



IntechOpen

Parallel Manipulators, towards New Applications

Edited by Huapeng Wu



PARALLEL MANIPULATORS

TOWARDS NEW APPLICATIONS

EDITED BY
HUAPENG WU

Parallel Manipulators, towards New Applications

<http://dx.doi.org/10.5772/61>

Edited by Huapeng Wu

© The Editor(s) and the Author(s) 2008

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, 2008 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

Parallel Manipulators, towards New Applications

Edited by Huapeng Wu

p. cm.

ISBN 978-3-902613-40-0

eBook (PDF) ISBN 978-953-51-5828-8

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

4,000+

Open access books available

116,000+

International authors and editors

120M+

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



Preface

In recent years, parallel kinematics mechanisms have attracted a lot of attention from the academic and industrial communities due to potential applications not only as robot manipulators but also as machine tools. Generally, the criteria used to compare the performance of traditional serial robots and parallel robots are the workspace, the ratio between the payload and the robot mass, accuracy, and dynamic behaviour. In addition to the reduced coupling effect between joints, parallel robots bring the benefits of much higher payload-robot mass ratios, superior accuracy and greater stiffness; qualities which lead to better dynamic performance. The main drawback with parallel robots is the relatively small workspace.

A great deal of research on parallel robots has been carried out worldwide, and a large number of parallel mechanism systems have been built for various applications, such as remote handling, machine tools, medical robots, simulators, micro-robots, and humanoid robots.

This book opens a window to exceptional research and development work on parallel mechanisms contributed by authors from around the world. Through this window the reader can get a good view of current parallel robot research and applications.

The book consists of 23 chapters introducing both basic research and advanced developments. Topics covered include kinematics, dynamic analysis, accuracy, optimization design, modelling, simulation and control of parallel robots, and the development of parallel mechanisms for special applications. The new algorithms and methods presented by the contributors are very effective approaches to solving general problems in design and analysis of parallel robots.

The goal of the book is to present good examples of parallel kinematics mechanisms and thereby, we hope, provide useful information to readers interested in building parallel robots.

Editor

Huapeng Wu

Institute of Mechatronics and Virtual Engineering
Lappeenranta University of Technology
Finland

Contents

Preface	VII
1. Control of Cable Robots for Construction Applications Alan Lytle, Fred Proctor and Kamel Saidi	001
2. Dynamic Parameter Identification for Parallel Manipulators Vicente Mata, Nidal Farhat, Miguel Díaz-Rodríguez, Ángel Valera and Álvaro Page	021
3. Quantifying and Optimizing Failure Tolerance of a Class of Parallel Manipulators Chinmay S. Ukidve, John E. McInroy and Farhad Jafari	045
4. Dynamic Model of a 6-dof Parallel Manipulator Using the Generalized Momentum Approach António M. Lopes and Fernando Almeida	069
5. Redundant Actuation of Parallel Manipulators Andreas Müller	087
6. Wrench Capabilities of Planar Parallel Manipulators and their Effects Under Redundancy Flavio Firmani, Scott B. Nokleby, Ronald P. Podhorodeski and Alp Zibil	109
7. Robust, Fast and Accurate Solution of the Direct Position Analysis of Parallel Manipulators by Using Extra-Sensors Rocco Vertechy and Vincenzo Parenti-Castelli	133
8. Kinematic Modeling, Linearization and First-Order Error Analysis Andreas Pott and Manfred Hiller	155
9. Certified Solving and Synthesis on Modeling of the Kinematics. Problems of Gough-Type Parallel Manipulators with an Exact Algebraic Method Luc Rolland	175
10. Advanced Synthesis of the DELTA Parallel Robot for a Specified Workspace M.A. Laribi, L. Romdhane and S. Zegloul	207

11. Size-adapted Parallel and Hybrid Parallel Robots for Sensor Guided Micro Assembly	225
Kerstin Schöttler, Annika Raatz and Jürgen Hesselbach	
12. Dynamics of Hexapods with Fixed-Length Legs	245
Rosario Sinatra and Fengfeng Xi	
13. Cartesian Parallel Manipulator Modeling, Control and Simulation	269
Ayssam Elkady, Galal Elkobrosy, Sarwat Hanna and Tarek Sobh	
14. Optimal Design of Parallel Kinematics Machines with 2 Degrees of Freedom	295
Sergiu-Dan Stan, Vistrian Maties and Radu Balan	
15. The Analysis and Application of Parallel Manipulator for Active Reflector of FAST	321
Xiao-qiang Tang and Peng Huang	
16. A Reconfigurable Mobile Robots System Based on Parallel Mechanism	347
Wei Wang, Houxiang Zhang, Guanghua Zong and Zhicheng Deng	
17. Hybrid Parallel Robot for the Assembling of ITER	363
Huapeng Wu, Heikki Handroos and Pekka Pessi	
18. Architecture Design and Optimization of an On-the-Fly Reconfigurable Parallel Robot	379
Allan Daniel Finistauri, Fengfeng (Jeff) Xi and Brian Petz	
19. A Novel 4-DOF Parallel Manipulator H4	405
Jinbo Wu and Zhouping Yin	
20. Human Hand as a Parallel Manipulator	449
Vladimir M. Zatsiorsky and Mark L. Latash	
21. Mobility of Spatial Parallel Manipulators	467
Jing-Shan Zhao, Fulei Chu and Zhi-Jing Feng	
22. Feasible Human-Spine Motion Simulators Based on Parallel Manipulators	497
Si-Jun Zhu, Zhen Huang and Ming-Yang Zhao	

Control of Cable Robots for Construction Applications

Alan Lytle, Fred Proctor and Kamel Saidi
*National Institute of Standards and Technology
United States of America*

1. Introduction

The Construction Metrology and Automation Group at the National Institute of Standards and Technology (NIST) is conducting research to provide standards, methodologies, and performance metrics that will assist the development of advanced systems to automate construction tasks. This research includes crane automation, advanced site metrology systems, laser-based 3D imaging, calibrated camera networks, construction object identification and tracking, and sensor integration and process control from Building Information Models. The NIST RoboCrane has factored into much of this research both as a robotics test platform and a sensor/target positioning apparatus. This chapter provides a brief review of the RoboCrane platform, an explanation of control algorithms including the NIST GoMotion controller, and a discussion of crane task decomposition using the Four Dimensional/Real-time Control System approach.

1.1 The NIST RoboCrane

RoboCrane was first developed by the NIST Manufacturing Engineering Laboratory's (MEL) Intelligent Systems Division (ISD) in the late 1980s as part of a Defense Advanced Research Project Agency (DARPA) contract to stabilize crane loads (Albus et al., 1992). The basic RoboCrane is a parallel kinematic machine actuated through a cable support system. The suspended moveable platform is kinematically constrained by maintaining tension due to gravity in all six support cables. The support cables terminate in pairs at three vertices attached to an overhead support. This arrangement provides enhanced load stability over beyond traditional lift systems and improved control of the position and orientation (pose) of the load. The suspended moveable platform and the overhead support typically form two opposing equilateral triangles, and are often referred to as the "lower triangle" and "upper triangle," respectively.

The version of RoboCrane used in this research is the Tetrahedral Robotic Apparatus (TETRA). In the TETRA configuration, all winches, amplifiers, and motor controllers are located on the moveable platform as opposed to the support structure. The upper triangle only provides the three tie points for the cables, allowing the device to be retrofitted to existing overhead lift mechanisms. Although the TETRA configuration is presented in this chapter, the control algorithms and the Four Dimensional/Real-time Control System (4D/RCS), for 3D + time/Real-time Control System, task decomposition are adaptable to

many different crane configurations. The functional RoboCrane design can be extended and adapted for specialized applications including manufacturing, construction, hazardous waste remediation, aircraft paint stripping, and shipbuilding. Figure 1 depicts the RoboCrane TETRA configuration (a) and the representative work volume (b). Figure 2 shows additional retrofit configurations of the RoboCrane platform, and Figure 3 shows implementations for shipbuilding (Bostelman et al., 2002) and aircraft maintenance.

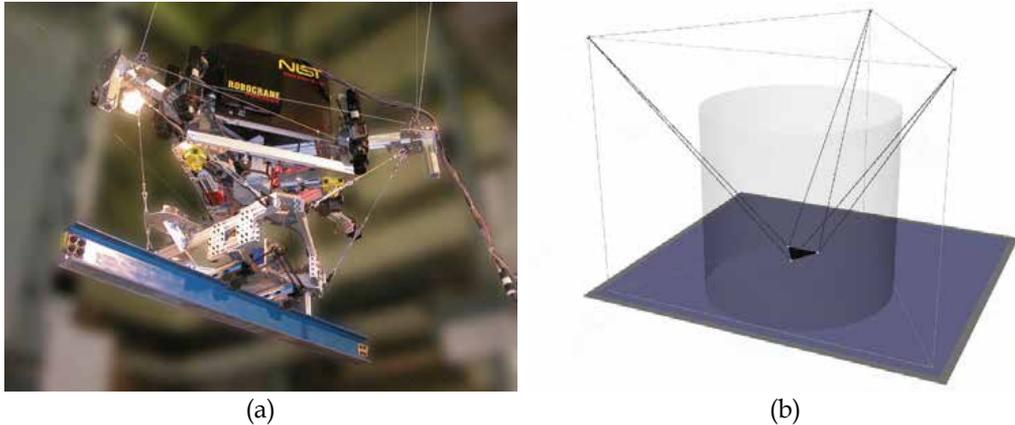


Fig. 1. RoboCrane - TETRA configuration (a); Rendering of the RoboCrane environment. The shaded cylinder represents the nominal work volume (b).

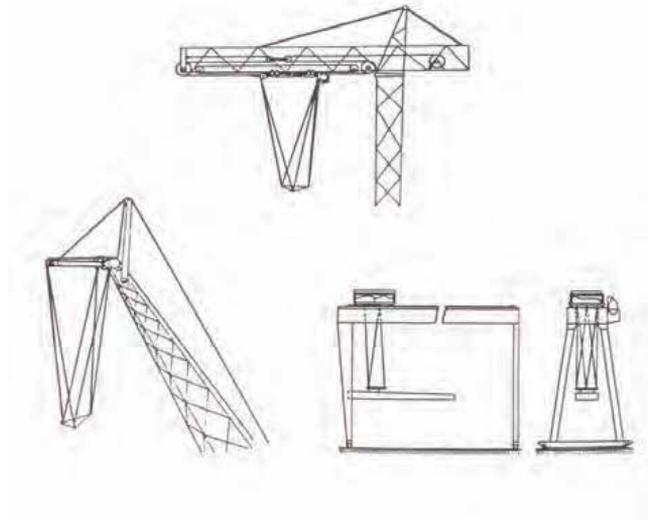


Fig. 2. Illustrations of RoboCrane in possible retrofitted configurations: Tower Crane (top), Boom Crane (lower left) and Gantry Bridge Crane (lower right).

1.2 Motivation for current research

Productivity gains in the U.S. construction sector have not kept pace with other industrial sectors such as manufacturing and transportation. These other industries have realized their productivity advances primarily through the integration of information, communication,

automation, and sensing technologies. The U.S. construction industry lags these other sectors in developing and adopting these critical, productivity-enhancing technologies. Leading industry groups, such as the Construction Industry Institute (CII), Construction Users Roundtable (CURT) and FIATECH, have identified the critical need for fully integrating and automating construction processes.

Robust field-automation on dynamic and cluttered construction sites will require advanced capabilities in construction equipment automation, site metrology, 3D imaging, construction object identification and tracking, data exchange, site status visualization, and design data integration for autonomous system behavior planning. The NIST Construction Metrology and Automation Group (CMAG) is conducting research to provide standards, methodologies, and performance metrics that will assist the development, integration, and evaluation of these technologies. Of particular interest are new technologies and capabilities for automated placement of construction components.



Fig. 3. The NIST Flying Carpet – a platform for ship access in drydocks (a) and the NIST Aircraft Maintenance Project (AMP) – a platform for aircraft access in hangars (b).

2. RoboCrane kinematics

From (Albus et al., 1992), given an initial condition where the overhead support and the suspended platforms are represented by parallel, equilateral triangles with centers aligned along the vertical axis Z, (see Figure 4), the positions of the upper triangle with vertices A, B, and C and lower triangle with vertices D, E, and F are expressed as

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} -b \\ -\frac{1}{3}b\sqrt{3} \\ -h \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} b \\ -\frac{1}{3}b\sqrt{3} \\ -h \end{bmatrix} & \mathbf{C} &= \begin{bmatrix} 0 \\ \frac{2}{3}b\sqrt{3} \\ -h \end{bmatrix} \\
 \mathbf{D} &= \begin{bmatrix} 0 \\ -\frac{2}{3}a\sqrt{3} \\ 0 \end{bmatrix} & \mathbf{E} &= \begin{bmatrix} a \\ \frac{1}{3}a\sqrt{3} \\ 0 \end{bmatrix} & \mathbf{F} &= \begin{bmatrix} -a \\ \frac{1}{3}a\sqrt{3} \\ 0 \end{bmatrix}
 \end{aligned} \tag{1}$$

With the positions of the vertices of triangles ABC and DEF as described in equations (1), when the lower platform is moved to a new position and orientation (D'E'F') through a translation of

$$\mathbf{U} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \quad (2)$$

and a rotation of

$$R_{yz}(\gamma, \theta, \phi) = R_z(\phi) \cdot R_x(\theta) \cdot R_y(\gamma) \quad (3)$$

the cable lengths can be expressed as

$$\begin{aligned} \mathbf{L}_1 &= \begin{bmatrix} -b + \frac{2}{3}aQ_{12}\sqrt{3} - u_x \\ -\frac{1}{3}b\sqrt{3} + \frac{2}{3}aQ_{22}\sqrt{3} - u_y \\ -h + \frac{2}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} & \mathbf{L}_2 &= \begin{bmatrix} -b + \frac{2}{3}aQ_{12}\sqrt{3} - u_x \\ -\frac{1}{3}b\sqrt{3} + \frac{2}{3}aQ_{22}\sqrt{3} - u_y \\ -h + \frac{2}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} \\ \mathbf{L}_3 &= \begin{bmatrix} b - aQ_{11} - \frac{1}{3}aQ_{12}\sqrt{3} - u_x \\ -\frac{1}{3}b\sqrt{3} - aQ_{21} - \frac{1}{3}aQ_{22}\sqrt{3} - u_y \\ -h - aQ_{31} - \frac{1}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} & \mathbf{L}_4 &= \begin{bmatrix} -aQ_{11} - \frac{1}{3}aQ_{12}\sqrt{3} - u_x \\ \frac{2}{3}b\sqrt{3} - aQ_{21} - \frac{1}{3}aQ_{22}\sqrt{3} - u_y \\ -h - aQ_{31} - \frac{1}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} \\ \mathbf{L}_5 &= \begin{bmatrix} aQ_{11} - \frac{1}{3}aQ_{12}\sqrt{3} - u_x \\ \frac{2}{3}b\sqrt{3} + aQ_{21} - \frac{1}{3}aQ_{22}\sqrt{3} - u_y \\ -h + aQ_{31} - \frac{1}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} & \mathbf{L}_6 &= \begin{bmatrix} -b + aQ_{11} - \frac{1}{3}aQ_{12}\sqrt{3} - u_x \\ -\frac{1}{3}b\sqrt{3} + aQ_{21} - \frac{1}{3}aQ_{22}\sqrt{3} - u_y \\ -h + aQ_{31} - \frac{1}{3}aQ_{32}\sqrt{3} - u_z \end{bmatrix} \end{aligned} \quad (4)$$

where

$$\begin{aligned} \mathbf{L}_1 &= \mathbf{A} - \mathbf{D}' & \mathbf{L}_2 &= \mathbf{B} - \mathbf{D}' & \mathbf{L}_3 &= \mathbf{B} - \mathbf{E}' \\ \mathbf{L}_4 &= \mathbf{C} - \mathbf{E}' & \mathbf{L}_5 &= \mathbf{C} - \mathbf{F}' & \mathbf{L}_6 &= \mathbf{A} - \mathbf{F}' \end{aligned} \quad (5)$$

and Q_{ij} represents an element in the following rotation matrix:

$$\mathbf{Q} = \begin{bmatrix} \cos(\gamma)\cos(\phi) - \sin(\gamma)\sin(\theta)\sin(\phi) & -\cos(\theta)\sin(\phi) & \sin(\gamma)\cos(\phi) + \cos(\gamma)\sin(\theta)\sin(\phi) \\ \cos(\gamma)\sin(\phi) - \sin(\gamma)\sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) & \sin(\gamma)\sin(\phi) - \cos(\gamma)\sin(\theta)\cos(\phi) \\ -\sin(\gamma)\cos(\theta) & \sin(\theta) & \cos(\gamma)\cos(\theta) \end{bmatrix} \quad (6)$$

Therefore, for any new desired pose of the moving platform described by equations (2) and (3), the required cable lengths to achieve that pose can be calculated by the inverse kinematic equations shown in equations (4).

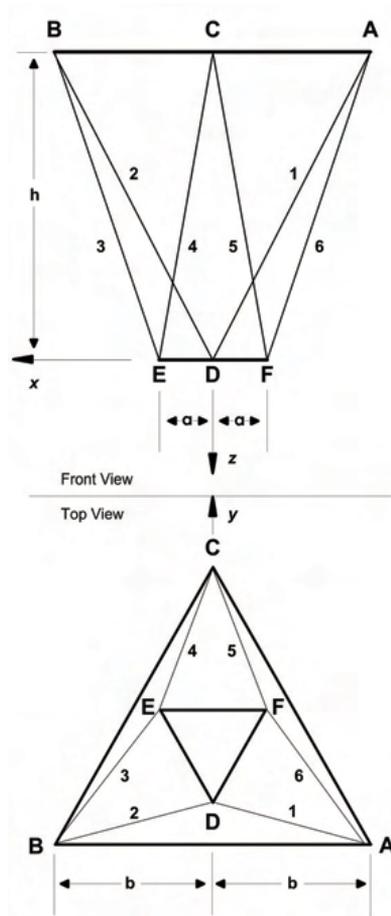


Fig. 4. Graphical representation of the RoboCrane cable support structure.

3. Measuring RoboCrane pose

The controller's estimate of the actual pose of RoboCrane differs from the actual pose due to several sources of error. Position feedback is provided through motor encoders that measure rotational position. Cable length is computed by multiplying the rotational position by the winch drum radius, with a suitable scale factor and offset.

However, the winch drum radius is not constant, but varies depending on the amount of cable that has already been wrapped around the drum, increasing its radius. It is possible to keep track of this and change the radius continually, by building a table that relates motor rotational position with effective radius.

Another source of error is that the cable length is affected by sag due to gravity. This sag depends on the pose of the platform and its load. Compensation can be achieved using an iterative process that begins with the nominal cable lengths, computes the platform pose using the forward kinematics equations, and determines the tensions on each of the cables using the transpose of the Jacobian matrix and the weight of the platform. The tensions can be used to generate the actual catenary curve of the cable, taking its nominal length as the

length of the hanging catenary curve. This process is repeated iteratively, with the nominal cable length as the fixed arc length of the catenary, and the chord between its endpoints as the continually revised length used by the forward kinematics.

Calibration errors in the mounting points of the ends of the cables further contribute to pose error. In practice these are not fixed points, but vary as the angles of the cables change the contact point to the pulleys or eye bolts that affix the ends. Even if these contact points were constant, their actual locations can be difficult to measure with precision, given their large displacement over a typical work volume.

Given these many sources of error, it is desirable to be able to measure the pose of the platform directly. There are many commercial systems for this purpose. An initial approach to external measurement implemented on RoboCrane uses a laser-based, large-scale, site measurement system (SMS).

3.1 The site measurement system (SMS)

A laser-based site measurement system (SMS) is used to track RoboCrane's pose and to measure object locations within the work volume. The SMS uses stationary, active-beacon laser transmitters and mobile receivers to provide millimeter-level position data at an update rate of approximately 20 Hz. This technology was chosen based upon a combination of factors including indoor/outdoor operation, accuracy, update-rate, and support for multiple receivers.

Each SMS transmitter emits two rotating, fanned laser beams and a timing pulse. Elevation is calculated from the time difference between fanned beam strikes. Azimuth is referenced from the timing pulse. The field of view of each transmitter is approximately 290° in azimuth and $\pm 30^\circ$ in elevation/declination.

Similar to GPS, the SMS does not restrict the number of receivers. Line-of-sight to at least two transmitters must be maintained by each receiver in order to calculate that receiver's position. The optical receivers each track up to four transmitters and wirelessly transmit timing information to a base computer for position calculation.

For tracking RoboCrane's pose, four laser transmitters are positioned and calibrated on the work volume perimeter, and three SMS receivers are mounted on RoboCrane near the vertices of the lower triangle. The receiver locations are registered to the manipulator during an initial setup process in the local SMS coordinate frame. A transmitter and an optical receiver are shown in Figure 5. The SMS receivers mounted on RoboCrane are shown in Figure 6.



Fig. 5. An SMS laser transmitter (a) and an SMS optical receiver (b).

The drawback of these systems is the added cost, and the need to maintain lines-of-sight between the platform and transmitters, potentially interfering with intended use. The benefits of accurate pose measurement are often significant enough to warrant their use. In the first implementation of the SMS to track RoboCrane, position estimates were obtained at several stopping points during RoboCrane's trajectory, and these estimates were used as coarse correction factors for the encoder positions. Current work is focused on a dynamic tracking approach to eliminate the need for stopping points.



Fig. 6. The SMS on RoboCrane showing a close-up view of one of the three receivers.

3.2 Dynamic pose measurement

A commanded pose will generally result in a different actual pose due to various sources of system error such as those discussed previously. This relationship is depicted as

$$\mathbf{N} \rightarrow \mathbf{X} \rightarrow \mathbf{A} \quad (7)$$

or, in matrix form,

$$\mathbf{NX} = \mathbf{A} \quad (8)$$

where \mathbf{N} is the commanded pose, \mathbf{X} is the perturbation that includes all the sources of error, and \mathbf{A} is the actual pose that results. The effects of \mathbf{X} can be cancelled by commanding an adjusted pose, \mathbf{N}^* , where

$$\mathbf{N}^* = \mathbf{NX}^{-1} \quad (9)$$

Using the adjusted pose allows us to achieve the original desired pose since

$$\mathbf{N}^* \mathbf{X} = \mathbf{N} \quad (10)$$

In general, most of the sources of error are unknown and variable, so computing \mathbf{X}^{-1} a priori is not feasible. However, \mathbf{X}^{-1} can be estimated by comparing a previously commanded

adjusted pose, \mathbf{N}^* , with the resulting actual pose, \mathbf{A}^* , as measured by the SMS. For time step, (i-1)

$$\mathbf{N}_{i-1}^* \mathbf{X}_{i-1} = \mathbf{A}_{i-1}^* \quad (11)$$

And the inverse of \mathbf{X}_{i-1} can be calculated as

$$(\mathbf{X}_{i-1})^{-1} = (\mathbf{A}_{i-1}^*)^{-1} \mathbf{N}_{i-1}^* \quad (12)$$

For the current time step, (i), the commanded adjusted pose can be calculated as

$$\mathbf{N}_i^* = \mathbf{N}_i (\mathbf{X}_{i-1})^{-1} \quad (13)$$

where \mathbf{N}_i is the desired pose for the current time step and \mathbf{X}_{i-1} is the perturbation from the previous time step. Therefore,

$$\mathbf{N}_i^* \mathbf{X}_i \approx \mathbf{N}_i \quad (14)$$

If the platform is moving, then the cancellation is not perfect, since we are trying to cancel this time step's unknown perturbation transform with the inverse from the previous time step, which will be slightly different. If the platform is stationary, the two converge and the cancellation becomes perfect.

Platform motion has a more pronounced practical effect due to measurement latency for \mathbf{A} . When computing \mathbf{X}^{-1} , it is important that the \mathbf{N} and \mathbf{A} poses are synchronized. If the measured \mathbf{A} pose lags the nominal \mathbf{N} pose, then the compensation will have the effect of *leading* the motion. When speed slows, this leading will become an overshoot, and the platform will oscillate.

In the presence of measurement latency, one solution is to only compute the compensating transform \mathbf{X}^{-1} when the platform is stationary. With this method, the platform is moved into an area of interest, held stationary for at least the latency period, and \mathbf{X}^{-1} is computed. From that point, iteration is suppressed, and the compensating transform is constant. As the platform moves away from the compensation point, its accuracy diminishes.

If the latency is constant and can be measured, a solution is to keep a time history of nominal poses and their associated inverse transforms, and look back into this history by the amount of latency to associate a pair \mathbf{N} and \mathbf{X}^{-1} to the latent \mathbf{A} measurement. If the measurements can be timestamped, then the same technique can be supplemented with timestamps to make the association. This technique can be used in the presence of variable measurement latency.

Controller latency also has an effect on the accuracy of the compensating transform. Figure 7 shows the magnitude of the translation portion of the compensating transform during tests with four different trajectory cycle times. In each test, the platform speed varied from 1 cm/s to 10 cm/s. These tests were done with a simulated measurement system that simulates actual position from the servo position run through the forward kinematics. In this case, the compensating transform should be small, and in fact it goes to zero as the

motion pauses between each speed setting. It is apparent from these figures that as the platform speed increases, the magnitude of the compensating transform increases, as is expected from servo lag. It is also apparent that as cycle time increases, so does the magnitude of the transform. This is due to the uncertainty between when nominal position is registered by the controller, and when it is read out some fraction of a period later.

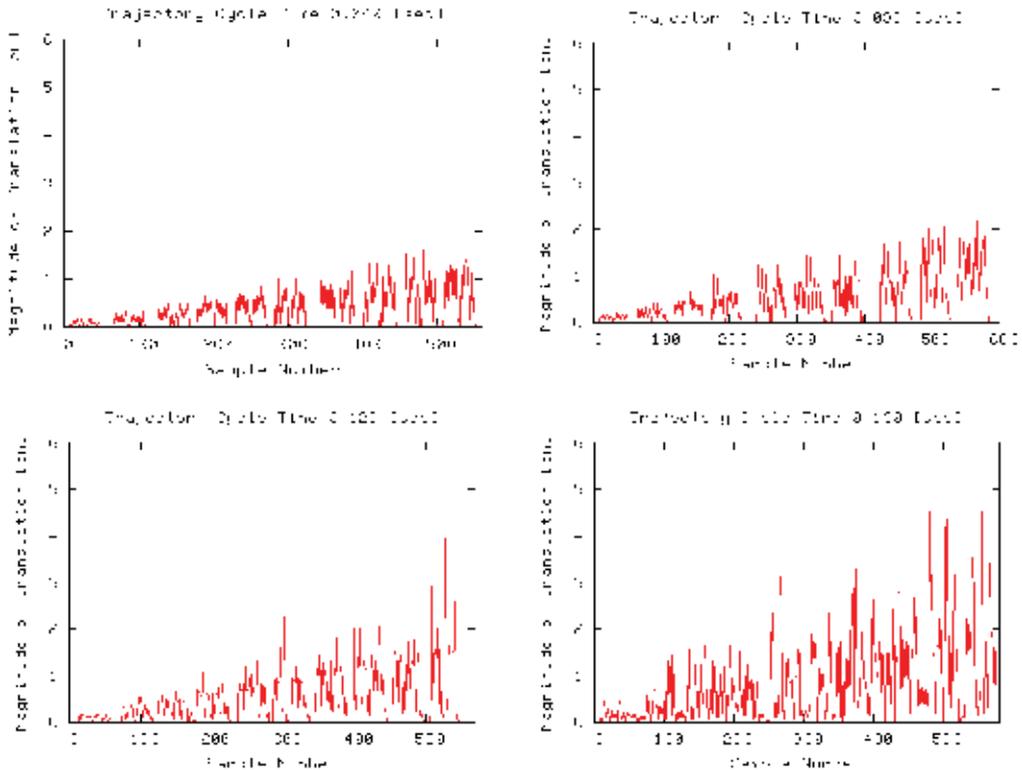


Fig. 7. Compensating transform magnitude (translation only) for four different trajectory cycle times. As the trajectory cycle time increases, the magnitude increases, and becomes more noisy as a result of the increased uncertainty in the latency between control output and compensation. (Note: Figures intended as qualitative examples of cycle time effects.)

