 2000; Saadat & Nan, 2002), based on mathematical representations of their internal physical behavior.

Robotic manipulative operations for deformable objects often rely on the object deformation model. However the operations may result in failure because of unexpected deformation of the objects during the manipulation process. Thus, automatic handling of deformable objects requires that the evaluation of the deformation of these objects is performed in advance using the object models to ensure that the manipulative operation is successful in the real application. Furthermore, it is important to plan tasks and derive their strategies by analyzing the manipulative processes using deformable object models. Beyond performing only simulations, in (Shimoga & Goldenberg, 1996) a soft finger is modeled using the Kelvin model in which a spring and damper are placed in parallel. The deformation parameters were experimentally calculated in a first phase, and then used in the Kelvin model with the desired impedance parameters to successfully control the impedance of a soft fingertip. In another experiment the physical interaction between a deformable fingertip and a rigid object was modeled and controlled by (Anh *et al.*, 1999) based on a comprehensive dynamical notations.

In fact, deformable objects change their shapes during manipulation and display a wide range of responses to applied interaction forces because of their different physical properties. This is due to their nonlinearity attributes and other uncertainties, such as friction, vibration, hysteresis, and parameter variations. To cope with this problem, one approach is to estimate the shape of the deformable object by calculating an internal model and simulating the object behavior. Such internal model could be static or dynamic (Abegg *et al.*, 2000). As examples from the work on static and dynamic modeling, in (Hirai *et al.*, 1994) they calculated a static model for the object and obstacle in 2D, while in (Wakamatsu *et al.*, 1995) they calculated the same but in 3D. In (Zheng & Chen, 1993) they emphasized on trajectory generation based on a static model for a flexible load. Using a similar static modeling approach, the problem of insertion tasks is tackled in (Zheng *et al.*, 1991) with a flexible peg modeled as a slender beam. In the work presented in (Kraus & McCarragher, 1996), they followed the same static modeling guidelines such that no dynamic analysis is considered. However, in contrast to other works on static modeling they considered the use of force feedback to control manipulator motions. In the paper of (Wakamatsu *et al.*, 1997), they extended the ideas employed in static modeling to derive a dynamic model of a deformable linear object. Other modeling techniques were also reported in the literature, for example, in (Nguyen & Mills, 1996) they considered using lumped parameter model. In (Wu *et al.*, 1996; Yukawa *et al.*, 1996) they investigated the problem with a distributed parameter model solution.

However, it is difficult to build an exact model of deformable objects. Thus, for some researchers modeling can be highly depending on imitating and simulating the skills of human expertise when dealing with such objects. In this case the robot motion during task execution can be divided into several primitives, each of which has a particular target state to be achieved in the task context. These primitives are called skills. An adequately defined skill can have enough generality to be applied to various similar tasks. Accordingly, different control strategies are required for the robot arm to manipulate in an autonomous manner the different kinds of objects according to the specified application. Most of the

previous research works on deformable objects involve the modeling and controlling of 1D deformable linear objects, such as beams, cables, wires, tubes, ropes, and belts. Some of the skill-based modeling and manipulation for handling deformable linear objects has been reported, for example, by (Henrich *et al.*, 1999) where they analyzed the contact states and point contacts of a deformable linear object with regard to manipulation skills. The problem of picking up linear deformable objects by experimentation is discussed in (Remde *et al.*, 1999a). The problem of inserting a flexible beam into a hole is examined in (Nakagaki *et al.*, 1995) using a heuristic approach to guide the manipulator motion.

Finite-elements modeling techniques were also used to model deformable objects characteristics and to simulate the physical interaction. A framework is described in (Luo & Nelson, 2001) based on FEM modeling that fuses vision and force feedback for the control of highly linear deformable objects in form of active contours, or snakes, to visually observe changes in object shape during the manipulation process. The elastic deformation of a sheet metal part is modeled in (Li *et al.*, 2002) using FEM and a statistical data model. The results from this model are used to minimize the part's deformation. In (Kosuge *et al.*, 1995), they used FEM modeling to examine the problem of controlling the static deformation of a plate when handled by a dual manipulation system. In one of the recent efforts, a finite-elements modeling technique was reported by (Garg & Dutta, 2006), where a model is developed to control the grasping and manipulation of a deformable object based on internal force requirements. In this model the object deformation is related to fingertip force, and based on impedance control of the end-effector.

However, modeling of 3D deformable objects for robotic manipulation has not been widely addressed in the literature so far. This results from its inherent complexity and the fact that a majority of researchers hope to tackle the simpler 1D modeling problem before generalizing it to a 3D modeling solution. Among the very few research efforts on 3D modeling of deformable objects is the pioneering work reported by (Howard & Bekey, 2000) who developed a generalized solution to model and handle 3D unknown deformable objects. This work benefited from a dynamic model originally introduced by (Reznik & Laugier, 1996) to control the deformation of a deformable fingertip. The model used in (Howard and Bekey, 2000) to represent the viscoelastic behavior is derived from dividing the object into a network of interconnected particles and springs according to the Kelvin model. Then by using Newtonian equations, the particles motion is used to calculate the deformation characteristics based on neural networks. Other interesting methods for modeling 3D deformable objects are based on probing the object to extract the deformation characteristics with the aid of vision. One of these methods was developed in (Lang *et al.*, 2002) to acquire deformable models of elastic objects in an interactive simulation environment where an integrated robotic facility was designed to probe the deformable object in order to acquire measurements of interactions with the object. Another method of probing and vision tracking was proposed in (Cretu *et al.*, 2008) to model deformable objects geometric and elastic properties. The approach uses vision and neural networks to select only a few relevant sampling points on the surface of the object and guides the acquisition of deformation characteristics through tactile probing on these selected points. The measurements are combined to accurately represent the 3D deformable object in terms of shape and elastic behavior.

3.1.3 Deformable Object Grasping and Contact Modeling

Nowadays, an important goal of robotic systems is to achieve stable grasp and manipulation of objects whose attributes and deformation characteristics are not known a priori. To establish contact and grasp modeling for deformable objects, the concepts of rigid force and form closure, as well power grasp, were extended to accommodate deformable objects. In (Wakamatsu *et al.*, 1996) the effort was to extend the concept of force closure for rigid objects with unbounded applied forces to deformable objects with bounded applied forces. Wakamatsu *et al.* introduced the concept of bounded force closure, which is defined as grasps that can resist any external force within the bound. They considered a candidate grasp and external forces within a bound that can deform and displace the deformed part. In (Prattichizzo *et al.*, 1997) the focus is on the dynamics of the deformable objects during the process of power grasp. A geometric approach is adopted to derive a control law decoupling the internal force control action from the object dynamics. More recently, a new framework for grasping of deformable parts in assembly lines was proposed in (Gopalakrishnan & Goldberg, 2005) based on form closure for grasping deformable parts. In this framework a measure of grasp quality is defined based on balancing the potential energy needed to release the part against the potential energy that would result in plastic deformation. Other attempts were reported on grasping using soft fingers, such as the work in (Shimoga & Goldenberg, 1996), to design systems with force control based on grasping with soft fingers. In (Tremblay & Cutkosky, 1993) they also used a deformable fingertip but equipped with a dynamic tactile sensor which was able to detect slippage. The paper of (Inoue & Hirai, 2008) is an up-to-date reference on soft finger modeling and grasping analysis.

3.2 Robotic Interaction Control with Deformable Objects

In early robotic systems designed to manipulate deformable objects, the problem of interaction control was solved mainly in two different ways. The robotic system to handle deformable object was either designed based on force and grasp stability control, or force control versus deformation control. A control strategy based on PID control was proposed in (Mandal & Payandeh, 1995) to maintain stable contact against a compliant 1D surface. In (Meer & Rock, 1994) they used impedance control to manipulate flexible objects in 2D. A force and position control scheme was developed in (Chiaverini *et al.*, 1994) capable of regulating a manipulator in contact with an elastically compliant surface using PID control. In the paper of (Patton *et al.*, 1992) they used an adaptive control loop to generate correct tension on a 2D deformable object where stiffness is designated as the adaptive variable. In (Luo & Ito, 1993) the researchers developed an adaptive control algorithm such that the robot manipulator was able to maintain continuous interaction with a 1D deformable surface. In the work of (Seraji *et al.*, 1996) a dual-mode control scheme using both compliance and force control was applied to establish a desired force on a 1D deformable surface. In the research effort of (Yao & Tomizuka, 1998) they used a robust combination of force and motion control to enable a robot manipulator to apply a force against a 1D nonlinear compliant surface. A feedback regulator was developed in (Siciliano & Villani, 1997) which only required force and position measurements to be fed into the control loop to handle a compliant surface. In another framework handling compliant surfaces with unknown stiffness, (Chiaverini *et al.*, 1994) introduced a parallel force/position control solution. In (Li *et al.*, 2008) researchers investigated solving the problem of interaction with

unknown deformable surface within an adaptive compliant force/motion control framework. The deformable object elasticity parameters were identified as a mass-spring system. Based on an intelligent setup for dynamic modeling introduced in (Katic & Vukobratovic, 1998) a PD controller was developed which allows a manipulator to apply a constant force on a 1D deformable object without having prior knowledge of its deformation. In the work reported by (Al-Jarrah & Zheng, 1998) a controller is set initially to command the manipulator to bend a 1D deformable object into a desired configuration in an intelligent compliant motion framework. In the work of (Venkataraman *et al.*, 1993), a neural network was used to address the problem of deformation parameters identification. Similarly, a fuzzy logic based control system was introduced by (Tarokh & Bailey, 1996). In (Arai *et al.*, 1993) the problem of deformable object manipulation in terms of both positioning and orientation of 2D objects was addressed. In their work, the desired trajectory was produced by controlling the torque. This control scheme was improved later in (Arai *et al.*, 1997) by using recurrent neural network as a forward model. The focus in (Kim & Cho, 2000) was on solving the misalignment problem in flexible part assembly using neural networks. Finally, a real-time eye-in-hand system was introduced by (Terauchi *et al.*, 2008) in which impedance control is used to cope with the flexible interaction and a neural network is used to learn the impedance parameters. A review of the intelligent control techniques applied for deformable object cases can be found in (Katic & Vukobratovic, 2003).

Overall, these systems require explicit models of the object which include in-depth knowledge about mass/object dynamics and deformability, and therefore, a complex force sensory system is required to measure the position and force on the object. However, dexterous grasping and manipulation of a deformable object must be performed robustly despite uncertainties in the robot environment where deformable objects are neither exactly located nor modeled. This leads to higher flexibility, and can improve speed and precision of the task execution. A number of recent research efforts focus on improving both the task quality and its range of feasibility by using integrated vision and force based control schemes. In such dexterous manipulation it is important to consider the difference between the way of handling rigid and deformable objects. This leads to a major distinction between the definitions of grasping and manipulation respectively (Hirai *et al.*, 2001). While the manipulation of a rigid object requires only the control of its location, the manipulation of a deformable object requires controlling both the location of the object and its deformation. In the handling process of rigid objects, grasping and manipulation can be performed independently. Grasping of a rigid object requires the control of grasping forces only, while manipulation of a freely moving rigid object results in the change of its position and orientation. On the other hand, grasping and manipulation interfere with each other in the handling of deformable objects. Grasping forces yield the deformation of a non-rigid object, which may simultaneously change the shape and location of the object. Hence contact between fingers and the object may be lost and grasping may be compromised due to the deformation at the fingertips. Therefore, in the handling of deformable objects, grasping and manipulation must be performed in a collaborative way.

3.3 Interaction under Combined Vision, Force and Tactile Feedback

The way of automating robotic manipulators to handle deformable objects in an unknown configuration typically involves an initial exploratory action by vision sensors to guide the

robot arm toward the object, then visual information must be complemented by force/tactile measurements collected when a tactile probe or a dexterous hand comes in contact with the surface of the object. This supplementary data refines the knowledge about the position and orientation of the object and can provide an estimate of its elasticity or viscosity characteristics. All available information must be merged into a coherent model in order to allow the tuning of the feedback control loop that will guide the dexterous grasping and manipulation processes. Finally, tactile probing should continue during the operation using tactile sensors mounted on the fingertips to provide the necessary tactile sensitivity and sufficient dexterity to perform skillful manipulations of the deformable objects which may be of irregular shape and composition. Furthermore, visually monitoring of the task provides the necessary feedback to gauge how well the manipulator performed the task, or if an error has occurred, such as slippage. It is generally recognized that employing a multi-sensory system is the most effective way to model the deformation and estimate the object's shape and its attributes during the manipulation.

Vision systems can be used to detect the shape as well as to select proper picking points. Force/tactile sensors can also detect the shape or the contact. The contact state transitions based on force and vision sensors was studied in (Abegg *et al.*, 2000). They presented a systematic approach to manipulating a deformable linear object by capturing the transition graph representing the possible poses of a linear deformable object in contact with a convex polyhedron. Neurocomputing was used on tactile data in (Molina *et al.*, 2007) to model in real-time the stiffness of unknown deformable objects in the form of an anthropomorphic finger. Earlier attempts using vision systems for guiding a manipulator motion were concerned about making a knot with a rope (Inoue & Inaba, 1983), about estimating the 3D pose of deformable object using stereoscopic vision (Byun & Nagata, 1996), or about estimating the shape of a flexible beam while inserting it into a hole (Nakagaki *et al.*, 1996). Force/torque sensors were used also in (Kraus & McCarragher, 1997) to estimate the buckling of a linear deformable object when being inserted into a hole.

In recent efforts to solve the interaction control problem using multi-sensory feedback, a robust control law was developed in (Hirai *et al.*, 2001) for manipulation of 2D deformable parts using tactile and vision feedback to control the motion of a deformable object with respect to the position of selected reference points. Following this positioning approach, multiple points on a deformable object are guided to the final position. In a later study (Huang *et al.*, 2005), a position/force hybrid control method that incorporates visual information with force control was introduced to enable a robot arm with a flexible tool in the form of a hose to perform the contact process with the unknown 2D deformable object. Recent developments in (Foresti & Pellegrino, 2004) focused on automating the way of handling deformable objects using vision techniques only. Their vision system works along with a hierarchical self-organizing neural network to select proper grasping points in 2D.

3.4 Deformable Objects Manipulation in the Industry

In the recent years, robotic manipulation of deformable objects has been demonstrated in a variety of biomedical applications as well as in various manufacturing processes, especially in the electronic and electrical industry, as well as in the automotive, the aerospace, the leather, textile and garment, and in the food processing industries. In biomedical and industrial applications, there exist many manipulative operations that deal with different types of deformable objects ranging from viscoelastic objects, such as in a tele-surgery

operation, to industrial materials, such as string-like flexible objects, rubber parts, fabrics, paper sheets, and foods. In (Saadat & Nan, 2002) a detailed survey is reported about deformable objects applications based on the type of industry and on the complexity of the deformable object considered, that is 1D, 2D, or even 3D in very few cases. These applications were collected from the fabric and garments, aerospace, automotive, shoe/leather, food processing, and medical industries. Several similar applications of deformable objects manipulation were also presented in (Henrich & Worn, 2000) with the focus on 1D and 2D objects. A comprehensive review of robotic manipulation in the food industry, where it requires handling flexible and irregular food products, was conducted in (Chua *et al.*, 2003).

One of the early efforts to automate the process of handling linear deformable objects was reported by (Chen & Zheng, 1991) where a vision system was used to calculate deflection of a beam and the trajectory of the end-effector was computed for insertion of the beam into a hole and tracking the beam deformation. A numerical differentiation method was introduced to estimate the amount of deformation. This method of deformation assessment was used in developing strategies for beams assembly in (Zheng *et al.*, 1991). The focus of these authors in (Chen & Zheng, 1995) was also on inserting a flexible beam into a hole. The main contribution of this work is to maintain a minimum tool motion. In another research done by (Nakagaki *et al.*, 1996), a vision system was employed to determine the shape of the inserting beam and the force acting on the beam is calculated by minimizing the potential energy under geometric constraints. Through a further study by the same group (Nakagaki *et al.*, 1997) the amount of deformation of the object was determined by measuring its shape using stereovision, and the force acting on the tip using a force/torque sensor with less computation efforts. In the work presented in (Hirai *et al.*, 1995), the focus was in guiding a robotic manipulator in an industrial application to mimic an expert inserting a deformable hose onto a plug. In this experiment, force and vision sensors were used to examine the deformation of the hose during the task. In (Yue & Henrich, 2001) they addressed the problem of handling deformable linear objects to avoid acute vibration. In (Al-Yahmadi & Hsia, 2005) a sliding mode control scheme was proposed to enable cooperative manipulators generating forces and moments required to handle a flexible beam along a predefined trajectory. Their nonlinear model was capable of suppressing the existing vibrations due to manipulation. The work in the paper (Huang *et al.*, 2008) solved the problem of the mating process of electric cables in the context of linear deformable objects. In this paper they proposed a piecewise linear force model based on a combination of mass-spring systems to describe the matting process.

In the efforts related to automotive industry, an algorithm was derived in (Remde *et al.*, 1999b) which was successful at moving a robot gripper to a position close to the object in order to perform picking up and hanging of linear deformable objects. Their method required only one point on the object to perform the gripping. In a related automotive process, in (Abegg *et al.*, 2000) a feature-based visual control was developed. Sensory-motor primitives were used to evaluate the assembled part features and recognize its state. In another method discussed in (Byun & Nagata, 1996) the evaluation of the shape of the object is considered. This method is based on feature matching obtained from two images of an object from two different cameras. In (Kraus & McCarragher, 1997) they used a hybrid position/force solution in a sheet metal bending process. This method avoids the building of internal forces and exploits the natural impedance provided to the manipulator. Other

researchers have used multi-robots for the assembly of flexible sheet metal in the automotive industry. In the work carried out in (Mills & Ing, 1996), two robots were employed to grasp a flexible sheet metal payload. They used a dynamic model with the control method in order to implement an automobile body assembly. In (Sun *et al.*, 1998) the researchers used a hybrid position/force controller to operate two cooperative robots holding an aluminum sheet while performing the desired movement. The presence of an internal force control helped to avoid any possible damage to the system.

Applications of deformable objects handling was also reported in fabrics, garment, fur, rope, leather and the shoe industries. Tying knots in ropes and linking knitted fabrics are considered as another category of deformable linear objects for industrial applications. In the work conducted in (Hopcroft *et al.*, 1991), the robotic system described was able to tie knots in ropes of many different sizes, stiffnesses, and initial configurations. The manipulation of ropes as deformable objects which exhibit hysteresis was studied in (Matsuno *et al.*, 2005). They proposed a method to express a rope status using a topological model. On the same topic, in (Saha & Isto, 2007) a topological 3D probabilistic roadmap approach was introduced to plan the manipulator motion to tie knots. In (Wada *et al.*, 2001) they proposed a robust manipulation system using a simple PID controller for linking of knitted fabrics. In this swing operation multiple points on the deformable object should be guided to their location simultaneously to handle both bending and stretch deformations. A strategy has been derived in (Ono *et al.*, 1995) for unfolding a fabric piece based on cooperative sensing of touch and vision. A robotic system driven by a PD controller for garment inspection purpose was developed in (Yuen *et al.*, 2008). In the work carried out in (Foresti & Pellegrino, 2004), the researchers focused their attention on automating the way of picking furs through using vision techniques only. Their vision system can successfully select proper grasping points on fur pieces. In a leather industrial process, (Tout & Reedman, 1990) reported the use of a vision system capable of recognizing and distinguishing flat leather components with irregular shapes. In a roughing process for leather and shoe-making industry, a manufacturing process was developed in (Spencer, 1996) where the vision system was able to calculate the roughing trajectory and tool orientation. A rubber belt assembly system was implemented in (Miura & Ikeuchi, 1995) using vision sensor and they succeeded in automating the belt-pulley assembly process in 3D.

In the space industry, the problem of the manipulation of large flexible structures by space cooperative manipulators was addressed in (Tzeranis *et al.*, 2005). Their control algorithm ensures the application of the required forces on the flexible structure to perform the desired maneuvers while minimizing the effect of vibrations. Other than the references mentioned above, another set of publications dealing with interesting issues of applications involving 1D and 2D deformable objects manipulation were also reported. Amongst these applications, methods were proposed to extract and handle fruits and vegetables with specialized grippers in (Naghdy & Esmaili, 1996; Davis *et al.* 2008), to manipulate linear objects in (Zheng & Chen, 1993; Schmidt & Henrich, 2001; Schlechter & Henrich, 2002; Yue & Henrich, 2002; Acker & Henrich, 2003), to handle postal sacks in (Kazerooni & Foley, 2005), to manipulate bound book pages in (Young & Nourbakhsh, 2004), for knotting manipulation in (Wakamatsu *et al.*, 2002; 2004), and to automate unfolding fabrics in (Ono, 2000).

Despite substantial developments reported in the robotic manipulation process of industrial deformable objects, solving this problem still constitutes a challenging area of research because of the complexity of interactions between the deformable object and the manipulator, especially in the 3D case. Also the challenges are increased in some applications where hazardous or harsh environments require extra autonomy to completely remove human intervention. Several applications of this type are emerging for security and in space applications where it is difficult or impossible to operate safely without using advanced intelligent robotic systems. The harsh environment of space, the significant costs of life support systems for human beings and of "man-rating" space systems for safety, and the communications problems caused by the immense distances involved in interplanetary travel have given space programs additional incentives to develop systems of total automation beyond those commonly employed in industry. The sheer magnitude of many interesting applications, such as establishing green houses on the moon, requires massive automation and enhancement to the way robot manipulators deal with deformable objects.

4. Dexterous End-effectors

Dexterous end-effectors in form of a multi-fingered gripper or an anthropomorphic hand have a crucial role to play to support the manipulation of deformable objects. In order to meet the requirements imposed by the interaction with deformable objects, dexterous hands must have a sufficient number of degrees of freedom and be equipped with tactile sensors on their fingertips and palm.

A wide variety of robotic grippers/hands, both for industrial robotic applications and for humanoid robots, are already available to researchers, and some even start to be integrated into commercial applications. (Biagiotti *et al.*, 2004; Alba *et al.*, 2005) review some of the available state-of-the-art dexterous hands summarizing their features and specifications. (Okamura *et al.*, 2000) also surveys dexterous manipulation algorithms that can be applied to the case of deformable objects. Researchers consider the human hand as the reference model in the development of robotic hands where the focus is kept on either anthropomorphic design or dexterous design. Therefore, to decide on the suitable design, anthropomorphism, dexterity, speed and force/tactile sensing ability are the major factors to trade-off. However, in some applications the hand is also combined with data glove interface to allow for haptic tasks to be performed.

Any of these hands has the potential to be mounted on a suitable manipulator to develop an autonomous system for the purpose of handling deformable objects. However, there are still important issues that arise from the use and integration of such dexterous hands with off-the-shelf robotic manipulators. The main one comes from the fact that most robot arms commercially available do not offer an open architecture. Many newly developed dexterous hands also suffer from the same limitation, while their architecture, both mechanical and electronics, may not be suitable for a large range of deformable objects applications.

Beyond conducting laboratory experimentation, dexterous hands are expected to perform a wide range of applications in complex scenarios of rigid objects manipulation (Carrozza *et al.*, 2002; Namiki *et al.*, 2003a; Rothling *et al.*, 2007). For that reason, more sophisticated modeling and control schemes need to be achieved using their extended sensor information. The position of the object relative to the hand, and the point of contact between the manipulated object's surface and the finger remain the essential information to monitor.