Whenever a new X^{-1} transform is written to the controller, it has the potential to cause a jump in motion. To prevent this, transforms are “walked in” according to speed and acceleration limits. A large change in the transform will appear as a relatively quick but controlled move to the new, more accurate position. The effect of compensation is illustrated in Figure 8. The square path in the lower left of the figure is the uncompensated path, which is offset and slightly skewed from the ideal path due to kinematic miscalibration. Shortly after the second pass around the square path, compensation was turned on and its effects walked in over several seconds. This interval appears as the two line segments connecting the square paths. The square path in the upper right is the compensated path, whose adherence to the nominal edges at 0 cm and 10 cm is quite good.

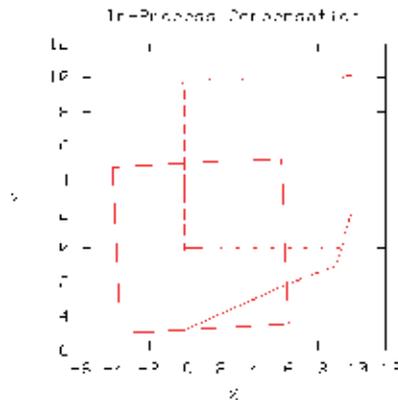


Fig. 8. Effect of in-process compensation The lower left square path is uncompensated and differs due to kinematic miscalibration. The upper right path is compensated. The connecting path results applying the compensation over time to avoid impulsive jumps.

When compensation is turned off, the last compensating transform remains in use. As the platform moves away from the point at which this transform was calculated, the compensation becomes less accurate. This is shown in Figure 9.

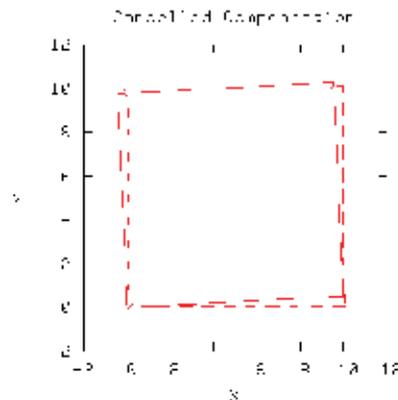


Fig. 9. Trajectory drift after cancelling in-process compensation. The correction was made at location (0,0), and no further updates were performed.

4. RoboCrane control

4.1 GoMotion controller description

The RoboCrane controller is a two-level hierarchy. The bottom level is servo control, which takes position setpoints for the cable lengths at a period of 1 millisecond, and runs a proportional-integral-derivative (PID) controller using feedback from encoders mounted on the motors to generate drive signals. The top level is trajectory planning, which takes desired goal poses and plans smooth Cartesian motion along a linear path, taking into account speed, acceleration and jerk constraints. The trajectory planner executes at a period of 10 milliseconds, calculating intermediate poses that are run through the inverse kinematic

equations to generate cable lengths sent to the servo controllers. Joint mode control is also possible, with goals specified in terms of desired cable lengths. The inverse kinematics are not needed in this case.

Servo control is divided among six similar modules, each running PID control with extensions that handle velocity and acceleration feedforward terms, output biasing, deadband and saturation detection for anti-windup of integral gain. A software application programming interface (API) localizes how the servo modules connect to specific hardware such as commercial input/output boards for encoder feedback and digital-to-analog conversion, open-loop stepper motors or distributed input/output. The servo modules run periodically at 10 times the period of the trajectory planner. Interpolation between setpoints sent by the trajectory planner is done using either linear, cubic or quintic polynomial interpolation of the setpoint over time, depending on application needs.

Trajectory planning is done following *S-curve* velocity profiling with specified velocity, acceleration and jerk. S-curve profiling has the advantage of bounding jerk, when compared with trapezoidal velocity profiling with abrupt changes in acceleration. S-curve profiling has seven motion phases, as shown in Figure 10.

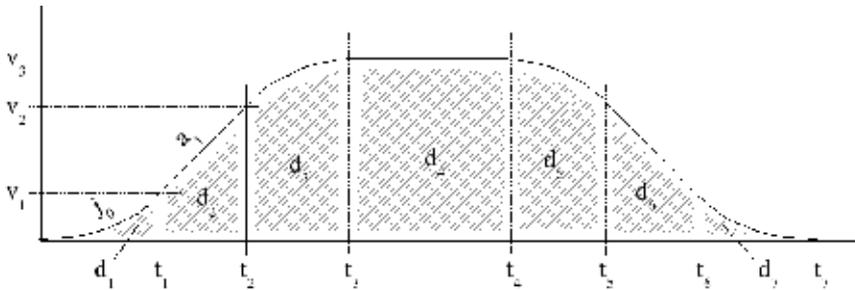


Fig. 10. S-curve velocity profile.

Here, v_3 , a_1 and j_0 are the specific maximum velocity, acceleration and jerk, respectively. At each trajectory time step, the distance increment is computed as the area under the S-curve for that time interval.

In joint position control mode (individual cable actuation), trajectory planning is done for each cable independently. Given a desired target cable length, the S-curve profile is computed and distances are computed each trajectory period. These distances are sent to the servo module for that joint for interpolation and tracking. Coordinated joint position control is possible, in which a set of six target cable lengths comprises the goal. Six trajectory profiles are computed, and five of the six are scaled so that their final arrival time matches the time of the longest move.

In Cartesian position control mode, motion control is split into translation and rotation vectors. The translation vector is a three-element vector with X, Y and Z components pointing to the target location, with associated velocity, acceleration and jerk along the path. The rotation vector is a three-element vector about which the overall rotation from the current orientation to the target orientation takes place. The magnitude of this vector is the amount of rotation. Angular velocity, acceleration and jerk are used to generate a profile for this portion of the move. One of the two profiles is scaled to match the time of the longer of the two so that the translation and rotation arrive at the same time. At each trajectory cycle,

the translation and rotation are computed, run through the inverse kinematics equations, and sent as a set of target cable lengths for interpolation and tracking by the servo modules. Motion along circular arcs is also supported. Rotational motion is planned as before. Translational motion is planned along the arc, where the distance under the S-curve profile is the distance along the arc. Aside from this geometric distinction, circular motion is the same as linear motion.

4.2 Initialization

When the controller begins executing, it assumes that the cable length measurements are uncalibrated. Cable length limits are invalid, as is any notion of the Cartesian pose of the platform or its limits. The controller allows individual cables to be moved independently, but inhibits Cartesian motion and cable length limit checking. Before any of these can take place, the platform must be “homed” to establish the offset between the initial arbitrary measurement of cable length (typically zero) and its true length.

In systems that lack a way to absolutely measure either cable lengths or Cartesian pose at startup, a homing procedure is used. There are several variations in this method. In one, fiducial marks are made on each cable, which when aligned with an associated mark on the platform denote that the cable is at a known length. The operator must manually jog each cable to align the marks, and indicate that the home condition has been met. The controller then computes an offset that is added to the raw feedback from the motor encoder to yield the known length value.

Another homing technique is to bring the platform to a known Cartesian location, such as level and oriented properly atop a mark on the floor. This requires manually moving the platform by adjusting cable lengths, which is unintuitive. In practice, the operator moves each cable so that the platform is relatively close to the home location, and falsely indicates that the cables are homed. Cartesian motion is then enabled, and the operator moves in Cartesian space for the final alignment. During this falsely-homed period, the platform motion will be skewed, but is usually close enough for intuitive positioning.

Homing is a time-consuming manual procedure. If the platform's Cartesian pose can be measured directly, such as with the SMS, then homing is not necessary. In this case, the controller is provided with the actual Cartesian position, which it runs through the inverse kinematics to get the cable lengths. The difference between these computed cable lengths and the uncalibrated lengths from the motor encoders is the offset used to calibrate the feedback.

4.3 Control modes

The RoboCrane controller supports various control modes. Teleoperation allows an operator to drive the platform directly, using a keyboard, mouse or joystick. Automatic control allows the execution of scripted trajectories.

Teleoperated Control: In teleoperated control, the operator uses a convenient input device, such as a keyboard, mouse or joystick, to move the platform directly. Typically a joystick is used, since it is most intuitive. This can be performed in either joint (i.e., cable lengths) or Cartesian space. With cable lengths, the operator selects a cable, and shortens or lengthens the cable according to the deflection of the joystick. If the controller has been homed, the Cartesian position is continually updated using the forward kinematics. Cable length motion is typically used only when homing the platform, since it results in unintuitive platform motion.

In Cartesian space, the operator uses the joystick to drive the platform in any of the X, Y and Z directions, or to rotate about these directions. The controller supports two reference frames: the world frame, with coordinates affixed to the unmoving ground; and the platform (or tool) frame, with coordinates affixed to the moving platform. World mode is typically used to position the platform near an area of interest, or to drive it along features in the world, such as the floor or walls. Tool mode is used to position the platform by driving it along axes aligned with grippers or tooling, so that approaches and departures can be made along arbitrary directions. The controller supports the definition of arbitrary tool coordinate systems, so that one tool can be dropped off, another picked up, and motion with respect to the new tool axes can be accomplished.

In world mode, Cartesian speeds from the joystick are converted into cable speeds using the inverse Jacobian. Given a desired Cartesian velocity of RoboCrane, \mathbf{V} , and using the inverse Jacobian¹ matrix, \mathbf{J}^{-1} , the cable speed vector, $\dot{\mathbf{L}}$, can be calculated as

$$\dot{\mathbf{L}} = \mathbf{J}^{-1}\mathbf{V} \quad (15)$$

where $\dot{\mathbf{L}}$ is the 6x1 cable speed matrix, \mathbf{J}^{-1} is the 6x6 inverse Jacobian transform matrix, and \mathbf{V} is the 6x1 Cartesian velocity vector (Tsai, 1999). The calculated cable speeds are transformed into winch motor rotation rates that are sent to the winches. Each motor encoder keeps track of the number of motor shaft revolutions and that number is directly related to cable length. The six cable lengths are then used to calculate a new Jacobian matrix, which is used the next time velocity commands are sent.

Since the inverse Jacobian matrix is calculated based on the instantaneous Cartesian pose of RoboCrane, the initial pose of RoboCrane must be known. This initial pose can be calculated by directly measuring the cable lengths and performing the forward kinematic calculations, or by placing RoboCrane in a predefined home pose at the beginning of each teleoperation session and initializing the cable lengths to preset values.

Speed changes are clamped to lie within acceleration limits, so that abrupt changes in joystick position do not impart abrupt changes in motor speed. Cartesian position and orientation limits are applied, so that attempts to drive the platform outside a limit will be inhibited.

Automatic Control: With automatic control, motions in either cable or Cartesian space can be scripted in programs. These programs can be written by hand, or generated by off-line programming systems that can automate the generation of complex tasks throughout a large work volume. This is accomplished through a third level in the hierarchy, the Job Cell level. This level interfaces to the motion controller using the same interface as the teleoperation application, but sending discrete moves instead of teleoperation speeds.

There are two basic modes of automatic control, either in cable space or in Cartesian space. Cable space motions are less common, and would be used to drive individual cables during maintenance activities. Cartesian space motions are primarily used in applications. As with

¹ The Jacobian transform (or simply Jacobian), \mathbf{J} relates the velocities of the joints of a manipulator to the velocities (translational and rotational) of its end-effector, $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, where $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}$ are the velocity vectors of the joints and end-effector, respectively (Tsai, 1999).

Cartesian teleoperation, programmed Cartesian moves can be done either with respect to the world frame or the tool frame. A representative program is

```
# rotate to 30-degree yaw at 1, -2, 3
movew 1 -2.0 3.0 0 0 30.0
# move along the tool's Y axis 10 cm
movet 0 0.1 0 0 0 0
```

World motions are absolute (although they can be incremental), while tool motions are strictly incremental, since the tool origin moves along with the tool.

5. High level control

5.1 4D/RCS overview

The NIST RCS (Albus, 1992) methodology describes how to build control systems using a hierarchy of cyclically executing control modules. In (Bostelman et al., 1996), RCS was applied to a RoboCrane implementation. At the lowest level of the hierarchy, each control module processes input from sensors, builds a world model, and generates outputs to actuators in response to commands from its supervisory control module. These functional components of a control module are termed sensory processing (SP), world modeling (WM) and behavior generation (BG), respectively. The servo control of a motor is a common example of a control module at the lowest level. Here, the sensor may be a motor shaft position encoder, the actuator is the motor shaft, the command is a desired setpoint for the shaft position, and the behavior may be the execution of a simple PID control algorithm. The SP function may simply be reading and scaling input from the encoder device, and the WM function may be maintaining a filtered estimate of the shaft position. Typical cycle times for such control modules are on the order of a millisecond.

One or more of these lowest-level control modules may be subordinate to a control module at the next level up in the hierarchy, termed the supervisor. In our example, the SP function at this level may simply provide each motor shaft position to the WM function, which would compute the overall position and orientation of the device's controlled point, perhaps the tool on a robot. The BG function may smoothly transform goal points to motor trajectories based on speed, acceleration and jerk. Here, goal points may arrive at variable intervals from the higher-level supervisor, one that may be reading them from a program file. Cycle times increase by about an order of magnitude for control modules that are one level higher in the hierarchy. For this trajectory planner, the cycle time would be about 10 ms.

A full RCS hierarchy would include additional lower-level control modules for individual tools, and control modules at higher levels of the hierarchy may coordinate the actions of many robots and auxiliary equipment. RCS has found its richest application in the area of mobile robotics. Here the SP functions include not just motors but cameras, 3D imaging systems (e.g. laser scanner), GPS and other navigation sensors. WM functions build maps of various resolutions and maintain symbolic representations of the world. BG functions reason on the symbolic representations, planning optimal paths around known features and reacting to sensed obstacles.

An RCS design differs from functional design or object-oriented design in that it begins with a task analysis of the system to be controlled. Here the designer identifies the tasks to be

performed at the top level, and then breaks each task down into subtasks that are performed by the subordinates. Usually the designer does not have complete freedom to determine the task breakdown, as some of the components that make up the system may have been reused from prior projects. In this case, the tasks must be expressed in terms of the available subtasks. Task analyses are helped enormously by considering scenarios that include system startup, shutdown, normal use and changes between various modes of operation. Often these scenarios bring to light the need for tasks that are not apparent from the original conception of the system.

An example of a comprehensive task analysis for the design of an automatic road vehicle controller can be found in (Barbera et al., 2004). The designers considered hundreds of scenarios listed in a manual of military driving, including lane changes, passing and intersection rules. What is made obvious by this analysis is that the top- and bottom-level tasks are relatively simple, while the tasks in the middle are the most complex. Other examples of task analyses for unmanned vehicle systems can be found in the latest version of RCS (known as 4D/RCS) (Albus et al., 2002).

Implementation of RCS control modules is done conceptually using state tables, which can then be programmed in any general-purpose computer language using conditionals or switch statements. The NIST RCS Library documents the software tools available for programming in C++ or Java. A detailed handbook (Gazi, 2001) covers the entire RCS analysis, design and programming using several examples and the RCS Library tools.

5.2 Crane task decomposition

Designing a new RCS-based controller for RoboCrane began by first identifying the requirements of the controller. The overall goal of the RoboCrane controller was defined as follows: to plan and execute tasks required for automated construction-material handling and/or building construction.

Controller Requirements: In order to accomplish its goal the RoboCrane controller needed to provide the following:

- Autonomous, semi-automated, and teleoperated modes of operation
- RoboCrane tool-point (i.e., platform) position and velocity control modes
- RoboCrane tool-point motion in joint, Cartesian, as well as other user-definable coordinate systems
- Cross-platform code portability (but still dependent on the real-time operating system)
- Adaptability to other robot/crane hardware
- Sensor-based collision avoidance

System Scope: Although the motivation for developing a controller was to be able to use it to control various cable-driven robots and to accomplish various tasks, the initial scope of the controller was limited to the following:

- Smooth and stable motion of the NIST RoboCrane
- Perform a steel beam pick and place task
- Construct a structure whose shape is limited by RoboCrane's current range of motion
- Connect the beam to the holder using drop-in connectors
- Carry beams whose size falls within RoboCrane's current load-carrying capabilities
- Communicate to RoboCrane using the current field bus architecture
- Operate under a real-time Linux operating system

- Use the built-in incremental winch motor encoders as well as the laser-based positioning system to determine RoboCrane's pose, but include the ability to add other sensors for pose determination in the future
- Acquire the steel beam and holder poses using the current laser-based positioning system

Task Decomposition: The next step in the RCS controller design process is to conduct a task decomposition of the controller's overall goal. RoboCrane's overall goal was divided into several subtasks, which were consequently also broken down into smaller tasks. This process continued until the lowest level tasks involved sending commands to the RoboCrane hardware (e.g., setting motor voltages). This is the lowest level of control that the controller can provide.

Figure 11 shows a sample task tree diagram resulting from the task decomposition process. In this figure the physical task of picking and placing a steel beam (as part of a steel erection sequence) is decomposed into 3 levels of subtasks. In keeping with the RCS architecture, each sublevel is responsible for planning and executing a smaller portion of the overall pick-and-place task. The lowest level is responsible for maintaining a commanded joint (or motor) velocity (or position). The next level up is responsible for generating and executing a series of n waypoints (i.e., positions and orientations in time) for the RoboCrane platform. The next higher level generates and executes the necessary commands to accomplish a segment of the pick-and-place operation. Finally, the highest level in Figure 11 is responsible for coordinating the execution of the segments that make up the overall pick-and-place task. This highest level also receives commands from higher levels (not shown in Figure 11) which coordinate the pick-and-place task with other tasks such as attaching a beam to a structure, picking and placing a column, and etc.

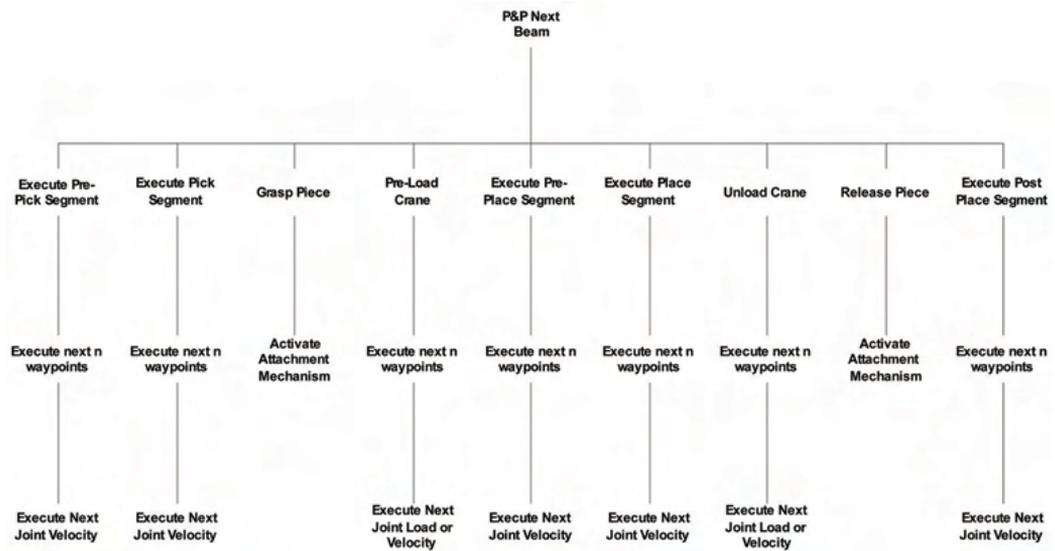


Fig. 11. Task tree diagram for the pick-and-place next beam task.

In addition to the physical tasks represented in the task tree diagram of Figure 11, other non-physical tasks are required in order to accomplish a pick-and-place operation. These

include tasks such as detecting obstacles, calculating collision free paths, etc. These tasks were also captured and broken down into 3 levels of subtasks, but are not included in Figure 11

State Tables: Following the task decomposition process the commands going into and out of each task, that are represented in the task tree diagram of Figure 11, are listed in a state table format. A state table (or state transition table) describes all possible input and output states (and actions) of a finite state machine. Table 1 shows a state table for the pick and place next beam task. The command that starts the execution of this task has the same name as the task itself and is also the title of the state table. The state table columns (from left to right) represent the input state numbers, the conditions that must be met to change the state, the output state numbers, and the output commands that are sent to lower level tasks, respectively.

When the pick and place next beam command is issued by a higher level task, the controller examines the state table shown in Table 1. The initial state of the pick and place next beam task is S0 and the first condition that is checked is whether the received command is new. If it is a new command, the state of the task is changed to S1 and the status of the task is changed to indicate that it is executing.

Pick and Place Next Beam	
S0 New Command	S1 Hold - Status=Executing
S1 Conditions Good to Move to Pre-Pick Pose	S2 Move to Pre-Pick Pose
S1 Timed out	S0 Hold - Status=Error
S2 Conditions Good to Move to Pick Pose	S3 Move to Pick Pose
S3 Conditions Good to Grasp	S4 Grasp Beam
S4 Conditions Good to Pre-Load Crane	S5 Pre-Load Crane
S5 Conditions Good to Move to Pre-Place Pose	S6 Move to Pre-Place Pose
S6 Conditions Good to Move to Place Pose	S7 Move to Place Pose
S7 Conditions Good to Unload Crane	S8 Unload Crane
S8 Conditions Good to Release	S9 Release Beam
S9 Conditions Good to Move to Post Place Pose	S10 Move to Post Place Pose
S10 At Post Place Pose	S0 Hold - Status=Done

Table 1. State table for the pick and place next beam task.

The next time the above state table is checked (i.e., during the next execution cycle of its corresponding control module) the new state of the task is S1, and the conditions that must be met are whether it is acceptable to move RoboCrane to the beam's pre-pick pose, or whether enough time has elapsed that something must be wrong. There may be one or more sub-conditions that must be satisfied in order to determine whether it is acceptable to proceed, but these can be aggregated into one description in the state table. If the conditions are met, the state of the task is changed to S2 and the command to move to the pre-pick pose is sent to a lower-level task. If time has expired, the state of the task is changed to S0 and an error is reported. Each lower level task that receives an output command reports its status back to the higher level task that issued the command until it finishes executing or

encounters an error. This process continues until all of the commands in the state table have been executed, at which point the pick and place next beam task is considered completed and the state of the table is reset to S0. For brevity, only a single timeout condition is shown in Table 1. In practice, numerous checks of this sort are made throughout the state table. Once the state tables for all of the tasks identified through the task decomposition process are completed they are organized into control modules as described next and implemented in software following the RCS guidelines.

Control Modules: As indicated in the prior RCS description, the commands in the task tree diagram of Figure 11 are organized into multiple levels. Each level's tasks may be grouped together into one or more modules responsible for coordinating and executing the tasks within it. Some of the critical modules (such as the servo algorithms) run as real-time processes within the operating system, while other less critical modules (such as long term path planning) run as non-deterministic processes.

Figure 12 shows the control architecture for the RoboCrane controller. The four levels above the software/hardware demarcation line in Figure 12 correspond to the four levels of Figure 11. The tasks have been grouped into the control modules shown. For example, the bottom level tasks of Figure 11 are grouped into the six "Servo" modules in Figure 12.

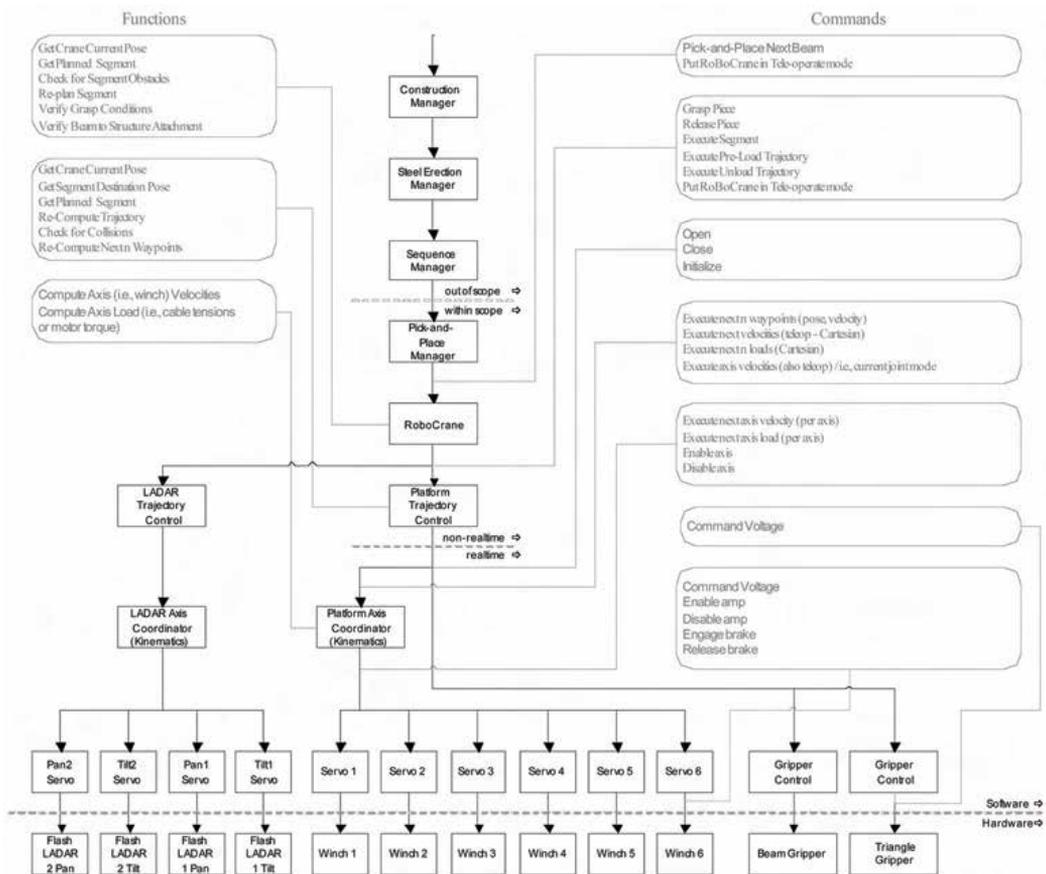


Fig. 12. RoboCrane controller architecture diagram.

Each of these modules are responsible for executing a servo algorithm which accepts the actual and desired positions (or velocity) of a winch motor as inputs and calculates a command voltage which maintains the desired position (or velocity). An alternate configuration would be to group the six servo modules into one.

Figure 12 also shows that the RoboCrane controller is part of a larger control architecture which includes four higher-level modules. For example, at the level above the RoboCrane controller would be a Pick-and-Place Manager that would actually command RoboCrane to perform the pick-and-place operation. The commands sent down by each module to a lower-level module are shown in the light gray boxes on the right. Some of the functions (or non-physical tasks) that each module performs are also shown in the light gray boxes on the left. The control modules above the Pick-and-Place Manager are also included in the figure. Finally, Figure 12 also includes modules for controlling the 3D imaging systems. These modules are responsible for coordinating the sensor orientations with the RoboCrane platform's motion in order to maintain a desired part of RoboCrane's environment within the combined sensors' field of view.

6. Conclusion

This chapter presented new research developments at NIST in control algorithms and controller design for parallel robots applied to Construction applications. In particular, this research focused on the NIST RoboCrane platform for automated placement of construction components. This work was the first to demonstrate the use of a laser-based site measurement system for 6 degree-of-freedom tracking of a robotic crane, and presented new methods for incorporating pose estimation errors in a compensation transform for the NIST GoMotion controller. Finally, this work presented task decomposition approaches for analyzing and automating construction crane operations based on a NIST 4D/RCS approach.

7. References

- Albus, J., Huang, H., Messina, E., Murphy, K., Juberts, M., Lacaze, A., Balakirsky, S., Shneier, M., Hong, T., & Scott, H. (2002). 4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems. *National Institute of Standards and Technology, Gaithersburg, MD, NISTIR 6912.*
- Albus, J.S., Bostelman, R.V., & Dagalakis, N.G. (1992). The NIST ROBOCRANE, A Robot Crane. *Journal of Robotic Systems, July.*
- Barbera, T., Albus, J., Messina, E., Schlenoff, C., & Horst, J. (2004). How task analysis can be used to derive and organize the knowledge for the control of autonomous vehicles. *Robotics and Autonomous Systems 49(1-2), 67-78.*
- Bostelman, R., Jacoff, A., Dagalakis, N., & Albus, J. (1996). RCS-Based RoboCrane Integration. *Proc. Intelligent Systems: A Semiotic Perspective, Gaithersburg, MD, Oct, 20-23.*
- Bostelman, R., Shackleford, W., Proctor, F., Albus, J., & Lytle, A. (2002). The Flying Carpet: A Tool to Improve Ship Repair Efficiency. *American Society of Naval Engineers Symposium, Bremerton, WA, Sept, 10-12.*

- Gazi, V. (2001). The RCS Handbook: Tools for Real-time Control Systems Software Development. Wiley.
- Tsai, L.W. (1999). Robot Analysis: The Mechanics of Serial and Parallel Manipulators. Wiley-Interscience.