However, the modeling and control problem of dexterous hand has been solved in general using simplified approaches based on heuristic rules inferred from practical task execution. This is because the mechanical model complexity and system nonlinearity make the optimization problem mathematically difficult to resolve. The development of fundamental optimized dexterous control methods for rigid objects manipulation was explored in (Yin *et al.*, 2003) within the framework of hybrid control scheme. The problem was solved in (Biagiotti *et al.*, 2003) based on impedance control with less mathematical constraints. More recently, in (Takahashi *et al.*, 2008), they developed an alternative robust force and position control for adaptive and stable grasping of solid objects with unknown mass and stiffness. Their algorithm can switch among the two position control and force control modes according to the amount of external force applied in order to achieve a successful grasp. In (Prats *et al.*, 2008) they applied the task frame formulation for manipulators (Baeten & De Schutter, 2003) at the hand level introducing a framework for dexterous manipulation of everyday rigid objects in household chores. Also, recent work has been presented to control a dexterous multi-fingered hand in deformable object manipulation application in (Minyong *et al.*, 2007; Mouri *et al.*, 2007) where hybrid impedance and force control strategies were applied to imitate the movements of human hand. In the experiments carried out, the stiffness characteristics of the object can be estimated in terms of mass-damped spring system's parameters using the impedance perception method introduced in (Kikuuwe & Yoshikawa, 2005). A robust and intuitive controller for multi-fingered hand was proposed in (Wimboeck *et al.*, 2006) to manipulate a deformable object where its motion and grasp forces are estimated based on a mass-spring system.

From the instrumentation point of view, recent vision techniques developed for dexterous hand grasping and manipulation of rigid objects can be found in (Saxena *et al.*, 2008). The grasping of unknown objects is learnt without need for it to be modeled. Grasping based on a kinematic model built from interactive perception is also achieved in (Katz & Brock, 2008). Multi-fingered dexterous hand grasping and force control using tactile feedback was also investigated in the literature for the rigid objects case. The work in (Maekawa *et al.*, 1996) and that in (Morales *et al.*, 2007) experimented with the integration of force/torque sensor combined with the tactile feedback. As a new trend, researchers started to consider in their setups and platforms a combination of vision with force/torque and tactile sensing to improve the grasping performance and stability of the multi-fingered hand (Allen *et al.*, 1999; Kragic *et al.*, 2003). The focus in (Namiki *et al.*, 2003b) was on developing dexterous hands with fast response grasping ability based on a suitable design and parallel sensory feedback to cope with the sudden dynamic changes in the environment. Several recent publications (Miller *et al.*, 2005; Ciocarlie *et al.*, 2008 a,b; Tegin *et al.*, 2008) introduce dexterous grasping techniques under prosthetic frameworks where the robotic hand gets more and more to resemble the human hand performance and dexterity. In some cases, the robotic hand is trained using real human interaction data. Finally, in order to evaluate the multi-fingered hand grasp quality and stability, measurement methods were established in the form of performance indices in (Kim *et al.*, 2004), which greatly help generalize and compare the development of the technology.

5. Conclusion

In an attempt to support the ongoing effort of development for robotic solutions to the manipulation of deformable objects with multi-sensory feedback, this chapter reviewed the major trends adopted over the last decades in autonomous robotic interaction, which remains mainly guided by vision and force/tactile sensing. This extensive survey aimed at providing and classifying a critical list of relevant references that broadly cover the field. Starting from an overview of classical modeling and control techniques with application to the robotic manipulation of rigid objects, the review investigated how these approaches are being extended to the case of deformable objects manipulation. The main issues related with the significant differences between rigid and non-rigid objects were highlighted and consideration was given to a wide range of solutions that have been proposed, often in direct correspondence with a specific application.

It is noticeable that most of the control methods available in the literature are applied to manipulate 1D and 2D deformable objects. The study of how to control a robot arm to handle a 3D deformable object still remains an open subject. Only a few early attempts to produce a generalized approach for handling 3D deformable objects were reported. Also, most of the proposed solutions currently available address the modeling problem of 3D deformable objects without attempting to solve the control problem simultaneously. Furthermore, the manipulation process does not involve any dexterity considerations.

The study of these aspects is essential for the current effort of the robotic research community to establish a novel framework for the purpose of dexterous handling of 3D deformable objects. It involves the development of sophisticated multi-sensory systems to work in coordination with a robot arm and hand, taking into account their mechanical structure and control scheme, that influence the accuracy, and the dexterity. The integration of such complementary techniques will ensure that more elaborate manipulation can be achieved in the near future.

6. References

- Abegg, F. ; Remde, A. & Henrich, D. (2000). Force and Vision Based Detection of Contact State Transitions, in *Robot Manipulation of Deformable Objects*, D. Henrich and H. Worn, (Eds.), Springer-Verlag, London.
- Acker, J. & Henrich, D. (2003). Manipulating Deformable Linear Objects: Characteristic Features for Vision-based Detection of Contact State Transitions, *Proc. IEEE Int. Symp. on Assembly and Task Planning*, pp.204-209.
- Alba, D. ; Armad, M. & Ponticelli, R. (2005). An Introductory Revision to Humanoid Robot Hands, in *Climbing and Walking Robots*, Springer Berlin Heidelberg.
- Al-Jarrah, O. & Zheng, Y. (1998). Intelligent Compliant Motion Control, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 28, pp. 116-122.
- Allen, P. K. ; Miller, A. T. , Oh, P. Y. & Leibowitz, B. S. (1999). Integration of Vision, Force and Tactile Sensing for Grasping, *Int. J. of Intelligent Machines*, Vol. 4, pp. 129-149.
- Al-Yahmadi, A. S. & Hsia, T.C. (2005). Modeling and Control of Two Manipulators Handling a Flexible Beam, *Proc. of World Academy of Science, Eng. and Tech.*, Vol. 6, pp. 147-150.
- Anderson, R. J. & Spong, M. W. (1988). Hybrid Impedance Control of Robotic Manipulators, *IEEE J. of Robotics and Automation*, Vol. 4, pp. 549-556.

- Anh, N. P. T.; Arimoto, S., Han, H. Y. & Kawamura, S. (1999). Control of Physical Interaction Between a Deformable Finger-tip and a Rigid Object, *Proc. IEEE Int. Conf. on System, Man, and Cybernetics*, pp. 812-817.
- Arai, F.; Rong, L. & Fukuda, T. (1993). Trajectory Control of Flexible Plate using Neural Network, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 155-160.
- Arai, F.; Niu, H. & Fukuda, T. (1997). Performance Improvement of Flexible Material Handling Robot by Error Detection and Replanning, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2938-2943.
- Bachiller, M.; Cerrada, C. & Cerrada, J. A. (2007). Designing and Building Controllers for 3D Visual Servoing Applications under a Modular Scheme, in *Industrial Robotics: Programming, Simulation and Applications*, K. H. Low, (Ed.), *pro literature Verlag*.
- Baeten, J. & De Schutter, J. (2003). Integrated Visual Servoing and Force Control - The Task Frame Approach, *Springer Tracts in Advanced Robotics*, Vol. 8, B. Siciliano, O. Khatib, and F. Groen, (Eds.), Springer, Berlin.
- Barbagli, F.; Salisbury, K. & Devengenzo, R. (2003). Enabling Multi-finger, Multi-hand Virtualized Grasping, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 809-815.
- Biagiotti, L.; Liu, H., Hirzinger, G. & Melchiorri, C. (2003). Cartesian Impedance Control for Dexterous Manipulation, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3270-3275.
- Biagiotti, L.; Lotti, F., Melchiorri, C. & Vassura, G. (2004). *How Far is the Human Hand: A Review on Anthropomorphic Robotic End-effectors*, University of Bologna, Italy.
- Bicchi, A. (1995). On the Closure Properties of Robotics Grasping, *Int. J. of Robotics Research*, Vol. 14, pp. 319-334.
- Bicchi, A. (2000). Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Toward Simplicity, *IEEE Trans. on Robotics and Automation*, Vol. 16, pp. 652-662.
- Bicchi, A. & Kumar, V. (2000). Robotic Grasping and Contact: A Review, *Proc. IEEE Int. Conf. On Robotics and Automation*, pp. 348-353.
- Buss, M.; Hashimoto, H. & Moore, J. (1996). Dexterous Hand Grasping Force Optimization, *IEEE Trans. on Robotics and Automation*, Vol.12, pp. 406-418.
- Byars, E. F.; Snyder, R. D. & Plants, H. L. (1983). *Engineering Mechanics of Deformable Bodies*, Harper & Row Publishers, New York.
- Byun, J. E. & Nagata, T. (1996). Determining the 3D Pose of a Flexible Object by Stereo Matching of Curvature Representations, *J. of Pattern Recognition*, Vol. 29, pp. 1297-1308.
- Carrozza, M.C.; Vecchi, F., Roccella, S., Barboni, L., Cavallaro, E., Micera, S. & Dario, P., (2002). The ADAH Project: An Astronaut Dexterous Artificial Hand to Restore the Manipulation Abilities of the Astronaut, *Proc. 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*.
- Castano, A. & Hutchinson, S. (1994). Visual Compliance: Task-directed Visual Servo Control, *IEEE Trans. on Robotics and Automation*, Vol. 10, pp. 334-342.
- Chen, C. & Zheng, Y. (1991). Deformation Identification and Estimation of One-dimensional Objects by Using Vision Sensors, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2306-2311.
- Chen, M. & Zheng, Y. (1995). Vibration-free Handling of Deformable Beams by Robot End-effectors, *J. of Robotic Systems*, Vol. 12, pp.331-347.

- Chen, Y. C.; Walker, I. D. & Cheatham, J. B. (1993). Grasp Synthesis for Planar and Solid Objects, *J. of Robotic Systems*, Vol. 10, pp. 153-186.
- Chiaverini, S. ; Siciliano, B. & Villani, L. (1994). Force/Position Regulation of Compliant Robot Manipulators, *IEEE Trans. on Automatic Control*, Vol. 39, pp. 647-652.
- Chua, P. Y.; Ilschner, T. & Caldwell, D. G. (2003). Robotic Manipulation of Food Products – a Review, *Int. J. of Industrial Robot*, Vol. 30, pp. 345-354.
- Ciocarlie. M. T. & Allen, P. K. (2008a). On-Line Interactive Dexterous Grasping, *Proc. Int. Conf. on Haptics: Perception, Devices and Scenarios*, pp. 104-113.
- Ciocarlie, M. T. ; Clanton, S. T. , Spalding, M. C. & Allen, P. K. (2008b), Biomimetic Grasp Planning for Cortical Control of a Robotic Hand, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2271-2276.
- Cornellà, J. ; Suarez, R., Carloni, R. & Melchiorri, C. (2008). Dual Programming Based Approach for Optimal Grasping Force Distribution, *J. of Mechatronics*, Vol. 18, pp. 348-356.
- Cretu, A. -M. ; Payeur, P. & Petriu, E. M. (2008). Neural Network Mapping and Clustering of Elastic Behavior from Tactile and Range Imaging for Virtualized Reality Applications, *IEEE Trans. on Instrumentation and Measurement*, Vol. 57, no 9, pp. 1918-1928.
- Cutkosky, M. R. (1989). On Grasp Choice, Grasp model, and the Design of Hands for Manufacturing Task, *IEEE Trans. on Robotics and Automation*, Vol. 5, pp. 269-279.
- Davis, S. ; Gray, J.O. & Caldwell, D. G. (2008). An End-effector Based on the Bernoulli Principle for Handling Sliced Fruit and Vegetables, *Robotics and Computer-Integrated Manufacturing*, Vol. 24, pp. 249-257.
- Deng, L.; Janabi-Sharifi, F. & Hassanzadeh, I. (2005). Comparison of Combined Vision/Force Control Strategies for Robot Manipulators, *Proc. of SPIE Int. Symp. Optomechatronic Technologies: Optomechatronic Systems Control Conference*, pp 1-12.
- Ding, D. ; Liu, Y. H. & Wang, S. (2001). Computation of 3D Form-closure Grasps, *IEEE Trans. on Robotics and Automation*, Vol. 17, pp. 515-522.
- Diolaiti, N. ; Melchiorri, C. & Stramigioli, S. (2005). Contact Impedance Estimation for Robotic Systems, *IEEE Trans on Robotics*, Vol. 21, pp. 925-935.
- Foresti, G. L. & Pellegrino, F. A. (2004). Automatic Visual Recognition of Deformable Objects for Grasping and Manipulation, *IEEE Trans. on Systems, Man, and Cybernetics: Applications and Reviews*, Vol. 34, pp. 325-333.
- Garg, D. P. & Kumar, M. (2003) . Object Classification via Stereo Vision Sensing in a Flexible Manufacturing Work Cell, *Proc. IEEE Int. Conf. on Mechatronics and Machine Vision in Practice*.
- Garg S. & Dutta, A. (2006). Grasping and Manipulation of Deformable Objects Based on Internal Force Requirements, *Int. J. of Advanced Robotic Systems*, Vol. 3, pp. 107-114.
- Gibson, S. F. & Mirtich, B. (1997). *A Survey of Deformable Modeling in Computer Graphics*, MERL Technical Report, TR97-19.
- Gopalakrishnan, K. & Goldberg, K. (2005). D-Space and Deform Closure Grasps of Deformable Parts, *Int. J. of Robotics Research*, Vol. 24, pp. 899-910.
- Hager, G.; Hutchinson, S. & Corke, P. (1996). Tutorial TT3: A Tutorial on Visual Servo Control, *Proc. Conf. IEEE Robotics and Automation*.
- Hashimoto, K. (2003). A Review on Vision-based Control of Robot Manipulators, *Advanced Robotics*, Vol. 17, pp. 969-991.

- Henrich, D. ; Ogasawara, T. & Worn, H. (1999). Manipulating Deformable Linear Objects-Contact States and Point Contacts, *Proc. IEEE Int. Symp. on Assembly and Task Planning*, pp. 198 - 204.
- Henrich, D. & Worn, H. (Eds.) (2000)., *Robot Manipulation of Deformable Objects*, Springer-Verlag, London.
- Hill, J. & Park, W. T. (1979). Real-time Control of a Robot with a Mobile Camera, *Proc. Int. Symp. on Industrial Robots*, pp. 233-246.
- Hirai, S. ; Wakamatsu, H. & Iwata, K. (1994). Modeling of Deformable Thin Parts for Their Manipulation, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2955-2960.
- Hirai, S. ; Noguchi, H. & Iwata, K. (1995). Transplantation of Human Skillful Motion to Manipulators in Insertion of Deformable Tubes, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1900-1905.
- Hirai, S. ; Tsuboi, T. & Wada, T. (2001). Robust Grasping Manipulation of Deformable Objects, *Proc. IEEE Int. Symp. On Assembly and Task Planning*, pp. 411-416.
- Hogan, N. (1987). Stable Execution of Contact Tasks Using Impedance Control, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1047-1054.
- Hopcroft, J. ; Kearney, J. & Krafft, D. (1991). A Case Study of Flexible Object Manipulation, *Int. J. of Robotics Research*, Vol. 10, pp. 41-50.
- Hosoda, K. ; Igarashi, K. & Asada, M. (1998). Adaptive Hybrid Control for Visual and Force Servoing in an Unknown Environment, *IEEE Robotics and Automation Magazine*, Vol. 5, pp. 39-43.
- Howard, A. H. & Bekey, G. (2000). Intelligent Learning for Deformable Object Manipulation, *J. Autonomous Robots*, Vol. 9, pp. 51-58.
- Huang, J.; Todo, I. & Yabuta, T. (2005). Position/Force Hybrid Control of a Manipulator with a Flexible Tool Using Visual and Force Information, in *Cutting Edge Robotics*, V. Kordic, A. Lazinica, and M. Merdan, (Eds.), pp. 611-628, ARS/pIV, Germany.
- Huang, J.; Fukuda, T. & Matsuno, T. (2008). Model-Based Intelligent Fault Detection and Diagnosis for Mating Electric Connectors in Robotic Wiring Harness Assembly Systems, *IEEE/ASME Trans. on Mechatronics*, Vol. 13, pp. 86-94.
- Hutchinson, S. ; Hager, G. & Corke , P. (1996). A Tutorial on Visual Servo Control, *IEEE Trans. on Robotics and Automation*, Vol. 12, pp. 651-670.
- Inoue H. & Inaba, M. (1983). Hand-eye Coordination in Rope Handling, *Robotics Research*, M. Brady, and R. Paul, (Eds.), pp. 163-174. *The MIT Press*.
- Inoue T. & Hirai, S. (2008). Experimental Investigation of Mechanics in Soft-fingered Grasping and Manipulation, in *Experimental Robotics*, O. Khatib, V. Kumar, and D. Rus, (Eds.), Springer-Verlag, Berlin.
- Ishikawa, T.; Sakane, S., Sato, T. & Tsukune, H. (1996). Estimation of Contact Position Between a Grasped Object and the Environment Based on Sensor Fusion of Vision and Force, *Proc. IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 116-123.
- Javad, D. & Najarian, S. (2005). Advances in Tactile Sensors Design/Manufacturing and its Impact on Robotics Applications - A Review, *Int. J. of Industrial Robot*, Vol. 32, pp. 268-281.
- Kaneko, K.; Harada, K. & Kanehiro, F. (2007). Development of Multi-Fingered Hand for Life-size Humanoid Robots, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 913-920.

- Katic, D. & Vukobratovic, M. (1998). A Neural Network Based Classification of Environment Dynamics Models for Compliant Control of Manipulation Robots, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 28, pp. 58-69.
- Katic, D. & Vukobratovic, M. (2003). *Intelligent Control of Robotic Systems*, Kluwer.
- Kawai, H. ; Murao, T. & Fujita, M. (2008). Passivity-based Dynamic Visual Force Feedback Control for Fixed Camera Systems, *Proc. IEEE Int. Symp. on Intelligent Control*, pp. 426-431.
- Katz, D. & Brock, O. (2008). Manipulating Articulated Objects with Interactive Perception, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 272-277.
- Kazerooni, H. & Foley, C. (2005). A Robotic Mechanism for Grasping Sacks, *IEEE Trans. on Automation Science and Engineering*, Vol. 2, pp. 111-120.
- Khatib, O. & Burdick, J. (1986). Motion and Force Control of Robot Manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1381-1386.
- Kikuuwe, R. & Yoshikawa, T. (2005). Robot Perception of Impedance, *Journal of Robotic Systems*, Vol. 22, pp. 231-247.
- Kim, J. Y. & Cho, H. S. (2000). A Neural Net-based Assembly Algorithm for Flexible Parts Assembly, *J. of Intelligent and Robotic Systems*, Vol. 29, pp. 133-160.
- Kim, B. H.; Yi, B. J. , Oh, S. R & Suh, I. H. (2004). Non-dimensionalized Performance Indices Based on Optimal Grasping for Multi-fingered Hand, *Mechatronics*, Vol. 14, 255-280.
- Kosuge, K.; Sakai, M. ,Kanitani, K. , Yoshida, H. & Fukuda, T. (1995). Manipulation of a Flexible Object by Dual Manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 318-322.
- Kragic, D. & Christensen, H. I. (2002). Survey on Visual Servoing for Manipulation, Technical Report, Computational Vision and Active Perception Laboratory, Fiskartorpsv 15 A 10044 Stockholm, Sweden, Tech. Rep. ISRNKTHA/NA/P-02/01-SE.
- Kragic, D.; Crinier, S., Brunn, D. & Christensen, H. I. (2003). Vision and Tactile Sensing for Real World Tasks, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1545-1550.
- Kraus W. & McCarragher, B. J. (1996). Force Fields in the Manipulation of Flexible Materials, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2352-2357.
- Kraus W. & McCarragher, B. J. (1997). Case Studies in the Manipulation of Flexible Parts Using a Hybrid Position/Force Approach, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 367-372.
- Lang, J.; Pai, D. & Woodham, R. (2002). Acquisition of Elastic Models for Interactive Simulation, *Int. J. of Robotics Research*, Vol. 21, pp. 713-733.
- Lewis, F.; Dawson, D. & Abdallah, C. (2004). *Robot Manipulator Control: Theory and Practice*, Marcel Dekker, New York.
- Li, Y. & Kao, I. (2001). A Review of Modeling of Soft-contact Fingers and Stiffness Control for Dexterous Manipulation in Robotics, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3055-3060.
- Li, H. F. ; Ceglarek, D. & Shi, J. (2002). A Dexterous Part Holding Model for Handling Compliant Sheet Metal Parts, *ASME Trans., J. of Manufacturing Science and Engineering*, Vol. 124, pp. 109-118.
- Li, Z.; Chen, W. & Luo, J. (2008). Adaptive Compliant Force-Motion Control of Coordinated Non-holonomic Mobile Manipulators Interacting with Unknown Non-rigid Environments, *J. of Neurocomputing*, Vol. 71, pp. 1330-1344.

- Lippiello, V. ; Siciliano, B. & Villani , L. (2007a). A Position-based Visual Impedance Control for Robot Manipulators, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2068-2073.
- Lippiello, V. ; Siciliano, B. & Villani , L. (2007b). Robot Force/Position Control with Force and Visual Feedback, *Proc. of European Control Conference*, pp. 3790-3795.
- Liu, G. & Li, Z. (2004). Real-time Grasping-force Optimization for Multifingered Manipulation: Theory and Experiments, *IEEE/ASME Trans. on Mechatronics*, Vol. 9, pp. 65-77.
- Liu, Y.; Lam, M. & Ding, D. (2004). A Complete and Efficient Algorithm for Searching 3D Form-closure Grasps in the Discrete Domain, *IEEE Trans. Robotics and Automation*, Vol. 20, pp. 805-816.
- Luo Y. & Nelson, B. J. (2001). Fusing Force and Vision Feedback for Manipulating Deformable Objects, *J. of Robotic Systems*, Vol. 18, pp. 103-117.
- Luo, Z. & Ito, M. (1993). Control Design of Robot for Compliant Manipulation on Dynamic Environments, *IEEE Trans. on Robotics and Automation*, Vol. 9, pp. 286-296.
- Maekawa, H., Tanie, K. & Komoriya, K. (1996). Dynamic Grasping Force Control Using Tactile Feedback for Grasp of Multifingered Hand, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2462-2469.
- Mandal, N. & Payandeh, S. (1995). Control Strategies for Robotic Contact Tasks: An Experimental Study, *J. Robotic Systems.*, Vol. 12, pp. 67-92.
- Mason, M. T. & Salisbury, J. K. (1986). Robots Hands and the Mechanics of Manipulation , *The MIT Press*, MA.
- Mason, M. T. (2001). Mechanics of Robotic Manipulation, *The MIT Press*.
- Matsuno, T. ; Tamaki, D., Arai, F. & Fukuda, T. (2005). Manipulation of Deformable Linear Objects with Knot Invariant to Classify Condition, *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 893-898.
- McClamroch, N. H. (1986). Singular Systems of Differential Equations as Dynamic Models of Constrained Robot Systems, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 21-28.
- Meer, D. W. & Rock, S. M. (1994). Experiments in Object Impedance Control for Flexible Objects, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1222-1227.
- Melchiorri, C. & Vassura, G. (1992). Mechanical and Control Features of the University of Bologna Hand Version 2, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 187-193.
- Mezouar, Y. ; Prats, M. & Martinet, P. (2007). External Hybrid Vision/Force Control, *Proc. Int. Conf. on Advanced Robotics*.
- Miller, A. ; Allen, P. K., Santos, V. & Valero-Cuevas, F. (2005). From Robot Hands to Human Hands: A Visualization and Simulation Engine for Grasping Research, *Int. J. Industrial Robot*, Vol. 32, pp. 55-63.
- Mills, J. K. & Ing, G. L. J. (1996). Dynamic Modeling and Control of a Multi-robot System for Assembly of Flexible Payloads with Applications to Automotive Body Assembly, *J. of Robotic Systems*, Vol. 13, pp. 817-836.
- Minyong, P. ; Mouri, K., Kitagawa, H. , Miyoshi, T. & Terashima, K. (2007). Hybrid Impedance and Force Control for Massage System by Using Humanoid Multi-fingered Robot Hand, *Proc. IEEE/RSJ Int. Conf. on Systems, Man and Cybernetics*, pp. 3021-3026.
- Mirza K. & Orin, D. E. (1990). Control of Force Distribution for Power Grasp in the Digits System, *Proc. IEEE Conf. on Decision and Control*, pp. 1960-1965.