Dynamic Parameter Identification for Parallel Manipulators

Vicente Mata¹, Nidal Farhat¹, Miguel Díaz-Rodríguez²,
Ángel Valera³ and Álvaro Page⁴,
*Universidad Politécnica de Valencia, Departamento de Mecánica y Materiales Valencia¹,
Universidad de los Andes, Facultad de Ingeniería,
Departamento de Tecnología y Diseño Mérida²,
Universidad Politécnica de Valencia, Departamento de Ingeniería
de Sistemas y Automática Valencia³,
Universidad Politécnica de Valencia, Departamento de Física Aplicada Valencia⁴,
España^{1,3,4},
Venezuela²*

1. Introduction

The information provided by robot manufacturers regarding the dynamic parameters of robotic systems (the inertial properties of the links and friction parameters at the kinematic joints) is limited and even nonexistent. For instance, friction parameters are generally not provided. Thus, it is necessary to develop efficient procedures for their measurement. The direct measurement of these parameters is not practical since it would imply disassembling the robot. On the other hand, obtaining these parameters from the CAD models has the disadvantage that some parts can not be modeled in full detail and parameters that depend on operational conditions, like friction, can not be determined. For these reasons, parameters identification has turned out to be a widely accepted technique for determining the dynamic parameters. This chapter provides an overview of parameters identification processes applied to parallel manipulators. Practical implementation issues are also considered. In addition, an approach that considers the identification problem as a nonlinear constrained optimization problem is presented. Moreover, an evaluation of the accuracy of the solution of parameters is also addressed.

The importance of inertial and friction parameters lies in their application in most of the recent literature for advanced model-based control algorithms. The accuracy of dynamic parameters plays an important role in the precision, performance, stability and robustness of these control algorithms (Khalil & Dombre, 2002). On the other hand, they are important in dynamic simulation. It is known that the validation of the direct dynamic problem depends considerably on the precision of the dynamic parameters of the mechanical system. An accurately modeled robot permits the substitution of the real mechanical system by a virtual one thus avoiding the expensive experimental tests used to adjust the operational parameters for this system (Hiller et al., 2002). Additionally, another important field in which accurate knowledge of the dynamic parameters is needed is in path planning

algorithms that take into account robot dynamics. The predicted forces depend greatly on the accuracy of the estimated inertial and friction parameters. Hence, inaccurate estimates of the dynamic parameters may lead to an overloaded robot (dynamically or statically), which is the case in approximately 50% of industrial robots (Swevers et al., 2002).

Initially, dynamic parameter identification procedures for estimating the dynamic parameters of open loop mechanical structures were developed in the middle eighties (Khosla & Kanade, 1985; Atkeson et al., 1986; Olsen & Bekey, 1986; Gautier & Khalil, 1988). Since then, they have been widely used and several contributions to serial robot dynamics application control and simulation have been made.

Identification procedures can be classified into two main groups: indirect and direct approaches. On the one hand, indirect procedures act sequentially in several steps. In each step, parameters of a different nature (basically friction and some inertial terms) are identified by means of specifically designed experiments. On the other hand, in the direct approach, all the parameters are identified in the same stage. A detailed comparison between the direct and the indirect approaches, applied to a PUMA industrial robot, can be found in Benimeli et al. (2006). The indirect approach has the disadvantage that errors due to the noise in the measured data are being accumulated throughout the different stages (Khalil & Dombre, 2002). Moreover, it is difficult to maintain the working conditions constant not only throughout these stages, but also within the same one.

For parallel manipulators, the direct approach has been applied (Renaud et al., 1993; Guegan et al., 2003; Farhat et al., 2008). Meanwhile, the indirect approach has been proposed (Grotjahn et al., 2004; Abdellatif et al., 2007). However, apart from error accumulation in each step, the separation of the parameters of a different nature is not straightforward as for open chain manipulators. Due to the fact that the direct approach allows parameters identification in one single experiment, removing the accumulation of error between steps, this chapter will be focus on the direct approach applied to parallel manipulators.

The first part of the chapter deals with conventional direct dynamic parameters identification processes. Thus, the dynamic model, suitable for identification purposes, is developed in its linear form with respect to the dynamic parameters. Due to the fact that the number of parameters is usually greater than the dimension of the equations of motion, an overdetermined system is developed. This overdetermined system is rank deficient, therefore it has to be reduced to another equivalent system that only contains independent columns. These columns correspond to a subset of parameters called the base parameters. Reduction process can be held symbolically (Khalil & Bennis, 1995) or numerically (Gautier, 1991). For experiment design and in order to reduce the sensitivity of the system to the noise signal, procedures have been developed for the trajectories to be executed by the manipulator (Gautier & Khalil, 1992; Swevers et al., 1997). Finally, the dynamic system in its reduced matrix form is solved for the base parameters using the Least Square Method (LSM).

The parameters dynamic identification procedure outlined in the previous paragraph has two main disadvantages: firstly, results could contain non-physically feasible parameters and secondly, it is also limited to linear friction models. Non-physical feasibility can be detected by obtaining a base parameters solution that does not have any physical interpretation when compared with corresponding combinations of the inertial parameters; masses lower than zero or non positive-definite local inertial matrices (Yoshida & Khalil, 2000). This issue not only affects the stability of some of the advanced model-based control algorithms, but is also crucial in the dynamic simulation tasks.

The second part of the chapter will focus on two identification procedures. First, a procedure based on the parameters identification formulated as a nonlinear constrained optimization problem is reviewed. This approach allows not only the implementation of nonlinear friction models to model friction phenomenon at robot joints, but also the consideration of constraint equations in order to ensure the physical feasibility of the identified parameters. The second procedure is established upon the accuracy of the parameter solution, which is called here the *identifiability* of the parameters. Experiments will be held on a class of parallel manipulator. The main conclusions and further research concerning parameters identification for parallel manipulators are presented at the end.

2. Dynamic model

The starting point of the identification process depends on obtaining the dynamic model of the mechanical structure in its linear form with respect to the inertial and friction parameters. For this purpose, the dynamic model can be developed basically by two methods (Kozłowski, 1998); the integral and the differential methods. The integral method is derived from the energy equation and requires measurements of positions, velocities and applied forces on the actuated joints. Measurements of accelerations are not required. This method has been applied on serial manipulators (Gautier & Khalil, 1988; Sheu & Walker, 1989; Khalil et al., 1990; Sheu & Walker, 1991) and extended for parallel manipulators (Bhattacharya et al., 1997). Olsen and others also used the integral method (Olsen & Petersen, 2001; Olsen et al., 2002) where they proposed the use of the *maximum likelihood* method instead of the conventional Least Squares Method (LSM) in the identification process.

On the other hand, the differential method takes the advantage of the equations of motion as a base in the development of the identification process algorithms. As a result, acceleration appears explicitly and needs to be measured. It is known that the equations of motion can be constructed implementing various dynamic principles. Models suitable for the identification process have been developed by means of; the Newton-Euler formulation (Luh et al., 1980; Atkeson et al., 1986; Olsen & Bekey, 1986; Khosla, 1989), the Lagrange formulation (Ha et al., 1989; Sheu & Walker, 1991), Jourdain's principle of Virtual Power (Grotjahn et al., 2004), Gibbs-Appell equations of motion (Benimeli et al., 2003) and recently (Hardeman et al., 2006), a finite element based approach along with Jourdain's principle of Virtual Power was used to develop an automatic generation of the dynamic models for identification.

In order to study the advantages/disadvantages of the integral and differential methods, a comparison was carried out. The experiments were held considering a two degrees of freedom serial manipulator (Prufer et al., 1994). From the results, they concluded that the differential method has advantages over the integral one. Thus, the differential model is used here for parameters identification of parallel manipulators.

2.1 Rigid body model

The equation of motion that describes the dynamic behavior of an open chain manipulator can be obtained by means of Gibbs-Appell equations of motion. For a serial robot (Mata et al., 2002), the rigid body dynamic model can be written as follows,

$$\tau_k = \sum_{i=k}^n \left\{ \left(\frac{\partial}{\partial \ddot{q}_k} \right)^T \cdot \left[{}^i \mathbf{I}_{G_i} \cdot {}^i \dot{\omega}_i + {}^i \tilde{\omega}_i \times ({}^i \mathbf{I}_{G_i} \cdot {}^i \tilde{\omega}_i) \right] \right\} + \sum_{i=k}^n \left[\left(\frac{\partial}{\partial \ddot{q}_k} \right)^T \cdot m_i \cdot {}^i \ddot{\mathbf{r}}_{G_i} \right] \quad (1)$$

where τ_k and \ddot{q}_k are the generalized forces and accelerations of the joint k , respectively, ${}^i \dot{\omega}_i$ is the angular acceleration, and ${}^i \ddot{\mathbf{r}}_{G_i}$ is the acceleration of the center of gravity, m_i is the mass and ${}^i \mathbf{I}_{G_i}$ is the inertial tensor of the center of gravity, and the superscript/subscript (i) stands for the link number. All of them expressed with respect to the i^{th} local reference frame. The Denavit-Hartenberg modified convention has been considered for modeling the system and $i = k..n$ indicates the sum over all links above joint k , including itself.

For dynamic parameters identification, a linear form with respect to dynamic parameters is necessary. To this end (Atkeson et al., 1986), the linear acceleration of the center of gravity of the i^{th} body is expressed as a function of the linear acceleration of the link coordinate frame i^{th} . Moreover, the link inertial tensor is also expressed about the link coordinate frame by means of parallel axis theorem.

In addition, the following notations are introduced. On the one hand, the cross product $\vec{a} \times \vec{b}$ is expressed using the skew symmetric matrix \tilde{a} . Thus, $\vec{a} \times \vec{b} = \tilde{a} \cdot b$ where,

$$\tilde{a} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \text{ and } \vec{b} = [b_x \quad b_y \quad b_z]^T \quad (2)$$

On the other hand, $\mathbf{B} \cdot \vec{a} = \hat{a} \cdot \vec{B}$ where,

$$\hat{a} = \begin{pmatrix} a_x & a_y & a_z & 0 & 0 & 0 \\ 0 & a_x & 0 & a_y & a_z & 0 \\ 0 & 0 & a_x & 0 & a_y & a_z \end{pmatrix} \text{ and } \vec{B} = [B_{xx} \quad B_{xy} \quad B_{xz} \quad B_{yy} \quad B_{yz} \quad B_{zz}]^T \quad (3)$$

By doing the above mentioned, upon substituting (2)-(3) in (1) and using some vector identities, the dynamic model linear with respect to dynamic parameters can be written as (Mata et al., 2005),

$$\tau_k = \begin{cases} {}^k \bar{\mathbf{z}}_k^T \cdot \sum_{i=k}^n {}^k \mathbf{R}_i \left[{}^i \hat{\eta}_i \cdot {}^i \bar{\mathbf{I}}_i + ({}^i \tilde{\mathbf{r}}_{O_k O_i} \cdot {}^i \tilde{\eta}_i - {}^i \ddot{\mathbf{r}}_{O_i}) \cdot m_i \cdot {}^i \tilde{\mathbf{r}}_{O_i G_i} + {}^i \tilde{\mathbf{r}}_{O_k O_i} \cdot {}^i \ddot{\mathbf{r}}_{O_i} \cdot m_i \right] & k : R \\ {}^k \bar{\mathbf{z}}_k^T \cdot \sum_{i=k}^n {}^k \mathbf{R}_i \cdot (m_i \cdot {}^i \ddot{\mathbf{r}}_{O_i} + {}^i \tilde{\eta}_i \cdot m_i \cdot {}^i \tilde{\mathbf{r}}_{O_i G_i}) & k : P \end{cases} \quad (4)$$

where ${}^i \hat{\eta}_i = ({}^i \dot{\omega}_i + {}^i \tilde{\omega}_i \cdot {}^i \hat{\omega}_i)$ y ${}^i \tilde{\eta}_i = ({}^i \tilde{\omega}_i \cdot {}^i \tilde{\omega}_i + {}^i \dot{\omega}_i)$.

In equation (4), ${}^i \tilde{\mathbf{r}}_{O_i G_i}$ locates the center of mass of the i^{th} link with respect to its own reference system. ${}^i \ddot{\mathbf{r}}_{O_i}$ is the acceleration of the origin of the i^{th} local reference system. ${}^k \mathbf{R}_i$ is the rotation matrix between the i^{th} and k^{th} reference systems. ${}^i \tilde{\mathbf{r}}_{O_k O_i}$ is the position vector from the i^{th} reference system to the k^{th} reference system with respect to the i^{th} reference

system. ${}^k\bar{z}_k = [0 \ 0 \ 1]^T$. P and R denote the type of the corresponding joint, revolute or prismatic, respectively. Equation (4) can be written in the following matrix form,

$$\mathbf{K}(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi}_{rb} = \bar{\tau} \quad (5)$$

\mathbf{K} can be denoted as the observation matrix of a single configuration; this matrix depends on the generalized kinematic variables. The vector of dynamic parameters $\bar{\Phi}_{rb}$ regroups the elements of the inertia tensor $[\bar{I}_{xx} \ \bar{I}_{xy} \ \bar{I}_{xz} \ \bar{I}_{yy} \ \bar{I}_{yz} \ \bar{I}_{zz}]^T$, calculated with respect to the local reference frame, the mass m and the first mass moments with respect to the center of gravity $[m_x \ m_y \ m_z]^T$, for all the bodies contained in the system.

Equation (5) can be applied for the dynamic parameter identification of open chain manipulators. Under other circumstances, such as parallel manipulators, its application is not straightforward. For parallel manipulators, the dynamic model can be obtained by making a cut at one or more joints so that the manipulator can be dealt with as an open-chain mechanical systems with a tree structure. By doing so, equation (5) can be applied for the several open chain mechanical systems obtained after the cut. However, the constraint equations representing the union at the cutted joints should be fulfilled. These equations have the following form,

$$f_i(q_1, q_2, \dots, q_n) = 0 \quad i = 1, 2, \dots, m \quad (6)$$

where (q_1, q_2, \dots, q_n) are the generalized coordinates and m is the number of independent constraint equations. The degree of freedom (n_{DOF}) of the system is obviously $(n-m)$. Taking the first and the second time derivatives of the previous equations, the following the acceleration constraint equations are obtained,

$$\mathbf{A}(\bar{q}) \cdot \ddot{\bar{q}} - \bar{b}(\bar{q}, \dot{\bar{q}}) = \bar{0} \quad (7)$$

where \mathbf{A} is the Jacobian matrix of the constraint equations with respect to the generalized coordinates, and \bar{b} is a vector that contains all the terms that remain after removing all the acceleration dependent terms from the acceleration constraint equations. Regrouping the terms of the matrix \mathbf{A} , according to the coordinated partition, in independent/dependent generalized accelerations,

$$[\mathbf{A}_i \ \mathbf{A}_e] \begin{bmatrix} \ddot{\bar{q}}_i \\ \ddot{\bar{q}}_e \end{bmatrix} = \bar{b} \quad (8)$$

where \mathbf{A}_i and \mathbf{A}_e are obtained when the above mentioned coordinated partition is applied to the Jacobian matrix of the constraint equations.

Similarly, regrouping equation (5) but this time according to the independent and dependent generalized coordinates,

$$\mathbf{K}_i \cdot \bar{\Phi}_{rb} = \bar{\tau}_i \quad \mathbf{K}_e \cdot \bar{\Phi}_{rb} = \bar{\tau}_e \quad (9)$$

The subindices i and e refer to independent and dependent generalized coordinates respectively.

In a similar way to that introduced in (Udwadia & Kalaba, 1998), and starting from equation (4) and equation (5), it can be proved that the dynamic equation for parallel manipulators in its linear form with respect to the dynamic parameters can be written in the form,

$$[\mathbf{K}_i - \mathbf{X}^T \cdot \mathbf{K}_e] \cdot \vec{\Phi}_{rb} = \vec{\tau}_i - \mathbf{X}^T \cdot \vec{\tau}_e \quad (10)$$

where $\mathbf{X} = \mathbf{A}_e^{-1} \cdot \mathbf{A}_i$.

If the dependent generalized forces correspond to passive joints then, they can be dropped from the equation. Hence, equation (6) is reduced to,

$$[\mathbf{K}_i - \mathbf{X}^T \cdot \mathbf{K}_e] \cdot \vec{\Phi}_{rb} = \vec{\tau}_i \quad (11)$$

The observation matrix for a given trajectory can be found by appending this equation over all the configurations (n_{pts}) of the trajectory. This gives,

$$\begin{bmatrix} [\mathbf{K}_i - \mathbf{X}^T \cdot \mathbf{K}_e]_1 \\ [\mathbf{K}_i - \mathbf{X}^T \cdot \mathbf{K}_e]_2 \\ \vdots \\ [\mathbf{K}_i - \mathbf{X}^T \cdot \mathbf{K}_e]_{n_{pts}} \end{bmatrix} \cdot \vec{\Phi}_{rb} = \begin{bmatrix} [\vec{\tau}_i]_1 \\ [\vec{\tau}_i]_2 \\ \vdots \\ [\vec{\tau}_i]_{n_{pts}} \end{bmatrix} \quad (12)$$

The left-hand side of this equation is the observation matrix for a given trajectory (\mathbf{W}_{rb}) and the right-hand side is the corresponding applied forces ($\vec{\tau}$), or in a compact form,

$$\mathbf{W}_{rb}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}) \cdot \vec{\Phi}_{rb} = \vec{\tau} \quad (13)$$

2.2 Friction models

Several friction models have been proposed in the literature (Olsson et al., 1998). The classical friction model used for identification is a linear model which includes Coulomb and viscous friction. Equation (14) represents a general asymmetrical linear model (Armstrong, 1988) for both Coulomb and viscous friction,

$$\mathbf{F}_f = \begin{cases} F_c^+ + F_v^+ \dot{q} & \dot{q} > 0 \\ -F_c^- - F_v^- \dot{q} & \dot{q} < 0 \end{cases} \quad (14)$$

where F_c and F_v stand for the Coulomb and viscous friction coefficients, respectively. (+ve) and (-ve) superscripts correspond to the velocity sign. Considering different coefficients for the different velocity sign, this model is asymmetrical. Applying the friction model to all joints and for all the configurations (n_{pts}),

$$\mathbf{W}_f(\dot{\vec{q}}) \cdot \vec{\Phi}_f = \vec{\tau}_f \quad (15)$$

The previous equation is applicable in the case of linear friction model. In the other case, if the friction at joints seems to have nonlinear tendency, nonlinear friction models can be used. In the identification process different nonlinear friction models have been used. For example the following models (Grotjahn et al., 2001),

$$F_2(v) = F_C + F_{v_1} v + F_{v_2} \operatorname{atan}(F_{v_3} v) \quad (16)$$

where, F_C, F_{v_1}, F_{v_2} and F_{v_3} are friction model parameters. On the other hand, another model proposed (Farhat, 2006) for friction modeling has the form,

$$F_f = \begin{cases} F_c^+ + F_v |v|^\delta & v > 0 \\ -F_c^- - F_v |v|^\delta & v < 0 \end{cases} \quad (17)$$

where, F_c , F_v are Coulomb and viscous parameters and δ a geometry dependent variable that takes into account the *Stribeck effect*.

2.3 Actuator dynamics

In some cases, a considerable part of the actuator torque is consumed by accelerating or decelerating its rotor inertia (J_{ri}) and its driven system (for instance, a ball screw drive, J_s). Then, the rotor and the driving system inertia have to be considered. The corresponding equation for the actuator of the i^{th} joint can be written as follows,

$$\tau_{ri} = (J_{ri} + J_s) \cdot \ddot{q}_i \quad (18)$$

Equations (18) is linear. The actuator dynamic for all the joints and for all the configurations (n_{pts}) can be expressed in matrix form as,

$$\mathbf{W}_r(\ddot{\mathbf{q}}) \cdot \bar{\Phi}_r = \bar{\tau}_r \quad (19)$$

2.4 Complete robot model

If only linear friction models are considered in the identification process, equations (13), (15), and (18) can be grouped. Thus the complete dynamic model of the manipulator can be express as follows,

$$[\mathbf{W}_{rb} \quad \mathbf{W}_f \quad \mathbf{W}_r] [\bar{\Phi}_{rb}^T \quad \bar{\Phi}_f^T \quad \bar{\Phi}_r^T]^T = \mathbf{W} \cdot \bar{\Phi} = \bar{\tau} \quad (20)$$

In equation (20), \mathbf{W} is the observation matrix of the system and $\bar{\Phi}$ is the vector grouping all the dynamic parameters. In the case that nonlinear friction models are considered, the complete dynamic model of the manipulator can be expressed in the form,

$$[\mathbf{W}_{rb} \quad \mathbf{W}_r] [\bar{\Phi}_{rb}^T \quad \bar{\Phi}_r^T]^T + (\bar{F}_i - \mathbf{X}^T \cdot \bar{F}_i) = \mathbf{W} \cdot \bar{\Phi} + (\bar{F}_i - \mathbf{X}^T \cdot \bar{F}_i) = \bar{\tau} \quad (21)$$

where \bar{F}_i and \bar{F}_c stand for the friction in the independent and dependent joints, respectively. These vectors include the friction parameters ($\bar{\Phi}_f$). $\bar{\Phi}$, in this case, contains the body and the actuator dynamic parameters.

2.5 Base parameters

An important characteristic of the system expressed by equation (20) is that some inertial parameters do not affect the dynamics of the manipulator and others have a relative effect with respect to the other parameters on the external generalized forces. Mathematically, this can be expressed by the presence of zero columns in the observation matrix (\mathbf{W}) and the dependence that exist between others. Hence, this system never could be considered as a determined one. The minimal number of parameters that are needed to determine uniquely the dynamics of the system has to be calculated. This set of parameters is known as the base parameters and can be considered as a combination with the others that have an effect. The number of base parameters is equal to the rank of the observation matrix.

The base parameter combination can be calculated principally in two ways; analytically (Khalil & Bennis, 1995) or numerically using the Singular Values Decomposition (SVD) or QR factorization (Gautier, 1991). The analytical analysis will not be considered here since for a parallel manipulator its application is not direct. Analytical calculation of the base parameters have already been presented for close chains mechanical systems (Khalil & Bennis, 1995), however, the method is applicable only to some particular topologies. The use of SVD is characterized by its precision, while the QR factorization by its low computational cost. The first case is of interest to us since the precision of the resulting inertial parameters identified is more important.

As an example, consider the 3-DOF RPS (revolute, prismatic and spherical joints) parallel manipulator depicted in Fig. (1), this manipulator (a virtual model and an actual one) are used here for the experimental evaluation of the dynamic parameter identification process.

As can be seen in the figure, this parallel manipulator consists of a fixed base and a moving platform interconnected by three RPS limbs. The axes of rotation of the revolute joints are assumed to share the same plane of the base. Spherical joints are modeled as three successive revolute joints with the corresponding axes of rotation passes through the center of the spherical joint. The linear motion of each prismatic joint of the actual parallel robot is achieved through a ball screw linear actuator driven by a DC motor.

This parallel robot consists of 7 bodies: 3 limbs in which two of them contain two bodies and one with 3 bodies including the platform. Then, considering linear friction models and for a trajectory of n_{pts} configurations, the manipulator dynamic model in its linear form is appended in the following matrix form,

$$\mathbf{W}_{(n_{DOF} \cdot n_{pts}) \times (70 + n_{fric} + n_j)} \cdot \bar{\Phi}_{(70 + n_{fric} + n_j) \times 1} = \bar{\tau}_{(n_{DOF} \cdot n_{pts}) \times 1} \quad (22)$$

As described before and because of the dependence between the inertial parameters the SVD has been used to reduce this model to its base form (reduced form) expressed by the following equation,

$$\mathbf{W}_{red}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \cdot \bar{\Phi}_{base} = \bar{\tau} \quad (23)$$

where \mathbf{W}_{red} is the reduced matrix and $\bar{\Phi}_{base}$ is the base parameters vector. This vector is composed of the 25 parameters listed in Table (1). It has been found that friction parameters, as well as the screw and rotor inertial parameters, are linearly independent.



Fig. 1. The 3-DOF RPS parallel robot; ADAMS model on the left and Actual 3-RPS parallel robot on the right (built at the Polytechnic University of Valencia).

No	Base Parameters	No	Base Parameters
1	$m_x(1)$	14	$m_x(4)$
2	$m_y(1)^*, **$	15	$m_y(4)^*$
3	$I_{zz}(1)+I_{yy}(2)^*, **$	16	$I_{yy}(5)+ I_{zz}(4)^*$
4	$m_x(2)$	17	$m(5)-2.531m_y(3)+m(3)+m(2)^*, **, †$
5	$m_z(2)^*, **$	18	$m_x(5)$
6	$I_{xx}(3)-0.3952m_y(3)^*, **$	19	$m_z(5)^*$
7	$I_{xy}(3)+0.2282m_y(3)$	20	$m_x(6)$
8	$I_{xz}(3)$		
9	$I_{yy}(3)+0.3952m_y(3)-0.2082(m(3)+m(2)) \quad *, **$	21	$m_y(6)^*$
10	$I_{yz}(3)$	22	$I_{yy}(7)+ I_{zz}(6)^*$
11	$I_{zz}(3)-0.2082(m(3)+m(2))^*, **$	23	$m(7)+2.531m_y(3)^*, **, †$
12	$m_x(3)+0.5774m_y(3)-0.4563(m(3)+m(2)) \quad *, **, †$	24	$m_x(7)$
13	$m_z(3)$	25	$m_z(7)^*$

Table 1. Rigid body base parameters of a 3-RPS robot.

3. Experiment design

An accurate and efficient dynamic parameter identification process requires special design experiments. The trajectories to be executed by the robot not only have to be able to reduce the sensitivity of the identification solution to the noise signal, but also the data processing (position and forces) needs to be kept as simple and accurate as possible.

The condition number of a matrix can be considered as an upper limit for input-output error transmissibility (Horn & Johnson, 1985). Therefore one of the criteria for designing a “well excited” trajectory is to minimize the condition number of the matrix \mathbf{W}_{red} . This can be treated as an optimization process where the objective function can be written as,

$$f(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, \ddot{\bar{\mathbf{q}}}) = \text{Cond}(\mathbf{W}_{\text{red}}) \quad (24)$$

As can be observed from equation (24), the variables of the optimization process are the generalized coordinates and their time derivatives. Several approaches have been proposed (Armstrong, 1989; Gautier & Khalil, 1992; Presse & Gautier, 1993) in which the trajectory is parameterized. However, the finite Fourier series trajectory parameterization proposed by (Swevers et al., 1996) is the most widely implemented.

$$\begin{aligned} q_i(t) &= q_{i0} + \sum_{j=1}^{n_H} \left[\frac{a_{ij}}{2\pi f \cdot j} \sin(2\pi f \cdot j \cdot t) - \frac{b_{ij}}{2\pi f \cdot j} \cos(2\pi f \cdot j \cdot t) \right] \\ \dot{q}_i(t) &= \sum_{j=1}^{n_H} \left[a_{ij} \cos(2\pi f \cdot j \cdot t) + b_{ij} \sin(2\pi f \cdot j \cdot t) \right] \\ \ddot{q}_i(t) &= \sum_{j=1}^{n_H} \left[-2\pi f \cdot j \cdot a_{ij} \cdot \sin(2\pi f \cdot j \cdot t) + 2\pi f \cdot j \cdot b_{ij} \cdot \cos(2\pi f \cdot j \cdot t) \right] \end{aligned} \quad (25)$$

where t is the time, q_{i0} , a_{ij} and b_{ij} are the coefficients of the Fourier series that will be the new variables of the optimization process, n_H is the harmonic number and f is the fundamental frequency.

Generally, there exist many limitations on the movement of the actual manipulator. These can be converted to constraints in the optimization process. Depending on their complexity with respect to the variables of the optimization process they could be linear or nonlinear. For example, limitations on the displacements and the velocity of the prismatic joints (actuators) – if exist – are linear, meanwhile limitations to avoid the singularity regions or physical constraints such as the aperture angle of the spherical joints are nonlinear.

It will be important to mention that the condition number of the observation matrix is not the only criteria for finding exciting trajectories. In a statistical frameworks the covariance matrix of the maximum likelihood has been used (Swevers et al., 1997). The criteria for optimization is called *d-optimality criterion* and has the following form,

$$f(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, \ddot{\bar{\mathbf{q}}}) = -\log(\det(\mathbf{W}^t \Sigma^{-1} \mathbf{W})) \quad (26)$$

where Σ is the diagonal covariance matrix of the measured actuator forces. Recently, this criteria was used for the case in which the parameterization of the trajectories was based on combining Fourier series and polynomial functions (Park, 2006).

As mentioned previously above, the exciting trajectories must allow simple and accurate data processing. This can be achieved if measurement position is fitted to the finite Fourier series.

For the considered parallel manipulator presented in the previous section, the exciting trajectories were obtained according to the criteria of minimization of the condition number of the reduced observation matrix. The trajectory was parameterized by means of a finite Fourier series with 11 harmonics functions. Limitations on the movement of the actual manipulator that were introduced as constraints in the optimization process can be outlined as follows,

1. Limitations on the stroke of the linear actuators, upper and lower bounds.
2. Limitations on the velocities of the linear actuators, upper and lower bounds.
3. Limitation on the aperture angle of the spherical joints. Observe that this is a sufficient condition to avoid singularity regions of this manipulator.

The optimization process was carried out in FORTRAN environments using a nonlinear Sequential Quadratic Programming (SQP) optimization routine provided by the NAG commercial library.

It is important to mention that in the optimization process, one can take use of methods for scaling the reduced observation matrix. The scaled matrix can be obtained as follow,

$$(\mathbf{W}_{\text{red}})_{\text{eq}} = \mathbf{R} \cdot \mathbf{W}_{\text{red}} \cdot \mathbf{C} \quad (27)$$

where \mathbf{R} is the $r \times r$ diagonal matrix and \mathbf{C} is the $b \times b$ diagonal matrix, both including row and column scaling factors on their diagonal, respectively. r and b are the dimensions of \mathbf{W}_{red} . To keep on the benefits of the condition number reduction of \mathbf{W}_{red} , scaling must be introduced into the identification process. Another approach that can be used for scaling \mathbf{W}_{red} is the normalization of the matrix by means of the division of its columns by their norms. The use of scaling factors or the normalization rescale has to be taken into account in the identified model parameters afterwards.

For the purpose of experimental parameter identification, several trajectories were found. For example, one of the trajectories that have been obtained had the condition number of 595. It is depicted in Fig. (2).

4. Physical feasibility

Once an optimized trajectory has been found and it is applied to the manipulator, data is measured so that the dynamic system represented by equation (23) is ready to be solved. At this point an important issue in the dynamic parameters identification arises: the physical feasibility of the identified parameters. Because of the incompleteness of the dynamic model and the existence of noise in the measurements, the solution of equation (23) by LSM, or the identified base parameters, is susceptible to have no physical interpretation when compared with corresponding combinations of the inertial parameters; masses lower than zero or non positive-definite local inertial matrices (Yoshida & Khalil, 2000). This issue not only affects the stability of some of the advanced model-based control algorithms, but it is also a crucial one in the case of dynamic simulations.

To insure the physical feasibility of the identified parameters a nonlinear *constrained* optimization process, instead of the LSM, could be implemented to solve equation (1). The corresponding physical feasibility constraint equations have the following form,

$$\begin{aligned} m_i &> 0 \\ I_{xx}^g &> 0 \\ I_{xx}^g I_{yy}^g - (I_{xy}^{xy})^2 &> 0 \\ \det({}^i \mathbf{I}_{G_i}) &= I_{xx}^g I_{yy}^g I_{zz}^g + 2 I_{xy}^g I_{xz}^g I_{yz}^g - I_{xx}^g (I_{yz}^g)^2 - I_{yy}^g (I_{xz}^g)^2 - I_{zz}^g (I_{xy}^g)^2 > 0 \end{aligned} \quad (28)$$

where $I_{xx}^g, I_{xy}^g, I_{xz}^g, I_{yy}^g, I_{yz}^g$ and I_{zz}^g are the components of body inertia matrix ${}^i \mathbf{I}_{G_i}$ calculated with respect to its center of gravity.

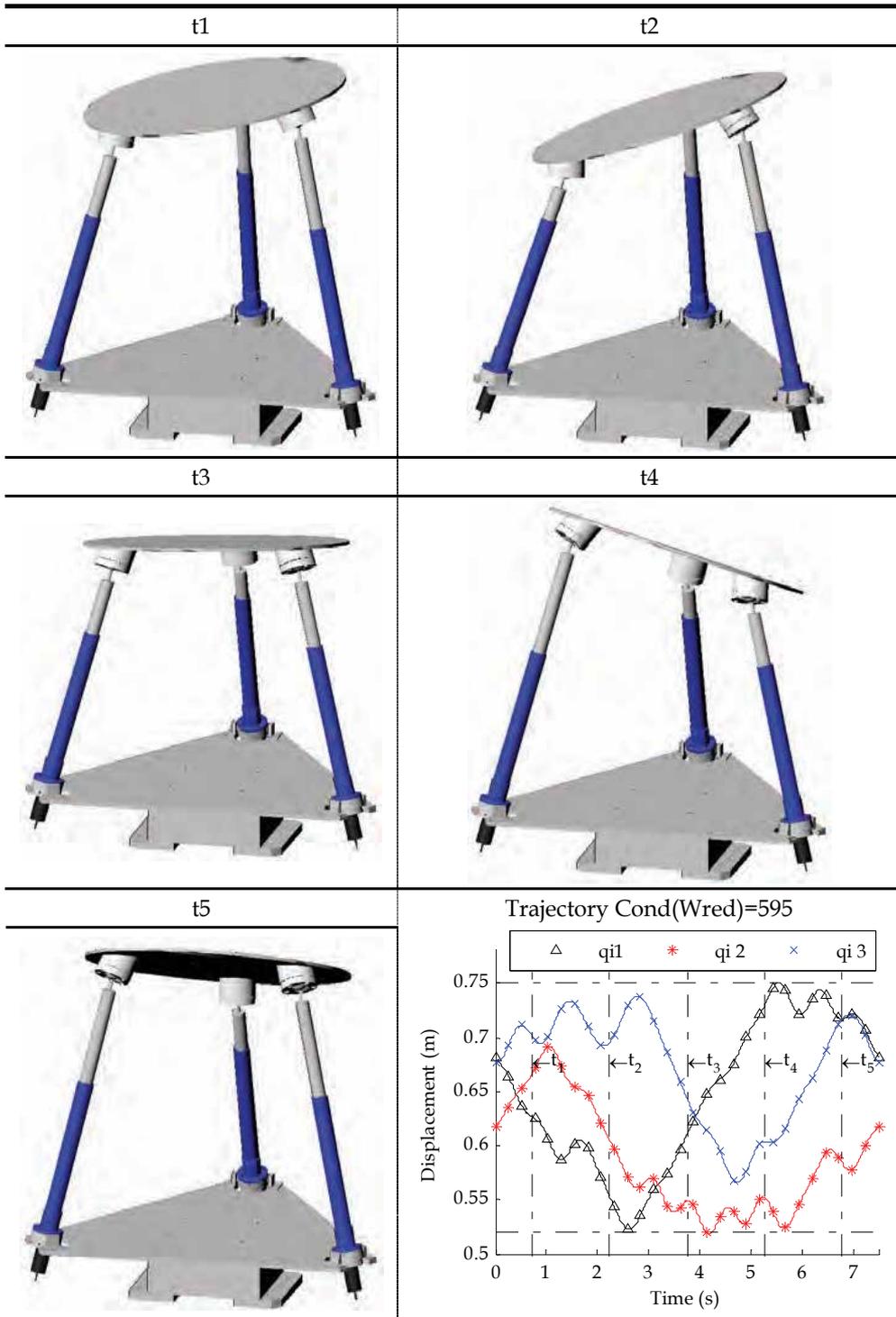


Fig. 2. An optimized trajectory with a Condition Number of 595.

Note that in equation (28), local inertia matrixes are calculated with respect to the center of gravity, meanwhile the inertia matrix terms of the identified base parameters are calculated with respect to the origin of the local reference system. Hence, to verify the physical feasibility of a set of numerical values of the base parameters, they must be compared with the corresponding combinations of the inertial parameters. That is to say, if it is possible to obtain a set of inertial parameters that verify the equality between the linear combinations and the numerical values of the base parameters then this set of base parameters – numerical values – is judged to be physically feasible, and vice versa. A scheme was proposed (Yoshida & Khalil, 2000) to judge whether the values of the base parameters correspond to a set of physically feasible inertial parameters, or not. This scheme evaluates the physical feasibility after the parameter identification process has been carried out. Other authors subject formulate the identification process as a nonlinear constrained optimization problem (Mata et al., 2005; Farhat et al., 2008). This approach allows on the one hand, ensuring the physical feasibility of the identified parameters. On the other hand, it permits the implementation of nonlinear friction models, like the one expressed by equations (16) and (17), to model friction phenomenon at robot joints. To this end, the objective function of the identification problem has the form,

$$f(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}, \bar{\tau}) = \bar{\tau} - \underbrace{\left(\mathbf{W}_{\text{red}}^I \right)_{\text{eq}} \cdot \left(\Phi_{\text{base}}^I \right)}_{\text{Linear part}} + \underbrace{\left(\bar{\mathbf{F}}_f - \mathbf{X}^T \cdot \bar{\mathbf{F}}_f \right)}_{\text{Nonlinear friction vector}}, \quad (29)$$

constrained by equation (28) applied to all links constituting the robot.

5. Practical issues

5.1 Measurements

Recall that the inputs of the identification process are the external forces, positions and there time derivatives, all measured for an optimized trajectory. Generally, the external forces or the torque exerted by motors is not directly available. An acceptable approach is to assume a linear relation between the current and the torque,

$$\tau_m = K_m i_m \quad (30)$$

where τ_m is the motor torque, i_m is the motor current and K_m is the torque constant which can be found, among other methods, by means of a previously performed experiments on the dismantled motors or by in situ experiments (Corke, 1996).

On the other hand, for kinematic variables, most industrial robots are provided with a precise position sensor. However, the direct measurement of joints velocities and joints accelerations is not normally available. Thus, they are obtained by taking the time derivative of the measured positions. These derivatives could be obtained numerically using central difference algorithms (Khalil & Dombre, 2002). For the velocity,

$$\dot{q}_i(t) = \frac{(q_i(t+1) - q_i(t-1))}{2\Delta t} \quad (31)$$

Other approaches (Swevers et al., 1996) suggest, using the exciting trajectory found by the optimization process. In this manner the new trajectory coefficients are found by refitting

the measured positions to the Fourier series. Therefore, the first and second time derivatives could be obtained analytically. Other authors (Gautier et al., 1995) propose filtering the measured positions by a low pass filter before taking the derivative numerically.

The authors of this chapter made a comparison between three different methods for finding the time derivatives (Diaz-Rodriguez et al., 2007). The comparison was carried out over a 3-RPS parallel manipulators which was equipped with accelerometers located at the generalized independent coordinated. The following methods were used,

1. Filtering the measured positions by a low pass filter before taking the derivative numerically (Gautier et al., 1995)
2. Refitting the measured positions to the Fourier series used in the trajectory optimization process (Swevers et al., 1996)
3. Approach based on local fitting (Page et al., 2006). A local regression of the measured trajectory was executed using a third order polynomial. When performing the local regression the whole set of samples were considered, but with different statistical weights. After applying the local regression, the velocities and accelerations were derived from each polynomial.

The main conclusions of the work indicated that the three methods give quite similar results for the analyzed robot. Therefore, provided that well excited trajectories are used in the identification, any of these methods can be used in a deterministic framework.

5.2 Dynamic model reduction

If the geometry of robot parts is taken into account, some rigid body base parameters have zero values or values close to it. Consider for instance the 3-RPS robot (Fig. (1)) where the links connected to the base have a cylindrical geometry. It can be supposed – with a degree of certainty that the gravity center of these links lies on an axis parallel to the actuator movement. In this case, the corresponding axes of the local reference system attached to the body is in (y) direction, so the parameter related to the (x) position of the gravity center can be expected to have values close to zero. The same assumption is applied to links connected to the moving platform. Therefore, parameters 1, 4, 14, 18, 20, 24 from Table (1) can be removed. Moreover, it is possible to consider as well the form of the platform which is circular and flat, and by doing so parameters 7, 8, 10, 13 can also be removed. This reduced model is highlighted by (*) in Table (1). Another simplification can be applied if the parallel manipulators symmetry is considered. In Table (1) the rigid body base parameters of this case is highlighted by (**). It is important to mention that the columns of the observation matrix, associated with base parameters that consider the symmetry, have to be added in order to develop a model which properly describes the dynamic behavior of the robot. Table (2) summarizes the different models that have been proposed here.

5.3 Identifiability of the base parameters

When a direct parameter identification process is experimentally performed, two sources of error become apparent. On the one hand, not all the aspects of the robot can be modeled in detail (modeling discrepancies). On the other hand, noise in measurements is present. These errors lead to the fact that not all the base parameters can be properly identified. This apparently occurs when the independent contribution of some parameters to the generalized forces is smaller than the measurement noise or the modeling discrepancies.

The study of which parameters can be properly identified can be established on analyzing the relative standard deviation (σ_{pi}) of each parameter (Khalil & Dombre, 2002b) along with considering the physical feasibility (Yoshida & Khalil, 2000). Physical feasible base parameters with $\sigma_{pi} < 15\%$ are considered properly identified.

Model	Nº of Rigid Body Parameters	Parameter removed from Table (1)
1	25	-
2	15	1, 4, 7, 8, 10, 13, 14, 18, 20 and 24
3	9	1, 4, 7, 8, 10, 13, 14, 15, 16, 19, 18, 20, 21, 22, 24 and 25

Table 2. Different rigid body models for the dynamic parameter identification process of the 3-DOF RPS parallel manipulator.

6. Application to a 3-RPS robot

In this section, the results of the identification process, implemented by the authors over a 3-RPS parallel manipulator, are presented. In the first part, the approach based on considering the physical feasibility in the identification process is presented. In the second part of the section, the identifiability of the dynamic parameter for the 3-RPS robot is evaluated. In both sections, the identification process is validated using a simulated manipulator built by making use of the ADAMS dynamic simulation program. After that, it is applied over a real one constructed at the Polytechnic University of Valencia, see Fig. (1).

6.1 Identification considering the physical feasibility

Because of the noise in the input data and/or the discrepancies between the actual parallel robot and the dynamic model used in the identification process, some of the inertial parameters obtained using LSM methods result physically unfeasible. Thus, the necessity for a constrained optimization process to ensure physical feasibility appears clearly. In this subsection, the results are shown as a comparison between the original actuator forces and those calculated using the identified dynamic parameters in the case of; a) linear friction models and. b) nonlinear friction models.

The dynamic model of this manipulator, trajectory optimization and identification process were built in FORTRAN programming language with the aid of the NAG library and the NLPQL Sequential Quadratic Programming subroutine (Schittkowski, 2000).

Simulated Robot

In the simulated robot, nonlinear friction model is considered at all the joints of the robot. It can be represented by the following relation,

$$F(v) = F_C + (F_S - F_C)e^{|v/v_s|^{\delta_s}} + F_v v \quad (32)$$

Where F_C , F_S and F_v are the Coulomb, static, and viscous friction coefficients, respectively, v_s is the Stribeck velocity and δ_s is the stiction transition velocity. This model consists of five parameters and captures the Coulomb, static, viscous and Stribeck friction forces (Olsson et al., 1998). After calculating the external original forces, errors are introduced assuming a normal distribution producing the perturbed forces. Now, based on these perturbed forces,

identification is carried out considering asymmetric linear friction models for all joints, using LSM and optimization, and symmetric nonlinear friction models using optimization. The corresponding identification errors are shown in Table (3). where ε_{RA} is the Relative Absolute Error and is defined as,

$$\varepsilon_{RA} = \frac{\sum_i |\tau_{idnt_i} - \tau_i^*|}{\sum_i |\tau_i^* - \bar{\tau}^*|} \quad (33)$$

where, τ^* and τ_{idnt} are the actual applied force and those calculated using the dynamic model applying identified dynamic parameters, and $\bar{\tau}^*$ is the average of τ^* .

ε_{RA} (%)	Perturbed	Original
Linear (LSM)	11.96	8.26
Linear (Opt.)	12.44	8.85
Nonlinear (Opt.)	11.02	7.28

Table 3. Identification errors based on simulated manipulator.

As can be seen in Table (3), considering the case where linear friction models were used in the identification process, when the physical feasibility had been ensured, i.e. identification by optimization, the error increased. On the contrary, when this was accompanied by the nonlinear friction models, the results were improved considering both the perturbed and the original forces. Note that in this step, identification was carried out simulating both types of the mentioned identification process error sources.

Actual robot

Now, after verifying the identification process over the simulated manipulator, the results are shown in detail considering the identification of the dynamic parameters of the actual manipulator. Starting with the optimization process for the exciting trajectory and changing the initial estimations, different optimized trajectories were obtained. An example of such an optimized trajectory is that one presented previously in Fig. (2). Hereafter, the identified dynamic parameters were obtained basing on another optimized trajectory with a corresponding condition number of 638.

A PID controller is used in order to determine the control actions. The control actions were applied with a frequency of 100Hz, at which measurement were also taken. The total duration of the optimized trajectory is 7.5s. Trajectories were repeated several times, the applied control actions were averaged and then a second order lowpass digital Butterworth filter was applied. For the identification process, 75 configuration points are extracted every 0.1s.

When the LSM was used in the identification process, a non physically feasible base parameters were found. Hence, the identification process was held using the nonlinear constrained optimization process where the physical feasibility of the obtained inertial parameters was ensured. Fig. (3) shows a graphical comparison between the actual forces and those calculated using the dynamic parameters identified by LSM and optimization considering asymmetric linear friction models at the prismatic joints.

In order to justify the use of nonlinear friction models in the identification process rather than those which are linear, as the friction phenomenon in the considered joints has this tendency, a thorough error comparison was made. This is established considering three different sets of dynamic parameters identified by: LSM, optimization in the linear case if linear friction models are considered and optimization in the nonlinear case. In order to make an overall judgment, error comparison takes place over the same trajectory used in the identification process (that one of a condition number of 638) and others which are excited, including the low velocity one. The resulted calculated error in each case is shown in Table (4).

As can be observed from Table (4), considering identification by the optimization case and excluding the trajectory used in the identification process, the errors in the predicted applied forces considering the identified nonlinear friction models are lower than those corresponding to the linear friction ones for all trajectories. This shows that the dynamic model that includes nonlinear friction models has a better overall response. On the other hand, the dynamic parameters obtained by the LSM give the lowest error for all of the test trajectories. However, the calculated errors using the identified parameters found by means of the test trajectories became bigger, and almost doubled, contrary to those calculated by optimization, which kept the same order. Furthermore, the identified dynamic parameters using the optimization process are physically feasible.

6.2 Identifiability of the base parameters

The identification process, as has been pointed out in the previous section, has the ability to obtain a physical feasibility solution; however, the constrained optimization problem is cumbersome. This occurs because constraint equations are functions of the terms of the inertia tensor calculated with respect to the center of gravity of the corresponding body, and the linear relation between the base parameter vector and the physical parameters is not just one.

In addition, the solution of the nonlinear problem does not guarantee that the set of physical parameters found has been identified accurately. Another approach that can be used for parameter identification is to evaluate the physical feasibility after the identification process has been carried out (Yoshida et al., 1996) along with the statistical analysis of variances in the resulting parameters. Hence, two aspects are verified: Base Parameters with $\sigma_{pi} < 15\%$ and physical feasibility. If the parameter accomplishes these criteria, the parameter is considered properly identified.

For example, here, the identifiability of the dynamic parameters of a 3-RPS parallel robot is addressed considering a simulated manipulator whose inertial parameters have been obtained from the CAD models and the friction parameters has been obtained from an indirect parameter identification process performed by the authors (Farhat et al., 2006). Noise was added to the generalized forces as well as the independent generalized coordinates and their time derivatives.

The three models previously introduced in Table (2) were used in the identification process. Friction was identified using symmetrical linear models that include Coulomb and viscous frictions. When the parameters identified by using Model 1 were analyzed, only 4 of the 34 parameters, including friction and rotor and screw inertias, had σ_{pi} lower than 15%, and some of the identified parameters were physically unfeasible. This can be demonstrated in Table (5) (marked by *), where it can be seen the values of parameters of the simulated and the identified models, respectively.

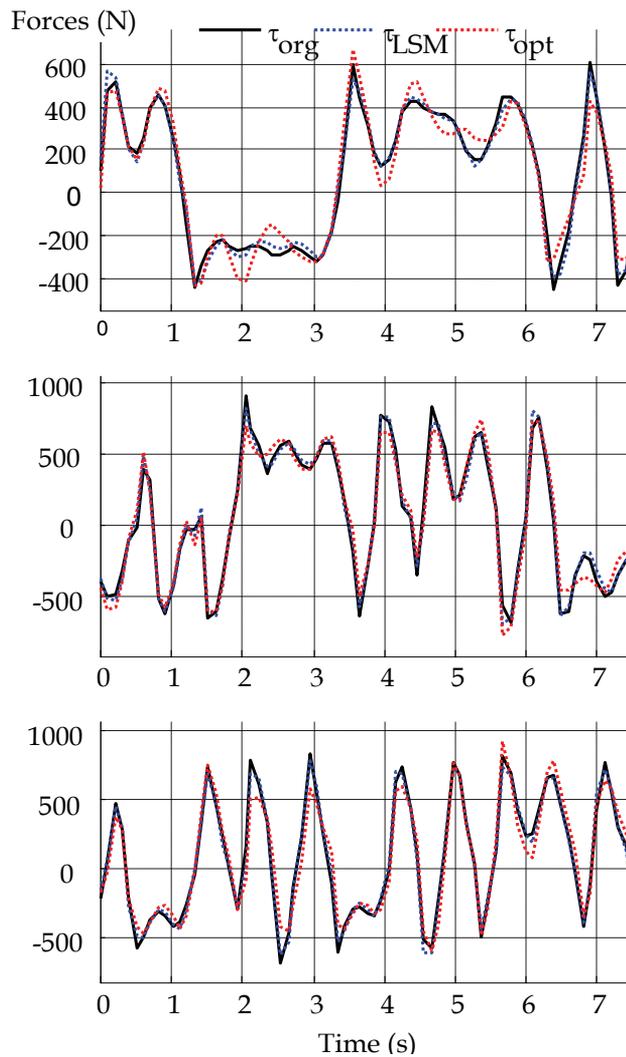


Fig. 3. Results from the LSM (τ_{LSM}) and the optimization process (τ_{opt}).

ε_{RA} (%)		Trajectory condition number					
Friction model		563	638	718	601	492	LowVel
Linear	LSM	12.8	9.48	14.6	17.9	17.9	15.5
	Opt	26.1	20.5	24.4	23.9	23.9	19.2
Nonlin	Op	24.7	21.2	23.5	23.4	23.4	18.0

Table 4. Error comparison considering linear and nonlinear friction models.

Results of the number of parameters properly identified, when model 1-3 was used in the identification process, are listed in Table (6). The table includes also the average relative error of the identified parameter relative to the exact parameter (ε_{AV}),

$$\varepsilon_{AV} = \frac{1}{n_p} \sum \left| \frac{\Phi_i - \hat{\Phi}_i}{\Phi_i} \right| \quad (34)$$

where Φ_i the exact values of the parameters and $\hat{\Phi}_i$ is the vector containing the identified parameters. An interesting fact is that despite that Model 1 achieving the lowest ε_{RA} , the corresponding ε_{AV} value was the highest. In addition, only 4 parameters were properly identified. The difference between models 2 and 3 in ε_{RA} was about 1.5%. As 12 parameters are identifiable, identification was performed using only these parameters (Model 4). This model includes 3 inertial parameters of the links related to the platform and marked by (†) in Table (1).

Parameter	Exact Values	Identified Values	σ_{pi} %
$I_{zz}(4)+I_{yy}(5)$	0.1555	-86.2016*	80.1596
$I_{zz}(6)+I_{yy}(7)$	0.1555	-66.6355*	62.0087
$Fv(1)$	3272.0	3296.73	2.5677
$Fc^+(1)$	227.96	1659.0181	53.3048
$Fc(1)$	228.04	-1210.55*	73.1859
$J_r(1)+J_s(1)$	483.10	505.03	13.8778

Table 5. Some of the base parameters identified using Model 1

Model	ε_{RA} %	ε_{AV} %	Number of Parameters
1	4.57	5.37	37/4
2	4.58	2.94	24/12
3	4.63	3.06	18/12
4	4.73	3.17	12/12

Table 6. Results of ε_{RA} and ε_{AV} from different models.

This result could indicate that, because of the topology of the parallel manipulator and in the presence of measurement noise, 12 parameters from which 3 are of the links inertial parameters can be used for modeling and simulating the 3-RPS parallel manipulator behavior.

Following the same procedure, a dynamic parameters identification process was applied over an actual parallel 3-RPS manipulator. The resulted ε_{RA} values and the number of parameters properly identified are shown in Table (7). Comparing Table (6) and Table (7), the level of ε_{RA} in the actual manipulator was doubled, but the numbers of parameter properly identified was found similar (12 for Model 4). The identified links base parameters of Model 1 and 4 are presented in Table (8) along with those of the simulated manipulator. As can be observed, the identified parameters of the actual manipulator using model 4 and the original CAD values of the simulated manipulator are comparable. Contrary to those identified using Model 1 where a significant difference appears.

The fact that 12 parameters can be properly identified is reasonable. On the one hand the topology itself of the parallel manipulator, does not allow finding well-excited trajectories. Additionally, some base parameters have a little contribution to the dynamic behavior of the

model; for example, during the movement the accelerations of the limbs are smaller than the platform. On the other hand, the friction of the linear actuator of the real manipulator was found to be high, this difficult even more the identifiability of the base parameters of the links.

Finally, Model 4 was validated. Parameters obtained from one trajectory were used to compute the forces for another one that had not been used for identification. Fig. (4) depicts this comparison. As can be seen, the estimated and measurements forces are very close.

Model	ϵ_{RA} %	Number of Parameter
1	8.40	37/2
2	8.43	24/9
3	8.53	18/12
4	8.62	12/12

Table 7. ϵ_{RA} from actual 3-RPS Manipulator.

Base Parameter	CAD	Real Manipulator Model 4	Real Manipulator Model 1
$m_x(3)+0.5774m_y(3) - 0.4563(m(3)+m(2))$	-2.47	-2.59	1.16
$m(5)- 2.531m_y(3)+m(3)+m(2)$	10.83	13.72	-3.29
$m(7)+2.531m_y(3)$	5.42	6.95	-0.557

Table 8. Rigid Body Base Parameters Model 1 vs Model 4.

7. Conclusions and further research

In this chapter, the problem of the identification of inertia and friction parameters for parallel manipulators was addressed. In the first part of the chapter an overview of the identification process applied to parallel manipulators was presented. First, the dynamic model was obtained in a systematic way starting from the Gibbs-Appell equations of motion. This dynamic model was reduced to a subset of parameters called base parameters by means of SVD. After that and to ensure the minimal input/output error transmissibility, approaches to obtaining optimized trajectories that have to be used in the identification process were presented. In the second part of the chapter, a direct identification approach was implemented on a 3-DOF RPS parallel manipulator considering the physical feasibility of the identified inertial parameters. To this end, a procedure based on a nonlinear constrained optimization problem has been reviewed. In addition, nonlinear friction models were included in the dynamic formulation subjacent to the identification process. In the last part of the chapter, a study of the identifiability of the base parameters was presented. It based on both analyzing the relative standard deviation of each parameter and considering its physical feasibility. For this approach a simulated manipulator was necessary for studying and evaluating models used in the identification process. For future research, a systematical approach is expected to be found, based on statistical frameworks and physical feasibility, for studying the identifiability of the dynamic parameter without the necessity of a simulated manipulator. Concepts presented here for parameter identification of parallel

manipulators can be extended to other areas. For instance, vehicle components (Butz et al., 2000; Serban & Freeman, 2001; Chen & Beale, 2003; Sujan & Dubowsky, 2003) and ultimately the novel humanoid systems (Gordon & Hopkins, 1997; Silva et al., 1997; Kraus et al., 2005).