- Miura, J. & Ikeuchi, K. (1995). Assembly of Flexible Objects without Analytical Models, *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp.77-83.
- Molina, J. L. P. ; Gonzalez, A. G. , Moran, J. C., Coronado, J. L. & Gorce, P. (2007). A Neural Tactile Architecture Applied to Real-time Stiffness Estimation for a Large Scale of Robotic Grasping Systems, *J. of Intelligent and Robotic Systems*, Vol. 49, pp. 311-323.
- Morales, A.; Prats, M., Sanz, P. & Pobil, A. P. (2007). An Experiment in the Use of Manipulation Primitives and Tactile Perception for Reactive Grasping, *Manipulation Workshop: Sensing and Adapting to the Real World*.
- Morel, G. ; Malis, E. & Boudet, S. (1998). Impedance Based Combination of Visual and Force Control, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1743-1748.
- Moreno, M. A. ; Yu, W. & Poznyak, A. S. (2001). Stable 3D Visual Servoing: An Experimental Comparison, *Proc. IEEE Int. Conf. on Control Applications*, pp. 218-223.
- Mouri, K.; Terashima, K., Minyong, P., Kitagawa, H. & Miyoshi, T. (2007). Identification and Hybrid Impedance Control of Human Skin Muscle by Multi-fingered Robot Hand, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2895-2900.
- Naghdy, F. & Esmaili, M. (1996). Soft Fruit Grading Using a Robotics Gripper, *Int. J. of Robotics and Automation*, Vol. 11, pp. 93-101.
- Nakagaki, H. ; Kitagaki, K. & Tsukune, H. (1995). Study of Insertion Task of a Flexible Beam into a Hole, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 330-335.
- Nakagaki, H. ; Kitagaki, K. , Ogasawara, T. & Tsukune, H. (1996). Study of Insertion Task of a Flexible Wire into a Hole by Using Visual Tracking Observed by Stereo Vision, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3209-3214.
- Nakagaki, H. ; Kitagaki, K. , Ogasawara, T. & Tsukune, H. (1997). Study of Deformation and Insertion Tasks of a Flexible Wire, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2397-3402.
- Natale, C. (2003). Interaction Control of Robot Manipulators: Six-degrees-of-freedom Tasks, *Springer Tracts in Advanced Robotics*, Vol. 3, Springer, Berlin.
- Namiki, A. ; Hashimoto K. & Ishikawa, M. (2003a). A Hierarchical Control Architecture for High-Speed Visual Servoing, *Int. J. of Robotics Research*, Vol. 22, pp. 873-888.
- Namiki, A. ; Imai, Y., Ishikawa, M. & Kaneko, M. (2003b). Development of a High-speed Multifingered Hand System and Its Application to Catching, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2666-2671.
- Nelson, B. J. & Khosla, P. K. (1996). Force and Vision Resolvability for Assimilating Disparate Sensory Feedback, *IEEE Trans. on Robotics and Automation*, Vol. 12, pp. 714-731.
- Nguyen, V. D. (1988). Constructing Force-closure Grasps, *Int. J. of Robotics Research*, Vol. 7, pp. 3-16.
- Nguyen, W. & Mills, J. K. (1996). Multi-robot Control for Flexible Fixtureless Assembly of Flexible Sheet Metal Auto Body Parts, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2340-2345.
- Okamura, A. M.; Smaby, N. & Cutkosky, M. R. (2000). An Overview of Dexterous Manipulation, *Proc. IEEE Int Conf on Robotics and Automation*, pp. 255-262.
- Olsson, T.; Johansson, R. & Robertsson, A. (2004). Flexible Force-vision Control for Surface Following Using Multiple Cameras, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pp. 798-803.

- Ono, E.; Kita, N. & Sakane, S. (1995). Strategy for Unfolding a Fabric Piece by Cooperative Sensing of Touch and Vision, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 441-445.
- Ono, E. (2000). Automated Swing System and Unfolding Fabric, in *Robot Manipulation of Deformable Objects*, D. Henrich and H. Worn, (Eds.), Springer-Verlag, London.
- Papanikolopoulos, N. ; Khosla, P. K. & Kanade ,T. (1993). Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision, *IEEE Trans. on Robotics and Automation*, Vol. 9, pp. 14-35.
- Patton, R. Swern, F. , Tricamo, S. & Van Der Veen A. (1992). Automated Cloth Handling Using Adaptive Force Feedback, *J. of Dynamic Systems Measurements and Control*, Vol. 114, pp. 731-733.
- Payeur, P. ; C. Pasca, Cretu, A.-M. & Petriu, E.M. (2005). Intelligent Haptic Sensor System for Robotic Manipulation, *IEEE Trans. on Instrumentation and Measurement*, Vol. 54, pp. 1583-1592.
- Pentland, A. & Williams, J. (1989). Good Vibrations: Modal Dynamics for Graphics and Animation, *ACM SIGGRAPH Computer Graphics*, Vol. 23, pp. 215-222.
- Pentland, A. & Sclaroff, S. (1991). Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 715-729.
- Pichler, A. & Jagersand, M. (2000). Uncalibrated Hybrid Force-Vision Manipulation, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1866-1871.
- Pires, J. N. (2006). Robot Manipulators and Control Systems, in *Industrial Robots Programming Building Applications for the Factories of the Future*, Springer.
- Peer, A.; Stanczyk, B., Unterhinninghofen, U. & Buss, M. (2006). Tele-assembly in Wide Remote Environments, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.
- Pomares, J.; Garcia, G.J. & Torres, F. (2007). A Robust Approach to Control Robot Manipulators by Fusing Visual and Force Information, *J. of Intelligent and Robotic Systems*, Vol. 48, pp. 437-456.
- Prats, M. ; Martinet, P., del Pobil, A. P. & Lee, S. (2008). Robotic Execution of Everyday Tasks by Means of External Vision/Force Control, *Intelligent Service Robotics*, pp. 253-266.
- Prattichizzo, D. Mercorelli, P. Bicchi, A. & Vicino, A. (1997). On the Geometric Control of Internal Forces in Power Grasps, *Proc. IEEE Conf. on Decision and Control*, pp. 1942-1947.
- Raibert, M. H. & Craig, J. J. (1981). Hybrid Position/Force Control of Manipulators, *ASME J. of Dynamic Sys. Meas. Control*, Vol. 102, pp. 126-133.
- Remde, A.; Henrich, D. & Worn H. (1999a). Picking Up Deformable Linear Objects with Industrial Robots, *ISR Int. Symp. on Robotics*.
- Remde, A.; Henrich, D. & Worn H. (1999b). Manipulating Deformable Linear Objects Contact State Transitions and Transition conditions, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1450-1455.
- Reznik, D. & Laugier, C. (1996). Dynamic Simulation and Virtual Control of a Deformable Fingertip, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1669-1674.
- Rothling, F; Haschke, R. , Steil, J. & Ritter, H. (2007).. Platform Portable Anthropomorphic Grasping with the Bielefeld 20-DOF Shadow and 9-DOF TUM Hand, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2951-2956.

- Saadat, M. & Nan, P. (2002). Industrial Applications of Automatic Manipulation of Flexible Materials, *Int. J. of Industrial Robot*, vol. 29, pp. 434-442.
- Saha, M. & Isto, P. (2007). Manipulation Planning for Deformable Linear Objects, *IEEE Trans. on Robotics*, Vol. 23, pp. 1141-1150.
- Salisbury, J. K. & Roth, B. (1983). Kinematic and Force Analysis of Articulated Mechanical Hands, *ASME J. of Mech. Transm. Autom. Des. Design*, Vol. 105, pp.33-41.
- Saut, J.; Remond, C. Perdereau, V. & Drouin, M. (2005). Online Computation of Grasping Force in Multi-fingered Hands, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1223-1228.
- Saxena, A. ; Driemeyer, J. & Ng, A. Y. (2008), Robotic Grasping of Novel Objects using Vision, *Int. J. of Robotics Research*, Vol. 27, pp. 157-173.
- Schiele, A. & De Bartolomei, M. (2006). A Ground Station for Human-Robot-Interaction Research with Exoskeleton, *Proc. Workshop on Advanced Space Technologies for Robotics and Automation*.
- Schlechter, A. & Henrich, D. (2002). Manipulating Deformable Linear Objects: Manipulation Skill for Active Damping of Oscillations, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1541-1546.
- Schmidt, T. W. & Henrich, D. (2001). Manipulating Deformable Linear Objects: Robot Motions in Single and Multiple Contact Points, *Proc. IEEE Int. Symp. on Assembly and Task Planning*, pp.435-441.
- Seraji, H.; Lim, D. & Steele, R. (1996). Experiments in Contact Control, *J. of Robotic Systems*, Vol. 13, pp. 53-73.
- Shimoga, K. B. (1996). Robot Grasp Synthesis Algorithms: A Survey, *Int. J. Robotics Research*, Vol. 15, pp. 230-266.
- Shimoga, K. B. & Goldenberg, A. A. (1996). Soft Robotics Fingertips I: A Comparison of Construction Materials, *Int. J. of Robotics Research*, Vol. 15, pp. 320-334.
- Siciliano, B. & Villani, L. (1997). An Output Feedback Parallel Force/Position Regulator for Robot Manipulator in Contact with a Compliant Environment, *Systems and Control Letters*, Vol. 29, pp. 295-300.
- Siciliano, B. & Villani, L. (1999). *Robot Force Control*, Kluwer Academic Publishers, MA.
- Siciliano, B. & Khatib, O. (2008). *Handbook of Robotics*, Springer.
- Smits, R. ; Bruyninckx, H., Meeussen, W., Baeten, J., Slaets, P. & De Schutter, J. (2006). Model Based Position-Force-Vision Sensor Fusion for Robot Compliant Motion Control, *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp 501-506.
- Smits, R. ; Fioravanti, D. , De Laet, T. , Allotta, B. , Bruyninckx, H. & De Schutter, J. (2008). Image-Based Visual Servoing with Extra Task Related Constraints in a General Framework for Sensor-Based Robot Systems, in *Unifying Perspectives in Computational and Robot Vision*, D. Kragic and V. Kyrki , (Eds.) , Springer US.
- Spencer, J. E. Jr. (1996). Robotics Technology and the Advent of Agile Manufacturing Systems in the Footwear Industry, *Assembly Automation*, Vol. 16, pp. 10-15.
- Sun, D. ; Mills, J, K, & Liu, Y. H. (1998). Hybrid Position and Force Control of Two Industrial Robots Manipulating a Flexible Sheet: Theory and Experiment, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1835-1840.

- Takahashi, T. ; Tsuboi, T., Kishida, T., Kawanami, Y., Shimizu, S. Iribe, M., Fukushima, T. & Fujita, M. (2008). Adaptive Grasping by Multi Fingered Hand with Tactile Sensor Based on Robust Force and Position Control, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 264-271.
- Tarokh, M. & Bailey, S. (1996). Force Tracking with Unknown Environment Parameters Using Adaptive Fuzzy Controllers, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 270-275.
- Tegin, J. & Wikander, J. (2005). Tactile Sensing in Intelligent Robotic Manipulation- A Review, *Int. J. of Industrial Robot*, Vol. 32, pp. 64-70.
- Tegin, J.; Ekvall, S. , Kragic, D., Wikander, J. & Iliev, B. (2009). Demonstration-Based Learning and Control for Automatic Grasping, *Intelligent Service Robotics*, pp. 23-30.
- Terauchi, M. ; Tanaka, Y. & Tsuji, T. (2008). Real-Time Hand and Eye Coordination for Flexible Impedance Control of Robot Manipulator, in *RobVis 2008*, G. Sommer and R. Klette, (Eds.), *Springer-Verlag*, Berlin.
- Terzopoulos, D. ; Platt, J., Barr, A. & Fleischer, K. (1987).. Elastically Deformable Models, *Proc. of the Annual ACM SIGGRAPH '87 Conference*, Vol. 21. pp. 205-214.
- Thomas, U. ; Molkenstruck, S., Iser, R. & Wahl, F. M. (2007). Multi Sensor Fusion in Robot Assembly Using Particle Filters, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3837-3943.
- Tremblay, M. R.; Cutkosky, M. R. (1993). Estimating Friction Using Incipient Slip Sensing During a Manipulation Task, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 363-368.
- Trinkle, J. C.; Abel, J. M. & Paul, R. P. (1988). An Investigation of Frictionless Enveloping Grasping in the Plane, *Int. J. of Robotics Research*, Vol. 7, pp. 33-51.
- Tokumoto, S. ; Fujita, Y. & Hirai, S. (1999). Deformation Modeling of Viscoelastic Objects for Their Shape Control, *Proc. IEEE Int. Conf. on Robotics & Automation*, pp. 767-772.
- Tout, N. R. & Reedman, D. C. (1990). Intelligent Automated Assembly of the Upper Parts of Shoes, *Proc. IMechE Int. Conf. Mechatronic: Designing Intelligent Machines*, pp. 103-108.
- Tzeranis, D. ; Ishijima, Y. & Dubowsky, S. (2005). Manipulation of Large Flexible Structural Modules by Space Robots Mounted on Flexible Structure, *Proc. Int. Sym. on Artificial Intelligence, Robotics and Automation in Space*.
- Vassura, G. & Bicchi, A. (1989). Whole Hand Manipulation: Design of an Articulated Hand Exploiting all its Parts to Increase Dexterity, in *Robots and Biological Systems*, NATO-ASI Series, P. Dario, G. Sandini and P. Aebischer, (Eds), *Springer-Verlag*, Berlin.
- Venkataraman, S.; Gulati, S., Barhen, J. & Toomarian, N. (1993). A Neural Network Based Identification of Environment Models for Compliant Control of Space Robots, *IEEE Trans. on Robotics and Automation*, Vol. 9, pp. 685-697.
- Vukobratovic, M. ; Potkonjak, V. & Matijevic , V. (2003). Dynamics of Robots with Contact Tasks, *Kluwer Academic Publishers*.
- Wada, T. ; Hirai, S., Kawamura, S. & Kamiji, N. (2001). Robust Manipulation of Deformable Objects by a Simple PID Feedback, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.85-90.
- Wakamatsu, H. ; Hirai, S. & Iwata, K. (1995). Modeling of Linear Objects Considering Bend, Twist and Extensional Deformations, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 433-438.

- Wakamatsu, H.; Hirai, S. & Iwata, K. (1996). Static Analysis of Deformable Object Grasping Based on Bounded Force Closure, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3324-3329.
- Wakamatsu, H. ; Matsumura, T. & Hirai, S. (1997). Dynamic Analysis of Rod-like Object Deformation Towards their Dynamic Manipulation, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 196-201.
- Wakamatsu, H. ; Tanaka, Y., Tsumaya, A., Shirase, K. & Arai, E. (2002). Representation and Planning of Deformable Linear Object Manipulation Including Knotting, *Proc. IEEE Int. Conf. on Industrial Technology*, pp.1321-1326.
- Wakamatsu, H.; Tsumaya, A., Arai, E. & Hirai, S. (2004). Planning of One-handed Knotting/Raveling Manipulation of Linear Objects, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1719-1725.
- Watanabe, T. & Yoshikawa, T. (2007). Grasping Optimization Using a Required External Force Set, *IEEE Trans. on Automation Science and Engineering*, Vol. 4, pp. 52-66.
- Weiss, L.; Sanderson, A. & Neuman, C. (1987). Dynamic Sensor Based Control of Robots with Visual Feedback, *IEEE J. of Robotics and Automation*, Vol. 3, pp. 404-417.
- Wimboeck, T. ; Ott, C. & Hirzinger, G. (2006). Passivity-based Object-level Impedance Control for a Multifingered Hand, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4621-4627.
- Wu, J. ; Luo, Z. ,Yamakita, M. & Ito, K. (1996). Adaptive Hybrid Control of Manipulators on Uncertain Flexible Objects, *J. of Advanced Robotics*, Vol. 10, pp. 469-485.
- Xiao, D.; Ghosh, B. K. , Xi, N. & Tarn, T. J. (2000). Sensor-Based Hybrid Position/Force Control of a Robot Manipulator in an Uncalibrated Environment, *IEEE Trans. on Control Systems Technology*, Vol. 8, pp. 635-645.
- Yao, B. & Tomizuka, M. (1998). Adaptive Robust Motion and Force Tracking Control of Robot Manipulators in Contact with Compliant Surfaces with Unknown Stiffness, *Trans. ASME*, Vol. 120, pp. 232-240.
- Yin, Y. ; Luo, Z. , Svinin, M. & Hosoe, S. (2003). Hybrid Control of Multi-Fingered Robot Hand for Dexterous Manipulation, *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 3639-3644.
- Yoshikawa, T. (1986). Dynamic Hybrid Position/Force Control of Robot Manipulators-Description on Hand Constraint and Calculation of Joint Driving Force, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1393-1398.
- Young, J. & Nourbakhsh, I. (2004). Low Overhead Manipulation of Bound Book Pages, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 4326-4331.
- Yue, S. & Henrich, D. (2001). Manipulating Deformable Linear Objects: Attachable Adjustment-motions for Vibration Reduction, *J. of Robotic Systems*, Vol. 18, 375-389.
- Yue, S. & Henrich, D. (2002). Manipulating Deformable Linear Objects: Sensor-based Fast Manipulation During Vibration, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2467-2472.
- Yuen, C. W. M. ; Fung, E. H. K. , Wong, W. K. , Hau, L. C. & Chan, L. K. (2008). Application of Smart System to Textile Industry: Preliminary Design of a Smart Hanger for Garment Inspection, *Journal of the Textile Institute*, Vol. 99, pp. 569-580.
- Yukawa, T. ; Uchiyama, M. & Inooka, H. (1996). Stability of Control System in Handling of a Flexible Object by Rigid Arm Robots, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2332-2339.

- Zhang, H.; Chen, H., Nenchev, D. N. & Xi, N. (2006). Automated Robot Programming Based on Sensor Fusion, *Int. J. of Industrial Robot*, Vol. 33, pp. 451-459.
- Zheng, Y. F. ; Pei, R. & Chen, C. (1991) Strategies for Automatic Assembly of Deformable Objects, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2598-2603.
- Zheng, Y. F. & Chen, M. Z. (1993). Trajectory Planning of Two Manipulators to Deform Flexible Beams, *Proc. Int. Conf. on Robotics and Automation*, pp. 1019-1024.

Task analysis and kinematic design of a novel robotic chair for the management of top-shelf vertigo

Giovanni Berselli, Gianluca Palli, Riccardo Falconi,
Gabriele Vassura and Claudio Melchiorri
*University of Bologna
Italy*

1. Introduction

The present work addresses the task analysis and the kinematic design of a novel robotic chair to be used for diagnosing and treating Benign Paroxysmal Positional Vertigo (BPPV).

The task analysis process is based on: 1) Direct specifications given by well-trained doctors; 2) Theoretical considerations built upon a mathematical model describing BPPV dynamics.

BPPV is the most frequent cause of vertigo and it is characterized by intense symptoms of dizziness and nausea of short duration which are induced by a postural variation of the patient. A common name for BPPV is "top-shelf vertigo", since dizziness often occurs when tilting back the head, such as when looking at objects on the top shelf. In the elder patients, it represents a frequent cause of equilibrium loss and consequent possible traumatic injuries. Sometimes BPPV symptoms disappear spontaneously but more often they tend to last for months or even years if not adequately treated. It was estimated that 20% of the patients that complain about dizziness are affected by BPPV (approximately 64 in 100,000 people per year in Northern America (Baloh et al., 1989; Froehling et al., 1991; Honrubia et al., 1996)).

BPPV is a malfunction of the vestibular system which is the sensory structure that mainly contributes to the equilibrium. The vestibular system is situated inside the cranium and comprises three fluid-filled semicircular canals (SCC) namely Horizontal Canal (HC), Posterior Canal (PC) and Anterior Canal (AC) (Figure 1(a)) whose duty is the perception of the body's angular motion.

Even though data relating SCC's mutual orientation are sparse, multiple studies of inter-SCC angles concluded that canals are close to perpendicular. Anyway, the relative orientation of the SCC is characterized by a certain degree of disparity within the population of individuals and it can be described by a mean value and its standard deviation (Della Santina et al., 2005; Shinichiro et al., 2005). In media, AC lies on a plane inclined approximately 45° with respect to the body's sagittal plane, PC lies on a plane inclined 45° but in the opposite direction, HC is perpendicular to the other two canals and it is inclined approximately $15\text{-}20^\circ$ with respect to the body's horizontal plane (Figure 1(b)).

The body rotation is sensed through a sensory membrane (*cupola*) which is deflected by the action of the fluid contained in the SCC. Two identical sensory structures are symmetrically disposed with respect to the body's sagittal plane (six SCC in total).

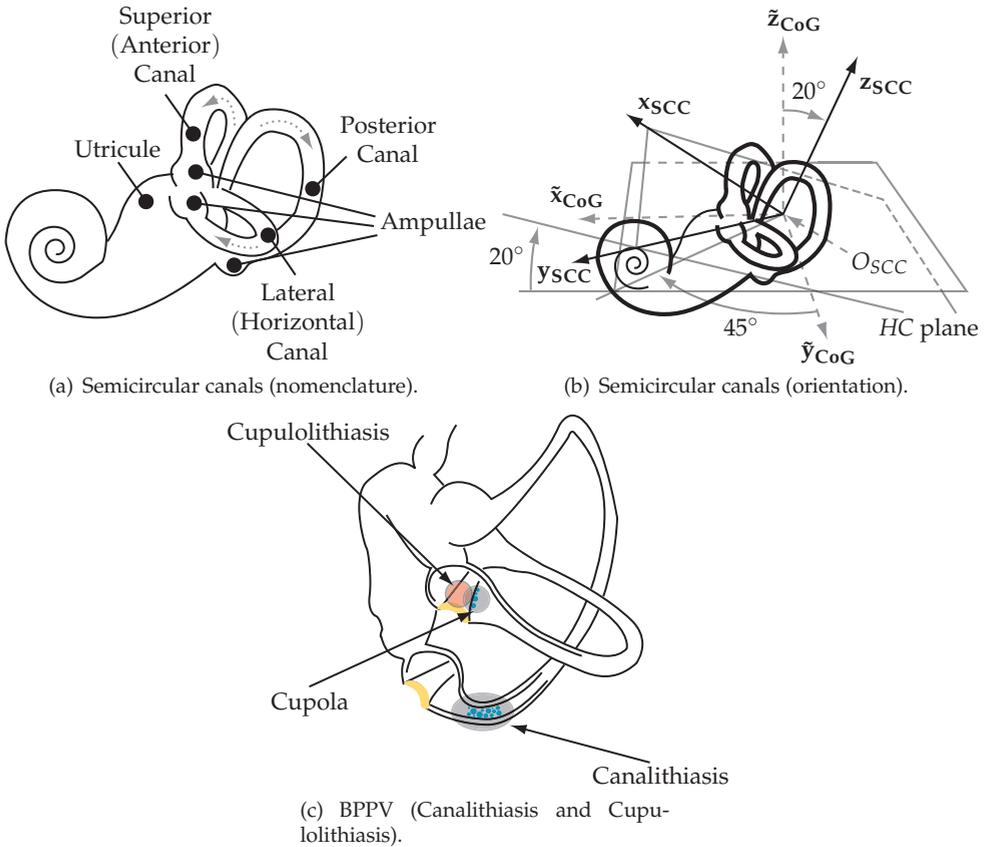


Fig. 1. Human vestibular system and BPPV.

The physiological cause of BPPV is the abnormal presence of calcite particles, named *otoconia* or *otoliths*, within the SCC. It is believed that the presence of otoconia within the SCC could interfere with their normal functioning and could induce a false sensation of motion when in reality the head is kept still.

Concerning the genesis of BPPV, the otoconia separate from their original location within the inner ear and are brought into the canals by normal head movements in everyday life. The medical community have proposed two mechanism to explain BPPV: *cupulolithiasis*, in which otoconia attach directly to the cupula, and *canalithiasis*, in which otoconia settle through the canals and exert a fluid pressure across the cupula (Figure 1(c)). A third and minor theory states that BPPV is caused by calcite dust which is gradually absorbed by the soft tissues causing BPPV symptoms to disappear spontaneously.

Different techniques can be used for BPPV diagnosis and treatment. Most techniques consist in a series of movements of the cranium of the patient in different spatial positions and orientations (repositioning maneuvers or RM, (Brandt & Daroff, 1980; Epley, 1992; Pagnini et al.,

1989; Semont et al., 1980)). RM are executed manually by the doctor and are designed so as to drive otoconia all the way around and out of the canal, so that they settle in the *vestibule/utricle* (Figure 1(a)). As the otoconia reach this position BPPV symptoms disappear.

The success of the RM in most clinical cases has raised an emerging consensus that canalithiasis is the more likely mechanism for BPPV. On the other hand, recent publications (House & Honrubia, 2003) suggest that both canalithiasis and cupulolithiasis are possible.

Concerning BPPV diagnosis, the side (left or right) and the SCC affected by BPPV can be predicted through the observation of the direction of an anomalous eye movement (nystagmus). The nystagmus is evoked by the vestibular-ocular reflex (Raphan et al., 1979; Robinson, 1977) as a result of abnormal stimulation caused by the presence of otoconia within the interested canal. It is believed (Squires et al., 2004) that the size (and mass) of the otoconia dispersed within the SCC is proportional to the intensity of the nystagmus.

Alike the treatment methodology, the diagnosis is carried out through a series of maneuvers called diagnostic maneuvers (DM) during which the doctor recognizes the existence and typology of nystagmus and decides which RM has to be used as treatment. DM are carried out a second time after the treatment: the absence of nystagmus during a subsequent DM represents the marker of the healing.

It has been shown (Epley, 1992) that DM/RM are efficient in 80% of the cases when executed by a well trained doctor. Possible failures can be caused by: (1) a not correct diagnosis; (2) the presence of otoconia in more than one SCC at the same time; (3) a sensible variation of the SCC orientation with respect to the theoretic orientation upon which RM are designed; (4) otoconia which stick to the SCC walls during RM and therefore do not reach the desired final position. If RM are not successful, surgery becomes necessary.

In any case, manual maneuvers present a series of limits such as: (1) very poor repeatability of the trajectories and subsequent impossibility of standardization, (2) limited capability of moving and orienting the patient, especially in case of obese people or elderly with mobility hindrances (where abrupt body/neck's movements should be avoided).

In practice, the efficiency of the present therapeutical methods could be highly improved if:

- A sensorial system capable of recognizing/measuring the existence and typology of nystagmus is employed. Such sensors are commercially available (Figure 2(a)) but it is impossible to correlate their data with the instantaneous pose of the patient during manual RM. Therefore, those devices should be either integrated with a system capable of precisely manipulating the human body or provided with a position/orientation sensor.
- RM and DM are adapted to the need and specific SCC orientation of the single patient. A 3D model of the vestibular system (including SCC morphology, position and orientation within the head) can be made available via TAC or MRN (Figure 2(b)) and specific software (Arnold et al. (1997), Figure 2(c)).
- A purposely designed robotic system is available with the capability of generating fully controllable trajectories with adequate kinematic and dynamic characteristics and adequate repeatability (Nakayama & Epley, 2005).

Besides the improved efficiency of the maneuver, a manipulator purposely designed for the study of BPPV could give a better insight of the physiological mechanisms that cause this disease. Past research has been focused on quantitative modeling of healthy vestibular activity only whereas the discussions around BPPV are nowadays still qualitative. A quantitative analysis is therefore necessary to fully understand the dynamics of top-shelf vertigo and to

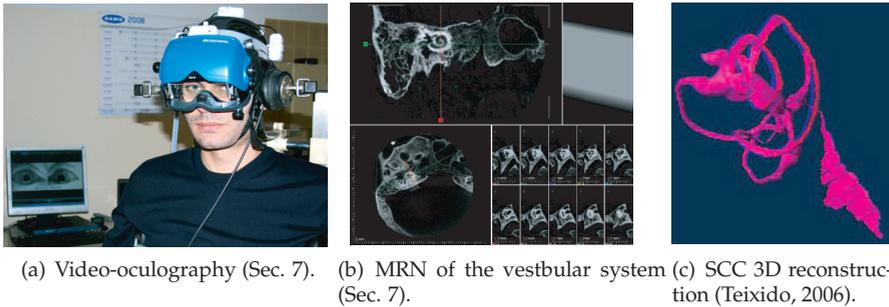


Fig. 2. Different tools used in the study of the vestibular activity.

find unambiguous treatment methodology.

On the basis of the aforementioned considerations, this chapter focuses on: 1) The development of a mathematical model which describes the dynamics of BPPV to be used for a reasonable planning of efficient RM; 2) The topological and dimensional synthesis of a serial robot for manipulating the human body and capable of overcoming those limitations which arise during manual RM. The outcome of this work will be useful for selecting the manipulator's actuation system and for optimizing its mechanical structure. On the other hand, some considerations outlined in the following sections are generally interesting when existing RM are critically compared in terms of efficacy.

Note that, if not explicitly reported, all the angles are indicated in radians, the lengths are in meters and the functions' time dependence is omitted for brevity.

2. General aspect of BPPV treatment methodologies

Figures 3(a), 3(b), 3(c) report, as instances, the schematics of a well known DM, namely the Dix-Hallpike DM (Dix & Hallpike, 1952), and of two RM, namely the Epley and the Semont maneuver (Epley, 1992; Semont et al., 1980). A survey of known DM and RM can be found in (Parnes et al., 2003).

In the Dix-Hallpike DM, the patient is brought from supine to seated position keeping the head turned 45° on one side and extended about 20° backward. If the test is positive the maneuver will cause the eyes to jump upward and twist.

The Epley RM, for treating BPPV of PC, involves sequential movements of the head into four different positions, keeping each position for roughly 30 seconds.

The Semont RM, for treating BPPV of PC, is a procedure where the patient is quickly moved from lying on one side to lying on the other. Despite being 90% effective after 4 treatment sessions, it is a vigorous maneuver that can be dangerous when treating elderly people and it is not currently favoured in the US. Moreover, the same inertial forces applied during this RM cannot be easily applied when treating BPPV of HC (where another RM is used i.e., the Pagnini RM (Pagnini et al., 1989)).

From a kinematic point of view, the existing maneuvers are classifiable in:

- RM that rely on the interaction between otoliths and gravity and therefore are based upon otoconia's sedimentation within the SCC. A typical example is the Epley RM. From a kinematic point of view, these maneuvers are based on the orientation of the

human body with respect to gravity and can therefore be realized with an RR structure (neglecting possible joints' limits).

- RM that rely both on otoconia sedimentation and on inertial forces due to rapid postural variation of the patient. It is believed that the inertial forces are capable of breaking the particle-walls interactions that could prevent the otoconia from moving under the effect of gravity. Typical examples are the Semount and the Pagnini RM. From a kinematic point of view, these maneuvers are based on both orienting and positioning the human body with specified trajectories.

As it has been underlined in the previous section, it is possible to enhance RM' efficacy by adding a certain degree of flexibility in the execution of different poses which would allow for the investigation of manually unfeasible maneuvers.

Several cases of robots employed for human handling can be found in the fields of virtual simulation or entertainment; for instance, the RoboCoaster from Kuka Roboticker GmbH (Kuka, 2004), is a six axis serial robot capable of carrying two passengers in programmable ride sequences. A more specific example is the Omniax Positioning System (OPS) designed and manufactured at the Portland Otologic Clinic by Eng. William Scott following Dr. J. M. Epley specifications (Nakayama & Epley, 2005) precisely for BPPV's diagnosis and treatment. The system is basically a moving chair that controls the patient's orientation by providing 360° rotations along two fixed axis plus a third axis for adjusting the pitch plane. By using the OPS, Nakayama and Epley have already demonstrated that a 360° maneuverability and repeatability can improve capabilities for managing BPPV. However, in their work, nothing is said about the application of controlled major inertial forces neither about the possibility of specifying the otoconia trajectory with respect to the SCC position. In practice, being an orienting device, the OPS cannot overcome the limitations of every RM relying on otoconia sedimentation.

In particular, the otoconia is an irregularly shaped rigid particle which might stick to the SCC soft walls. Therefore: 1) Controlled inertial forces could be useful to break the particle-wall interaction (as reported in Squires et al. (2004)); 2) A treatment strategy which prevents the otoconia from touching the SCC as the maneuver has started could increase the rate of success of non-invasive therapies.

The next section further addresses this issue.

3. Dynamic Model for BPPV

This section describes an idealized dynamic model of a single otoconium (hereafter also referred to as "the particle") moving inside a SCC. The aim of the model is to numerically simulate the particle's motion in the case of canalithiasis (Figure 1(c)). An analytical description of the mechanics of cupulolithiasis can be found in (Rajguru et al., 2004).

Similarly to the model proposed by Obrist & Hegemann (2008), a series of simplifying assumptions are made:

- The analysis of the otoconium motion is limited to regions where the SCC can be modeled as a circular toroid. This is in contrast with (Squires et al., 2004) where a SCC with varying cross sectional area is considered;
- The otoconium is considered as a sphere whose radius is negligible with respect to the SCC cross section;
- The radius of the SCC cross section is negligible with respect to the curvature of the circular toroid (Squires et al., 2004). This assumption allows to consider the SCC as a straight circular cylinder when calculating the Stokes drag force acting on the particle.

It is worth saying that an accurate prediction of the otoconium motion is not achievable for the many simplifications that have been made and for the uncertainties of the parameters describing the SCC and otoconium dynamics. However, a rough estimation of the particle trajectory is useful for verifying the effects of the maneuvers performed during BPPV therapy. A suitable compendium of the effects acting on the otoconium inside the SCC can be found in Obrist & Hegemann (2008) and Squires et al. (2004).



(a) Dix-Hallpike diagnostic maneuver.



(b) Epley repositioning maneuver.



(c) Semont repositioning maneuver.

Fig. 3. BPPV diagnostic and treatment maneuvers. Courtesy of St. Orsola Hospital, Bologna (Sec. 7).

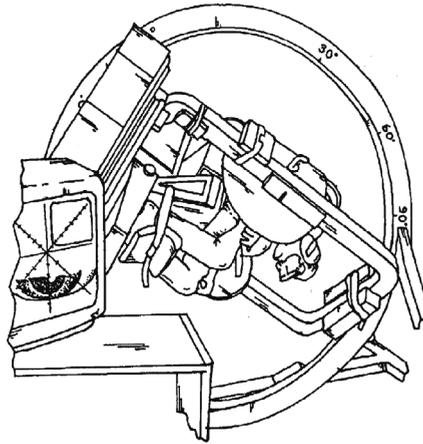


Fig. 4. Omniax positioning system (Nakayama & Epley, 2005).

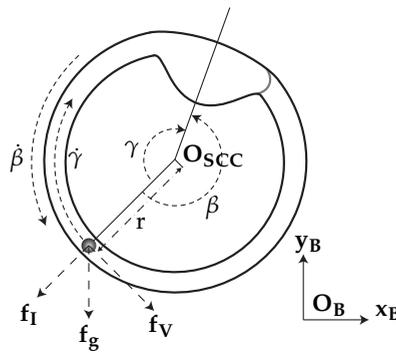


Fig. 5. Forces acting on the otoconium during an SCC planar motion.

Referring to Figure 5, the otoconium is subjected to the gravity field (the force f_g), to the inertial forces arising during its motion (the force f_I), and to the hydrodynamic effects due the presence of the SCC fluid that generates a Stokes drag (the force f_V caused by the relative motion of otoconium within the SCC).

In Squires et al. (2004): 1) Data that correlate the otoconium mass with the intensity of the nystagmus are provided; 2) An expression for the SCC fluid viscosity coefficient with respect to the otoconium's distance from the SCC walls is theorized. A cutoff particle-wall lubrication gap is also considered resembling the effects of the SCC fluid viscosity during the movement of the otoconium along the SCC walls. Similarly, in this chapter, such distance-dependent behavior has been included leading to an over-damped response of the otoconium during a possible impact with the SCC walls.

The model considered hereafter, differently from (Obrist & Hegemann, 2008), takes into account the motion of SCC and otoconium on a plane perpendicular to the SCC axis (conceived as a circular toroid). Thus, the center of the SCC, (point O_{SCC} in Figure 5) can move on the

Description	Symbol	Value	Unit
SCC mass (due to control)	m_c	1	kg
SCC inertia moment (//)	I_{c_z}	1	$\text{kg} \cdot \text{m}^2$
SCC mean radius	r_{mean}	$3.2 \cdot 10^{-3}$	m
SCC inner wall radius	r_{min}	$3.12 \cdot 10^{-3}$	m
SCC outer wall radius	r_{max}	$3.28 \cdot 10^{-3}$	m
SCC wall stiffness	k_c	10^{-3}	$\text{N} \cdot \text{m}^{-1}$
SCC wall damping	b_c	10^{-6}	$\text{N} \cdot \text{s} \cdot \text{m}^{-1}$
Particle mass	m_o	$3.88 \cdot 10^{-12}$	kg
Particle density	ρ_o	$2.7 \cdot 10^3$	$\text{kg} \cdot \text{m}^{-3}$
Particle radius	r_o	$7 \cdot 10^{-6}$	m
Fluid (endolymph) viscosity	η	10^{-6}	$\text{m}^2 \cdot \text{s}^{-1}$
Fluid (endolymph) density	ρ	10^3	$\text{kg} \cdot \text{m}^{-3}$
SCC-particle gap	r_{gap}	$1 \cdot 10^{-6}$	m
Gravity acceleration	g	9.81	$\text{m} \cdot \text{s}^{-2}$

Table 1. Parameters of the SCC-otoconium dynamic model.

Cartesian plane along the \mathbf{x}_B and \mathbf{y}_B directions (Figure 5, where $(\mathbf{xyz})_B$ is an absolute frame attached to the ground), and the SCC is free to rotate with respect to an axis that passes through its center and is perpendicular to the motion plane.

Referring to Figure 5, the angular position and angular velocity of the SCC are indicated by γ and $\dot{\gamma}$ respectively. The otoconium is considered as a sphere of constant radius with concentrated mass at its center and free to move inside the SCC.

It is convenient, from the point of view of modeling the system motion, to express the otoconium position with respect to the SCC center in polar coordinates. The position of the otoconium with respect to the SCC center is then indicated by the angular position β (measured with respect to a frame attached to \mathbf{O}_{SCC} and axis parallel to \mathbf{x}_B and \mathbf{y}_B) and the distance r form the center ($\dot{\beta}$ and \dot{r} indicating the time derivatives of such variables). The physical parameters considered in the dynamic model and the nomenclature used in the following are reported in Tab. 1.

Let us define:

- $\mathbf{q}_c = [x \ y \ \gamma]^T$ as the vector of coordinates of the SCC center \mathbf{O}_{SCC} with respect to $(\mathbf{xyz})_B$ ($\dot{\mathbf{q}}_c, \ddot{\mathbf{q}}_c$ being first and second time derivative of \mathbf{q}_c);
- $\mathbf{q}_o = [\beta \ r]^T$ as the vector of the otoconium coordinates ($\dot{\mathbf{q}}_o, \ddot{\mathbf{q}}_o$ being first and second time derivative of \mathbf{q}_o);
- $\boldsymbol{\tau}_c$ as the vector of the forces applied to the SCC (or, more generally, to the patient body);
- $C_{\gamma\beta} = \cos(\gamma + \beta)$;
- $S_{\gamma\beta} = \sin(\gamma + \beta)$.

By taking into account the forces \mathbf{f}_g , \mathbf{f}_I and \mathbf{f}_V and using the Euler-Lagrange formalism, the dynamics of the system can be described as follows:

$$\mathbf{M}_c \ddot{\mathbf{q}}_c = \boldsymbol{\tau}_c \quad (1)$$

$$\begin{aligned} \mathbf{M}_o(\mathbf{q}_o) \ddot{\mathbf{q}}_o + \mathbf{M}_{co}(\mathbf{q}_c, \mathbf{q}_o) \ddot{\mathbf{q}}_c + \mathbf{C}_o(\mathbf{q}_o, \dot{\mathbf{q}}_o) \dot{\mathbf{q}}_o \\ + \mathbf{C}_{co}(\mathbf{q}_o, \dot{\mathbf{q}}_o, \mathbf{q}_c, \dot{\mathbf{q}}_c) \dot{\mathbf{q}}_c + \mathbf{D}(\mathbf{q}_o) \dot{\mathbf{q}}_o \\ + \mathbf{g}(\mathbf{q}_o, \mathbf{q}_c) + \mathbf{b}(\mathbf{q}_o, \dot{\mathbf{q}}_o) = \mathbf{0} \end{aligned} \quad (2)$$

where

$$\begin{aligned}
 \mathbf{M}_c &= \begin{bmatrix} m_c & 0 & 0 \\ 0 & m_c & 0 \\ 0 & 0 & I_{c_z} \end{bmatrix} \\
 \mathbf{M}_o(\mathbf{q}_o) &= \begin{bmatrix} m_o r^2 & 0 \\ 0 & m_o \end{bmatrix} \\
 \mathbf{M}_{co}(\mathbf{q}_c, \mathbf{q}_o) &= \begin{bmatrix} -m_o r C_{\gamma\beta} & -m_o r S_{\gamma\beta} & m_o r^2 \\ -m_o S_{\gamma\beta} & m_o C_{\gamma\beta} & 0 \end{bmatrix} \\
 \mathbf{C}_o(\mathbf{q}_o, \dot{\mathbf{q}}_o) &= \begin{bmatrix} m_o r \dot{r} & m_o r (\dot{\gamma} + \dot{\beta}) \\ -m_o r (\dot{\gamma} + \dot{\beta}) & 0 \end{bmatrix} \\
 \mathbf{C}_{co}(\mathbf{q}_o, \dot{\mathbf{q}}_o, \mathbf{q}_c, \dot{\mathbf{q}}_c) &= \begin{bmatrix} 0 & 0 & m_o r \dot{r} \\ 0 & 0 & m_o r (\dot{\gamma} + \dot{\beta}) \end{bmatrix} \\
 \mathbf{D}(\mathbf{q}_o) &= \begin{bmatrix} d_\beta(r) r^2 & 0 \\ 0 & d_r(r) \end{bmatrix} \\
 \mathbf{g}(\mathbf{q}_o, \mathbf{q}_c) &= \begin{bmatrix} g m_o r S_{\gamma\beta} \\ -g m_o C_{\gamma\beta} \end{bmatrix} \\
 \mathbf{b}(\mathbf{q}_o, \dot{\mathbf{q}}_o) &= \begin{cases} \begin{bmatrix} 0 \\ -k_c(r - r_{\max}) - b_c \dot{r} \end{bmatrix} & \text{if } r \geq r_{\max} \\ \begin{bmatrix} 0 \\ k_c(r - r_{\min}) + b_c \dot{r} \end{bmatrix} & \text{if } r \leq r_{\min} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise} \end{cases}
 \end{aligned}$$

- \mathbf{M}_c represents the inertia matrix of the SCC. Note that it is considered diagonal and constant. Further details about this assumption can be found in Section 5;
- $\mathbf{M}_o(\mathbf{q}_o)$ represents the otoconium inertia matrix and it is diagonal since the otoconium is assumed to be a sphere whose mass is concentrated at its center (as stated above);
- $\mathbf{M}_{co}(\mathbf{q}_c, \mathbf{q}_o)$ represents the matrix that describes the coupling between the SCC and the otoconium accelerations meaning that the reference system of the SCC is not an inertial frame;
- $\mathbf{C}_o(\mathbf{q}_o, \dot{\mathbf{q}}_o)\dot{\mathbf{q}}_o$ and $\mathbf{C}_{co}(\mathbf{q}_o, \dot{\mathbf{q}}_o, \mathbf{q}_c, \dot{\mathbf{q}}_c)\dot{\mathbf{q}}_c$ describe the Coriolis and the centrifugal effects respectively which act on the otoconium and which are due to SCC motion and otoconium motion itself;
- $\mathbf{g}(\mathbf{q}_o, \mathbf{q}_c)$ is a generalized gravity force vector, i.e. the force vector \mathbf{f}_g as expressed in polar coordinate.
- $\mathbf{b}(\mathbf{q}_o, \dot{\mathbf{q}}_o)$ describes the repulsive effects of the SCC walls by using a simple model that includes the wall's stiffness, k_c , and damping, b_c . The parameters k_c and b_c have been chosen by trial and error so as to obtain, in simulation, a reasonable system behavior when an impact occurs between the otoconium and the SCC wall;
- $\mathbf{D}(\mathbf{q}_o)$ describes the Stokes drag coefficients.

Concerning the $\mathbf{D}(\mathbf{q}_o)$ matrix, its elements can be defined as (Squires et al., 2004):

$$\begin{aligned} d_s &= 6\pi\eta\rho r_o \\ r_{mg} &= r_{min} + r_{gap} \\ r_{Mg} &= r_{max} - r_{gap} \end{aligned}$$

$$d_\beta(r) = d_s \quad (3)$$

$$d_r(r) = \begin{cases} d_s & \text{if } r_{mg} < r < r_{Mg} \\ d_s \frac{r_{gap}}{r - r_{min}} & \text{if } r_{min} \leq r \leq r_{mg} \\ d_s \frac{r_{gap}}{r_{max} - r} & \text{if } r_{Mg} \leq r \leq r_{max} \end{cases} \quad (4)$$

where $r_{gap} = 10^{-6}$ [m] (note that $r_{gap} < r_o$). In this way, the particle is free to slide along the SCC wall, as required by the therapeutic maneuvers which relies on otoconia sedimentation (Squires et al., 2004). In addition, the increasing value of the radial damping coefficient in the wall proximity is used to emulate adherence effects between the particle and the SCC wall.

As previously stated, the value of the coefficient d_s is determined assuming that, from the point of view the hydrodynamic effects, the otoconium is a sphere whose radius is negligible with respect to the SCC cross section and that the radius of the SCC cross section is negligible with respect to the SCC curvature, $1/r_{mean}$.

Note that, during normal SCC functioning, an head movement causes the motion of the SCC fluid and a consequent deflection of the cupola (see Figure 1(a)). However, in canalithiasis, the very small time scale of the fluid inertial effects with respect to the otoconium's motion make such effects negligible in all those cases where just the otoconium's trajectory is of interest (Squires et al., 2004).

In addition, during manual RM, the otoconium's inertial forces could be considered negligible with respect to the gravitational force and to the Stokes drag. On the other hand, such effects are hereafter considered in order to allow the study of the SCC-particle relative motion along all the directions and to study possible particle-wall detaching maneuvers.

An important point to note is the particular structure of the dynamic eqs. (1) and (2). While the motion of the otoconium is influenced by the motion of the SCC, the vice-versa is not true thanks to the negligible mass of the otoconium with respect to the patient body. It follows that eqs. (1) and (2) are not cross-coupled, and that the motion of the SCC is independent. This means that only eq. (2) is necessary in order to compute the otoconium trajectory whenever the trajectory of the SCC is given (in terms of position, velocity and acceleration of all its three coordinates x , y and γ).

3.1 Manual unfeasible maneuvers: a case study

To show the effectiveness of maneuvers that exploit the rotation of the SCC along an axis passing through \mathbf{O}_{SCC} and perpendicular to the $\mathbf{x}_B\mathbf{y}_B$ plane (Figure 5), a significant case study is considered.