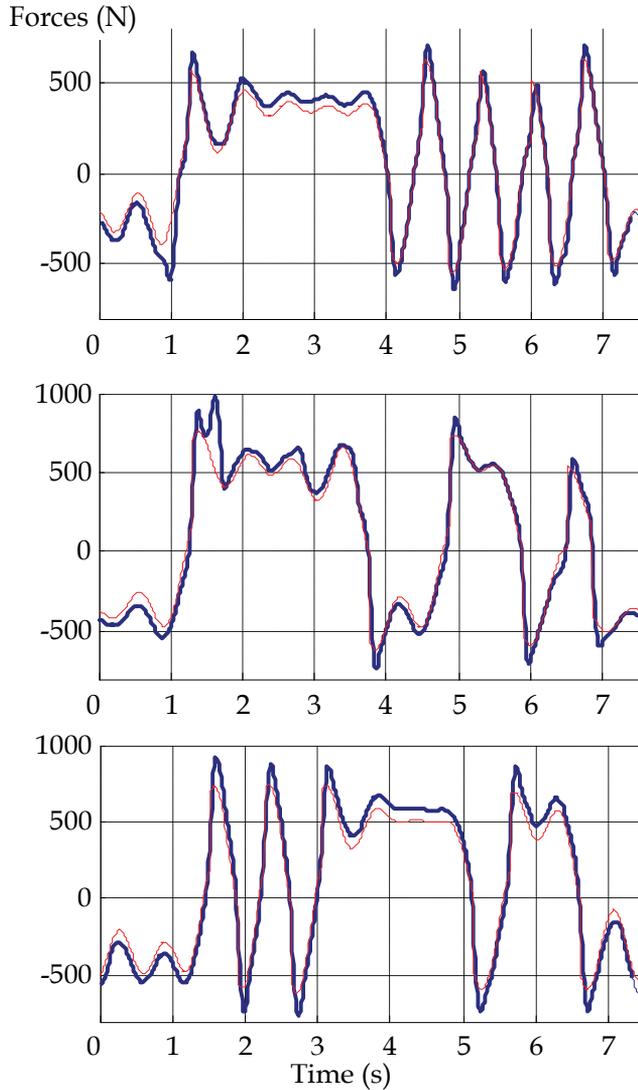


Fig. 4. Measurement Forces (-red) and Forces from Identified Parameter (-o blue)

8. Acknowledgements

This research has been supported by the Spanish Government grants project DPI2005-08732-C02-01 and DPI2006-14722-C02-01, cofinanced by EU FEDER funds. The third author thanks the University of Los Andes Scholarship Programs for helping to finance the junior doctoral studies.

9. References

- Abdellatif H., Grotjahn M. & Heimann B. (2007). Independent Identification of Friction Characteristics for Parallel Manipulators. *Journal of Dynamic Systems, Measurement and Control*, Vol. 127, No. 3, pp. 294-302.
- Armstrong B. (1988). Friction: experimental determination modeling and compensation. *IEEE Int. Conf. on Robotics and Automation*, pp. 1422-1427, Philadelphia, PA.
- Armstrong B. (1989). On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics. *International Journal of Robotics Research*, Vol. 8, No. 6, pp. 28-48.
- Atkeson C. G., An C. H. & Hollerbach J. M. (1986). Estimation of Inertial Parameters of Manipulator Loads and Links. *International Journal of Robotics Research*, Vol. 5, No. 3, pp. 101-119.
- Benimeli F., Mata V., Farhat N. & Valera A. (2003). Experimental Set-Up and Some Results in Parameter Identification in Robots. *Proceedings of RAAD'03, 12th International Workshop on Robotics in Alpe-Adria-Danube Region*, Cassino, Italy.
- Benimeli F., Mata V. & Valero F. (2006). A comparison between direct and indirect dynamic parameter identification methods in industrial robots. *ROBOTICA*, Vol. 24, No. 5, pp. 579 - 590.
- Bhattacharya S., Hatwal H. & Ghosh A. (1997). An on-line parameter estimation scheme for generalized stewart platform type parallel manipulators. *Mechanism And Machine Theory*, Vol. 32, No. 1, pp. 79-89.
- Butz T., Stryk O. V., Vögel M., Wolter T.-M. & Chucholowski C. (2000). Parallel parameter estimation in full motor vehicle dynamics. *SIAM News*, Vol. 33, No. 4,
- Corke P. (1996). In situ Measurement of Robot Motor Electrical Constants. *ROBOTICA*, Vol. 14, No. 4, pp. 433-436.
- Chen K. Y. & Beale D. G. (2003). Base dynamic parameter estimation of a MacPherson suspension mechanism. *Vehicle System Dynamics*, Vol. 39, No. 3, pp. 227-244.
- Diaz-Rodriguez M., Mata V., Farhat N. & Provenzano S. (2007). Identificación de Parámetros Dinámicos de Robots Paralelos: Métodos de obtención de las variables cinemáticas a partir de la medición de la posición. *8º Congreso Iberoamericano de Ingeniería Mecánica*, Federación Iberoamericana de Ingeniería Mecánica, Cusco.
- Farhat N. *Identificación de Parámetros Dinámicos en Sistemas de Cadena Cerrada. Aplicación a Robot Paralelos* (PhD). Valencia, Universidad Politécnica de Valencia, 2006.
- Farhat N., Mata V., Page Á. & Valero F. (2008). Identification of dynamic parameters of a 3-DOF RPS parallel manipulator. *Mechanism and Machine Theory*, Vol. 43, No. 1, pp. 1-17.
- Gautier M. (1991). Numerical Calculation of the Base Inertial Parameters of Robots. *Journal of Robotic Systems*, Vol. 8, No. 4, pp. 485-506.
- Gautier M. & Khalil W. (1988). On the Identification of the Inertial Parameters of Robots. *Proceedings of the 27th Conference on Decision and Control*, pp. 739-755.
- Gautier M. & Khalil W. (1992). Exciting Trajectories for the Identification of Base Inertial Parameters of Robots. *International Journal of Robotics Research*, Vol. 11, No. 4, pp. 362-375.
- Gautier M., Khalil W. & Restrepo P. (1995). Identification of the dynamic parameters of a closed loop robot. *Proceeding of IEEE 33th Conferences on Robotics and Automation*, pp. 3045-2050, Nagoya, Japan.
- Gordon T. J. & Hopkins R. (1997). Parametric Identification of Multibody Models for Crash Victim Simulation. *Multibody System Dynamics*, Vol. 1, No. 1, pp. 112.
- Grotjahn M., Daemi M. & Heimann B. (2001). Friction and rigid body identification of robot dynamics. *International Journal of Solids and Structures*, Vol. 38, No. 10-13, pp. 1889-1902.

- Grotjahn M., Heimann B. & Abdellatif H. (2004). Identification of Friction and Rigid-Body Dynamics of Parallel Kinematic Structures for Model-Based Control. *Multibody System Dynamics*, Vol. 11, No. 3, pp. 273-294.
- Guegan S., Khalil W. & Lemoine P. (2003). Identification of the dynamic parameters of the Orthoglide. *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation, 2003*, pp. 3272- 3277, Ecole Centrale de Nantes, France.
- Ha I. J., Ko M. S. & Kwon S. K. (1989). An Efficient Estimation Algorithm for the Model Parameters of Robotic Manipulators. *Ieee Transactions on Robotics and Automation*, Vol. 5, No. 3, pp. 386-394.
- Hardeman T., Aarts R. G. & Jonker J. B. (2006). A finite element formulation for dynamic parameter identification of robot manipulators. *Multibody System Dynamics*, Vol. 16, No. 1, pp. 21-35.
- Hiller M., Bekes F. & Bertarm T. (2002). Mechatronic Design in Automotive Systems. *International Symposium on Multibody Systems and Mechatronics, Proceedings of MUSME 2002*, Mexico City.
- Horn R. A. & Johnson C. R. (1985). *Matrix Analysis*, The Press Syndicate of the University of Cambridge.
- Khalil W. & Bennis F. (1995). Symbolic Calculation of the Base Inertial Parameters of Closed-Loop Robots. *International Journal of Robotics Research*, Vol. 14, No. 2, pp. 112-128.
- Khalil W., Bennis F. & Gautier M. (1990). The Use of the Generalized Links to Determine the Minimum Inertial Parameters of Robots. *Journal of Robotic Systems*, Vol. 7, No. 2, pp. 225-242.
- Khalil W. & Dombre E. (2002). *Modeling Identification and Control of Robots*, Hermes Penton Ltd. ISBN 1-9039-9613-9, London.
- Khosla P. K. (1989). Categorization of Parameters in the Dynamic Robot Model. *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, pp. 261-268.
- Khosla P. K. & Kanade T. (1985). Parameter Identification of Robot Dynamics. *Proceedings of the 24th IEEE Conference on Decision and Control*, pp. 1754-1760.
- Kozlowski K. (1998). *Modelling and Identification in Robotics*, Springer-Verlag. ISBN 3-540-76240-X, London.
- Kraus C., Bock H.-G. & Mutschler H. (2005). Parameter Estimation for Biomechanical Models Based on a Special Form of Natural Coordinates. *Multibody System Dynamics*, Vol. 13, No. 1, pp. 111.
- Luh J. Y. S., Walker M. W. & Paul R. P. C. (1980). On-line Computational Scheme for Mechanical Manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 102, No., pp. 69-76.
- Mata V., Benimeli F., Farhat N. & Valera A. (2005). Dynamic parameter identification in industrial robots considering physical feasibility. *Journal of Advanced Robotics*, Vol. 19, No. 1, pp. 101-120.
- Mata V., Provenzano S., Cuadrado J. I. & Valero F. (2002). Inverse dynamic problem in robots using Gibbs-Appell equations. *Robotica*, Vol. 20, No. 1, pp. 59-67.
- Olsen H. B. & Bekey G. A. (1986). Identification of Robot Dynamics. *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 1004-1010, San Francisco.
- Olsen M. M. & Petersen H. G. (2001). A new method for estimating parameters of a dynamic robot model. *Ieee Transactions on Robotics and Automation*, Vol. 17, No. 1, pp. 95-100.
- Olsen M. M., Swevers J. & Verdonck W. (2002). Maximum likelihood identification of a dynamic robot model: Implementation issues. *International Journal of Robotics Research*, Vol. 21, No. 2, pp. 89-96.
- Olsson H., Åström K. J., Canudas-De-Wit C., Gäfvert M. & Lischinsky P. (1998). Friction Models and Friction Compensation. *European Journal of Control*, No. 4, pp. 176-195.

- Page A., Candelas P. & Belmar F. (2006). On the use of local fitting techniques for the analysis of physical dynamic systems. *European Journal of Physics*, Vol. 27, No. 2, pp. 273-279.
- Park K.-J. (2006). Fourier-based optimal excitation trajectories for the dynamic identification of robots. *ROBOTICA*, Vol. 24, No. 5, pp. 625-633.
- Presse C. & Gautier M. (1993). New Criteria of Exciting Trajectories for Robot Identification. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 907-912, 0-8186-3450-2, I E E E, COMPUTER SOC PRESS, LOS ALAMITOS, Atlanta.
- Prüfer M., Schmidt C. & Wahl F. (1994). Identification of robot dynamics with differential and integral models: a comparison. *IEEE International Conference on Robotics and Automation*, pp. 340-345, San Diego, CA, USA.
- Renaud P., Vivas A., Andreff N., Poignet P., Martinet P., Pierrot F. & Company O. (1993). Kinematic and dynamic identification of parallel mechanisms. *Control engineering practice*, Vol. 14, No. 9, pp. 1099-1109.
- Schittkowski K. NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems. In: *Annals of Operations Research* (1.7 ed.), 2000.
- Serban R. & Freeman J. S. (2001). Identification and identifiability of unknown parameters in multibody dynamic systems. *Multibody System Dynamics*, Vol. 5, No. 4, pp. 335-350.
- Sheu S. Y. & Walker M. W. (1989). Estimating the essential parameter space of the robot manipulator dynamics. *Proceedings of the 28th IEEE Conference on Decision and Control*, pp. 2135-2140, Tampa, FL, USA.
- Sheu S. Y. & Walker M. W. (1991). Identifying the Independent Inertial Parameter Space of Robot Manipulators. *International Journal of Robotics Research*, Vol. 10, No. 6, pp. 668-683.
- Silva M. P. T., Ambrósio J. A. C. & Pereira M. S. (1997). Biomechanical Model with Joint Resistance for Impact Simulation. *Multibody System Dynamics*, Vol. 1, No. 1, pp. 84.
- Sujan V. A. & Dubowsky S. (2003). An optimal information method for mobile manipulator dynamic parameter identification. *Ieee-Asme Transactions On Mechatronics*, Vol. 8, No. 2, pp. 215-225.
- Swevers J., Ganseman C., Deschutter J. & Vanbrussel H. (1996). Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, Vol. 10, No. 5, pp. 561-577.
- Swevers J., Ganseman C., Tukul D. B., Deschutter J. & Vanbrussel H. (1997). Optimal robot excitation and identification. *Ieee Transactions on Robotics and Automation*, Vol. 13, No. 5, pp. 730-740.
- Swevers J., Verdonck W., Naumer B., Pieters S. & Biber E. (2002). An experimental robot load identification method for industrial application. Vol. 21, No. 8, pp. 701-712.
- Udwadia F. & Kalaba R. (1998). The Explicit Gibbs-Appell Equation and Generalized Inverse Forms. *Quarterly of Applied Mathematics*, Vol. 56, No. 2, pp. 277-288.
- Yoshida K. & Khalil W. (2000). Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters. *International Journal of Robotics Research*, Vol. 19, No. 5, pp. 498-510.
- Yoshida K., Mayeda H. & Ono T. (1996). Base parameters for manipulators with a planar parallelogram link mechanism. Vol. 10, No. 1, pp. 105-137.

Quantifying and Optimizing Failure Tolerance of a Class of Parallel Manipulators

Chinmay S. Ukidve, John E. McInroy and Farhad Jafari
University of Wyoming, Laramie.
USA

1. Introduction

For any robotic system, fault tolerance is a desirable property. This work uses a comparative approach to investigate fault tolerance and the associated problem of reduced manipulability of robots. An important result in combinatorial matrix theory is first obtained. Its consequent modifications are then applied to the theory of fault tolerance of robotic manipulators.

It is shown that for a certain class of parallel manipulators, the mean squared relative manipulability over all possible cases of a given number of actuator failures is always constant irrespective of the geometry of the manipulator. A theorem formulates the value of the mean squared relative manipulability. It is shown that this value depends only upon the number of simultaneous joint failures, the nominal number of joint degrees of freedom and the nominal task degrees of freedom. It is difficult to predict specific failures at the design stage and as such failure of any actuator is considered equally likely. In this context, optimal fault tolerant manipulability is quantified. The theory is applied to a special class of parallel manipulators called Orthogonal Gough-Stewart Platforms (Orthogonal GSPs or OGSPs). A class of two-group symmetric OGSPs which inherently provide for optimal fault tolerant manipulability under a single failure is developed.

2. Background

Robotic manipulators have become popular in numerous applications. They have been employed in automation of industrial processes, underwater exploration, space exploration and innovative defence technologies. The nature of some of these applications makes human presence near manipulators difficult and in many cases, impossible. This is especially true for robots employed in remote and hazardous environments. The repair and maintenance tasks for such robots are extremely difficult. In such cases, operational reliability is of prime importance. Therefore, it is imperative to incorporate failure tolerance in system design. Under the occurrence of failures, fault tolerance enables the robotic system to maintain critical functioning with a reduced level of performance.

Current research efforts are focussed on developing techniques for designing fault tolerant manipulators and robotic vision systems.

Redundant manipulators are rapidly becoming a focus of research due to a multitude of potential advantages they provide. In serial robots, kinematic redundancy has been

employed for obstacle avoidance (Baillieul, 1990), dexterity optimization (Lewis & Maciejewski, 1992) and torque minimization (Hollerbach & Suh, 1987). In parallel manipulators, singularity avoidance (Kim et al., 2004) and stiffness improvement (Kock & Schumacher, 1998) are broad areas where kinematic redundancy has proven useful. Another significant attribute of redundancy that has come under recent investigation is fault tolerance.

Kinematic failures commonly occur in manipulators. The effect of such failures on manipulator performance depends upon the nature of failure and the nominal kinematic design of the manipulator. For instance, loss of an actuator can render a serial manipulator completely unmanipulable. On the other hand, parallel manipulators can be designed to retain kinematic stability under loss of actuators. Locking of actuators is another commonly observed failure phenomenon. In any failure scenario a robotic system loses partial or complete manipulability. Fault tolerance and the consequent problem of reduced manipulability have been studied by a number of researchers.

Maciejewski (Maciejewski, 1990) associates the concept of fault tolerance to manipulator configuration. Dexterity index is used as a measure of fault tolerance. Optimal fault tolerant configurations are defined using this measure.

Roberts and Maciejewski (Roberts & Maciejewski, 1996) propose a local measure to quantify fault tolerance of a manipulator pose in terms of a manipulability index. Their approach uses the singular value decomposition of the manipulator Jacobian matrix. They describe a direct relation between relative manipulability and the null-space of the Jacobian matrix. They propose relative manipulability index as a measure of fault tolerance.

Paredis, Au and Khosla (Paredis, Au & Khosla, 1994) consider fault tolerance with respect to manipulator workspace and reach. They define fault tolerant workspace of a manipulator and suggest task based design of manipulators. Their approach uses iterative techniques to design manipulators.

Ting, Tosunoglu and Tesar (Ting, Tosunoglu & Tesar, 1993) explore control algorithms for fault tolerant operation of manipulators.

McInroy, O'Brien and Neat (McInroy, O'Brien & Neat, 1999) propose a fault tolerant precision pointing strategy using a class of parallel manipulators called Gough-Stewart Platforms (GSPs) (Stewart, 1966). A GSP is used as a pointing platform to reject vibrations from a noisy spacecraft bus over all frequencies. At low frequencies two-axis or three-axis pointing method is used, while at high frequencies six-axis vibration isolation is employed. The benefits of this approach include broadband pointing stability without a high-bandwidth pointing sensor or destabilizing excitation of the high frequency structural modes. To incorporate fault tolerance, they propose a reconfiguration algorithm to compute a decoupling matrix which allows motion in 'off degrees of freedom' to compensate for failures.

This work uses a comparative approach to investigate fault tolerance and the associated problem of reduced manipulability. Following is a description of the main contributions.

An important result in combinatorial matrix theory is first obtained. Its consequent modifications are then applied to the theory of fault tolerance of robotic manipulators.

It is shown that for a certain class of parallel manipulators, the mean squared relative manipulability over all possible cases of a given number of actuator failures is always constant irrespective of the geometry of the manipulators. This work uses the

manipulability index suggested by Yoshikawa (Yoshikawa, 1985) and the resulting relative manipulability indices proposed by Roberts and Maciejewski (Roberts & Maciejewski, 1996). A theorem formulates the value of the mean squared relative manipulability. It is shown that this constant depends only upon the number of simultaneous failures, the nominal number of joint degrees of freedom and the nominal task degrees of freedom.

It is difficult to predict specific failures at the design stage and as such failure of any actuator is considered equally likely. From this perspective optimal fault tolerant manipulability for a given number of faults has been defined by Roberts and Maciejewski in (Roberts & Maciejewski, 1996). This work quantifies optimal fault tolerant manipulability based only upon the number of simultaneous failures, the nominal number of joint degrees of freedom and the nominal task degrees of freedom.

For parallel manipulators employed in micromanipulation, the workspace is very small. For such manipulators, the definition of optimal fault tolerant manipulability for a given number of faults carries a different interpretation. Such manipulators can be assumed to operate only at a particular pose and therefore the the same definition can be applied to the manipulator in general rather than to a specific pose. As an illustration, the concept of optimal fault tolerant manipulability is applied to a special class of parallel manipulators called Orthogonal Gough-Stewart Platforms (Orthogonal GSPs or OGSPs). This work develops a class of two-group symmetric OGSPs (McInroy & Jafari, 2006) which inherently provide for optimal fault tolerant manipulability under a single failure.

3. Quantifying optimal fault tolerant manipulability

3.1 Manipulability index

In the robotics standard, the Jacobian matrix mapping joint velocities, $\dot{\theta}$, to generalized end effector velocities, V , is denoted by J .

$$V = J\dot{\theta}. \quad (1)$$

A number of researchers have proposed different measures that quantify the manipulability of a manipulator. One such manipulability index, based on a matrix determinant, was proposed by Yoshikawa (Yoshikawa, 1985):

$$w(J) = \sqrt{\det(JJ^T)}. \quad (2)$$

Failures in manipulators can occur in various ways. In this work, only mechanical failures that cause a manipulator to lose an actuator are considered. The impact of such failures varies with different classes of parallel manipulators. In a wide class of parallel mechanisms, under a joint failure, the resulting manipulator Jacobian matrix, referred to as reduced Jacobian matrix, is given by the original Jacobian matrix except that the column corresponding to the failed joint is removed. Gough Stewart Platforms are a classic example of manipulators belonging to this class. On the contrary, in multi-fingered grasps, the impact of failures on the kinematic representation is a function of composite manipulability Jacobian matrix, described in (Wen & Wilfinger, 1999) and cannot be directly derived by elimination of rows or columns.

The mechanisms treated in this work belong to the former class, which will be characterized by $J \in \Omega$. Consider a nominal Jacobian matrix, J , with n actuators. A reduced Jacobian

matrix with i simultaneous actuator failures will be denoted by ${}_i J$. Note that this ${}_i J$ is not unique since there may be multiple ways in which i struts may fail. In order to identify all reduced Jacobian matrices uniquely, a subscript j will be used. Therefore, ${}_i J_j$, $j \in \{1, 2, \dots, {}^n C_i\}$, will describe the alternative i strut failure schemes. This representation will be more clear from the following example.

Suppose J denotes a manipulator with 3 actuators. Then ${}_1 J$ denotes a reduced Jacobian matrix with 1 actuator failure. There are ${}^3 C_1$ ways in which 1 actuator can fail at a time from 3 actuators. Here, ${}^n C_s$ denotes the usual combinatorial notation. Therefore, ${}_1 J_j$ with $j \in \{1, 2, 3\}$ completely represents all reduced Jacobian matrices.

To analyze the post fault performance of a manipulator the relative manipulability index, ${}_i r_j$, will be used:

$${}_i r_j = \sqrt{\frac{\det({}_i J_j {}_i J_j^T)}{\det(JJ^T)}}. \quad (3)$$

Clearly this index is normalized and the scaled translational and rotational components of the manipulator Jacobian matrix do not affect this value.

3.2 Optimal fault tolerant manipulability

For serial manipulators, (Roberts & Maciejewski, 1996) define optimally fault tolerant configurations have been defined as those J in which the relative manipulability index ${}_i r_j$ remains constant over all possible j , for a given i . A rigorous method is provided in (Roberts & Maciejewski, 1996) to calculate ${}_i r_j$ over all j for a given i . However, that does not allow direct determination of optimal ${}_i r_j$ over all j for a given i . The following theorem shows that for a given n and i the sum of squares of ${}_i r_j$ over all j is invariant. Consequently, a formulation is developed that determines optimal ${}_i r_j$ over all j (Ukidve, McNroy & Jafari, 2006).

Theorem 1. Let J be an $m \times n$ ($n > m$) Jacobian matrix representing the operating configuration of any manipulator having n actuators such that $J \in \Omega$. Then the mean squared relative manipulability over all possible failures j given that i ($i \leq (n - m)$) actuators fail at a time is constant and is given by

$$\sum_{j=1}^{{}^n C_i} \frac{{}_i r_j^2}{{}^n C_i} = \frac{1}{{}^n C_i} \sum_{j=1}^{{}^n C_i} \frac{\det({}_i J_j {}_i J_j^T)}{\det(JJ^T)} = \frac{{}^{(n-i)} C_m}{{}^n C_m}. \quad (4)$$

Proof: Note: Following identities in combinatorics are used in the proof.

$${}_f C_g = \frac{f!}{g!(f-g)!}$$

$${}_f C_g = {}_f C_{(f-g)}$$

Case 1: $i = (n - m)$.

Note

$$\sum_{j=1}^n r_j^2 = \sum_{j=1}^n r_j^2 = \sum_{j=1}^n r_j^2 = 1. \quad (5)$$

This equality follows from the Binet-Cauchy Theorem, because for $i = (n - m)$

$$\det(JJ^T) = \sum_{j=1}^n \det({}_i J_j J_j^T). \quad (6)$$

Case 2: $i < (n - m)$.

Let $c_k \in \mathbb{R}^n$, denote the k^{th} column of J ($k = 1$ to n). Then,

$$J = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix},$$

and

$$JJ^T = c_1 c_1^T + c_2 c_2^T + \dots + c_n c_n^T. \quad (7)$$

Let $(n - m) = t$.

By Binet-Cauchy Theorem,

$$\det(JJ^T) = \sum_{j=1}^n \det({}_t J_j J_j^T). \quad (8)$$

Suppose i actuators fail at a time from n actuators ($i < t$). This leads to a reduced Jacobian matrix ${}_i J_j$ with $j \in \{1, 2, \dots, {}^n C_i\}$. Without loss of generality, we may assume that the actuators corresponding to last i columns fail. This says that all the reduced Jacobian matrices can be completely expressed by ${}_i J_j$ with $j \in \{1, 2, \dots, {}^{n-i} C_t\}$. Note that although this loss of generality argument is not applicable to the reduced Jacobian matrix, equation (7) makes it clear that the argument is valid for the product of the reduced Jacobian matrix and its transpose. The number of actuators remaining is $(n - i)$. The number of ways in which the failures can happen at a time are ${}^n C_i$. In other words, ${}^n C_i = {}^n C_{(n-i)}$ is the number of ways in which $(n - i)$ un-failed actuators can be chosen from n actuators. So, there will be ${}^n C_i$ such reduced Jacobian matrices, ${}_i J_j$, where $j \in \{1, 2, \dots, {}^n C_i\}$.

Apply the Binet-Cauchy Theorem to a representative reduced Jacobian matrix ${}_i J_j$, denoted by ${}_i J_j^1$.

Then, we have

$$\det({}_i J_j^1 J_j^{1T}) = \sum_{j=1}^{(n-i)} \det({}_i J_j^1 J_j^{1T}). \quad (9)$$

Similarly,

$$\det({}_i J_{j_i}^2 J_{j_i}^{2T}) = \sum_{j=1}^{(n-i)C_m} \det({}_i J_{j_i}^2 J_{j_i}^{2T}). \quad (10)$$

In general,

$$\det({}_i J_j^n C_i J_j^n C_i^T) = \sum_{j=1}^{(n-i)C_m} \det({}_i J_j^n C_i J_j^n C_i^T). \quad (11)$$

Note that all of the ${}^{(n-i)}C_m$ terms appearing on the R.H.S. of equation (9) are exactly those terms on the R.H.S. of equation (8) which do not have the eliminated i columns in their submatrices, ${}_i J_j$. Moreover, each equation from ((9) - (11)) has one term on the L.H.S and ${}^{(n-i)}C_m$ terms on the R.H.S and there are ${}^n C_i$ such equations.

Using equations ((9) - (11)) to add all possible ${}^n C_i$ reduced Jacobian matrices ${}_i J_j^1, {}_i J_j^2, \dots, {}_i J_j^n C_i$ i.e. taking the summation of all possible ${}^n C_i$ reduced Jacobian matrices,

$$\begin{aligned} & \det({}_i J_{j_i}^1 J_{j_i}^{1T}) + \det({}_i J_{j_i}^2 J_{j_i}^{2T}) + \dots + \det({}_i J_j^n C_i J_j^n C_i^T) \\ &= \sum_{j=1}^{(n-i)C_m} \det({}_i J_{j_i}^1 J_{j_i}^{1T}) + \sum_{j=1}^{(n-i)C_m} \det({}_i J_{j_i}^2 J_{j_i}^{2T}) + \dots \\ & \dots + \sum_{j=1}^{(n-i)C_m} \det({}_i J_j^n C_i J_j^n C_i^T). \end{aligned} \quad (12)$$

Now comparing equation (8) with equation(12), each term in the R.H.S. of (12) is a term on the R.H.S. of (8). Choose a particular term on the R.H.S. of (8), for example $\det({}_i J_i J_i^T)$. This term will occur in only those $\det({}_i J_j^p J_j^{pT})$ ($p=1$ to ${}^n C_i$) for which ${}_i J_j^p$ contains exactly those n columns in ${}_i J_i$. For n given columns, the number of columns left to choose for ${}_i J_j$; with i columns eliminated is $(n-m)$; from which we choose the remaining $(n-m-i)$ columns. Therefore, the number of occurrences for $\det({}_i J_i J_i^T)$ will be ${}^{(n-m)}C_{(n-m-i)}$.