The maneuver starts with the particle in the lower equilibrium point (at the bottom of the SCC, $\beta = -\pi/2$) and in contact with the wall, see Figure 6(b), and it can be conceptually divided in three phases:

1. A translational movement in the downward y direction, having high acceleration for a very limited amount of time (Figures 6(b) and 6(c)). It is assumed that an high inertial force can break the particle-wall interaction in case the otoconium is stuck to the SCC

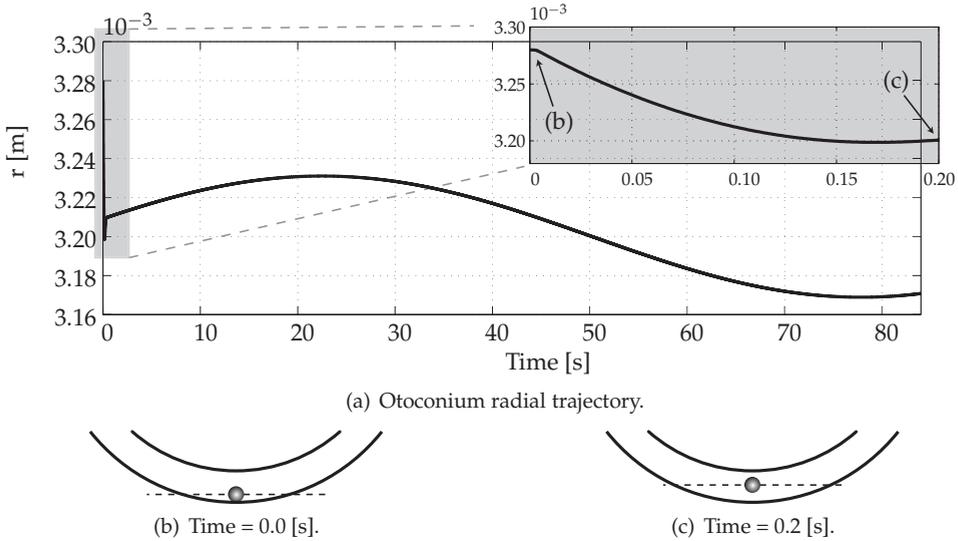


Fig. 6. Otoconium radial trajectory during the rotational maneuver and details of the detaching phase.

soft tissue. At the time the otoconium is free to move within the SCC, it will tend to maintain its absolute position thanks to the particle own inertia and to a decreasing damping force (see eq. (4) as the particle-wall distance increases.

2. Simultaneously, the patient body is rapidly rotated backward of an angle of $\pi/2$ [rad] (Figure 7(c)). This rapid rotation causes the particle to remain almost in the same position with respect to the SCC, thanks to the Stokes drag force.
3. A rotation of the patient body at a particular speed $\dot{\beta}_{eq}$, that allows to maintain the particle in a floating condition by establishing an equilibrium between the gravity and the Stokes drag (Figures 7(d)-7(f)). This rotation is maintained until the otoconium has reached the ampulla. The motion stops with the patient in the normal sitting position ($\gamma = 2\pi$, see Figure 7(f) and Figure 7(b)).

In the example reported in Figure 6, the SCC is subjected to an acceleration of $3g$ in the \mathbf{y}_B direction linearly decreasing to zero in 0.2 [s]. The final velocity along the \mathbf{y}_B axis is slowly recovered to zero during the remaining part of the maneuver (steps 2 and 3).

Note that Hain et al. (2005) report a peak tangential acceleration exerted in the \mathbf{x}_B direction (Figure 5) over 0.1 s during the Semont RM. The same authors claim that if there is clinical value to inertial forces acting on the otoconium, it would be most likely caused by breaking of canal-wall interactions and mobilizing an otoconium from the wall.

Assuming that the particle is located at the $r = r_{mean}$, the speed $\dot{\beta}_{eq}$ can be computed according to (Obrist & Hegemann, 2008; Squires et al., 2004) as:

$$\dot{\beta}_{eq} = \frac{|\mathbf{f}_g|}{d_{\beta}(r)r_{mean}} = \frac{m_o(1 - \rho/\rho_o)g}{6\pi\eta\rho r_o r_{mean}}$$

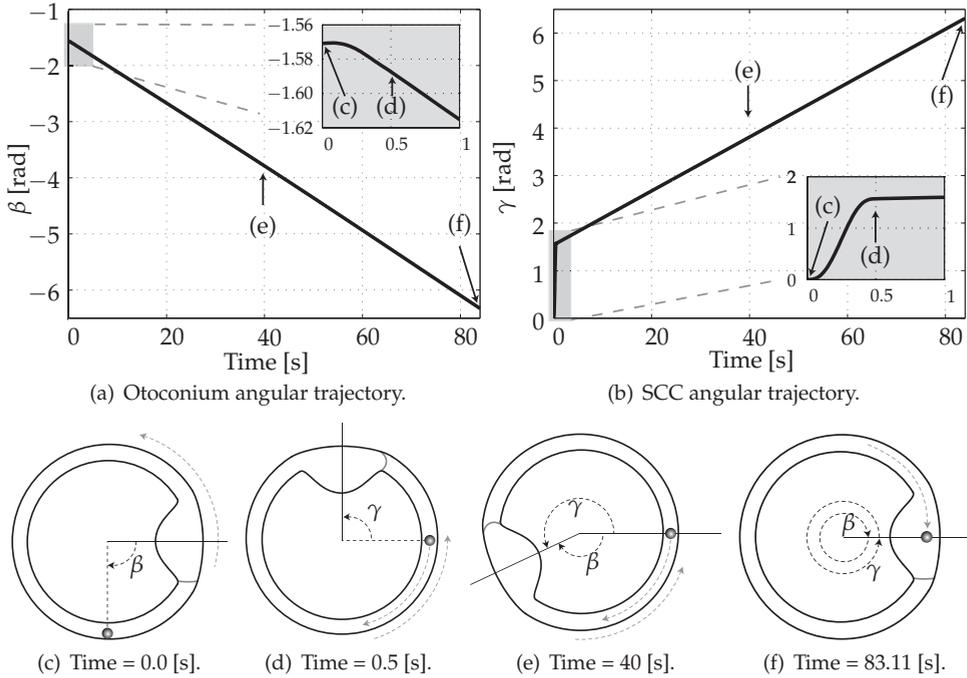


Fig. 7. Otoconium and SCC angular trajectory during the rotational maneuver.

Figure 6(a) shows the radial trajectory of the otoconium during the maneuver whereas Figure 7 reports the angular trajectory of SCC and particle. A detail of the first part of the trajectory is reported to highlight the detachment of the particle from the wall. Note that, during the remaining part of the maneuver, the radial position of the otoconium remains close to the SCC median radius (the oscillations are given by the uncertainties in the particle trajectory, especially during the detaching phase).

Moreover, the dynamic system (1)-(2) can be inverted so as to design the SCC trajectory (and, eventually, the necessary forces by means of (1)) when the desired otoconium trajectory $[\mathbf{q}_{o_d}, \dot{\mathbf{q}}_{o_d}, \ddot{\mathbf{q}}_{o_d}]$ is specified (where the subscript $_d$ stays for the desired trajectory instead of the actual one).

By inversion of eq. (2), the acceleration of the SCC can be then determined as:

$$\ddot{\mathbf{q}}_{c_d} = -\mathbf{M}_{co}^\dagger(\mathbf{q}_{c_d}, \mathbf{q}_{o_d}) [\mathbf{M}_o(\mathbf{q}_{o_d})\ddot{\mathbf{q}}_{o_d} + \mathbf{C}_o(\mathbf{q}_{o_d}, \dot{\mathbf{q}}_{o_d})\dot{\mathbf{q}}_{o_d} + \mathbf{C}_{co}(\mathbf{q}_{o_d}, \dot{\mathbf{q}}_{o_d}, \mathbf{q}_{c_d}, \dot{\mathbf{q}}_{c_d})\dot{\mathbf{q}}_{c_d} + \mathbf{D}(\mathbf{q}_{o_d})\dot{\mathbf{q}}_{o_d} + \mathbf{g}(\mathbf{q}_{o_d}, \mathbf{q}_{c_d}) + \mathbf{b}(\mathbf{q}_{o_d}, \dot{\mathbf{q}}_{o_d})] \quad (5)$$

where $\mathbf{M}_{co}^\dagger(\mathbf{q}_{c_d}, \mathbf{q}_{o_d})$ indicates the pseudo-inverse of the matrix $\mathbf{M}_{co}(\mathbf{q}_{c_d}, \mathbf{q}_{o_d})$. This last equation defines an ODE problem that can be easily solved numerically by double integration of (5) once the initial position/velocity of the SCC and the otoconium trajectory are known.

It is important to note that, since the $\mathbf{M}_{co}(\mathbf{q}_{cd}, \mathbf{q}_{od})$ is a 2×3 matrix, its null space can be defined as:

$${}^N\mathbf{M}_{co}(\mathbf{q}_{cd}, \mathbf{q}_{od}) = \text{Null} \{ \mathbf{M}_{co}(\mathbf{q}_{cd}, \mathbf{q}_{od}) \} = \begin{bmatrix} rC_{\gamma\beta} \\ rS_{\gamma\beta} \\ 1 \end{bmatrix} \quad (6)$$

Hence, the otoconium trajectory $[\mathbf{q}_{od}, \dot{\mathbf{q}}_{od}, \ddot{\mathbf{q}}_{od}]$ can also be achieved with a different choice of the SCC trajectory:

$${}^N\ddot{\mathbf{q}}_{cd} = \ddot{\mathbf{q}}_{cd} + {}^N\mathbf{M}_{co}(\mathbf{q}_{cd}, \mathbf{q}_{od})\lambda \quad (7)$$

where $\lambda \in \mathbb{R}$ is a suitable scalar coefficient used so as to select the desired SCC trajectory in the space of all the possible ones. The definition of the value of λ can be made on the base of different criteria. For instance the motion along a desired (penalized) direction can be minimized or the SCC center can be kept inside a desired region. It is worth reminding that (5) gives the minimum SCC acceleration that produces the desired otoconium trajectory.

4. Robotic chair kinematic design

Section 3 theoretically proves that particular RM could be used in order to firstly detach otoconia possibly stuck to the SCC's walls and then to drive them out of the canals while preventing further interactions with the canals' soft tissues. These particular RM require a rotation of an SCC along an axis passing through the point \mathbf{O}_{SCC} and perpendicular to the \mathbf{x}_{BYB} plane (Figure 5).

Therefore, starting from the experience of Nakayama & Epley (2005) and on the basis of the aforementioned considerations, the aim of this section is to define the kinematic structure of a serial robot which could add more flexibility in the execution of manually unfeasible RM when compared to existing solutions (e.g. the OPS or human-carrying industrial robots). In particular, the novel kinematic structure should be capable of practically applying the RM proposed in Section 3 to each one of the six SCC.

In summary, the considered serial linkage complies with the following general specifications:

- Capability to perform all existing RM based on otoconia sedimentation;
- Capability to perform unlimited rotations along the revolution axis of every SCC conceived hereafter as a circular toroid;
- Capability to apply controlled inertial forces on every SCC (similarly to the inertial forces applied during the Semont RM for treating the PC);
- Capability to reach a position where the moving chair would be easily accessible.

Other important issues are the safety and the ergonomics requirements as well as the overall dimensions that must be acceptable for usage in a hospital environment. Moreover, the psychological impact of these kind of machines on the elderly does not have to be undervalued. To this respect, a closed structure (like in Figure 4) has been discarded preferring the use of a serial manipulator. Despite the fact that serial structures are less rigid (and therefore less accurate), they dispose of a better workspace, better accessibility and are more "acceptable" by the patient in terms of human-robot interaction and user friendliness.

On the other hand, the adoption of commercial manipulators, alike an anthropomorphic robotic arm (Kuka, 2004), has been excluded as long as those structures do not guarantee the desired degree of flexibility. In fact, the existence of joints' limits and possible self collision

highly restricts the feasible rotations along the SCC revolution axis (as it can be proven by solving the inverse kinematic problems for such kind of manipulators). Therefore, differently from the conceptual design of a "general purpose" manipulator, it is necessary to kinematically design an "on-purpose" machine capable of complying with the aforementioned specifications.

Precisely, the topological and dimensional synthesis of the serial linkage has been achieved by means of a simplified Task Based Design (TBD) technique (Kim, 1992).

As previously proposed in the literature (Chedmail & Ramstein, 1996; Chen & Burdick, 1995; Kim & Khosla, 1993a; Yang & Chen, 2000), the TBD technique makes use of Genetic Algorithms (GA) (Goldberg, 1989) in order to determine a robot's kinematic structure which is capable of performing a given set of tasks. The robot's features to be determined include minimum number of degrees of freedom (MDOF), topology, and Denavit-Hartenberg parameters. For instance, TBD has been proven effective in determining assemblies of modular robots optimally suited to perform a specific assignment. In the contest of modular assemblies, both the topology of the serial chain and the links' length must be treated as non continuous variables. Hence, it is necessary to use an optimization method, such as the GA, which is capable of dealing with both highly nonlinear functions and discrete variables. In general, the optimization problem itself can be posed as unconstrained (as in Chedmail & Ramstein (1996)) or constrained (as in Kim & Khosla (1993a)). In the latter case different constraints can be applied e.g. reachability, joint limits, obstacle avoidance, dexterity measures. In the same way, the objective function to be optimized can be chosen in different manners such as workspace maximization, manipulability index maximization, degrees of freedom (DOF) minimization, mechanical constructability minimization. In this respect, global methods, as opposed to progressive ones, try to accomplish an optimum design in one step only by minimizing a weighted sum of the different requirements.

Similarly to the aforementioned example concerning modular robots, the optimization process presented hereafter deals with discrete variables (i.e. manipulator topology and discretized D-H parameters, see Section 4.2). The algorithm makes use of a progressive method which meets consecutive constraints and successive optimized solutions. Note that the robot kinematic design can be further improved by using a continuous optimization method (Avilés et al., 2000) once a possible robot's topology has been finalized.

4.1 Specification of tasks

In order to apply TBD techniques, a series of tasks must be specified analytically. For this purpose, three coordinate systems need to be defined (see Figure 8) as follows:

- $(xyz)_B$, an absolute frame attached to the ground;
- $(xyz)_{CoG}$, attached to the Center of Gravity (CoG) of the patient plus the moving chair, hereafter considered as a rigid body: $+z_{CoG}$ (the *yaw* or *horizontal rotation* axis) is a vertical axis pointing up, $+x_{CoG}$ (the *roll* axis) is perpendicular to $+y_{CoG}$ and $+z_{CoG}$ pointing anteriorly, and $+y_{CoG}$ (the *pitch* axis) points out the left ear;
- $(xyz)_{SCC}$ located on the intersecting point of the three revolution axes of each toroid, here considering left SCC only. z_{SCC} lies on the axis of the HC, y_{SCC} lies on the axis of the PC, x_{SCC} lies on the axis of the AC. SCC are considered as mutually orthogonal. Another coordinate system attached to right SCC can be defined in the same manner.

As previously stated (Figure 1(b)), the AC lies on a plane inclined approximately 45° with respect to sagittal plane $(xz)_{CoG}$, the PC lies on a plane inclined 45° but in opposite direction and the HC is perpendicular to the other two canals and inclined approximately $15-20^\circ$ with

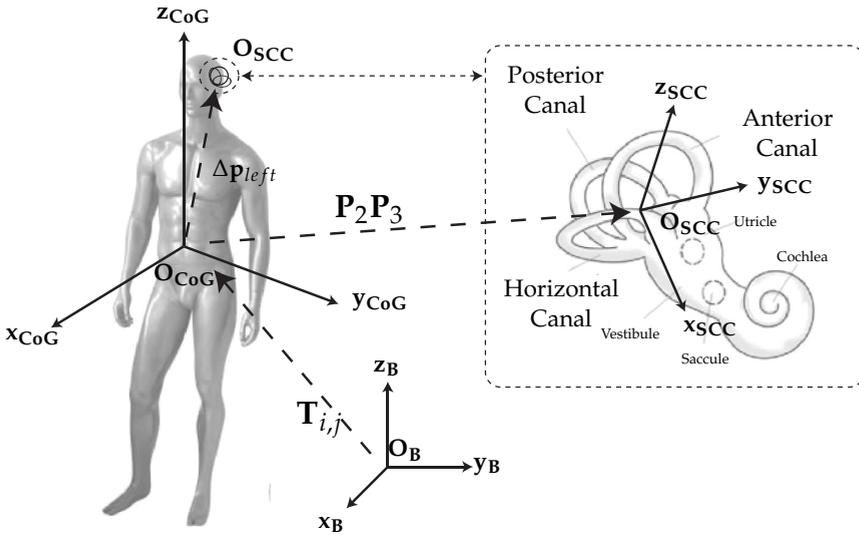


Fig. 8. Reference frames (solid lines) and relative homogeneous transformations (dotted lines).

respect to the horizontal plane $(xy)_{CoG}$. The desired trajectories that fulfill the requirements of the kinematic specifications are described by a finite set of tasks given as homogenous transformation matrices between $(xyz)_B$ and $(xyz)_{CoG}$. In the remaining part of the chapter the notations $R_x(\cdot)$, $R_y(\cdot)$ and $R_z(\cdot)$ will be used to address 3×3 rotational matrices with respect to x -, y - and z -axis respectively whereas $\bar{R}_x(\cdot)$, $\bar{R}_y(\cdot)$, $\bar{R}_z(\cdot)$ address the 4×4 homogeneous matrices associated with those same rotations.

Task set 1 – Rest position. This set contains only one task that depicts the chair in a position which can be easily reached by the patient. The task is described by the following homogeneous matrix:

$$T_{1,1} = \begin{bmatrix} R_z(90^\circ)R_y(45^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where h indicates the desired CoG height from the ground.

Task set 2 – Eccentric rotation. This set contains all the tasks that describe an eccentric rotation with variable radius.

$$T_{2,i} = \begin{bmatrix} R_y(-20^\circ)R_z(-\theta_i) & \rho \sin(\theta_i) \\ 0 & \rho \cos(\theta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where ρ is the radius of the circular trajectories, $T_{2,i}, i = 0 \dots N - 1$ are N tasks describing circular trajectories while maintaining the HC plane parallel to the ground, and $\theta_i \in \{\frac{\pi}{2N}i\}$ for $i = 0, \dots, N - 1$. Eccentric rotations can be used in order to apply controlled inertial forces

on the HC (similarly to the stimuli arising during the Semount RM for PC treatment).

Task set 3 – Existing clinical maneuvers. This task set collects the homogeneous matrices describing existing manual maneuvers. These maneuvers (Boniver, 1990) can be considered as a set of rotations along $(xyz)_{CoG}$ – axis. For instance, the task set that describes the Dix-Hallpike maneuver is given by the union of two concatenated subset with $N/2$ tasks each. The two subsets are described by the following matrices:

$$\mathbf{T}_{3,i}^{(1)} = \begin{bmatrix} & \mathbf{R}_z(\theta_i) & & 0 \\ & & & 0 \\ & & & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$\mathbf{T}_{3,i}^{(2)} = \mathbf{T}_{3,(N-1)/2}^{(1)} \begin{bmatrix} & \mathbf{R}_y(\theta_j) & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where $\theta_i \in \{-\frac{\pi}{2N}i\}$ and $\theta_j \in \{\frac{3\pi}{2N}j\}$ for $i, j = 0, \dots, \frac{N-1}{2}$.

Task set 4 – Rotation along SCC axis. Task set 4 creates a circular path of $(xyz)_{CoG}$ around the revolution axis of each SCC. Consider left SCC first. Starting from $(xyz)_{CoG}$ in rest position, let us define:

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & \Delta p_{left,x} \\ 0 & 1 & 0 & \Delta p_{left,y} \\ 0 & 0 & 1 & \Delta p_{left,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$\mathbf{P}_3 = \begin{bmatrix} \mathbf{R}_x(\Omega_{HC})\mathbf{R}_z(\Omega_{AC}) & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (13)$$

where $\Delta \mathbf{p}_{left} = \mathbf{O}_{SCC_{left}} - \mathbf{O}_{CoG} = [\Delta p_{left,x} \ \Delta p_{left,y} \ \Delta p_{left,z}]^t$ is the vector that identifies the position of $\mathbf{xyz}_{SCC_{left}}$ with respect to the patient's body frame, $\mathbf{O}_{SCC_{left}}$ and \mathbf{O}_{CoG} are the origins of $(xyz)_{SCC}$ and $(xyz)_{CoG}$ respectively, Ω_{HC} and Ω_{AC} are the orientations of HC and AC defined as rotations along $(xyz)_{CoG}\mathbf{P}_2$.

Let us define the following matrix as a design parameter :

$$\mathbf{P}_{BO} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h + \Delta p_{left,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

The rotations along the AC, HC and PC axis are respectively given by:

$$\text{AC : } \mathbf{T}_{4i} = \mathbf{P}_{BO}\overline{\mathbf{R}}_x(\theta_i)(\mathbf{P}_2\mathbf{P}_3)^{-1} \quad (15)$$

$$\text{HC : } \mathbf{T}_{5i} = \mathbf{P}_{BO}\overline{\mathbf{R}}_y(\pi/2)\overline{\mathbf{R}}_z(\theta_i)(\mathbf{P}_2\mathbf{P}_3)^{-1} \quad (16)$$

$$\text{PC : } \mathbf{T}_{6i} = \mathbf{P}_{BO}\overline{\mathbf{R}}_z(\pi/2)\overline{\mathbf{R}}_y(\theta_i)(\mathbf{P}_2\mathbf{P}_3)^{-1} \quad (17)$$

where $\theta_i \in \{2\pi i\}$ for $i = 0, \dots, N$.

Supposing a perfect symmetry of the SCC canals with respect to the body's sagittal plane, the tasks concerning the right SCC can be obtained by setting

$$\Delta \mathbf{p}_{right} = [\Delta p_{left,x} \quad -\Delta p_{left,y} \quad \Delta p_{left,z}]^T \tag{18}$$

At this stage, some design decisions have already been made:

- $(\mathbf{xy})_{\mathbf{B}}$ rest position and patient orientation could be left free under certain limits, whereas \mathbf{T}_{11} specifies a patient positioned over $(\mathbf{xyz})_{\mathbf{B}}$ origin with a given orientation.
- There is no need to request a certain orientation along $\mathbf{z}_{\mathbf{CoG}}$ as the initial pose for the set of existing manual RM whereas $(\mathbf{xy})_{\mathbf{B}}$ and $(\mathbf{xy})_{\mathbf{CoG}}$ are requested to be aligned as Task set 3 starts.
- The maneuvers in Task set 4 are conceived as rotations along a revolute pair with height $h + \Delta p_{left,z}$ from the ground when each set of rotations along each SCC could be exploited with different R-joints and in different spatial positions.

4.2 Problem Statement and Data Structures

The determination of the kinematic parameters for an optimized open-chain manipulator can be regarded as a generic optimization problem: minimize $\mathbf{f}(\mathbf{X})$, $\mathbf{X} \in \mathbf{S}$, where \mathbf{S} is the search space of possible solution points, subjected to a certain number of constraints.

Because of the high-dimensioned parameter space, an heuristic algorithm based on the theory of GA has been implemented in order to find a possible solution. Within GA terminology, the objective function to be optimized, $\mathbf{f}(\mathbf{X})$, is called *fitness function*, whereas an instance of a possible solution $\bar{\mathbf{X}}$ is called *individual*. The set of all the possible solutions at a given iteration of the algorithm is called *population*. Let us define:

$$\mathbf{X} = \begin{bmatrix} \mathbf{T}_0^B & \mathbf{0}_{4 \times 1} \\ \mathbf{DH} & \mathbf{type} \\ \mathbf{T}_{tool}^n & \mathbf{0}_{4 \times 1} \end{bmatrix}, \quad \text{where : [DH type] = } \begin{bmatrix} \alpha_1 & a_1 & \vartheta_1 & d_1 & R \text{ or } P \\ \alpha_2 & a_2 & \vartheta_2 & d_2 & R \text{ or } P \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & F \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_n & a_n & \vartheta_n & d_n & R \text{ or } P \end{bmatrix} \tag{19}$$

where $\mathbf{X}_{(n+8) \times 5}$ is a matrix representation of a serial robot, n is its number of DOF, \mathbf{T}_0^B , \mathbf{T}_{tool}^n are 4x4 homogenous matrices, $[\mathbf{DH type}]_{n \times 5}$ is an augmented matrix of D-H parameters with an appended column indicating joint type. D-H parameters are listed as follows: $[\alpha, a, \vartheta, d]$. Possible joints are 1-DOF joints (revolute (R) or prismatic (P)) and 0-DOF joints used as a slack variable (F) to model a manipulator with less DOF than the maximum allowed.