This is true for each term on the R.H.S. of(8). Furthermore, we also know that each term from the ${}^{(n-i)}C_m$ terms appearing on the R.H.S. of ${}^n C_i$ equations ((9) - (11)) occurs in the R.H.S. of equation (8).

Dividing equation (12) by (8), we have

$$\frac{\det({}_i J_{j_i}^1 J_{j_i}^{1T})}{\det(JJ^T)} + \frac{\det({}_i J_{j_i}^2 J_{j_i}^{2T})}{\det(JJ^T)} + \dots + \frac{\det({}_i J_j^n C_i J_j^n C_i^T)}{\det(JJ^T)} = {}^{(n-m)}C_{(n-m-i)}. \quad (13)$$

Therefore,

$$\sum_{j=1}^n r_j^2 = \sum_{j=1}^n \frac{\det({}_i J_{j_i} J_{j_i}^T)}{\det(JJ^T)} = {}^{(n-m)}C_{(n-m-i)}. \quad (14)$$

Dividing both sides by ${}^n C_i$ to take the mean and noting that,

$${}^{(n-m)}C_{(n-m-i)} = \frac{{}^{(n-i)}C_m^n}{{}^n C_m} C_i, \quad (15)$$

the result follows. #

This proof leads to the definition of optimal fault tolerant manipulability.

Definition: A manipulator operating about a single point in the workspace is said to be optimally fault tolerant for a given number of failures i if for all $j \in \{1, 2, \dots, C_i\}$

$${}_i r_j = c \quad (16)$$

where c is a constant.

It is clear from Theorem 1 that if post-fault relative manipulability for certain cases of failure are higher than the optimal fault tolerant manipulability value, then for other worst cases of failure, post-fault relative manipulabilities are extremely low. This has precisely been the motivation for developing optimally fault tolerant manipulators and the above definition arises as a direct consequence.

Corollary 2. A manipulator characterized by $J \in \Omega$ and operating about a single point in the workspace is optimally fault tolerant to i faults if

$${}_i r_j = \sqrt{\frac{{}^{(n-i)}C_m^n}{{}^n C_m}} \quad (17)$$

for all $j \in \{1, 2, \dots, C_i\}$.

Proof: Equation (16) defines manipulators with optimal fault tolerant manipulability for i faults. By Theorem 1, if each ${}_i r_j$ is constant, c is given by equation (1) #

Roberts and Maciejewski (Roberts & Maciejewski, 1996) present a singular value decomposition approach to identify fault tolerant configurations for serial manipulators and describes a rigorous method to determine whether a given nominal configuration possesses optimal fault tolerant manipulability. The above theorem states a formulation which directly gives the value of optimal manipulability under a given number of failures, for any manipulator with a given number of actuators. In fact, the theorem proves that irrespective of the operating configuration, all manipulators having the same number of actuators, have the same value of optimal manipulability under a given number of failures. This new idea plays a key role in the design of fault tolerant serial manipulators. The following example illustrates this point.

Suppose a fault tolerant serial manipulator is to be designed such that it has 3 degrees of freedom ($m=3$) and it is desired to have an optimal fault tolerant manipulability of 0.5 ($r_j=0.5$) under a single-actuator failure ($i=1$). This implies that the manipulator operating in the nominal configuration, should be able to sustain the failure of any actuator and retain half of its manipulability. Substituting all known values in equation (1), we get $n=4$. This means that the manipulator has to have 4 actuators in order to have optimal fault tolerant manipulability for single failure.

Another implication of this theorem is significant in terms of understanding post-fault behavior of any manipulator. From equation (14), it is clear that if the value of ${}_i r_j$ is quite close to $\frac{{}^{(n-m)}C_m}{{}^{(n-m-i)}C_m}$ for some j then the reduction in manipulability is far more pronounced if a possible failure combination corresponding to some other j were to occur. Taking this into account, it is possible to assemble the actuators in such a way that actuators which are more likely to fail (for example, actuators with actuators that are more likely to have manufacturing defects) correspond to those j which give

$${}_i r_j > \sqrt{\frac{{}^{(n-i)}C_m}{{}^n C_m}}.$$

This idea has a greater impact on the design of parallel manipulators. While actuator failures may cause a serial manipulator to stop functioning, actuator failures have a comparatively smaller effect on redundant parallel manipulators because they can retain kinematic stability. Therefore, the two consequences that can be applied to the design of serial manipulators are applicable to parallel manipulators as well.

The most significant area of investigation where the above results influence parallel manipulator design is the choice of geometry. This area will be explored in the Section 5.

3.3 Examples

Some specific examples provide more insight to understanding this concept of optimal fault tolerant manipulability.

Number of Nominal Actuators n	Number of Failures i	Sum of all possible ${}_i r_j^2$	Optimal Fault Tolerant Manipulability
7	1	1	0.377
8	1	2	0.500
8	2	2	0.189
9	1	3	0.577
9	2	3	0.288
9	3	3	0.109

Table 1. Optimal Fault-tolerant manipulability of 6-dof redundant manipulators

It is clear from Table 1 that as the number of failures increases, the optimal fault tolerant manipulability decreases drastically, regardless of the geometry. Consider the example of any 8-actuator manipulator suffering from 2 simultaneous failures. The sum of squares of relative manipulabilities is 2 for 28 failure possibilities. This means that if some post-fault relative manipulabilities are more than the optimal fault tolerant manipulability (0.189), then the worst case values are far less than 0.189. Therefore, irrespective of the geometry of the 8-actuator manipulator, for worst cases of failure the relative manipulabilities will have negligible values. Hence, it is important to design manipulators that are optimally fault tolerant to a given number of failures.

Table 1 also provides another important inference which is significant from the design perspective. Any redundant manipulator gives very low optimal fault tolerant manipulability values for more than one failures, and these values decrease drastically with number of failures. For example, for two failures in an octopod the optimal fault tolerant manipulability is 0.189 and, for two and three failures in a nanopod the optimal fault tolerant manipulabilities are 0.288 and 0.109 respectively. This means that under the hypothesis of equal probability of failure for each actuator, it is not practical to design manipulators optimally fault tolerant to more than one fault.

4. Symmetric orthogonal Gough Stewart platforms

4.1 Gough Stewart platforms

A Gough-Stewart Platform (GSP) is a parallel manipulator consisting of a base, a moving platform (or payload) and struts. The length of struts is controlled by actuators. The struts have spherical joints at the payload end and U joints at the base. To provide six degrees of freedom, six struts are commonly used. Figure 1 is a diagrammatic representation of a GSP. Payload attachment points and base attachment points are represented by p_i and q_i ($i \in \{1,2,3,4,5,6\}$) respectively.

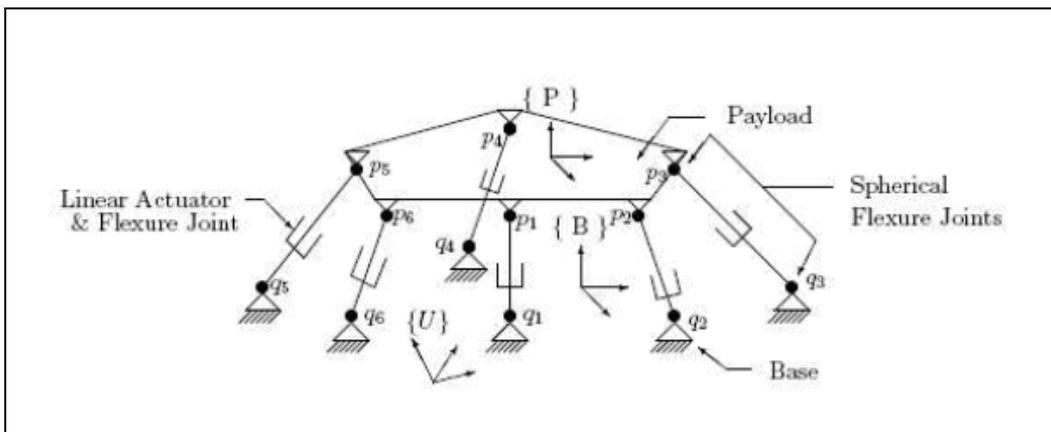


Fig. 1. Gough-Stewart Platform

OGSPs are a special class of GSPs that provide kinematic and dynamic decoupled control. Therefore, OGSPs are being widely used in commercial, military and space applications. Scientists at Northrop Grumman Space Technologies (NGST) are currently experimenting with an 8-strut OGSP. More recent applications of OGSPs include laser tracking and pointing, ultra-precise manipulation (McInroy & Jafari, 2006) and robotic surgery (Wapler et al., 2003). The very nature of these applications makes maintenance or repair of manipulators very difficult. Moreover, a single failure may compromise the fulfilment of objective or cause costly downtime. As a consequence, it is desirable to design OGSPs which can sustain failures, while retaining an acceptable level of manipulability. Figure 2 shows one of the flexure jointed hexapods at the University of Wyoming. It has a mutually orthogonal geometry.

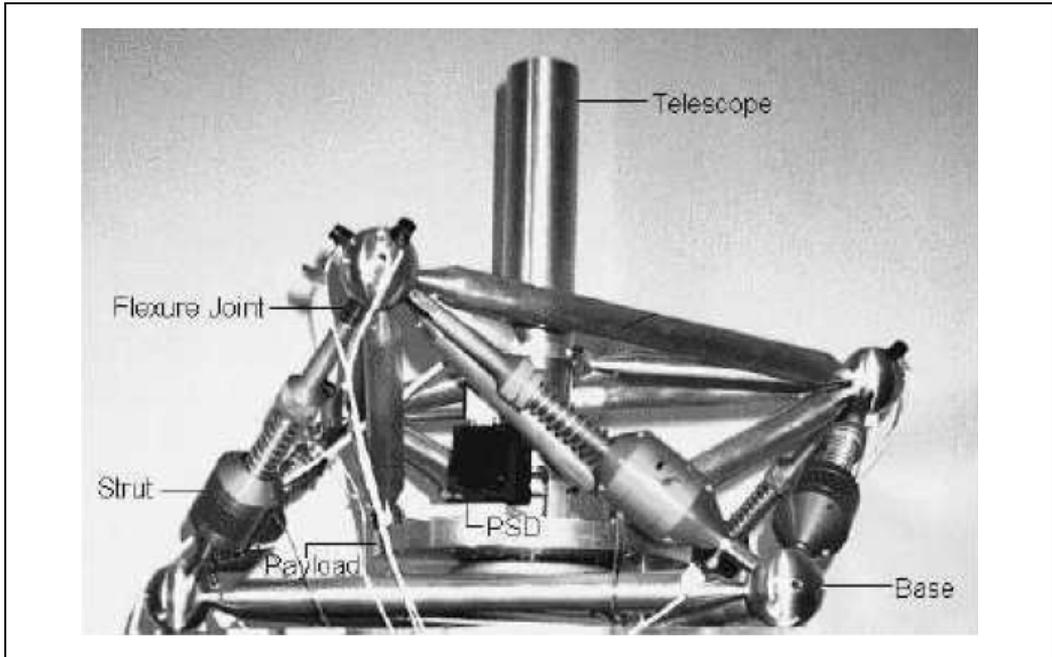


Fig. 2. A Flexure Jointed Hexapod at the University of Wyoming

Recent research has shown that symmetric groups of struts can be used to generate OGSPs having desired properties at their home position (McInroy & Jafari, 2006) and several new results have been obtained.

The following part of this section recapitulates important results from (McInroy & Jafari, 2006).

4.2 Kinematics of symmetric OGSPs

The inverse Jacobian, M , of a GSP maps the generalized velocity of the payload to the corresponding joint velocities of each strut ($\dot{\theta} = MV$). It has the form:

$$M = \begin{bmatrix} \bar{u}_1^T & \bar{v}_1^T \\ \vdots & \vdots \\ \bar{u}_l^T & \bar{v}_l^T \end{bmatrix} \quad (18)$$

where $\bar{u}_i, \bar{v}_i \in \mathbb{R}^3$, $\bar{v}_i = \bar{p}_i \times \bar{u}_i$. \bar{u}_i is the unit vector along strut i and $\bar{p}_i \in \mathbb{R}^3$ is the moving platform attachment point of strut i . Please refer to Figure 1. Note that, even though M is called the inverse Jacobian to comply with the robotics standard, its computation does not require inversion, thus it is well defined for all GSP.

Definitions: Let $M \in M_{l \times 6}(\mathbb{R})$. Write

$$M = \begin{bmatrix} U^T & V^T \end{bmatrix}$$

where $U, V \in M_{3 \times l}(\mathbb{R})$. We say $M \in \text{GSP}$, M is a Gough-Stewart Platform, if:

- $\text{diag}(U^T U) = [1 \ 1 \ \dots \ 1]$
- $\text{diag}(U^T V) = 0$

We say M is a Weighted Orthogonal Gough-Stewart Platform, $M \in \text{w-OGSP}$, if $M \in \text{GSP}$ and:

- $M^T K M$ is a diagonal matrix for a diagonal K .

Where $K = I$ these matrices become the Orthogonal Gough-Stewart Platforms.

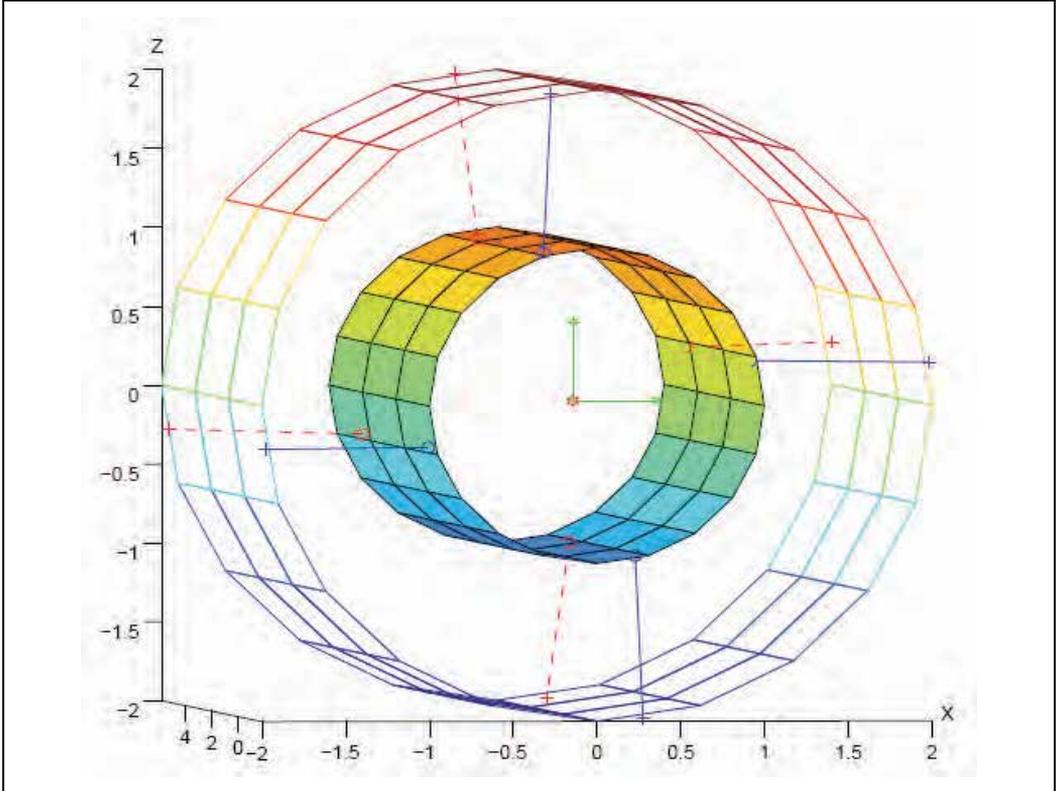


Fig. 3. [4 4] cylindrical OGSP with optimal fault tolerant manipulability

(McInroy & Jafari, 2006) develops properties and designs of symmetrical weighted OGSPs. Struts that are geometrically symmetrical are treated together, so the entire OGSP is decomposed into m different groups, with the i^{th} group having n_i struts. Then

$$\bar{n} = [n_1 \quad n_2 \quad \dots \quad n_m]^T$$

is a vector of positive integers describing the number of struts in each group. The total number of struts in the GSP is then $l = \sum_{i=1}^m n_i$. Let $\bar{u}_j, \bar{v}_j \in \mathbb{R}^3$ correspond to the i^{th} strut in group j . Let $U = [\bar{u}_{11} \ \bar{u}_{21} \ \dots \ \bar{u}_{n_1} \ \bar{u}_{12} \ \dots \ \bar{u}_{n_m}]$ and $V = [\bar{v}_{11} \ \bar{v}_{21} \ \dots \ \bar{v}_{n_1} \ \bar{v}_{12} \ \dots \ \bar{v}_{n_m}]$. A GSP can then be found for these struts by letting $M = [U^T \quad V^T]$.

Following is the summary of results in (McInroy & Jafari, 2006).

Proposition 3. Conditions (a) and (b) in the GSP definition are satisfied if

$$\vec{u}_{ij} = \begin{bmatrix} S_{\phi_{ij}} C_{\beta_{ij}} \\ S_{\phi_{ij}} S_{\beta_{ij}} \\ C_{\phi_{ij}} \end{bmatrix}, \vec{v}_{ij} = x_{ij} \vec{v}_{x_{ij}} + y_{ij} \vec{v}_{y_{ij}} \quad (19)$$

where $S_x = \sin x$, $C_x = \cos x$, $\phi_j, \beta_j, x_j, y_j \in \mathbb{R}$, and

$$\vec{v}_{x_{ij}} = \begin{bmatrix} S_{\beta_{ij}} \\ -C_{\beta_{ij}} \\ 0 \end{bmatrix}, \vec{v}_{y_{ij}} = \begin{bmatrix} C_{\phi_{ij}} C_{\beta_{ij}} \\ C_{\phi_{ij}} S_{\beta_{ij}} \\ -S_{\phi_{ij}} \end{bmatrix}. \quad (20)$$

Conversely, if $M \in \text{GSP}$, then M may be represented by a parameterization given by (19) and (20).

Theorem 4. Let all groups contain more than two struts, i.e. $\min_j n_j > 2$. Then $M \in \text{w-OGSP}$ if

- The same angle, ϕ_j , is used for all struts in group j , i.e. $\phi_i = \phi_j$,
- The same x component of \vec{v} , x_j , is used for all struts in group j , i.e. $x_i = x_j$,
- The same y component of \vec{v} , y_j , is used for all struts in group j , i.e. $y_i = y_j$,
- The same k , k_j , is used for all struts in group j , i.e. $k_i = k_j$,
- Struts in a group are rotated about the z-axis equal amounts, i.e. $\beta_i = \beta_j + \frac{2\pi(i-1)}{n_j}$,
- $A_x \vec{x} = 0$ and $A_y \vec{y} = 0$,

where

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \vec{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{bmatrix},$$

$$\vec{k} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix}, \vec{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix},$$

$$A_x = [k_1 n_1 S_{\phi_1} k_2 n_2 S_{\phi_2} \cdots k_m n_m S_{\phi_m}], \quad (21)$$

$$A_y = [k_1 n_1 S_{2\phi_1} k_2 n_2 S_{2\phi_2} \cdots k_m n_m S_{2\phi_m}]. \quad (22)$$

$\phi, \beta, k, \in \mathbf{R}$ may be freely chosen. \bar{x} and $\bar{y} \in \mathbf{R}^n$ may be freely chosen to satisfy (F). Furthermore, if σ_i^* denotes the i^* diagonal element of $M^T K M$, then

$$\sigma_1^2 = \sigma_2^2 = \frac{1}{2} \sum_{j=1}^m n_j k_j S_{\phi_j}^2, \quad (23)$$

$$\sigma_3^2 = \sum_{j=1}^m n_j k_j - 2\sigma_1^2, \quad (24)$$

$$\sigma_4^2 = \sigma_5^2 = \frac{1}{2} \sum_{j=1}^m n_j k_j (x_j^2 + y_j^2 C_{\phi_j}^2), \quad (25)$$

$$\sigma_6^2 = \sum_{j=1}^m n_j k_j y_j^2 S_{\phi_j}^2. \quad (26)$$

In (Aphale, 2006) robust fault tolerance is defined as the property by which the rank of M equals 6 or the number of struts remaining after failures, whichever is minimum. Not all geometric designs of OGSPs are robustly fault tolerant. In fact, it has been proved that [3 3 2] geometry gives the only robustly fault tolerant design for 8- strut (octopod) OGSPs. This means that [3 3 2] geometry is the only one wherein, if any two struts fail, the rank of M remains 6. While robust fault tolerance guarantees motion in 6 degrees of freedom for a n - strut platform under any $n - m$ failures ($m \leq (n - 6)$), experiments made on the University of Wyoming octopod clearly show that robustly fault tolerant designs suffer from serious post-fault stability problems due to poor conditioning. On the other hand, in many cases the design specifications may require a single failure tolerant architecture. For instance, in a typical case, it would be better to design an 8-strut OGSP which gives an optimal fault tolerant manipulability of 0.5 for a single failure, instead of designing a robustly fault tolerant 8-strut OGSP. This argument will be clearer from the example explained in the next section where a class of symmetric OGSPs having optimal fault tolerant manipulability is proposed.

5. Fault tolerant Gough Stewart platforms

5.1 Design

For parallel manipulators, the problem of inverse kinematics is easier to solve. Therefore, in most literature on parallel manipulators, the inverse Jacobian, M , is used for study.

Remark: In this work, it is assumed that the Jacobian relating joint and Cartesian motion is constant. This is equivalent to considering that the operation is about a single point, rather than across a workspace. The rationale for making this assumption is that there are several high precision OGSP applications which demand operation over a very small workspace. These include high precision motion control for telescopes, scanning microscopes, integrated circuit fabrication, stiffness, precision pointing and vibration isolation.

As mentioned in Section 3, [4 4] redundant OGSPs are currently under investigation by a number of researchers. This section develops a more general class of symmetric OGSPs with optimal fault tolerant manipulability under one fault.

A key characteristic of symmetric OGSPs is rotational invariance. Rotational invariance of groups of struts can be clearly understood with the help of Figure 3, Figure 4 and Figure 5.

Figure 3 represents a symmetric 8-strut OGSP, having M given as,

$$\begin{bmatrix} 0.8660 & 0.0000 & 0.5000 & 0.1369 & -0.5969 & -0.2372 \\ 0.0000 & 0.8660 & 0.5000 & 0.5969 & 0.1369 & -0.2372 \\ -0.8660 & 0.0000 & 0.5000 & -0.1369 & 0.5969 & -0.2372 \\ 0.0000 & -0.8660 & 0.5000 & -0.5969 & -0.1369 & -0.2372 \\ 0.0000 & 0.5000 & 0.8660 & -1.0338 & -0.2372 & 0.1369 \\ -0.5000 & 0.0000 & 0.8660 & 0.2372 & -1.0338 & 0.1369 \\ 0.0000 & -0.5000 & 0.8660 & 1.0338 & 0.2372 & 0.1369 \\ 0.5000 & 0.0000 & 0.8660 & -0.2372 & 1.0338 & 0.1369 \end{bmatrix}$$

It can be clearly seen that a strut failure in group 1 (Figure 4) or a strut failure in group 2 (Figure 5) causes the same effective change in manipulability.

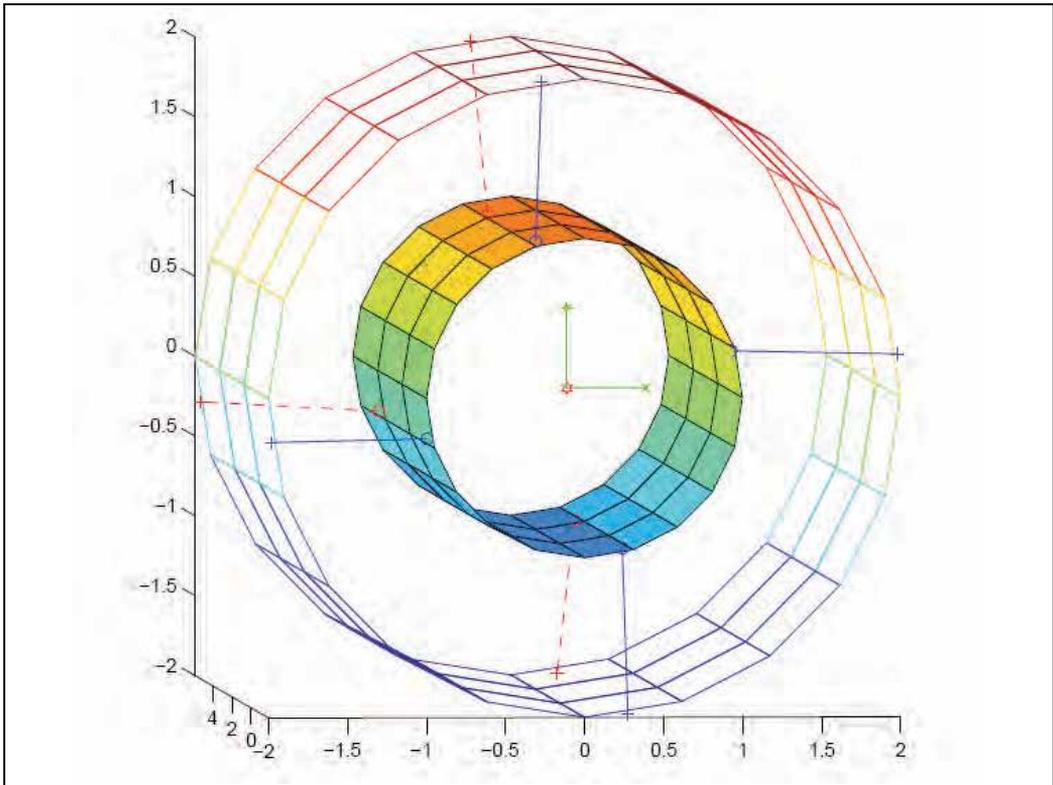


Fig. 4. [4 4] cylindrical OGSP with one failure in group 1.

This prominent feature provides symmetric OGSPs with inherent optimal fault tolerant manipulability under the occurrence of a failure. Furthermore, for symmetric OGSPs it is possible to estimate post-fault reduction in manipulability by knowing the geometry. This is explained in the following theorem.

Theorem 5. For a [p q] ($p > 3, q \geq 3$ or $q > 3, p \geq 3$) geometry, satisfying (A)- (F) in Theorem 4, the relative manipulability after a single failure in group [p] is given by ${}_i r_p$, where ${}_i r_p$ is the optimal fault tolerant manipulability under one fault for an OGSP with [p p] geometry. For the remaining cases of failure i.e. those corresponding to group [q], the relative manipulability is given by ${}_i r'_q$ where ${}_i r'_q$ is the optimal fault tolerant manipulability under one fault for an OGSP with [q q] geometry.