According to the Denavit-Hartenberg (D-H) convention, \mathbf{T}_0^B and \mathbf{T}_{tool}^n indicate position and orientation of robot base and tool with respect to the coordinate system attached to the first and last movable joints respectively. Considering robot tool position coincident with the origin of $(\mathbf{xyz})_{\mathbf{CoG}}$ means that just the orientation part of \mathbf{T}_{tool}^n needs to be specified. The links' length is described as a discrete variable varying from zero to a predefined maximum value and then divided into a finite number of parts. If the i -th joint is a revolute pair then ϑ_i is a pose variable and therefore not considered as a design parameter, if joint i -th is prismatic then d_i is the pose variable. Finally, if i -th joint is fixed, the corresponding row will be deleted in the evaluation process.

The candidate robots generation is based on a set of heuristic rules similar to those found in (Kim & Khosla, 1993a):

- **Kinematic simplicity:** α and θ (whenever the latest is considered as a design variable) can assume values belonging to the set $[0, \pm\pi/2, \pi]$. Concerning R-joints, at least one of the two variables representing length (a, d) is set to zero.
- **Redundancy avoidance:** R-joints described by D-H parameters of the type $[0, 0, \theta, d]$ cannot be followed by another R-joint, thus avoiding solutions where two revolute joints are mounted on the same axis.

4.3 Evaluation Procedure

The design process of the serial link chain is automated through an optimization procedure which allows a less subjective decision making progression and increases the performance with respect to an objective function defined in order to assess the benefit of a solution (Figure 9).

The type of searching method depends most of all on the type of variables to be dealt with (continuous, discrete or mixed) and on the type of problem (constrained or unconstrained). In this chapter, a GA is used to solve a constrained optimization process on the discrete variable. Exhaustive search techniques, which basically measures the benefit of each possible individual, could be used to find the exact optimal solution; whether this technique would be better suited for a specified problem is just a matter of computational time. Probabilistic search techniques, such as GA or simulated annealing, become a good choice when the search space is extremely large.

At each step a GA creates a new population of individuals using the individual or data structures of the current generation. It basically scores each current individual computing its fitness value; it then selects a set of parents based on their fitness (selection process) and produces a new generation starting from this given set of individuals. Individuals of the new generation are either taken from the selected parents without any change (elitism), randomly changing a single parent (mutation) or combining vector entries of different parents within the same class of substructures (i.e. avoiding as a result an individual with different data structure from the one reported). The algorithm stops when a given stopping criteria is met. In this chapter the only specified criteria is a limit on the number of generations.

Given a set of tasks, it is stated that a reachability constraint (RC) must be satisfied. For a particular task, the RC is said to be satisfied if: 1) There exists a solution of the inverse kinematic problem (IK) which presents a positional error norm and an orientational error norm lower than an appropriate threshold (Kim & Khosla, 1993b) 2) Such solution is found within a certain number of iterations. As long as the structure of the manipulator is not yet defined and the serial chain can assume a very high number of configurations, a numerical method to solve the IK problem is used; reference is made to the singularity robust IK method proposed by L. Kelmar and P. K. Khosla (Kelmar & Khosla, 1990). If the RC is not satisfied the fitness value is set to a very large number.

4.4 Design stages

The GA-based optimization procedure has been splitted in two different stages (Figure 9).

4.4.1 Minimized Degrees-of-Freedom approach (MDOF)

In the first design stage the fitness value is simply the number of DOF of the manipulator regardless of joints being revolute or prismatic. IK is computed for every task in series on

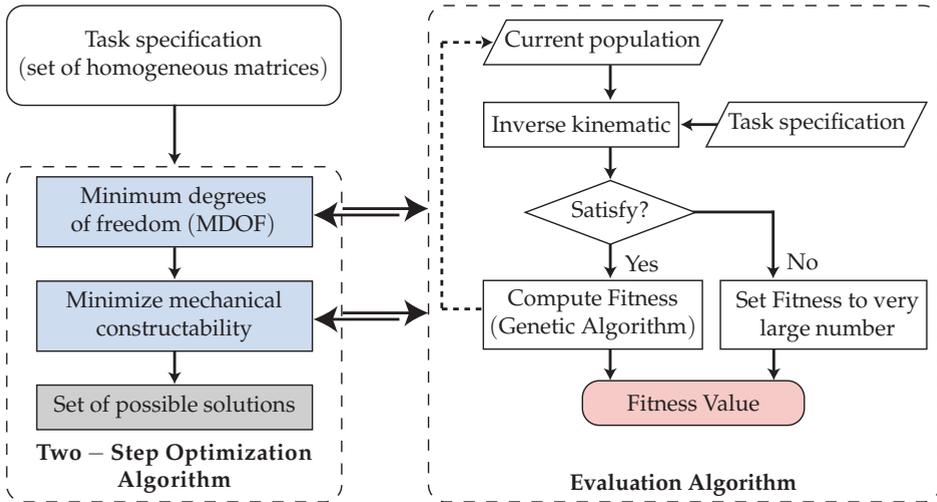


Fig. 9. Design steps and evaluation procedure.

structures where rows concerning fixed joints are previously deleted. If the RC is not satisfied the fitness value is set to a very large number and successive tasks are not evaluated. Prior to IK calculation, which is the most time consuming procedure, it is verified that:

$$r_i - \sum_{j=0}^n l_j < 0 \quad i = 1, \dots, N_{TOT} \quad \text{where} \quad \begin{cases} l_j \leq L_{max} & \text{if } j\text{-th joint is revolute} \\ l_j = 2L_{max} & \text{if } j\text{-th joint is prismatic} \end{cases} \quad (20)$$

where N_{TOT} is the total number of tasks, r_i is the Euclidian distance of i -th task from the robot base and l_j is j -th link length. $l_0 = \|\mathbf{O}_0 - \mathbf{O}_B\|$ (\mathbf{O}_0 and \mathbf{O}_k are supposed coincident, where \mathbf{O}_k is the origin of $(xyz)_k$ according to Chocron & Bidaud (1997)). After IK, i -th RC is considered not satisfied if:

$$\sqrt{a_j^2 + (\Delta d_{j,max}^2)} > 2L_{max} \quad (21)$$

where $\Delta d_{j,max} = |d_{j,max} - d_{j,min}|$ (i.e. during motion the prismatic joint has traveled a distance greater than $2L_{max}$, its initial length being set to a_j).

The result of this design step is the minimum number of DOF necessary to perform task specifications meaning that the algorithm has found an individual \bar{X} that represents a n -DOF kinematic structure able to perform every pose.

4.4.2 Mechanical Constructability Minimization.

The aim of this second design stage is the minimization of total link length and therefore of total robot’s mass. The fitness function is set to:

$$F(X) = \sum_{j=0}^n l_j, \quad \text{where} \quad \begin{cases} l_j = \sqrt{a_j^2 + d_j^2} & \text{if } j\text{-th joint is revolute} \\ l_j = \sqrt{a_j^2 + (\Delta d_{j,max}^2)} & \text{if } j\text{-th joint is prismatic} \end{cases} \quad (22)$$

	α [rad]	a [m]	θ [rad]	d [m]	R/P
Joint 1	$\pi/2$	0.00	θ_1	1.20	R
Joint 2	$3\pi/2$	0.10	θ_2	0	R
Joint 3	$3\pi/2$	0.00	π	d_3	P
Joint 4	$3\pi/2$	0.00	θ_4	0	R
Joint 5	$3\pi/2$	0.00	θ_5	0.20	R
Joint 6	π	0.00	θ_6	0	R

Table 3. Best solution’s Denavith-Hartenberg parameters and joint types.

The best found solution is reported in Table 3, the matrices \mathbf{T}_0^B and \mathbf{T}_{tool}^n being two identity matrices i.e. the absolute frame and the robot’s base frame are coincident and the frame attached to the last link is coincident with the robot’s tool frame. The solution’s representation is depicted in Figure 10. The proposed serial linkage is topologically similar to a Stanford manipulator presenting a particular wrist (not spherical). As long as no obstacle avoidance has been considered, the only useful information that can be found by the MDof approach is that the specified set of tasks could not be accomplished by less than 6-DOF robots or that the GA couldn’t find such solution applying the given set of heuristic rules (*mechanical simplicity and redundancy avoidance*).

A conceptual design of the six-DOF robot (patented by Berselli et al. (2007)) is depicted in Figure 11(a) whereas Figure 11(b) shows the manipulator performing a rotation along the revolution axis of the right AC.

In particular, the last two joints are used to replicate every existing manual maneuver and the first joint is used to apply controlled inertial forces (similar to the stimuli arising during the Semount RM) on the HC via eccentric rotation of the patient. The other DOF are used to control the trajectory of the otoconia within an SCC performing full rotation along the revolution axis of every SCC (as shown in Section 3). At this stage no care was taken concerning dynamic and structural analysis and optimization. Further steps for the development of a working prototype include decision making of possible motors, gears, bearings and couplings as well as cabling and material selection.

5. Discussion and future work

Given the manipulator’s kinematic model reported in Tab. 3, the movements of the SCC and of the patient body can be easily related to the motion of the manipulator. In particular, the dynamic model of eqs. (1) and (2) allows the study of the otoconium movements during maneuvers that can be potentially performed by the proposed serial linkage. In fact, the chosen kinematical structure allows to achieve unlimited rotations along the revolution axis of every SCC and to control the SCC planar movement in the x and y directions (Figure 5). Obviously, this kind of RM cannot be manually achieved.

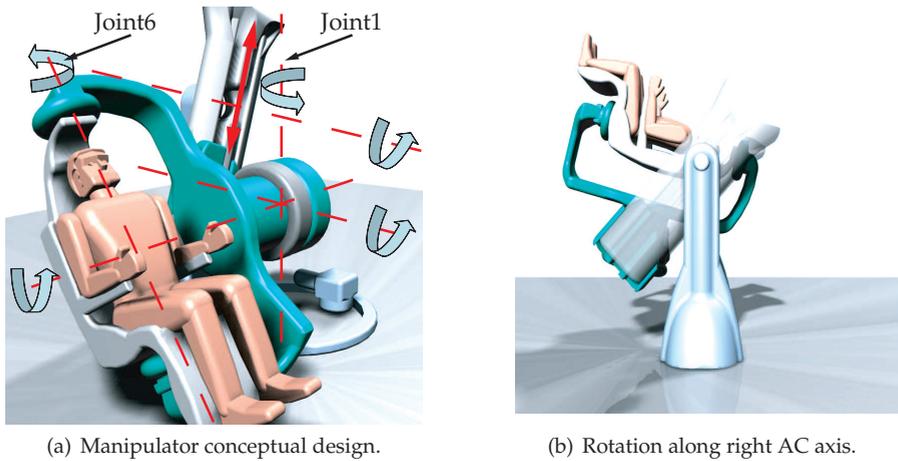


Fig. 11. Manipulator performing tasks.

Note that the choice of suitable strategies for controlling the manipulator (Siciliano & Khatib, 2008) makes it possible to assume that the dynamics of the patient's body movements is described by eq. (1). Therefore, \mathbf{M}_c can be assumed to be the identity matrix. This assumption implies that all the spurious effects that deviate the behavior of the SCC dynamics with respect to eq. (1) must be compensated by the manipulator controller.

It is clear that the otoconium trajectory must be carefully selected in order to achieve a SCC motion that: 1) It is tolerable by the patient (in terms of imposed acceleration); 2) It does not overcome the manipulator limits (in terms of possible pose and velocity/acceleration).

At last, it should be pointed out that eq. (1) is useful when determining the forces which are necessary to accomplish a given RM once the otoconium trajectory and the patient parameters are known.

6. Conclusions

This chapter proves the usability of a simplified Task Based Design technique as an aid in the synthesis of serial linkages and presents a novel robotic chair to be used in diagnosing and treating BPPV. The specification of the tasks to be performed by the chair has been based upon direct specifications given by well-trained doctors or upon the observation that BPPV therapy could be improved once the limits of the manual maneuvers are overcome. Supposing that it is possible to control the motion of a SCC along a plane perpendicular to the SCC revolution axis, an idealized BPPV's dynamic model has been used to show that manually unfeasible maneuvers can be optimized for better treating positional vertigo.

Therefore, as a response to a series of new requirements, the presented novel robotic chair is capable of performing both existing manual maneuver based on otoconia sedimentation and unlimited rotations along the revolution axis of every SCC while controlling the SCC planar motion.

To the best of the author's knowledge, the proposed solution kinematically differs from any existing device and could be used to enhance the rate of success of BPPV non-invasive therapies.

7. Acknowledgment

The authors gratefully acknowledge MedRob Project funded by the University of Bologna for supporting this work and the contribution of Prof. Giovanni Carlo Modugno and Dr. Cristina Brandolini from St. Orsola Hospital, University of Bologna.

8. References

- Arnold, B., Jäger, L. & Grevers, G. (1997). Visualization of inner ear structures by three-dimensional high-resolution magnetic resonance imaging, *Otology and Neurotology* **17**: 935–939.
- Avilés, R., Vallejo, J., Ajuria, G. & Agirrebeitia, J. (2000). Second order methods for the optimum synthesis of multibody systems, *Structural and Multidisciplinary Optimization* **19**: 192–203.
- Baloh, R. W., Sloane, P. D. & Honrubia, V. (1989). Quantitative vestibular function testing in elderly patients with dizziness, *Ear Nose Throat J.* **6**: 1–16.
- Berselli, G., Vassura, G. & Modugno, G. C. (2007). A serial robot for the handling of human body to be used in the diagnosis and treatment of vestibular lithiasis, *Italian Patent No. RM2007A000252, Issued for University of Bologna.* .
- Boniver, R. (1990). Benign paroxysmal positional vertigo. state of the art, *Acta Oto-rhinolaryngologica Belgica* **52**(4): 281–289.
- Brandt, T. & Daroff, R. B. (1980). Physical therapy for benign paroxysmal positional vertigo, *Arch. Otolaryngol.* **106**: 484–485.
- Chedmail, P. & Ramstein, E. (1996). Robot mechanisms synthesis and genetic algorithms, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3466–3471.
- Chen, M. & Burdick, J. (1995). Determining task optimal robot assembly configurations, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 132–137.
- Chocron, O. & Bidaud, P. (1997). Genetic design of 3d modular manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 223–228.
- Della Santina, C., Potyagaylo, V., Migliaccio, A., Minor, L. B. & Carey, J. P. (2005). Orientation of human semicircular canals measured by three-dimensional multiplanar ct reconstruction, *Journal of the Association for Research in Otolaryngology* **68**: 935–939.
- Dix, M. & Hallpike, C. (1952). Pathology, symptomatology and diagnosis of certain common disorders of the vestibular system, *Ann. Otol. Rhinol. Laryngol.* **61**: 987–1016.
- Epley, J. M. (1992). The canalith repositioning procedure - for treatment of benign paroxysmal positional vertigo, *Otolaryngol. Head Neck Surg.* **107**: 399–404.
- Falconi, R. & Melchiorri, C. (2007). Roboticad, an educational tool for robotics, *17th IFAC World Congress*, pp. 9111–9116.
- Falconi, R., Melchiorri, C., Macchelli, A. & Biagiotti, L. (2006). Roboticad: a matlab toolbox for robot manipulators, *8th International IFAC Symposium on Robot Control (Syroco)*, pp. 9111–9116.

- Froehling, D. A., Silverstein, M. D., Mohr, D. N., Beatty, C., Offord, K. P. & Ballard, D. J. (1991). Benign positional vertigo: Incidence and prognosis in a population-based study in olmsted county, Minnesota. *Mayo. Clin. Proc.*, Vol. 66, pp. 596–601.
- Goldberg, D. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley.
- Hain, T. C., Squires, T. M. & Stone, H. A. (2005). Clinical implications of a mathematical model of benign paroxysmal positional vertigo.
- Honrubia, V., Bell, T. S., Harris, M. R., Baloh, R. W. & Fisher, L. M. (1996). Quantitative evaluation of dizziness characteristics and impact on quality of life, *Am. J. Otol.*, Vol. 17, pp. 595–602.
- House, M. G. & Honrubia, V. (2003). Theoretical models for the mechanisms of benign positional paroxysmal vertigo, *T. Audiol. Neurootol.* 8: 91–99.
- Kelmar, L. & Khosla, P. (1990). Automatic generation of forward and inverse kinematics for a reconfigurable modular manipulator system, *Journal of Robotic Systems* 7(4): 599–619.
- Kim, J. (1992). *Task Based Kinematic Design of Robot Manipulators*, PhD thesis, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA.
- Kim, J. & Khosla, P. (1993a). Design of space shuttle tile servicing robot: An application of task based kinematic design, *IEEE International Conference on Robotics and Automation*, pp. 867–874.
- Kim, J. & Khosla, P. (1993b). A formulation for task based design of robot manipulators, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2310–2317.
- Kuka (2004). Six-dimensional fun the world first passenger-carrying robot.
- Nakayama, M. & Epley, J. (2005). Bppv and variants: Improved treatment results with automated, nystagmus-based repositioning, *Ornithology-Head and Neck Surgery* 133: 107–112.
- Obrist, D. & Hegemann, S. (2008). Fluid-particle dynamics in canalithiasis, *Journal of The Royal Society Interface* 5(27): 1215–1229.
- Pagnini, P., Nuti, D. & Vannucchi, P. (1989). Benign paroxysmal vertigo of the horizontal canal, *J. Otorhinolaryngol. Relat. Spec.* 1989: 161–170.
- Parnes, L., Agrawal, S. K. & Atlas, J. (2003). Diagnosis and management of benign paroxysmal positional vertigo (bppv).
- Rajguru, S. M., Ifediba, M. A. & Rabbitt, R. D. (2004). Three-dimensional biomechanical model of benign paroxysmal positional vertigo.
- Raphan, T., Matsuo, V. & Cohen, B. (1979). Velocity storage in the vestibulo-ocular reflex arc, *Exp. Brain Res.* 35: 229–248.
- Robinson, D. A. (1977). Linear addition of optokinetic and vestibular signals in the vestibular nucleus, *Exp. Brain Res.* 30: 447–450.
- Semont, A., Freyss, G. & Vitte, E. (1980). Curing the bppv with a liberatory maneuver, *Adv. Otorhinolaryngol.* 42: 290–293.
- Shinichiro, H., Hideaki, N., Koji, T., Akihiko, I. & Makito, O. (2005). Three dimensional reconstruction of the human semicircular canals and measurement of each membranous canal plane defined by reids stereotactic coordinates, *Annals of Othology, Rhinology and Larintology* 112-2: 934–938.
- Siciliano, B. & Khatib, O. (2008). *Handbook of Robotics*, Springer.
- Squires, T. M., Weidman, M. S., Hain, T. C. & Stone, H. A. (2004). A mathematical model for top-shelf vertigo: the role of sedimenting otoconia in BPPV, *Journal of Biomechanics* 37(8): 1137 – 1146.

- Teixido, M. (2006). Inner ear anatomy, *Delware Biotechnology Institute*, [online]. Available: www.dbi.udel.edu/MichaelTeixidoMD/.
- Yang, G. & Chen, I. (2000). Task-based optimization of modular robot configurations: M dof approach, *Mechanism and Machine Theory* **35**(4): 517–540.

A Wire-Driven Parallel Suspension System with 8 Wires (WDPSS-8) for Low-Speed Wind Tunnels

Yaqing ZHENG^{1,2}, Qi LIN^{1,*} and Xiongwei LIU³

1. Department of Aeronautics Xiamen University, Xiamen, 361005, China

2. College of Mechanical Engineering and Automation, Huaqiao University, Quanzhou, 362021, China

3. School of Computing, Engineering and Physical Sciences, University of Central Lancashire, Preston, UK

1. Introduction

As a new type of parallel manipulator, the wire-driven parallel manipulator has advantageous characteristics such as simple and reconfiguration structure, large workspace, high load capacity, high load/weight ratio, easy assembly/disassembly, high modularization, low cost and high speed. A new concept has been proposed by using the wire-driven parallel manipulator as aircraft model suspension system in low-speed wind tunnel tests for nearly 9 years. The authors of the context have undertaken over six years research work about a 6-degree-of-freedom (DOF) wire-driven parallel suspension system with 8 wires (WDPSS-8) for low-speed wind tunnels, and achieved some deep understanding. The attitude control and aerodynamic coefficients (static derivatives) of the scale model have been investigated both theoretically and experimentally. Two prototypes (WDPSS-8) have been developed and tested in low-speed wind tunnels. It is also found the possibility to use the prototypes (WDPSS-8) for the experiment of dynamic derivatives by successfully implementing the single-DOF oscillation control to the scale model. In order to investigate the feasibility of using the same wire-driven parallel suspension system for the static and dynamic derivatives experiments in low-speed wind tunnels, the research will go on in this direction. The research results, particularly the experiments of the dynamic derivatives, will provide some criterion of experimental data for the free flight and some effective experimental methods, which deal with the controllability capability of post stall maneuvers in the design of great aircrafts and a new generation of vehicles. Concerning the research outcomes, 4 projects including sponsored by NSFC (National Natural Science Foundation of China) have been finished and 21 papers(c.f. Appendix) have been published by our group.

* Corresponding authors: Qi LIN, E-mail: qilin@xmu.edu.cn

2. Background

2.1 The Traditional Rigid Suspension Systems for Wind Tunnels

Wind tunnel test is one important way to obtain the aerodynamic coefficients of the aircrafts. During the wind tunnel tests, it is necessary to support the scale model of the aircraft in the streamline flow of the experimental section of the wind tunnel using some kind of suspension system. The suspension system will have a lot of influences on the reliability of the results of the wind tunnel tests. The traditional rigid suspension systems have some unavoidable drawbacks for the blowing experiments of static and dynamic derivatives such as the serious interference of the strut on the streamline flow [1-6].

2.2 The Cable-Mounted Systems for Low-Speed Wind Tunnels

The cable-mounted systems for wind tunnel tests, developed in the past several decades, deal with the contradiction between the supporting stiffness and the interference on the streamline flow [7-12]. However their mechanism is not robotic and consequently quite different from wire-driven parallel suspension systems in attitude control schemes and force-measuring principle. In addition they can not be used in the dynamic derivatives experiments [12].

2.3 The Wire-Driven Parallel Suspension Systems (WDPSS) for Low-Speed Wind Tunnels

Instead, the free-flight simulation concept in wind tunnels through an active suspension, such as six-DOF wire-driven parallel suspension systems (WDPSS), is suitable to get the aerodynamic coefficients of the aircraft's model [13-16], which comes from the research improvement in wire-driven parallel manipulator and force control. Some successful achievements in this field have been made in the Suspension Active pour Soufflerie (SACSO) project supported by French National Aerospace Research Center (ONERA) for nearly 8 years. And they have been applied in vertical wind tunnel tests with a wind speed of 35 m/s for fighters at the first stage of their conceptual design. However the system can not be used in the experiment of dynamic derivatives [13, 14].

The goal of this context is to introduce some contributions in the field of wire-driven parallel suspension systems for static and dynamic derivatives of the aircraft model for low-speed wind tunnels. Under the sponsorship of NSFC (National Natural Science Foundation of China), the research work about a 6-DOF wire-driven parallel suspension system with 8 wires (WDPSS-8) for low-speed wind tunnels has been carried out by the authors over 6 years, and some deep and systematic results have been published [15-26]. The attitude control and aerodynamic coefficients (static derivatives) of the scale model have been investigated in theory and in experiment.

Two prototypes (WDPSS-8) have been built and tested in two different low-speed wind tunnels respectively [21,23,25]. And with the prototype, the single-DOF oscillation control of the scale model has been implemented successfully [23-26]. This shows it is possible to use a WDPSS for the experiment of static and dynamic derivatives in low-speed wind tunnel.

Concerning the possibility of using the same WDPSS to make the static and dynamic derivatives experiments in low-speed wind tunnels, a survey of the research work finished about some key issues of WDPSS-8 in wind tunnel experiments will be addressed. The research results, especially in the experiments of the dynamic derivatives, will provide some

criterion of experimental data for the free flight and some effective experimental methods about the controllability capability of post stall maneuvers in the design of a new generation of aircrafts and vehicles, which will help to provide a novel support system in the field of wind tunnel tests of aircrafts.

The rest of the Chapter is organized as follows: The key issues of WDPSS-8 for the experiments of static derivatives of the aircraft's model in Low-Speed Wind Tunnels (LSWT) are given in the next Section 3. The research results of WDPSS-8 for the experiments of dynamic derivatives of aircrafts in LSWT are presented in Section 4. Finally, discussions and future works are suggested in Section 5.

3 WDPSS-8 for Experiments of Static Derivatives of Aircrafts for Low-Speed Wind Tunnels

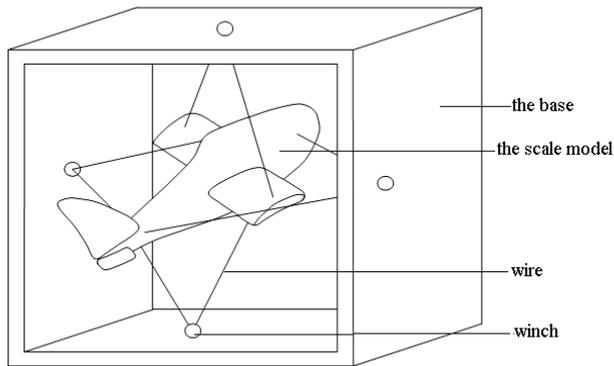
3.1 Two WDPSS-8 prototypes

3.1.1 A Manually operated WDPSS-8 prototype validated in a closed circuit wind tunnel

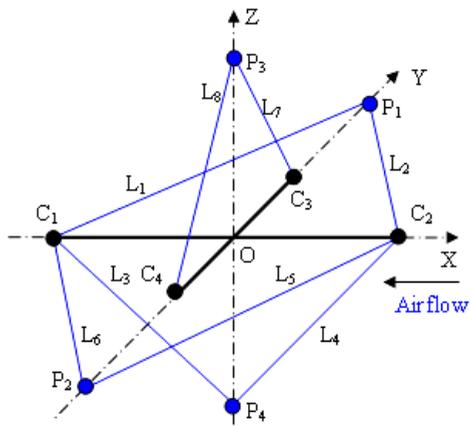
A wire-driven parallel manipulator is a closed-loop mechanism where the moving platform is connected to the base through wires by multitude of independent kinematic chains. The number of moving platform's degree of freedoms (DOFs) is defined as the dimension of linearity space which is positively spanned by all the screws of the structure matrix of the manipulator. So the moving platform of a 6-DOF completely or redundantly restrained wire-driven parallel manipulator is driven by at least 7 or more wires.

Meanwhile, a 6-DOF WDPSS is essential for free flight of the aircraft's model in a 3-dimensional space wind tunnel. Fig. 1(a) shows the concept of a 6-DOF WDPSS driven by 8 wires (WDPSS-8). Its geometric definition is shown in Fig. 1(b). A manually operated prototype of such a design shown in Fig.1 (c) is built and tested in a closed circuit LSWT, the geometric parameters of which are listed in Table 1. To implement the scheme for the attitude adjustment of the aircraft, a driving mechanism adjusted manually has been developed which allows the aircraft model to maneuver, i.e., to permit roll, pitch and yaw motion. For the WDPSS-8, each cable will be attached to a driving unit, which consists of a screw bar and a driving nut, as shown in Fig. 1(d). A commercial load cell interfaced to the cable shown in Fig. 1(e) is used to measure the tension of a cable. To avoid extra interference, the strain gage balance and driving unit are attached to the wind tunnel frame on the outside of the tunnel, as shown in Fig. 1(c) and (d). However, Fig. 1(f) shows the aircraft model mounted on a conventional strut supporter system in the same LSWT.

The WDPSS-8 prototype has been validated by wind tunnel tests in a wind speed of 28.8 m/s. It was found in the experiments that there is little vibration occurring at the end of the scale model, which is less than that in the corresponding traditional strut supporter system shown in Fig. 1(f). And it was also been found that the fundamental frequency of WDPSS-8 is smaller than that of the corresponding traditional strut supporter. It shows the rigidity of WDPSS is better than that of traditional strut s supporter. This phenomenon will be more serious when the model is bigger and heavier. Therefore, as a supporter system of scale models in low-speed wind tunnel test, WDPSS is more suitable for researching and developing new great aircraft.



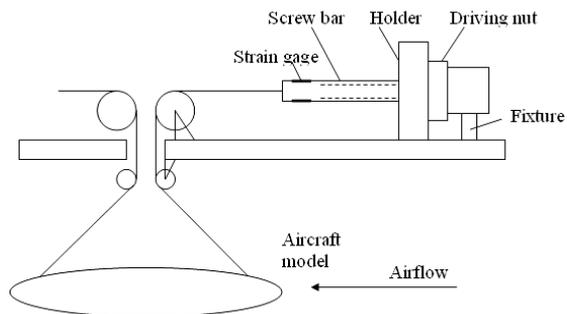
(a) The concept of wire-driven parallel suspension system with 8 wires (WDPSS-8)



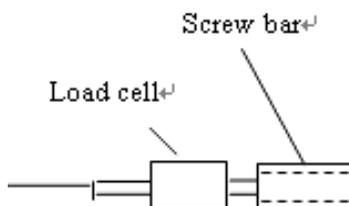
(b) Geometric definition of the WDPSS-8 prototype



(c) WDPSS-8 tested in closed LSWT with a wind speed of 28.8 m/s



(d) Driving unit of WDPSS-8



(e) Load cell interface



(f) traditional Strut supporter system

Fig. 1. Comparison of 2 different suspension systems

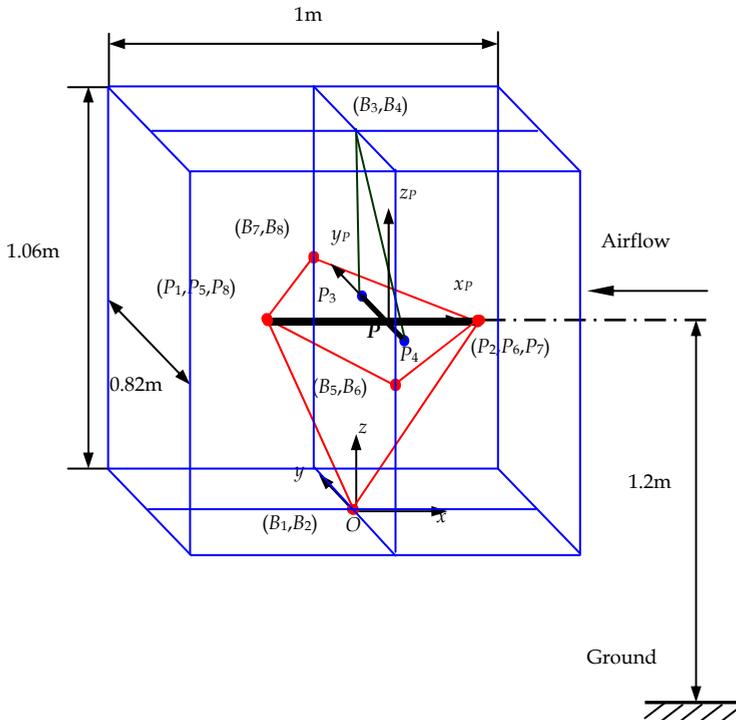
Pitch (°)	$C_1 (X,Y,Z)$	$C_2 (X,Y,Z)$	$C_3 (X,Y,Z)$	$C_4 (X,Y,Z)$
-6	-300, 0, -32	300, 0, 32	0, 605, -30	0, -605, -30
0	-302, 0, 0	302, 0, 0	0, 605, -30	0, -605, -30
6	-300, 0, 32	300, 0, -32	0, 605, -30	0, -605, -30
12	-295, 0, 63	295, 0, -63	0, 605, -30	0, -605, -30
	$P_1 (X,Y,Z)$	$P_2 (X,Y,Z)$	$P_3 (X,Y,Z)$	$P_4 (X,Y,Z)$
	0, 0, 420	0, 0, -420	0, 605, 0	0, -605, 0

Table 1. Geometric parameters of the WDPSS-8 prototype (unit :mm)

3.1.2 Another WDPSS-8 prototype tested in an open return circuit wind tunnel

To meet need of open wind tunnels, another kind of WDPSS has to be developed. Second WDPSS-8 presented in the context is one of them. The geometric definition of the WDPSS-8 is shown in Fig. 2(a). And its structural parameters are listed in Table 2.

A test platform about this WDPSS-8 for low-speed wind tunnels realized also is shown in Fig.2 (b) and Fig.2 (c), in which the 3 rotational attitude control of the scale model (yaw, roll and pitch) has been accomplished [27]. The corresponding prototype has been built shown in Fig. 3. During the wind tunnel testing, it is necessary to place the scale model using the suspension system in the experimental section of wind tunnels. And the attitude of the scale model must be adjustable. To give different attitude of the scale model in movement control, the inverse kinematics problem is required to be solved to deals with the calculation of the length of each cable correspond to the attitude wanted of the model. The solution to the problem will provide the data for the movement control experiment. The modeling of inverse pose kinematics of WDPSS-8 can be found in references [18, 22].



(a) Another geometric definition of WDPSS-8 prototype



(b) Prototype of WDPSS-8



(c) Circuit connecting in the control cupboard

Fig. 2. WDPSS-8 prototype for open return circuit wind tunnel

$P_1 (x_p, y_p, z_p)$	$P_2 (x_p, y_p, z_p)$	$P_3 (x_p, y_p, z_p)$	$P_4 (x_p, y_p, z_p)$
-150, 0, 0	120, 0, 0	0, 142.5, 0	0, -142.5, 0
$B_1 (X, Y, Z)$	$B_3 (X, Y, Z)$	$B_5 (X, Y, Z)$	$B_7 (X, Y, Z)$
0, 0, 0	0, 0, 1060	0, -410, 530	0, 410, 530

Table 2. Structural parameters of the WDPSS-8 prototype (unit: mm)

3.2 Calculation of the static derivatives

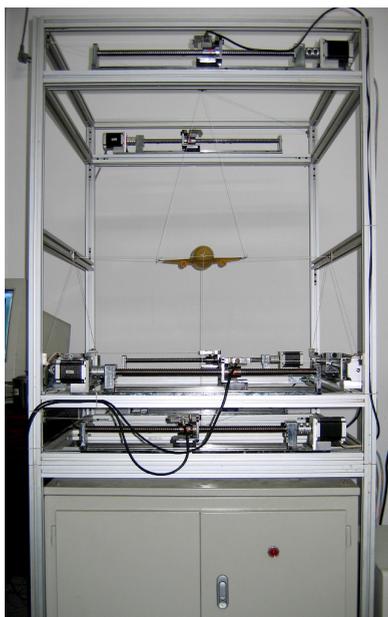


Fig. 3. Control experiment of attitude angle

Because the scale model moves in a quasi-static way during the LSWT experiment for the static derivatives, it is reasonable to calculate the aerodynamic force and torque exerted on it using the difference of the force and torque exerted on the scale model between without wind and with wind. As the preliminary research, the assumption that all constraints are perfectly applied with no resistance in pulleys or other mechanisms such as point-shaped joints which are required to maintain the geometry of the wires at the base and the scale model is given for the convenience. Maybe this is not practically the case, but it is reasonable because the attitude of the scale model is controlled and adjusted in a quasi-static way so that the errors about the mechanism configuration between without wind and with wind could easily limited to a range that can be neglected.

The static model of WDPSS-8 without wind can be expressed by:

$$J^T T + F_G = 0 \quad (1)$$

Here, T is a tension vector $(t_1 \dots t_8)^T$ with 8 components related to 8 wires without wind, 0 is a null vector with 6 components, J^T is the structural matrix of the manipulator, F_G is the gravity vector with 6 components.

The static model of WDPSS-8 with wind load can be expressed by:

$$J^T T_W + F_G + F_A = 0 \quad (2)$$

Here, F_A is the vector of aerodynamic force and torque with 6 components, and T_W is the tension vector composed of the tension of 8 wires with wind.

From Eqs.(1) and (2), it can be found that the equation $F_A = J^T(T - T_W)$ is satisfied.

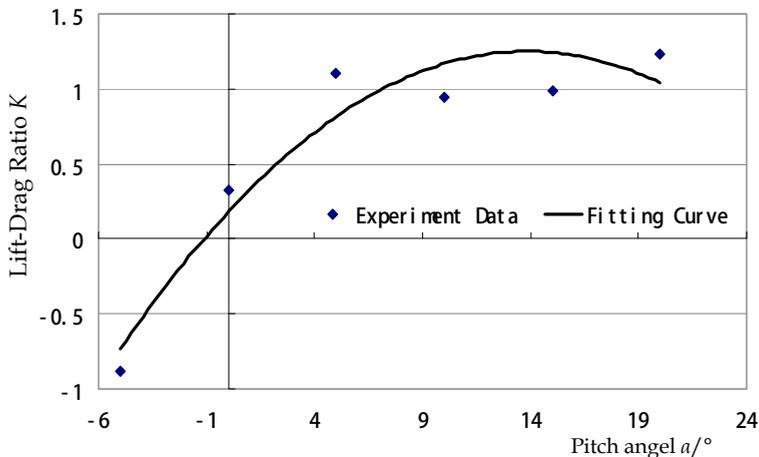
In order to calculate the static derivatives (related to F_A), the tension of all wires and the posture of the scale model need to be measured when the position of the scale model is controlled without wind and with wind. The experiment of static derivatives using second WDPSS-8 has been finished in an open return circuit low-speed wind tunnel, which will be stated in the following in detail.

The prototype of second WDPSS-8 has been set in an open return circuit low-speed wind tunnel for blowing test, as shown in Fig.7. The experimental section of the wind tunnel is rectangular with the width of 0.52 meter and the height of 0.50 meter. The space has a length of 1 meter long [26].

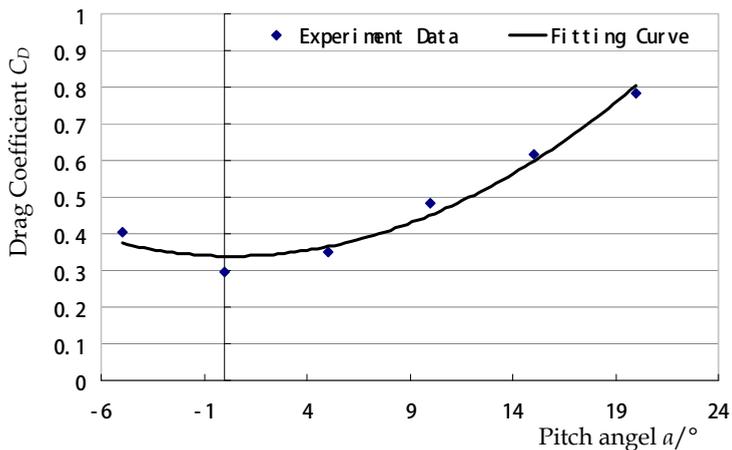


Fig. 7. Second WDPSS prototype in open return circuit LSWT for blowing test

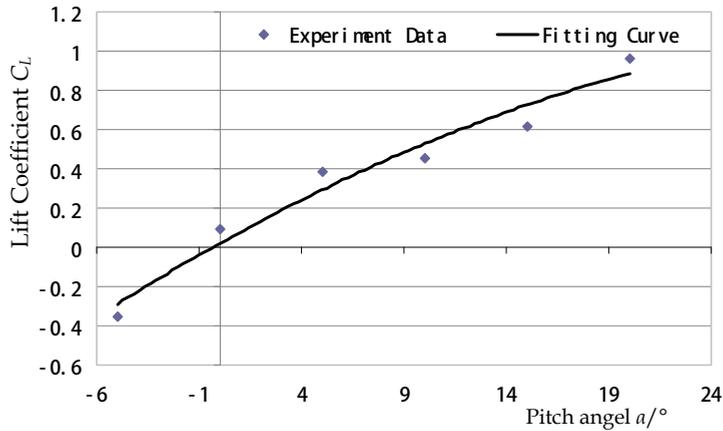
As shown in Fig.7, an airplane model is suspended by second WDPSS-8 in the experimental section of the open return circuit low-speed wind tunnel for tests. And the airflow speed can be adjusted among 0~50m/sec.



(a) Lift-Drag Ratio K vs. pitch angle a



(a) Drag coefficient C_D vs. pitch angle a



(a) Lift coefficient C_L vs. pitch angle a

Fig. 8. Aerodynamic parameter curves from wind tunnel test with WDPSS-8

Because the F_A is determined by T and T_W , every components $t_i(i=1,2,\dots,8)$ of them must be obtained in wind tunnel test. The force-measurement system in the WDPSS-8 consists of the power, force sensors, transducers, interface circuit and data acquisition card.

A group of wind tunnel tests has carried out and a series of experimental curves including lift coefficient C_L , drag coefficient C_D and lift/drag ratio K versus angle of pitch has been acquired by calculating the equation $F_A = J^T(T - T_W)$.

As shown in Fig.8, there are 3 experimental curves for wind tunnel testing of WDPSS-8 with a wind speed of 29.37m/s. Though there is no data about the standard model as a criterion, the curves are reasonable and suitable for expressing the aerodynamic characteristics of the airplane model.

4 WDPSS-8 for the Experiments of Dynamic Derivatives of the Aircraft model for Low-Speed Wind Tunnels

To get dynamic derivatives, the single-DOF oscillation control to the scale model with support system in wind tunnel and the calculation of dynamic derivatives are all very important steps. As a novel attempt, the former has been realized on the prototype of second WDPSS-8. And in theory, the calculating method for dynamic derivatives with WDPSS in low-speed wind tunnel has been also investigated.

In the following, the preliminary oscillation control of the scale model implemented in the test platform of WDPSS-8 will be stated at first, and then the calculation of dynamic derivatives will be given after based on the analysis of the dynamic modeling of the system and oscillation control scheme.

4.1 The preliminary oscillation control of the scale model

With the prototype of second WDPSS-8, the single-DOF oscillation control of the scale model has

been implemented successfully [23-26]. This shows it is possible to use WDPSS for the experiment of dynamic derivatives.

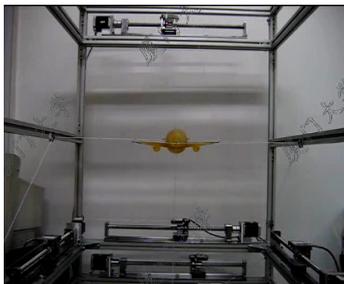


Fig. 9. The single-DOF pitch oscillation control of the scale model

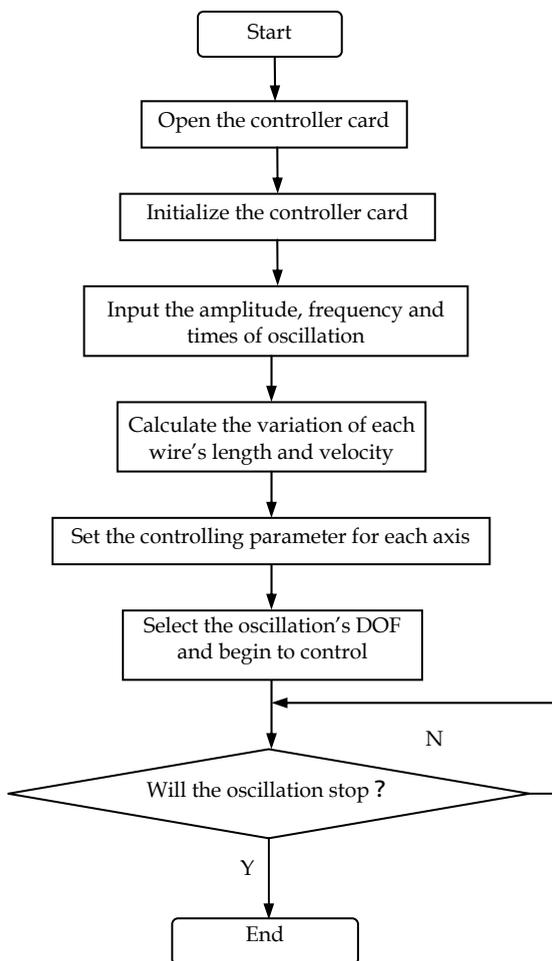


Fig. 10. Flow chart of oscillation control program

Attitude	Amplitude (°)	Frequency (Hz)
Pitch Angle	5	1.5
	10	0.625
Roll Angle	5	2
	10	1.25
Yaw Angle	5	0.625
	10	0.5

As shown in Fig.9, an airplane model suspend by the prototype of WDPSS-8 has been controlled to oscillate in the single-DOF (including pitch, roll and yaw). The amplitude ranges from 0 to 10 degree and the frequency ranges from 0 to 2 Hz for each kind of oscillation. The flow chart of oscillation control program is shown in Fig.10. According to the requirements for the experiments of dynamic derivatives, the oscillation control of the scale model is accomplished according to the suitable selection of the parameters for amplitude and frequency listed in Table 3. More detailed information and video about the experimental results can be found in the URL: <http://blog.sina.com.cn/AircraftEngineering> [27].

4.2 Dynamic modeling of the system and the scheme of oscillation control

The dynamic modeling to suspension system is necessary to design the control system for the oscillation of the scale model. In building the dynamic model, the assumptions are given as follows:

- The deformation of wires is so small that it may be neglected, and the mass of the wires can be neglected as well.
- The dynamics of the actuators is neglected to simplify the dynamics model of the manipulator.

In references [18, 20], the total dynamic modeling of WDPSS-8 is written as:

$$(\mathbf{M}_0 + \mathbf{J}^T \mathbf{M} \mathbf{J}) \ddot{\mathbf{X}} + (\dot{\mathbf{M}}_0 + \mathbf{J}^T \mathbf{M} \dot{\mathbf{J}} + \mathbf{J}^T \mathbf{B} \mathbf{J}) \dot{\mathbf{X}} = \mathbf{J}^T \boldsymbol{\tau} - \mathbf{F}_g \quad (3)$$

Here $\mathbf{M}_0 = \begin{bmatrix} (m_p \mathbf{I})_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{A}_{G(3 \times 3)} \end{bmatrix}$ is the inertia matrix of the scale model including any attached payload, m_p is the mass and \mathbf{A}_G is the inertia tensor about the gravity center. $\mathbf{M} = \text{diag}(m_1, \dots, m_8) \in \mathbb{R}^{8 \times 8}$ is the inertia matrix of the actuators, \mathbf{X} is vector of the posture of the scale model, $\mathbf{B} = \text{diag}(b_1, \dots, b_8) \in \mathbb{R}^{8 \times 8}$ is the matrix of viscous friction of

the actuators , $\tau = (\tau_1, \dots, \tau_8)^T \in \mathbb{R}^8$ is vector of the torque of the actuators , $F_g = (0, 0, m_p \cdot g, 0, 0, 0)^T$ is the gravity vector of the scale model, and $g=9.8$ (m/s²).

The total dynamic modeling is a highly coupled and redundantly restrained nonlinear system which should be decoupled and linearized.

A control law of actuator vector of motor is designed as follows:

$$\tau = (J^T)^{-1} (K_d(\dot{X}_d - \dot{X}) + K_v(\ddot{X}_d - \ddot{X}) + F_g) + v \tag{4}$$

Here, $J^T v=0$ is satisfied, moreover K_d and K_v are the different values of the control feedback gain without wind respectively. It can be proven that the control system is stable and robust with the control law mentioned above.