Proof: Consider a manipulator with [p q] ($p > 3, q \geq 3$ or $q > 3, p \geq 3$) geometry. Let M_p and M_q denote the inverse Jacobian corresponding to each group. Then the composite inverse Jacobian matrix M is given by

$$M = \begin{bmatrix} M_p \\ M_q \end{bmatrix}. \quad (27)$$

Consider the case that a single link in group [p] fails. Then from rank one perturbation of a matrix, we have

$$\det(M^T M) = \det(M'^T M') (1 + p_f (M'^T M')^{-1} p_f^T) \quad (28)$$

where p_f represents the row of M_p corresponding to the link failure and M' represents the inverse Jacobian matrix after failure. Then,

$$\frac{\det(M'^T M')}{\det(M^T M)} = \frac{1}{(1 + p_f (M'^T M')^{-1} p_f^T)}. \quad (29)$$

Using the Matrix Inversion Lemma for the expression on the R.H.S. of equation (29)

$$\frac{\det(M'^T M')}{\det(M^T M)} = 1 - (p_f (M'^T M' + p_f^T p_f)^{-1} p_f^T). \quad (30)$$

Using the formulation as in equation (7), we have

$$\frac{\det(M'^T M')}{\det(M^T M)} = 1 - (p_f (M^T M)^{-1} p_f^T). \quad (31)$$

Using conditions (A)- (F) given in Theorem 4, for a [p q] geometry with equal strut stiffness, we have

$$\varphi_{i1} = \varphi_p, \varphi_{i2} = \varphi_q,$$

$$\begin{aligned}x_{i1} &= x_p, x_{i2} = x_q, \\ y_{i1} &= y_p, y_{i2} = y_q,\end{aligned}\tag{32}$$

and

$$(M^T M)^{-1} = \text{diag}\left[\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \frac{1}{\sigma_3^2}, \frac{1}{\sigma_4^2}, \frac{1}{\sigma_5^2}, \frac{1}{\sigma_6^2}\right].\tag{33}$$

Note that p_f also has a trigonometric parametrization given by Proposition 3.

$$p_f^T = \begin{bmatrix} S_{\phi_p} C_{\beta_{ij}} \\ S_{\phi_p} S_{\beta_{ij}} \\ C_{\phi_p} \\ S_{\beta_{ij}} + C_{\phi_p} C_{\beta_{ij}} \\ -C_{\beta_{ij}} + C_{\phi_p} S_{\beta_{ij}} \\ S_{\phi_p} \end{bmatrix},\tag{34}$$

and substituting equation (32) in equations ((23)-(26)), we get

$$\sigma_1^2 = \sigma_2^2 = \frac{1}{2}(pS_{\phi_p}^2 + qS_{\phi_q}^2),\tag{35}$$

$$\sigma_3^2 = (p+q) - 2\sigma_1^2,\tag{36}$$

$$\sigma_4^2 = \sigma_5^2 = \frac{1}{2}(p(x_p^2 + y_p^2 C_{\phi_p}^2) + q(x_q^2 + y_q^2 C_{\phi_q}^2)),\tag{37}$$

and

$$\sigma_6^2 = (py_p^2 S_{\phi_p}^2 + qy_q^2 S_{\phi_q}^2).\tag{38}$$

Substituting equations ((35)-(38)) into equation(33), we get $(M^T M)^{-1}$ in terms of design parameters. Using this formulation of $(M^T M)^{-1}$ into equation (31), then substituting equation (34) in equation (31) and simplifying the complicated trigonometric expression, we get

$$\frac{\det(M'^T M')}{\det(M^T M)} = 1 - (p_f (M^T M)^{-1} p_f^T) = 1 - \frac{3}{p}.\tag{39}$$

It is important to note that this expression does not depend upon q or the particular geometric parameters ϕ_y, x_y, y_y and β_y .

Note that the optimal fault tolerant manipulability for any [p p] manipulator is given by equation (1) in Theorem 2. Hence,

$${}_1r_j = \sqrt{\frac{{}^{(2p-1)}C_6}{{}^{2p}C_6}} = \sqrt{1 - \frac{3}{p}}. \tag{40}$$

Since the choice of p does not cause any loss of generality, we have

$${}_1r'_j = \sqrt{\frac{{}^{(2q-1)}C_6}{{}^{2q}C_6}} = \sqrt{1 - \frac{3}{q}}. \tag{41}$$

#

Results from this Theorem are plotted in Figure 6. Figure 6 depicts the change in values of the relative manipulability, for different geometries, under the occurrence of one failure.

This Theorem proves the independence of the manipulability contributions of each symmetric group of a two-group OGSPs which may have different number of struts in each group. It is shown that within the group, any failure will give the same manipulability reduction even in any two-group OGSPs. Figure 6 depicts the change in relative manipulability under on failure, for symmetric OGSPs with different two-group geometrical designs.

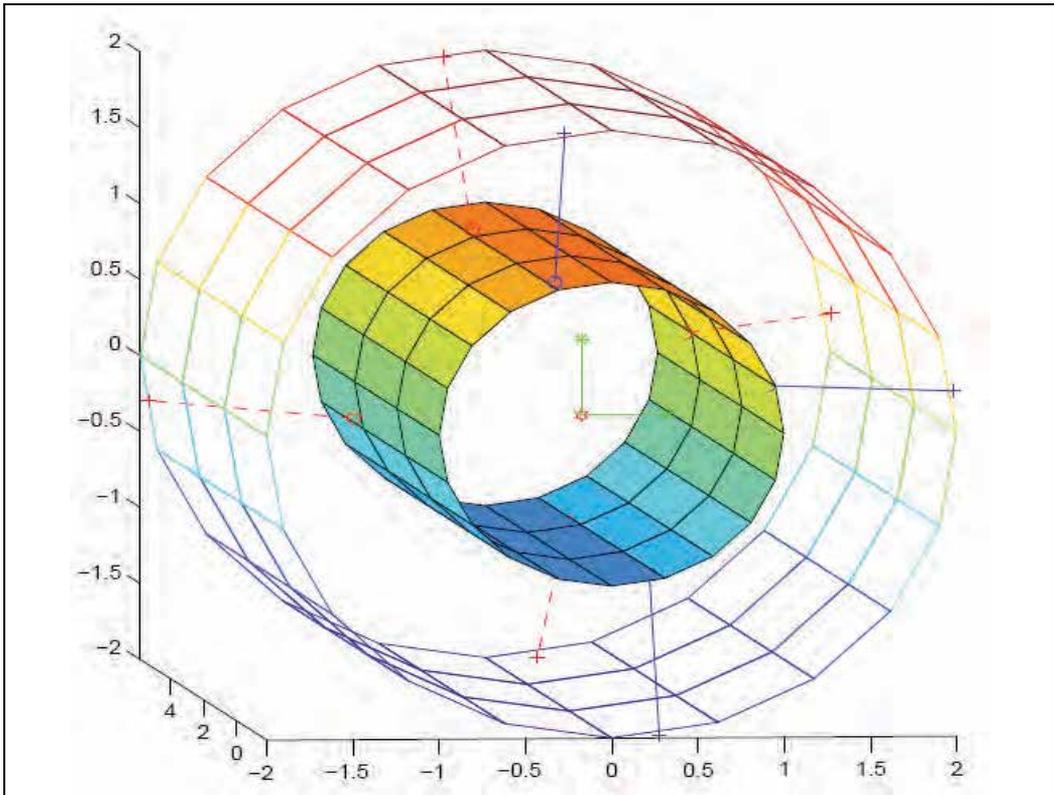


Fig. 5. [4 4] cylindrical OGSP with one failure in group 2.

Looking at Figure 6 it is now possible to estimate the level of post fault reduction in manipulability of symmetric OGSPs. Corollary 6 proves that all two-group OGSPs (i.e. with $[m \ m]$ ($m > 3$) geometries) possess optimal fault tolerant manipulability.

Corollary 6. Any $2s$ -strut OGSP with $[s \ s]$ ($s > 3$) geometry generated by Theorem 4 possesses optimal fault tolerant manipulability under one fault and its value is given by,

$${}^1r_j = \sqrt{\frac{{}^{(2s-1)}C_6}{2^s C_6}} \tag{42}$$

for all $j \in \{1,2,\dots, {}^{2s}C_s\}$.

Proof: Consider a manipulator with $[p \ q]$ ($p > 3, q \geq 3$ or $q > 3, p \geq 3$) geometry. Substitute $q = p = s$. Using Theorem 5,

$${}^1r_j = \sqrt{\frac{{}^{(2s-1)}C_6}{2^s C_6}} \tag{43}$$

for all $j \in \{1,2,\dots, {}^{2s}C_s\}$.

#

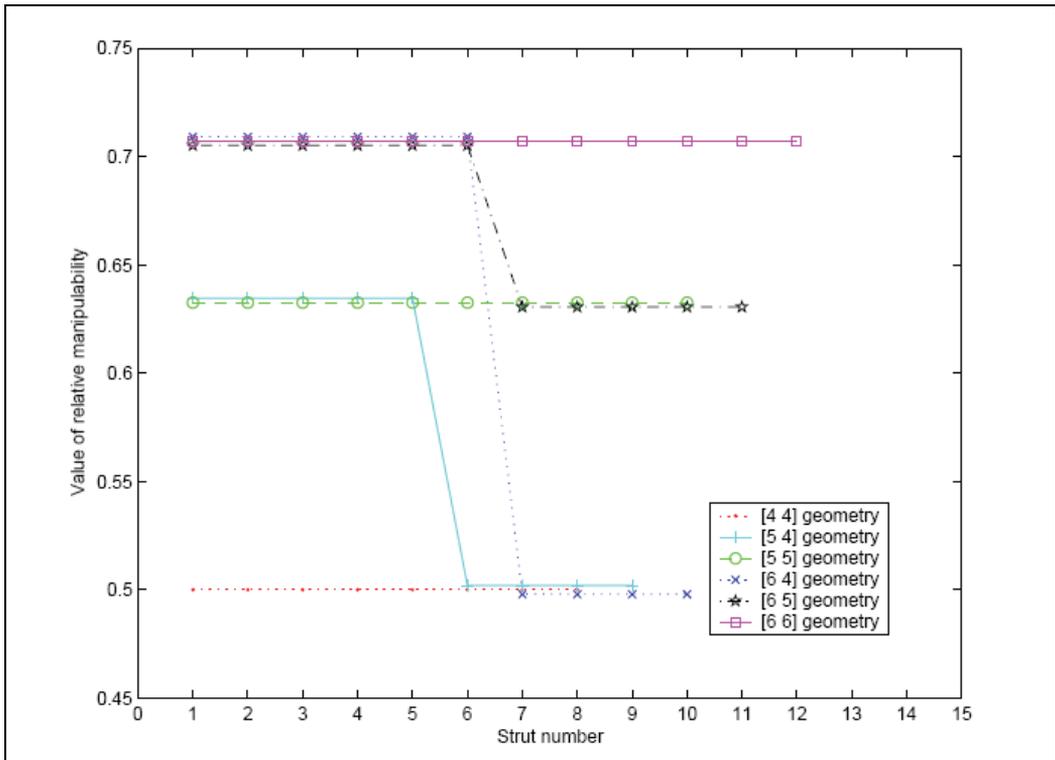


Fig. 6. Variation of the relative manipulability under a single failure, for various two group geometries

For the particular case of a symmetric 8-strut OGSP introduced at the beginning of this section,

$${}_1r_j = 0.5 \text{ for all } j \in \{1,2,\dots,8\}.$$

This inherent property possessed by symmetric OGSPs can be put to a significant advantage in design. Theorem 5 and Corollary (6) allow freedom of designing symmetric OGSPs with a high value of nominal manipulability. For example, by Corollary (6) it is seen that an 8-strut OGSP sustains any single-strut failure while retaining half of its nominal manipulability. The optimal fault tolerant manipulability of symmetric OGSPs makes them a suitable choice for critical applications where failure tolerance is necessary.

5.2 Singularities

While designing OGSPs with optimal fault tolerant manipulability, it is important to identify symmetric OGSPs which may be rendered singular under the occurrence of one fault. At the onset of singularity, unexpected motions are possible and the manipulator cannot be controlled. This is highly undesirable and potentially destructive. The following Theorem develops the necessary and sufficient condition to identify optimal fault tolerant OGSPs with potential singularity problems.

Theorem 7. Let M be the inverse Jacobian matrix of an OGSP with two groups. Then, ${}_iM^T M$ is singular if and only if the group in which failure occurs has at most 3 struts.

The following lemma is necessary to prove the Theorem.

Lemma 8. For any $m \times n$ matrix, M ,

$$\text{rank}(M^T M) = \text{rank}(M). \quad (44)$$

Proof of lemma: Clearly,

$$\text{rank}(M^T M) \leq \text{rank}(M). \quad (45)$$

Let $M^T Mx = 0$ for $x \in \mathbb{R}^n$.

Then,

$$\langle M^T Mx, x \rangle = \langle Mx, Mx \rangle = \|Mx\|^2 = 0. \quad (46)$$

Hence, $Mx = 0$.

#

Proof of Theorem: Suppose that ${}_iM^T M$ is singular. Then,

$$\text{rank}({}_iM^T M) \leq 5.$$

Proposition 7 in (Aphale, 2006) determines the rank of M for an OGSP, having p groups of struts:

$$\text{rank}(M) = \min(\sum_1^p \text{rank}(M_p), 6) \quad (47)$$

where M_p denotes the inverse Jacobian matrix of the p^{th} group. In the context of failures, this proposition directly implies

$$\text{rank}_i(M) = \min(\sum_1^p \text{rank}_f^p(M), 6) \quad (48)$$

where $^p M$ denotes the inverse Jacobian matrix of the p^{th} group having f strut failures within the group. That is, $\sum f = i$.

Applying Lemma 8 to equation (48), we have

$$\text{rank}_i(M^T M) = \min(\sum_1^p \text{rank}_f^p(M^T M), 6). \quad (49)$$

The nominal OGSP under consideration consists of two groups of struts. Hence,

$$\text{rank}_i(M^T M) = \min[\sum_{e=1}^2 \text{rank}_{f_e}^e(M^T M), 6] \quad (50)$$

where $f_1 + f_2 = i$. Theorem 1 in (Aphale, 2006) establishes that the maximum rank of the Jacobian matrix of a group of struts forming an OGSP is 3. Therefore, $^e M^T M$ is singular if the group in which any failure occurs has at most 3 struts. The converse is immediate. #

Remark: It is worthwhile to note that unitarily equivalent Jacobian matrices (and inverse Jacobian matrices) have the same manipulability, and it may be readily checked that all single failure reduced inverse Jacobian matrices of a 2s OGSP with an [s s] geometry generated by Theorem 4 are unitarily equivalent. This observation highlights the fact that these designs produce manipulators with optimal fault tolerant manipulability.

5.3 Application example: air borne laser (ABL)

Currently, feasibility of missile defense using an aircraft equipped with a high energy laser is being explored. At the concept level, the system uses a mirror inside the fuselage which focusses a beam from a megawatt-class chemical laser. Optic and beam control systems keeps the beam locked on a small supersonic target hundreds of kilometers away. It is believed that ABL can destroy hostile theater ballistic missiles while they are still in the highly vulnerable boost phase of flight before separation of the warheads. ABL can operate above the clouds, where it is possible to autonomously detect and track missiles as they are launched, using an onboard surveillance system. The defense system acquires the target, then accurately points and fires the laser with sufficient energy to destroy the missile. Airborne optical or electro-optical systems may be too large for all elements to be mounted on a single integrating structure, other than the aircraft fuselage itself. An eight-legged six-DOF OGSP (Octopod) is a perfect candidate to maintain the required alignment between

elements. However the various smaller integrating structures (benches) must still be isolated from high-frequency airframe disturbances that could excite resonances outside the bandwidth of the alignment control system. The combined active alignment and vibration isolation functions must be performed by flight-weight components, which may have to operate in a vacuum. The platform used must be able to perform the dual functions of low-frequency alignment and high-frequency isolation (Keinholz, 1999).

The manipulability requirements for OGSPs intended for such an application are very demanding and Aphale (Aphale, 2006) describes them in detail. It is also shown (Aphale, 2005) that OGSPs are capable of meeting the manipulability requirements, making them suitable for the ABL application. Failure tolerance is imperative for this missile defense application. Furthermore, it is difficult to predict specific failures at the design stage and as such failure of any actuator is considered equally likely. If an equal reduction of manipulability is desired under a failure of any strut, an OGSP with optimal fault tolerant manipulability is an excellent choice.

6. Conclusions and future work

6.1 Conclusions

This work proves that for a certain class of parallel manipulators functioning about a single point in its workspace, the mean squared relative manipulability over all possible cases of a given number of actuator failures is always constant irrespective of the geometry of the manipulator. In this context, optimal fault tolerant manipulability is defined and quantified using a simple algebraic formulation. The definition is more suited to parallel manipulators since they can retain kinematic stability under failures which constitute loss of actuators.

For micromanipulation, symmetric OGSPs can be designed to possess optimal manipulability under actuator failures. OGSP geometries that may be rendered singular due to faults can be identified and avoided. OGSPs with optimal fault tolerant manipulability are highly suitable for critical applications since they retain a reasonable and equal fault tolerant performance if any actuator fails. For example, Figure 3 illustrates a cylindrical [4 4] OGSP that can be used in aerospace applications with ABL. These OGSPs will provide operational reliability critical to the application.

6.2 Future work

Currently most OGSPs are seen to have a very small range of motion in the joint space. In such scenarios, the assumption that the Jacobian matrix remains constant with respect to time, is valid. Recent applications demand OGSPs with a larger range of motion. The assumption of the Jacobian being constant does not hold validity in such cases. Investigating the fault tolerant characteristics of a manipulator Jacobian which will take into account the change with respect to time can be of great practical importance. It has recently been shown (Roberts, Yu & Maciejewski, 2007) that, regardless of a manipulator's geometry or the amount of kinematic redundancy present in a manipulator, no fully spatial manipulator Jacobian can be equally fault tolerant to three or more joint failures. Due to these constraints in generalization, it would be useful to formulate manipulator Jacobian matrices that possess equal fault tolerance to specified scenarios involving multiple failures. In particular, weights can be assigned to relative manipulability indices corresponding to multiple failure

scenarios and optimized values of relative manipulability can be obtained based on the result derived in Theorem 1. Exploring the application of design and control techniques devised for OGSPs in areas of medical robotics and haptic interfaces can be considered. Robotics holds promise in standardized surgical procedures like eye surgery, knee surgery, etc. The theory developed thus far can be applied efficiently in medical applications where principles of robotics and computer vision combine towards a single objective. Multiple finger grasp mechanisms and other parallel manipulators have been considered for such applications. In these applications there is a need to withstand failures with almost no degradation in performance. It is possible to transfer many theories and techniques related to parallel manipulators to the analysis of multiple finger grasps with some modification.

It would be worthwhile to consider optimizing control for grasps such that fault tolerance can be achieved. Internal force calculations have been done for parallel mechanisms like multi-finger grasp mechanisms (Kerr & Roth, 1986). Internal force issues in other forms of parallel manipulators have also been explored (Lebret, Liu & Lewis, 1993) (Hiller and Schneider, 1997). Literature on the internal forces generated in GSPs is limited. OGSPs being a very recently defined class haven't been explored with respect to the internal forces they generate and need to withstand. With redundancy comes more number of actuators than the required minimum and a large number of constraints associated with them. Under failures, internal forces will be a major factor in the dynamics and control of OGSPs. Generating OGSPs that provide equal tolerance to failures with respect to the dynamic manipulability index seems feasible.

Finally, it is most important to recognize that the main contribution of this work is a combinatorial result in linear algebra. Numerous systems in various disciplines can be modeled by matrices. For instance, matrices are used to model power transmission and distribution systems. In matrix models where failures amount to elimination of rows and (or) columns, the theory of fault tolerance developed thus far would be useful and worthwhile extending.

7. References

- Aphale, S. (2006). *Designing orthogonal gough-stewart platforms with robust fault tolerance*, ProQuest / UMI, ISBN-10: 0542313596, Ph.D. Dissertation, University of Wyoming, Laramie, WY.
- Baillieul, J. (1996). Avoiding obstacles and resolving redundancy, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1698 - 1703, San Francisco, CA., April 1990.
- Hiller, M. & Schneider, M. (1997). Modeling, simulation and control of flexible manipulators, *European Journal of Mechanics*, vol. 16, 1997, page numbers 127-150.
- Hollerbach, J. M. & Suh, K. C. (1987). Redundancy resolution of manipulators through torque optimization, *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, August 1987, page numbers 308-316.
- Kerr, J. & Roth, B. (1986). Analysis of multifingered hands, *International Journal of Robotic Research*, vol. 4, no. 4, 1986, page numbers 3-17.

- Kim H. W. ; Lee J. H. ; Yi, B. J & Suh I. H. (2004). Singularity-free load distribution algorithms for a 6 dof parallel haptic device, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 298-304, New Orleans, LA., May 2004.
- Kock, S. & Schumacher W. (1998). A parallel x-y manipulator with actuation redundancy for high speed and active stiffness applications, *Proceedings of IEEE International Conference on Robotics and Automation*, , pp. 2295-2300, Leuven, Belgium, May 1998.
- Lebret, G.; Liu, K. & Lewis, F. L. Dynamic analysis and control of a stewart platform manipulator, *Journal of Robotic Systems*, vol. 10, no. 5, 1993, page numbers 629-655.
- Lewis, C. L. & Maciejewski, A. A. (1992). Dexterity optimization of kinematically redundant manipulators in presence of faults, *Proceedings of Fourth International Symposium on Robotics and Manufacturing*, pp. 279-284, Santa Fe, NM., November 1992.
- Maciejewski, A. A. (1990). Fault tolerant properties of kinematically redundant manipulators, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 638-642, Cincinnati, OH., May 1990.
- McInroy, J. E. ; O'Brien, J. F. & Neat, G. W. (1999). Precise, fault-tolerant pointing using a stewart platform, *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 1, March 1999, page numbers 91-95.
- McInroy, J. E. & Jafari, F. (2006). Finding symmetric orthogonal gough-stewart platforms, *IEEE Transactions on Robotics and Automation*, vol. 22, no. 5, October 2006, page numbers 880-889.
- Paredis, C. J. J. ; Au, W. K. F. & Khosla, P. K. (1994). Kinematic design of fault tolerant manipulators, *Computers Electrical Engg.*, vol. 20, no. 3, 1994, page numbers 211-220.
- Roberts, R. G. & Maciejewski, A. A. (1996). A local measure of fault tolerance for kinematically redundant manipulators, *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, August 1996, page numbers 543-552.
- Roberts, R. G.; Yu, H. G. & Maciejewski, A. A. (2007). Characterizing Optimally Fault-Tolerant Manipulators Based on Relative Manipulability Indices, *2007 International Conference on Intelligent Robots and Systems (IROS 2007)*, pp. 3925-3930, San Diego, CA., Oct. 29 - Nov. 2, 2007.
- Stewart, D. (1966). A platform with six degrees of freedom, *Proceedings of Institution of Mechanical Engineers*, Part 1, vol. 180, no. 15, 1966, page numbers 371-378.
- Ting, Y. ; Tosunoglu, S. & Tesar, D. (1993). A control structure for fault-tolerant operation of robotic manipulators, *Proceedings of IEEE International Conference on Robotics and Automation*, , pp. 684-690, Atlanta, GA., May 1993.
- Ukidve, C. S. ; McInroy, J. E. & Jafari, F. (2006). Orthogonal Gough-Stewart Platforms with optimal fault tolerant manipulability, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3801-3806, Orlando, FL., May 2006.
- Wapler, M. ; Urban, V.; Weisener, T.; Stallkamp, J. ; Durr, M. & Hiller, A. (2003). A stewart platform for precision surgery, *Transactions of the Institute of Measurement and Control*, vol. 25, no. 4, 2003, page numbers 329-334.

- Wen, J. T.-Y. & Wilfinger, L. S. (1999). Kinematic manipulability of general constrained rigid multibody systems, *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, June 1999, page numbers 558-567.
- Yoshikawa, T. (1985). Manipulability of robotic mechanisms, *International Journal of Robotics Research*, vol. 4, no. 2, 1985, page numbers 3-9.

Dynamic Model of a 6-dof Parallel Manipulator Using the Generalized Momentum Approach

António M. Lopes and Fernando Almeida
 UISPA – Unidade de Integração de Sistemas e Processos Automatizados,
 Universidade do Porto, Faculdade de Engenharia
 Portugal

1. Introduction

The dynamic model of a mechanical system relates the time evolution of its configuration (position, velocity and acceleration) with the forces and torques acting upon it. The inverse dynamic model is important for system control while the direct model is used for system simulation.

Serial structure manipulator dynamic modelling is a well established subject. So, recent developments have been oriented towards the improvement of numerical efficiency enabling their use in real-time control algorithms (Lilly, 1993; Naudet, 2003; Mata, 2002; Lee, 2005; Featherstone, 2000). Parallel structure manipulators present a more complex problem, and, usually, the model algorithms cannot be generalized. When used in a real-time control framework the resulting models must be simplified as they usually demand a very high computational effort.

The dynamic model of a parallel manipulator when operated in free space can be mathematically represented, in the Cartesian space, by a system of nonlinear differential equations that may be written in matrix form as

$$\mathbf{I}(\mathbf{x}) \cdot \ddot{\mathbf{x}} + \mathbf{V}(\mathbf{x}, \dot{\mathbf{x}}) \cdot \dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) = \mathbf{f} \quad (1)$$

$\mathbf{I}(\mathbf{x})$ being the inertia matrix, $\mathbf{V}(\mathbf{x}, \dot{\mathbf{x}})$ the Coriolis and centripetal terms matrix, $\mathbf{G}(\mathbf{x})$ a vector of gravitational generalized forces, \mathbf{x} the generalized position of the mobile platform or end-effector and \mathbf{f} the controlled generalized force applied on the end-effector:

$$\mathbf{f} = \mathbf{J}^T(\mathbf{x}) \cdot \boldsymbol{\tau} \quad (2)$$

where $\boldsymbol{\tau}$ is the generalized force developed by the actuators and $\mathbf{J}(\mathbf{x})$ is a jacobian matrix.

The dynamic model of a parallel manipulator is usually developed following one of two approaches (Callegari, 2006): the Newton-Euler or the Lagrange methods. The Newton-Euler approach uses the free body diagrams of the rigid bodies. The Newton-Euler equation is applied to each single body and all forces and torques acting on it are obtained. Do and Yang, and Reboulet and Berthomieu use this method on the dynamic modelling of a Stewart platform (Do & Yang, 1988; Reboulet & Berthomieu, 1991). They achieve their result introducing some simplifications on the legs models. Ji (Ji, 1994) presents a study on the

influence of leg inertia on the dynamic model of a Stewart platform. Mouly (Mouly, 1993) presents a simplified model for a variation of the Stewart platform, only taking into account the mobile platform. Dasgupta and Mruthyunjaya used the Newton-Euler approach to develop a closed-form dynamic model of the Stewart platform (Dasgupta & Mruthyunjaya, 1998). This method was also used by other researchers (Dasgupta & Choudhury, 1999; Khalil & Ibrahim, 2007; Riebe & Ulbrich, 2003; Khalil & Guegan, 2004; Guo & Li, 2006; Carvalho & Ceccarelli, 2001).

The Lagrange method describes the dynamics of a mechanical system from the concepts of work and energy. This method enables a systematic approach to the motion equations of any mechanical system. Nguyen and Pooran use this method to model a Stewart platform, modelling the legs as point masses (Nguyen & Pooran, 1989). Other researchers follow an approach similar to the one used by Nguyen and Pooran, but trying to increase the physical meaning of the obtained mathematical expressions (Liu et al., 1993; Lebret et al., 1993). Geng and co-authors (Geng et al., 1992) used the Lagrange's method to develop the equations of motion for a class of Stewart platforms. Some simplifying assumptions regarding the manipulator geometry and inertia distribution were considered. Lagrange's method was also used by others (Bhattacharya et al., 1998; Gregório & Parenti-Castelli, 2004; Caccavale et al., 2003).

Unfortunately the dynamic models obtained from these classical approaches usually present high computational loads. Therefore, alternative methods have been searched, namely the ones based on the principle of virtual work (Wang & Gosselin, 1998; Tsai, 2000; Li & Xu, 2005; Staicu et al., 2007), and screw theory (Gallardo et al., 2003).

In this paper the authors present a new approach to the problem of obtaining the dynamic model of a six degrees-of-freedom (dof) parallel manipulator: the use of the generalized momentum concept.

The manipulator under study may be seen as a variation of the Stewart platform, with the uniqueness of having all its actuators fixed to the base platform and only moving in a direction perpendicular to that base (Merlet & Gosselin, 1991). A prototype of this manipulator, the Robotic Controlled Impedance Device (RCID), was developed aiming a broad set of force-impedance control tasks. The obtained dynamic model requires a considerably lower computational effort than the one resulting from the use of classical Lagrange method.