It is noted that if another control law is used that can ensure the stability of the control system, the values of dynamic derivatives calculated from the control system that will be formulated in detail in the next section will be different. There may occur a question about the correctness of the method for the dynamic derivatives' calculation. However, it is regarded as reasonable after a balance of the analysis of the differences of control schemes and of their robustness is given. Much more work in the aspect will be discussed in the future work. Moreover the required repeatability of the control system will be indicated and investigated using some kind of tools like robustness.

4.3 Dynamic derivatives' calculation

In the test platform of WDPSS-8, the oscillation control of the scale model is controlled without wind and with wind respectively. According to the experimental data, the aerodynamic force and torque can be calculated by the dynamic equations of the system. Also the dynamic derivatives may be calculated by the real torques of motors without wind and with wind, which can be measured by the force sensors mounted on the axis of the motors.

Taking the pitch oscillation as an example, i.e., $X = \theta_p = \{0 \ 0 \ 0 \ 0 \ \theta_p \ 0\}^T = \{0 \ 0 \ 0 \ 0 \ \theta_{p0} \sin \omega t \ 0\}^T$, the total dynamic model without wind and with wind can be obtained. In fact, the single-DOF oscillation control of the scale model has been successfully executed on the WDPSS-8 prototype, the frequency of which is from 0~2 Hz and the amplitude of which is 5~10 degree, see Table 3[25]. In addition, the motion versus time tracks that differs by small fractions of a degree should be provided by a motion control system. To obtain accurate measurements of the dynamic derivatives, maybe a time-varying discrete control system should be built and investigated. As the preliminary research, it is only regarded as a time-constant continuous control system. Moreover, the specifications on the path of the single-DOF oscillation of the scale model matching with wind on and wind off should be considered to an extent that target precisions are related to expected accuracies of the measured derivatives, but this issue will not be discussed here. Much more work in the aspect will be discussed in the future work by the tools like the robustness of the control system.

□ Under the condition when the scale model has a pure pitch rotation without wind, Eq.(3) can be written as

$$(\mathbf{M}_0 + \mathbf{J}^T \mathbf{M} \mathbf{J}) \ddot{\theta}_p + (\dot{\mathbf{M}}_0 + \mathbf{J}^T \dot{\mathbf{M}} \mathbf{J} + \mathbf{J}^T \mathbf{B} \mathbf{J}) \dot{\theta}_p = \mathbf{J}^T \boldsymbol{\tau} - \mathbf{F}_g \quad (5)$$

From Eq.(5), $a \ddot{\theta}_p + b \dot{\theta}_p = c$ can be got. Here, a is the result of adding all the elements of the 5th row of Matrix $(\mathbf{M}_0 + \mathbf{J}^T \mathbf{M} \mathbf{J})$,

b is the result of adding all the elements of the 5th row of Matrix $(\dot{\mathbf{M}}_0 + \mathbf{J}^T \dot{\mathbf{M}} \mathbf{J} + \mathbf{J}^T \mathbf{B} \mathbf{J})$,

$c = K_d(\theta_{Pd} - \theta_p) + K_v(\dot{\theta}_{Pd} - \dot{\theta}_p)$, θ_{Pd} and $\dot{\theta}_{Pd}$ are the desired pitch angle and angular velocity of the scale model.

$M_y(t)_{off}$, defined as the oscillation torque vector of the system without wind when the scale model has a pure pitch rotation, satisfies

$$M_y(t)_{off} = (\overline{M_y})_{off} \sin(\omega t + \lambda) = a \ddot{\theta}_p + b \dot{\theta}_p = c \quad (6)$$

□ Under the condition when the scale model has a pure pitch rotation with wind, Eq.(3) can be written as

$$M_y^{\theta_p} \ddot{\theta}_p + M_y^{\dot{\theta}_p} \dot{\theta}_p + M_y^{\theta_p} \theta_p + (\mathbf{M}_0 + \mathbf{J}^T \mathbf{M} \mathbf{J}) \ddot{\theta}_p + (\dot{\mathbf{M}}_0 + \mathbf{J}^T \dot{\mathbf{M}} \mathbf{J} + \mathbf{J}^T \mathbf{B} \mathbf{J}) \dot{\theta}_p = \mathbf{J}^T \boldsymbol{\tau}' - \mathbf{F}_g \quad (7)$$

$M_y(t)_{on}$, defined as the oscillation torque vector of the system with wind when the scale model has a pure pitch rotation, satisfies

$$M_y(t)_{on} = (\overline{M_y})_{on} \sin(\omega t + \lambda) = K'_d (\theta_{Pd} - \theta_p) + K'_v (\dot{\theta}_{Pd} - \dot{\theta}_p) \quad (8)$$

From Eqs. (5) and (7), the following equation can be obtained,

$$M_y^{\theta_p} \ddot{\theta}_p + M_y^{\dot{\theta}_p} \dot{\theta}_p + M_y^{\theta_p} \theta_p = ((\overline{M_y})_{on} - (\overline{M_y})_{off}) \sin(\omega t + \lambda) \quad (9)$$

From Eqs.(6), (8) and (9), the following equation can be obtained,

$$((\overline{M_y})_{on} - (\overline{M_y})_{off}) \sin(\omega t + \lambda) = (K'_d - K_d) (\theta_{Pd} - \theta_p) + (K'_v - K_v) (\dot{\theta}_{Pd} - \dot{\theta}_p) \quad (10)$$

In fact, the 5th element of vector torque $\boldsymbol{\tau}$ and $\boldsymbol{\tau}'$ of the motors, τ_y and τ'_y , can be measured by tension sensors, which can be respectively expressed by

$$\tau_y = a_{1\tau_y} \sin \alpha t + b_{1\tau_y} \cos \alpha t; \tau'_y = a'_{1\tau_y} \sin \alpha t + b'_{1\tau_y} \cos \alpha t \quad (11)$$

Hence, Eq.(9) can be expressed by

$$((\overline{M}_y)_{on} - (\overline{M}_y)_{off}) \sin(\omega t + \lambda) = J_5(a'_{1\tau_y} - a_{1\tau_y}) \sin \alpha t + J_5(b'_{1\tau_y} - b_{1\tau_y}) \cos \alpha t.$$

Here J_5 is the result of adding all the elements of the 5th row of Matrix \mathbf{J}^T .

Owing to the 3 equations: $\theta_p = \theta_{p0} \sin \omega t$, $\dot{\theta}_p = \omega \theta_{p0} \cos \omega t$, $\ddot{\theta}_p = -\omega^2 \theta_{p0} \sin \omega t$, Eq.(9) can be rewritten as

$$M_y^{\theta_p} - \omega^2 M_y^{\ddot{\theta}_p} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\theta_{p0}} \cos \lambda = J_5(a'_{1\tau_y} - a_{1\tau_y});$$

$$M_y^{\dot{\theta}_p} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\omega \theta_{p0}} \sin \lambda = J_5(b'_{1\tau_y} - b_{1\tau_y})$$

Owing to the 3 equations: $M_y^{\dot{\theta}_p} = M_y^{\dot{\alpha}} + M_y^{\dot{\omega}_y}$; $M_y^{\theta_p} = M_y^{\omega_y}$; $M_y^{\ddot{\theta}_p} = M_y^{\alpha}$, Eqs.(9) may be rewritten as:

$$M_y^{\alpha} - \omega^2 M_y^{\dot{\omega}_y} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\theta_{p0}} \cos \lambda = J_5(a'_{1\tau_y} - a_{1\tau_y}) \quad (12)$$

$$M_y^{\dot{\alpha}} + M_y^{\dot{\omega}_y} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\omega \theta_{p0}} \sin \lambda = J_5(b'_{1\tau_y} - b_{1\tau_y}) \quad (13)$$

Eqs. (11) and (12) can also be rewritten as Eq.(14) and Eq.(15) respectively, as follows

$$m_y^{\alpha} - K^2 m_y^{\dot{\omega}_y} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\theta_{p0} q s b_A} \cos \lambda = \frac{J_5(a'_{1\tau_y} - a_{1\tau_y})}{\theta_{p0} q s b_A} \quad (14)$$

$(m_y^{\alpha} - K^2 m_y^{\dot{\omega}_y})$ is called In-Phase Pitch Oscillatory Derivatives. $K = \frac{\omega b_A}{V}$ is called reduced frequency, b_A is called mean aerodynamic chord length. V is called free-stream airflow velocity.

$$m_y^{\dot{\alpha}} + m_y^{\dot{\omega}_y} = \frac{(\overline{M}_y)_{on} - (\overline{M}_y)_{off}}{\theta_{p0} q s b_A K} \sin \lambda = \frac{J_5(b'_{1\tau_y} - b_{1\tau_y})}{\theta_{p0} q s b_A K} \quad (15)$$

$(m_y^{\dot{\alpha}} + m_y^{\dot{\omega}_y})$ is called Out-of-Phase Pitch Oscillatory Derivatives.

In the same way, the dynamic derivatives corresponding to the other 2 single-DOF oscillation (roll oscillation and yaw oscillation) also can be obtained.

Surely the analysis is just based on the theoretical aspects. The test platform of WDPSS-8 for the experiment of dynamic derivatives is still to be built and the precise measuring systems of the vibration angular displacement and the real torque of the motors should be designed and implemented.

5. Conclusions and Future Works

Basen on researching into WDPSS-8, after analysing and comparing 3 support systems in wind tunnel including the strut suspension system, cable mounted system and wire-driven parallel suspension system, the following conclusions can be acquired.

(1) Till now the strut support systems and rotary balances can be used in measuring the dynamic derivatives of the aircraft in low-speed wind tunnels successfully, wire-driven parallel suspension system has a great potentiality, but it is still under investigation.

(2) The cable mounted system is one of suitable method for measuring the static derivatives of the aircraft in LSWT. Though it allows a large supporting stiffness, small interference of the streamline flow and a high measuring precision for large attack angle, it can not be used in measuring the dynamic derivatives.

(3) Wire-driven parallel suspension system has opened a new horizon for measuring the static and dynamic derivatives of the aircraft in LSWT. Using the same system based on position control and force control in robotics, it allows to realize the free flight of the aircraft model and to obtain the static and dynamic derivatives. However, the results given in this Chapter can only be considered as a preliminary step in establishing feasibility, although the wire-driven parallel suspension system is a very interesting design, and it may be sufficiently developed into a routine practical system.

6. Acknowledgements

This research is sponsored by National Natural Science Foundation of China(Grant No. 50475099), the Youth Talents Creation Foundation of Fujian Province (Grant No. 2006F3083). and the Scientific Launch Foundation for the Talents of Huaqiao University (Grant No. 06BS218).

7. References

- [1]Sun, H.S., 1999, "The development of 96-test system for measuring dynamic derivatives at high angle of attack", *Journal of Experiments in Fluid Mechanics*, Vol.13,No.1,pp.31-37.[in Chinese]
- [2]Zhang, R.P., Wang, X.N., Li, Z.X., et al, 2006, "Investigation of sting support interference in low speed wind tunnel", *Journal of Experiments in Fluid Mechanics*, Vol.20,No.3,pp. 33-38.[in Chinese]
- [3] Roos, F.W., 2001, "Micro blowing for high-angle-of-attack vortex flow control on fighter aircraft", *Journal of Aircraft*, Vol.38,No.3,pp.454-457
- [4] Bernhardt, J.E., Williams, D.R., 2000, "Close-loop control of forebody flow asymmetry", *Journal of Aircraft*, Vol.37,No.3,pp.491-498
- [5] Cao, Y.H., Deng, X.Y., 1998, "Experimental study of aircraft fuselage aerodynamic forces at high attack angle", *Acta Aerodynamica Sinica*, Vol.16,No.4,pp. 394-399.[in Chinese]
- [6] Du, X.Q., Wang, X.J., Zhou, Y., 2003, "Investigation of support interference in low speed wind tunnel oscillation tests", *Experiments and Measurements in Fluid Mechanics*, Vol.17,No.2,pp.37-40.[in Chinese]

- [7] Bian, Y.Z., Li, X.R., Li, L., et al, 1999, "Development of model cable mount and strain gauge balance measuring system in low speed closed wind tunnel", *Experiments and Measurements in Fluid Mechanics*, Vol.13, No.3, pp.85-90. [in Chinese]
- [8] Shen, L.M., Shen, Z.H., Huang, Y., 1988, "A wire type-support system for high angle of attack test in low speed wind tunnel", *Experiments and Measurements in Fluid Mechanics*, Vol.12, No.4, pp.15-22. [in Chinese]
- [9] Bennett, R.M., Farmer, M.G., Mohr, R.L., et al, 1978, "Wind-tunnel technique for determining stability derivatives from cable-mounted models", *Journal of Aircraft*, Vol.15, No.5, pp. 304-310
- [10] Liu, T.F., Gao, F.X., Lv, W., 2001, "Mechanical property calculation of wire-support structure in wind tunnel experiment", *Mechanics in Engineering*, Vol.23, No.1, pp. 46-48. [in Chinese]
- [11] Wang, Y.K., Huang, T., Deng, X.Y., 2004, "The study of aerodynamic characteristics of afterbody and effect of cable-mounting", *Chinese Journal of Theoretical Applied Mechanics*, Vol.36, No.3, pp.257-264. [in Chinese]
- [12] Griffin, S.A., 1991, "Vane support system (VSS), a new generation wind tunnel model support system", AIAA91-0398, 1991
- [13] Lafourcade, P., Llibre, M., Reboulet, C., 2002, "Design of a parallel wire-driven manipulator for wind tunnels", *Proceedings of the Workshop on Fundamental Issues and Future Directions for Parallel Mechanisms and Manipulators*, Quebec City, Quebec, 2002, pp.187-194
- [14] <http://www.onera.fr/dcsd/sacso/index.php>.
- [15] Liu, X.W., Zheng, Y.Q., Lin, Q., 2004, "Overview of wire-driven parallel manipulators for aircraft wind tunnels", *Acta Aeronautica Et Astronautica Sinica*, Vol.25, No.4, pp.393-400. [in Chinese]
- [16] Zheng, Y.Q., 2004, "Research on key theoretical issues of wire-driven parallel kinematic manipulators and the application to wind tunnel support systems", PhD Dissertation, Quanzhou: Huaqiao University, 2004, pp.4-104. [in Chinese]
- [17] Zheng, Y.Q., Lin, Q., Liu, X.W., 2005, "Design methodology of wire-driven parallel support systems in the low speed wind tunnels and attitude control scheme of the scale model", *Acta Aeronautica Et Astronautica Sinica*, Vol.26, No.6, pp.774-778. [in Chinese]
- [18] Zheng, Y.Q., Lin, Q., Liu, X.W., 2007, "Initial test of a wire-driven parallel suspension system for low speed wind tunnels", In: *Proceedings of the 12th World Congress in Mechanism and Machine Science*, Besancon, France, June 17-21, 2007, Vol.5, pp.88-93
- [19] Lin, Q., Zheng, Y.Q., Liu, X.W., 2006, "Modeling and control of a wire-driven parallel support system with large attack angles in low speed wind tunnels", CD *Proceedings of 25th Congress of the International Council of the Aeronautical Sciences*, Hamburg, Germany, 3-8 September 2006
- [20] Zheng, Y.Q., 2006, "Feedback linearization control of a wire-driven parallel support system in wind tunnels", *Proceedings of Sixth International Conference on Intelligent System Design and Applications*, Jinan, Shandong, China, October 16-18, 2006

- [21] Liu, X.W., Q.Y., Agyemang, B.B., Zheng, Y.Q., Lin, Q., 2006, "Design of a wire-driven parallel suspension system for wind tunnel based virtual flight testing", Proceedings of the 7th International Conference on Frontiers of Design and Manufacturing, Guangzhou, China, June 19-22, 2006
- [22] Zheng, Y.Q., Lin, Q., Liu, X.W., 2006, "Kinematic Calibration of a Wire-Driven Parallel Support System in Wind Tunnels", China Mechanical Engineering, Vol.17, No.6, pp.551-554
- [23] Liang, B., Zheng, Y.Q., Lin, Q., 2007, "Attitude Control of the Scale Model of Wire-Driven Parallel Suspension Systems for Low-Speed Wind Tunnels", Forum on Key Technique of Large Aircraft and Academic Annual Meeting of Chinese Society of Aeronautics and Astronautics in 2007, Shenzhen, China, September 2-3, 2007
- [24] Hu, L., 2008, "Research on Wire-Driven Parallel Suspension Systems for Low-Speed Wind Tunnels", Master's Thesis, Quanzhou: Huaqiao University, October 2008. [in Chinese]
- [25] Lin, Q., Liang, B., Zheng, Y.Q., 2008, "Control Study on Model Attitude and Oscillation by Wire-Driven Parallel Manipulator Support System for Low-Speed Wind Tunnel", Journal of Experiments in Fluid Mechanics, Vol.22, No.3, pp.75-79. [in Chinese]
- [26] XIAO, Y. W. 2009, "Study on Model Aerodynamical Measurement with Wire-driven Parallel Suspension in Low-Speed Wind Tunnel", Master's Thesis, Xiamen: College of Physics and Mechanical & Electrical Engineering in Xiamen University, June 2009. (in Chinese)
- [27] <http://blog.sina.com.cn/AircraftEngineering>.

8. Appendix(the list of the publications of the authors in the field of wire-driven parallel suspension systems for low-speed wind tunnels)

- [1] LIU Xiongwei, ZHENG Yaqing, LIN Qi. Overview of Wire-driven Parallel Kinematic Manipulators for Aircraft Wind Tunnels(J). Acta Aeronautica ET Astronautica Sinica, 2004, 25(4) : 393-400(in Chinese, indexed by EI)
- [2] ZHENG Yaqing, LIN Qi, LIU Xiongwei. Design Methodology of Wire-Driven Parallel Support Systems in the Low Speed Wind Tunnels and Attitude Control Scheme of the Scale Model(J). Acta Aeronautica ET Astronautica Sinica, 2005, 26(6) : 774-778(in Chinese, indexed by EI)
- [3] LIU Xiongwei, ZHENG Yaqing, LIN Qi. Design of a Novel Wire-Driven Parallel Support System in a Low Speed Wind Tunnel and its Calibration Using Two Inclinometers. CD Proceedings of Lamdamap 7th International Conference, Cranfield Management Development Centre, Cranfield, Bedfordshire, UK, 27th-30th June, 2005
- [4] ZHENG Yaqing, LIN Qi, LIU Xiongwei. Kinematic Calibration of a Wire-Driven Parallel Support System in Wind Tunnels (J). China Mechanical Engineering, 2006, 17(6) : 551-554(in Chinese)
- [5] LIN Qi, ZHENG Yaqing, LIU Xiongwei. Modeling and Control of a Wire-Driven Parallel Support System with Large Attack Angles in Low Speed Wind Tunnels[C]. CD Proceedings of 25th Congress of the International Council of the Aeronautical Sciences, Hamburg, Germany, 3-8 September, 2006

- [6] LIU X W, Q Y, AGYEMANG B B, ZHENG Y Q, LIN Q. Design of a Wire-Driven Parallel Suspension System for Wind Tunnel Based Virtual Flight Testing[C]. Proceedings of the 7th International Conference on Frontiers of Design and Manufacturing, Guangzhou, China, June 19-22, 2006
- [7] ZHENG Y Q. Feedback Linearization Control of a Wire-Driven Parallel Support System in Wind Tunnels[C]. Proceedings of Sixth International Conference on Intelligent System Design and Applications, Jinan, Shandong, China, October 16-18, 2006(Indexed by ISTP)
- [8] LIANG Bin, ZHENG Yaqing, LIN Qi. Attitude Control of the Scale Model of Wire-Driven Parallel Suspension Systems for Low-Speed Wind Tunnels(C). Forum on Key Technique of Large Aircraft and Academic Annual Meeting of Chinese Society of Aeronautics and Astronautics in 2007 ,Shenzhen, China, September 2-3,2007 (in Chinese)
- [9] ZHENG Yaqing, HU Long. On a Wire-Driven Parallel Robot in Low-Speed Wind Tunnels Tests for the Aircrafts' Dynamic Derivatives with Large Attack Angles(J). Mechatronics Technology, 2007(S) : 236~239(in Chinese)
- [10] ZHENG Y Q, LIN Q, LIU X W. Initial Test of a Wire-Driven Parallel Suspension System for Low Speed Wind Tunnels[C]. 12th IFToMM World Congress, Besançon (France), June 18-21, 2007
- [11] HU Long, ZHENG Yaqing, LIN Qi, LIU Xiongwei. Dynamic Analysis of a Wire-Driven Parallel Manipulator for Low-Speed Wind Tunnels (J).Journal of Huaqiao University(Natural Science), 2008,29(2):184-189(in Chinese)
- [12] ZHENG Yaqing. Force-Measuring Experiment for the Scale Model of WDPSS in Low-Speed Wind Tunnel(J).Journal of Huaqiao University(Natural Science), 2009,30(2):119-122 (in Chinese)
- [13] LIN Qi, LIANG Bin, ZHENG Yaqing. Control Study on Model Attitude and Oscillation by Wire-driven Parallel Manipulator Support System for Low-Speed Wind Tunnel(J). Journal of Experiments in Fluid Mechanics,2008,22(3):75-79 (in Chinese, indexed by EI)
- [14] ZHENG Yaqing, LIN Qi, LIU Xiongwei, MITROUCHEV Peter. Wire-Driven Parallel Suspension Systems for Static and Dynamic Derivatives of the Aircraft in Low-Speed Wind Tunnels[C]. Proceedings of the 8th International Conference on Frontiers of Design and Manufacturing, Tianjin, China, Sept. 23-26, 2008
- [15] HU Long, ZHENG Yaqing. Static Stiffness Analysis of Wire-Driven Parallel Suspension Systems in Low-Speed Wind Tunnels[C].Proceedings of Annual Meeting of Fujian Province's Society of Mechanical Engineering in 2008, Xiamen, November 14th-16th,2008 (published in Journal of Xiamen University of Technology, 2008, (16) : 22-26) (in Chinese)
- [16] ZHENG Yaqing, LIN Qi, LIU Xiongwei, MITROUCHEV Peter. On Wire-Driven Parallel Suspension Systems for Static and Dynamic Derivatives of the Aircraft in Low-Speed Wind Tunnels[J]. Acta Aeronautica ET Astronautica Sinica, 2008 (in press, in Chinese, indexed by EI)
- [17] HU Long, ZHENG Yaqing. Analysis of Calculation Principle of Aerodynamic Derivatives of the Aircraft in Wire-Driven Parallel Suspension Systems for Low-Speed Wind Tunnels[J]. Journal of Huaqiao University (Natural Science), 2008 (in Chinese, Accepted)

- [18] ZHENG Yaqing, LIN Qi, LIU Xiongwei, MITROUCHEV Peter. Preliminary Step Towards Wire-Driven Parallel Suspension Systems for Static and Dynamic Derivatives of the Aircraft in Low-Speed Wind Tunnels[J]. The Journal of Engineering Research (Accepted)
- [19] HU Long. Research on Wire-Driven Parallel Suspension Systems for Low-Speed Wind Tunnels. Master's Thesis, Quanzhou: Huaqiao University, October 2008. [in Chinese]
- [20] LIANG B. System analysis and motion control of a 6-DOF wire-driven parallel manipulator with 3R3T type. Master's Thesis, Xiamen: College of Physics and Mechanical & Electrical Engineering in Xiamen University, June 2008. (in Chinese)
- [21] XIAO Y W. Study on Model Aerodynamical Measurement with Wire-driven Parallel Suspension in Low-Speed Wind Tunnel. Master's Thesis, Xiamen: College of Physics and Mechanical & Electrical Engineering in Xiamen University, June 2009. (in Chinese)
- [22] <http://blog.sina.com.cn/AircraftEngineering>