This paper is organized as follows. Section 2 describes the RCID parallel manipulator. Section 3 presents the manipulator dynamic model using the generalized momentum approach. In section 4 the computational effort of the RCID dynamic model is evaluated. Conclusions are drawn in section 5.

2. Parallel manipulator structure

The RCID is a 6-dof parallel mini-manipulator (Figure 1). Parallel manipulators are well known because of their high dynamic performances and low positioning errors (Chablat et al., 2004; Merlet, 2006). In the last few years parallel manipulators have attracted great attention from researchers involved with robot manipulators (Bruzzone, 2005), robotic end effectors (Vischer & Clavel, 2000), robotic devices for high-precision robotic tasks (Pernette, et al., 2000), machine-tools (Zhang & Gosselin, 2002), simulators (Kim et al., 2002), and haptic devices (Constantinescu et al., 2005).



Figure 1. Photography of the RCID

The mechanical structure of the RCID comprises a fixed (base) platform and a moving (payload) platform, linked together by six independent, identical, open kinematic chains (Figure 2). Each chain comprises two links: the first link (linear actuator) is always normal to the base and has a variable length, l_i , with one of its ends fixed to the base and the other one attached, by a universal joint, to the second link; the second link (fixed-length link) has a fixed length, L , and is attached to the payload platform by a spherical joint. Points B_i and P_i are the connecting points to the base and payload platforms. They are located at the vertices of two semi-regular hexagons, inscribed in circumferences of radius r_B and r_P , that are coplanar with the base and payload platforms (Figure 3).

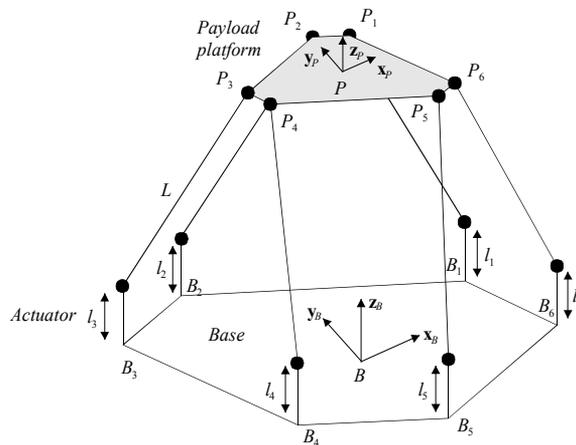


Figure 2. A schematic view of the RCID mechanical structure

For kinematic modelling purposes a right-handed reference frame $\{B\}$ is attached to the base. Its origin is located at point B , the centroid of the base. Axis \mathbf{x}_B is normal to the line connecting points B_1 and B_6 and axis \mathbf{z}_B is normal to the base, pointing towards the payload platform. The angles between points B_1 and B_3 and points B_3 and B_5 are set to 120° . The

separation angles between points B_1 and B_6 , B_2 and B_3 , and B_4 and B_5 are denoted by $2\phi_B$ (Figure 3). In a similar way, a right-handed frame $\{P\}$ is assigned to the payload platform. Its origin is located at point P , the centroid of the payload platform. Axis x_P is normal to the line connecting points P_1 and P_6 and axis z_P is normal to the payload platform, pointing in a direction opposite to the base. The angles between points P_1 and P_3 and points P_3 and P_5 are set to 120° . The separation angles between points P_1 and P_2 , P_3 and P_4 , and P_5 and P_6 are denoted by $2\phi_P$ (Figure 3). The main kinematic RCID parameters have been adjusted in order to maximize the manipulator dexterity (Lopes & Almeida, 1996) within a prescribed workspace: the payload platform may be positioned anywhere inside a sphere of radius 10 mm (centred at a point of the line witch contains axis z_B) and rotate $\pm 15^\circ$ around any axis containing the payload platform centre. This requires the actuators displacement of $\Delta l_i = 70$ mm approximately. The main kinematic parameters values are shown in Table 1.

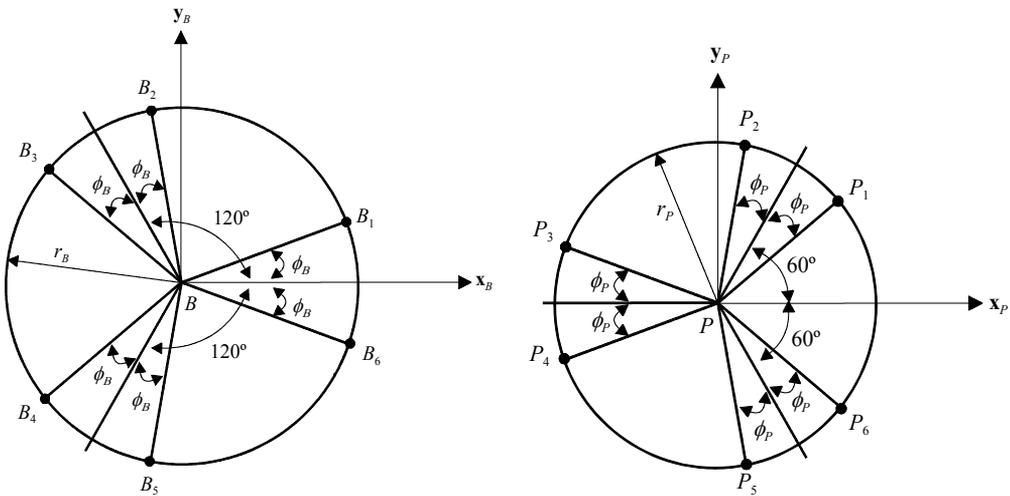


Figure 3. Position of the connecting points to the base and payload platforms

Parameter	Value
r_B	80 mm
r_P	40 mm
ϕ_B	15°
ϕ_P	0°
L	97.98 mm
Δl_i	70 mm

Table 1. RCID main kinematic parameters

The RCID prototype is powered by six DC rotary motors (28D11-222E.2, from PORTESCAP). A ball-screw based transmission converts motor rotation to actuator vertical translation. Linear position and acceleration of each actuator are measured, as well as Cartesian forces and moments applied to the payload platform. Actuators acceleration relative to the base platform is given by the difference between the signals of each actuator accelerometer and the base accelerometer. Potentiometric displacement transducers (RC13-

100 Bauform M, from MEGATRON), accelerometers (FA-208-15, Range $\pm 5g$, from EUROSENSOR) and a six-axis force/torque transducer (67M25A-I40, 200N, from JR3) are used.

The RCID mechanical structure has been produced using, whenever possible, standard mechanical components. Nevertheless, several small parts have been purposely designed and manufactured. Physically, the RCID mechanical structure comprises two fixed identical parallel platforms, which have been carefully aligned both angular and axially. The bottom platform supports the six DC electric motors. It also supports a flange to connect the RCID to an industrial robot. The ball-screw transmissions (from STAR), the linear actuators, and the potentiometric displacement transducers are located between the two fixed platforms. The universal joints are steel standard parts (from HUCO) using needle roller bearings. The fixed-length links and the double spherical joints have been purposely designed and manufactured. The payload platform supports the force/torque transducer, which has an ISO interface that may be used to attach a tool. The RCID maximal height is approximately 530 mm (when the payload platform is at its farthest position). Maximum diameter is approximately 265 mm (corresponding to the fixed platforms diameter), and total mass is about 9.7 kg. Maximum payload platform velocity along vertical direction is 220 mm/s and maximum payload capability is 5 kg.

For kinematic modelling purposes, and attaching frames {P} and {B} to the payload and base platforms, respectively, the generalized position of frame {P} relative to frame {B} may be represented by the vector:

$${}^B \mathbf{x}_{P|B|E} = [x_P \quad y_P \quad z_P \quad \psi_P \quad \theta_P \quad \varphi_P]^T = \begin{bmatrix} {}^B \mathbf{x}_{P(pos)|B} & {}^B \mathbf{x}_{P(o)|E} \end{bmatrix}^T \quad (3)$$

where ${}^B \mathbf{x}_{P(pos)|B} = [x_P \quad y_P \quad z_P]^T$ is the position of the origin of frame {P} relative to frame {B}, and ${}^B \mathbf{x}_{P(o)|E} = [\psi_P \quad \theta_P \quad \varphi_P]^T$ defines an Euler angle system representing orientation of frame {P} relative to {B}. The Euler angles constitute a minimal representation of a rigid body orientation (only three parameters). There exist twelve different Euler angle systems, according to the sequence of the performed elemental rotations (Sciavicco & Siciliano, 1996). The used Euler angle system corresponds to the basic rotations (Vukobratovic & Kircanski, 1986): ψ_P about \mathbf{z}_P ; θ_P about the rotated axis $\mathbf{y}_{P'}$; and φ_P about the rotated axis $\mathbf{x}_{P''}$. This is equivalent to a rotation of φ_P about \mathbf{x}_B , followed by a rotation of θ_P about \mathbf{y}_B , and a rotation of ψ_P about \mathbf{z}_B . The rotation matrix is given by:

$${}^B \mathbf{R}_P = \begin{bmatrix} C\psi_P C\theta_P & C\psi_P S\theta_P S\varphi_P - S\psi_P C\varphi_P & C\psi_P S\theta_P C\varphi_P + S\psi_P S\varphi_P \\ S\psi_P C\theta_P & S\psi_P S\theta_P S\varphi_P + C\psi_P C\varphi_P & S\psi_P S\theta_P C\varphi_P - C\psi_P S\varphi_P \\ -S\theta_P & C\theta_P S\varphi_P & C\theta_P C\varphi_P \end{bmatrix} \quad (4)$$

$S(\cdot)$ and $C(\cdot)$ correspond to the sine and cosine functions, respectively. The chosen Euler angle system introduces a representation singularity at $\theta_P = 90^\circ$, that is, outside the allowed RCID workspace.

The position and velocity kinematic models of the RCID are well known (Merlet & Gosselin, 1991), being obtainable from the geometrical analysis of the kinematics chains. The velocity kinematics is represented by the Euler angles jacobian matrix, \mathbf{J}_E , or the kinematics jacobian,

\mathbf{J}_C . These jacobians relate the velocities of the active joints, the actuators, with the generalized velocity of the mobile platform:

$$\dot{\mathbf{i}} = \mathbf{J}_E \cdot {}^B \dot{\mathbf{x}}_P|_{B/E} = \mathbf{J}_E \cdot \begin{bmatrix} {}^B \dot{\mathbf{x}}_{P(pos)}|_B \\ {}^B \dot{\mathbf{x}}_{P(o)}|_E \end{bmatrix} \quad (5)$$

$$\dot{\mathbf{i}} = \mathbf{J}_C \cdot {}^B \dot{\mathbf{x}}_P|_B = \mathbf{J}_C \cdot \begin{bmatrix} {}^B \dot{\mathbf{x}}_{P(pos)}|_B \\ {}^B \boldsymbol{\omega}_P|_B \end{bmatrix} \quad (6)$$

with

$$\dot{\mathbf{i}} = [\dot{i}_1 \quad \dot{i}_2 \quad \dots \quad \dot{i}_6]^T \quad (7)$$

$${}^B \boldsymbol{\omega}_P|_B = \mathbf{J}_A \cdot {}^B \dot{\mathbf{x}}_{P(o)}|_E \quad (8)$$

and (Vukobratovic & Kircanski, 1986)

$$\mathbf{J}_A = \begin{bmatrix} 0 & -S\psi_p & C\theta_p C\psi_p \\ 0 & C\psi_p & C\theta_p S\psi_p \\ 1 & 0 & -S\theta_p \end{bmatrix} \quad (9)$$

Vectors ${}^B \dot{\mathbf{x}}_{P(pos)}|_B \equiv {}^B \mathbf{v}_P|_B$ and ${}^B \boldsymbol{\omega}_P|_B$ represent, by that order, the linear and angular velocity of the mobile platform relative to $\{B\}$, and ${}^B \dot{\mathbf{x}}_{P(o)}|_E$ represents the Euler angles time derivative.

3. Dynamic modelling using the generalized momentum approach

The generalized momentum of a rigid body, \mathbf{q}_c , may be obtained from the following general expression:

$$\mathbf{q}_c = \mathbf{I}_c \cdot \mathbf{u}_c \quad (10)$$

Vector \mathbf{u}_c represents the generalized velocity (linear and angular) of the body and \mathbf{I}_c is its inertia matrix. Vectors \mathbf{q}_c and \mathbf{u}_c and inertia matrix \mathbf{I}_c must be expressed in the same frame of reference.

Equation (10) may also be written as:

$$\mathbf{q}_c = \begin{bmatrix} \mathbf{Q}_c \\ \mathbf{H}_c \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{c(tran)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{c(rot)} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{bmatrix} \quad (11)$$

where \mathbf{Q}_c is the linear momentum vector due to rigid body translation and \mathbf{H}_c is the angular momentum vector due to body rotation. $\mathbf{I}_{c(tran)}$ is the translational inertia matrix and $\mathbf{I}_{c(rot)}$ the rotational inertia matrix. \mathbf{v}_c and $\boldsymbol{\omega}_c$ are the body linear and angular velocities.

The inertial component of the generalized force acting on the body can be obtained from the time derivative of equation (10):

$$\mathbf{f}_{c(ine)} = \dot{\mathbf{q}}_c = \dot{\mathbf{I}}_c \cdot \mathbf{u}_c + \mathbf{I}_c \cdot \dot{\mathbf{u}}_c \quad (12)$$

with force and momentum expressed in the same frame.

3.1 Mobile platform modeling

The linear momentum of the mobile platform, written in frame {B}, $\mathbf{Q}_P|_B$, may be obtained from the following expression:

$$\mathbf{Q}_P|_B = m_P \cdot {}^B \mathbf{v}_P|_B = \mathbf{I}_{P(tran)} \cdot {}^B \mathbf{v}_P|_B \quad (13)$$

$\mathbf{I}_{P(tran)}$ is the translational inertia matrix of the mobile platform,

$$\mathbf{I}_{P(tran)} = \begin{bmatrix} m_P & 0 & 0 \\ 0 & m_P & 0 \\ 0 & 0 & m_P \end{bmatrix} \quad (14)$$

m_P being its mass.

The angular momentum, $\mathbf{H}_P|_B$, also written in frame {B}, is:

$$\mathbf{H}_P|_B = \mathbf{I}_{P(rot)}|_B \cdot {}^B \boldsymbol{\omega}_P|_B \quad (15)$$

$\mathbf{I}_{P(rot)}|_B$ represents the rotational inertia matrix of the mobile platform, expressed in the base frame {B}.

The inertia matrix of a rigid body is constant when expressed in a frame that is fixed relative to that body. Furthermore if the frame axes coincide with the principal directions of inertia of the body, then all inertia products are zero and the inertia matrix is diagonal. Therefore, the rotational inertia matrix of the mobile platform when expressed in frame {P} may be written as:

$$\mathbf{I}_{P(rot)}|_P = \begin{bmatrix} I_{P_{xx}} & 0 & 0 \\ 0 & I_{P_{yy}} & 0 \\ 0 & 0 & I_{P_{zz}} \end{bmatrix} \quad (16)$$

This inertia matrix can be written in frame {B} using the following transformation (Torby, 1984):

$$\mathbf{I}_{P(rot)}|_B = {}^B \mathbf{R}_P \cdot \mathbf{I}_{P(rot)}|_P \cdot {}^B \mathbf{R}_P^T \quad (17)$$

The generalized momentum of the mobile platform, expressed in frame {B}, can be obtained from the simultaneous use of equations (13) and (15):

$$\mathbf{q}_P|_B = \begin{bmatrix} \mathbf{I}_{P(tran)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{P(rot)}|_B \end{bmatrix} \cdot \begin{bmatrix} {}^B \mathbf{v}_P|_B \\ {}^B \boldsymbol{\omega}_P|_B \end{bmatrix} \quad (18)$$

where

$$\mathbf{I}_P|_B = \begin{bmatrix} \mathbf{I}_{P(tra)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{P(rot)}|_B \end{bmatrix} \quad (19)$$

is the mobile platform inertia matrix written in the base frame {B}.
The combination of equations (8) and (15) results into:

$$\mathbf{H}_P|_B = \mathbf{I}_{P(rot)}|_B \cdot \mathbf{J}_A \cdot {}^B \dot{\mathbf{x}}_{P(o)}|_E \quad (20)$$

Accordingly, equation (18) may be rewritten as:

$$\mathbf{q}_P|_B = \begin{bmatrix} \mathbf{I}_{P(tra)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{P(rot)}|_B \end{bmatrix} \cdot \begin{bmatrix} \mathfrak{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_A \end{bmatrix} \cdot \begin{bmatrix} {}^B \mathbf{v}_P|_B \\ {}^B \dot{\mathbf{x}}_{P(o)}|_E \end{bmatrix} \quad (21)$$

$$\mathbf{q}_P|_B = \mathbf{I}_P|_B \cdot \mathbf{T} \cdot {}^B \dot{\mathbf{x}}_P|_{B/E} \quad (22)$$

\mathbf{T} being a matrix transformation defined by:

$$\mathbf{T} = \begin{bmatrix} \mathfrak{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_A \end{bmatrix} \quad (23)$$

The time derivative of equation (22) results into:

$${}^P \mathbf{f}_{P(ine)}|_B = \dot{\mathbf{q}}_P|_B = \frac{d}{dt} (\mathbf{I}_P|_B \cdot \mathbf{T})^B \dot{\mathbf{x}}_P|_{B/E} + \mathbf{I}_P|_B \cdot \mathbf{T} \cdot {}^B \ddot{\mathbf{x}}_P|_{B/E} \quad (24)$$

${}^P \mathbf{f}_{P(ine)}|_B$ is the inertial component of the generalized force acting on {P} due to the mobile platform motion, expressed in frame {B}. The corresponding actuating forces, $\boldsymbol{\tau}_{P(ine)}$, may be computed from the following relation:

$$\boldsymbol{\tau}_{P(ine)} = \mathbf{J}_C^{-T} \cdot {}^P \mathbf{f}_{P(ine)}|_B \quad (25)$$

The same inertial component of the generalized force acting on {P} due to the mobile platform motion, but now expressed using the Euler angles system, can be found by pre-multiplying equation (24) by \mathbf{T}^T :

$${}^P \mathbf{f}_{P(ine)}|_{B/E} = \mathbf{T}^T \cdot {}^P \mathbf{f}_{P(ine)}|_B = \mathbf{T}^T \cdot \frac{d}{dt} (\mathbf{I}_P|_B \cdot \mathbf{T})^B \dot{\mathbf{x}}_P|_{B/E} + \mathbf{T}^T \cdot \mathbf{I}_P|_B \cdot \mathbf{T} \cdot {}^B \ddot{\mathbf{x}}_P|_{B/E} \quad (26)$$

and, in a similar way, the corresponding actuating forces, $\boldsymbol{\tau}_{P(ine)}$, may be computed from the relation:

$$\boldsymbol{\tau}_{P(ine)} = \mathbf{J}_E^{-T} \cdot {}^P \mathbf{f}_{P(ine)}|_{B/E} \quad (27)$$

This implies that:

$${}^P \mathbf{f}_{P(ine)}|_{B/E} = \begin{bmatrix} {}^P \mathbf{F}_{P(ine)}^T|_B & {}^P \mathbf{M}_{P(ine)}^T|_E \end{bmatrix}^T \quad (28)$$

Vector ${}^P \mathbf{F}_{P(in)e} |_{|_B}$ represents the force acting on the centre of mass of the mobile platform, expressed in the base frame, $\{B\}$, and vector ${}^P \mathbf{M}_{P(in)e} |_{|_E}$ represents the moment acting on the mobile platform, expressed using the Euler angles system. Thus, this representation does not allow a clear physical interpretation of ${}^P \mathbf{M}_{P(in)e} |_{|_E}$.

On the other hand, it can be said that

$${}^P \mathbf{f}_{P(in)e} |_{|_B} = \begin{bmatrix} {}^P \mathbf{F}_{P(in)e} |_{|_B} & {}^P \mathbf{M}_{P(in)e} |_{|_E} \end{bmatrix}^T \quad (29)$$

where ${}^P \mathbf{M}_{P(in)e} |_{|_B} = \mathbf{J}_A^{-T} \cdot {}^P \mathbf{M}_{P(in)e} |_{|_E}$ represents the moment acting on the mobile platform expressed, this time, in the base frame.

From equation (24) it can be concluded that two matrices playing the roles of the inertia matrix and the Coriolis and centripetal terms matrix are:

$$\mathbf{I}_P |_{|_B} \cdot \mathbf{T} \quad (30)$$

$$\frac{d}{dt} (\mathbf{I}_P |_{|_B} \cdot \mathbf{T}) \quad (31)$$

It must be emphasized that these matrices do not have the properties of inertia or Coriolis and centripetal terms matrices and therefore should not, strictly, be named as such. Nevertheless, throughout the paper the names “inertia matrix” and “Coriolis and centripetal terms matrix” may be used if there is no risk of misunderstanding.

On the other hand, from equation (26), the inertia matrix and the Coriolis and centripetal terms matrix, expressed in the Euler angles system, are:

$$\mathbf{I}_P |_{|_E} = \mathbf{T}^T \cdot \mathbf{I}_P |_{|_B} \cdot \mathbf{T} \quad (32)$$

$$\mathbf{V}_P |_{|_E} = \mathbf{T}^T \cdot \frac{d}{dt} (\mathbf{I}_P |_{|_B} \cdot \mathbf{T}) \quad (33)$$

3.2 Actuators modeling

As the RCID actuators can only move perpendicularly to the base plane, their angular velocity relative to frame $\{B\}$ is always zero. So, each actuator can be modelled as a point mass located at its centre of mass.

The linear momentum of each actuator along direction \mathbf{z}_B , q_{A_i} , is obtainable from:

$$q_{A_i} = m_A \cdot \dot{l}_i \quad (34)$$

where m_A is the mass and \dot{l}_i the velocity of actuator i .

Simultaneously considering the six actuators results into:

$$\mathbf{q}_A = \begin{bmatrix} q_{A_1} \\ q_{A_2} \\ \vdots \\ q_{A_6} \end{bmatrix} = m_A \begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \\ \vdots \\ \dot{l}_6 \end{bmatrix} = m_A \cdot \dot{\mathbf{l}} \quad (35)$$

The use of velocity inverse kinematics and transformation \mathbf{T} in equation (35) leads to:

$$\mathbf{q}_A = m_A \cdot \mathbf{J}_C \cdot \mathbf{T} \cdot \dot{\mathbf{x}}_P |_{B/E} \quad (36)$$

The inertial component of the actuating forces, $\boldsymbol{\tau}_{A(ine)}$, due to actuators translation may be obtained from the time derivative of equation (36):

$$\boldsymbol{\tau}_{A(ine)} = \dot{\mathbf{q}}_A = m_A \cdot \left(\dot{\mathbf{J}}_E \cdot \dot{\mathbf{x}}_P |_{B/E} + \mathbf{J}_E \cdot \ddot{\mathbf{x}}_P |_{B/E} \right) \quad (37)$$

Multiplying equation (37) by \mathbf{J}_C^T the inertial component of the generalized force acting on {P} due to actuators translation, expressed in frame {B}, is obtained as:

$${}^P \mathbf{f}_{A(ine)} |_B = m_A \cdot \mathbf{J}_C^T \cdot \dot{\mathbf{J}}_E \cdot \dot{\mathbf{x}}_P |_{B/E} + m_A \cdot \mathbf{J}_C^T \cdot \mathbf{J}_E \cdot \ddot{\mathbf{x}}_P |_{B/E} \quad (38)$$

The inertial component of the generalized force acting on {P} due to actuators translation, expressed using the Euler angles system, is:

$$\begin{aligned} {}^P \mathbf{f}_{A(ine)} |_{B/E} &= \mathbf{T}^T \cdot {}^P \mathbf{f}_{A(ine)} |_B = m_A \cdot \mathbf{T}^T \cdot \mathbf{J}_C^T \cdot \dot{\mathbf{J}}_E \cdot \dot{\mathbf{x}}_P |_{B/E} + m_A \cdot \mathbf{T}^T \cdot \mathbf{J}_C^T \cdot \mathbf{J}_E \cdot \ddot{\mathbf{x}}_P |_{B/E} \\ &= m_A \cdot \mathbf{J}_E^T \cdot \dot{\mathbf{J}}_E \cdot \dot{\mathbf{x}}_P |_{B/E} + m_A \cdot \mathbf{J}_E^T \cdot \mathbf{J}_E \cdot \ddot{\mathbf{x}}_P |_{B/E} \end{aligned} \quad (39)$$

The inertia matrix and the Coriolis and centripetal terms matrix, expressed in the Euler angles system, may be extracted from equation (39) as:

$$\mathbf{I}_{A|E(eq)} = m_A \cdot \mathbf{J}_E^T \cdot \mathbf{J}_E \quad (40)$$

$$\mathbf{V}_{A|E(eq)} = m_A \cdot \mathbf{J}_E^T \cdot \dot{\mathbf{J}}_E \quad (41)$$

These matrices represent the inertia matrix and the Coriolis and centripetal terms matrix of a virtual mobile platform that is equivalent to the six actuators.

3.3 Fixed-length links modeling

If the centre of mass of each fixed-length link, cm_L , is considered to be located at a constant distance b_{cm} from the fixed-length link to mobile platform connecting point (Figure 4) then its position relative to frame {B} is:

$${}^B \mathbf{p}_{L_i} |_B = {}^B \mathbf{x}_{P(pos)} |_B + {}^P \mathbf{p}_i |_B - \frac{b_{cm}}{L} \cdot \mathbf{a}_i \quad (42)$$

Equation (42) may be successively rewritten as:

$${}^B \mathbf{p}_{L_i} |_B = {}^B \mathbf{x}_{P(pos)} |_B + {}^P \mathbf{p}_i |_B - \frac{b_{cm}}{L} \cdot \left({}^B \mathbf{x}_{P(pos)} |_B - \mathbf{b}_i + {}^P \mathbf{p}_i |_B - \mathbf{d}_i \right) \quad (43)$$

$${}^B \mathbf{p}_{L_i} |_B = \left(1 - \frac{b_{cm}}{L} \right) \cdot {}^B \mathbf{x}_{P(pos)} |_B + \left(1 - \frac{b_{cm}}{L} \right) \cdot {}^P \mathbf{p}_i |_B + \frac{b_{cm}}{L} \cdot \mathbf{b}_i + \frac{b_{cm}}{L} \cdot \mathbf{d}_i \quad (44)$$

${}^B \mathbf{p}_{L_i | B}$ being a vector expressed in frame {B}.

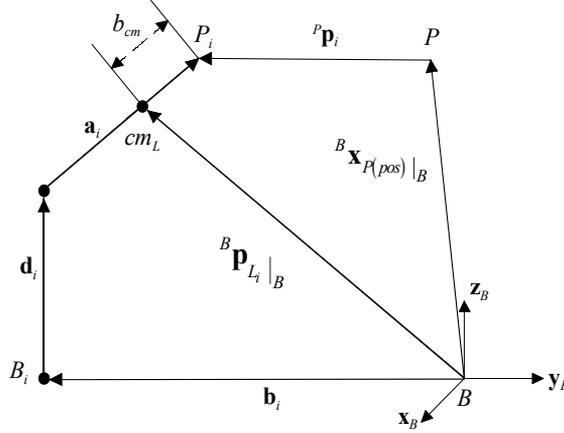


Figure 4. Position of the centre of mass of a fixed-length link i

The linear velocity of the fixed-length link centre of mass, ${}^B \dot{\mathbf{p}}_{L_i | B}$, relative to {B} and expressed in the same frame may be computed from the time derivative of equation (44):

$$\begin{aligned} {}^B \dot{\mathbf{p}}_{L_i | B} &= \left(1 - \frac{b_{cm}}{L}\right) \cdot \left({}^B \dot{\mathbf{x}}_{P(pos) | B} + {}^B \boldsymbol{\omega}_P | B \times {}^P \mathbf{p}_i | B\right) + \frac{b_{cm}}{L} \cdot \dot{\mathbf{d}}_i \\ &= \left(1 - \frac{b_{cm}}{L}\right) \cdot \left({}^B \dot{\mathbf{x}}_{P(pos) | B} + {}^B \boldsymbol{\omega}_P | B \times {}^P \mathbf{p}_i | B\right) + \frac{b_{cm}}{L} \cdot \dot{l}_i \cdot \mathbf{z}_B \end{aligned} \quad (45)$$

Equation (45) can be rewritten as:

$${}^B \dot{\mathbf{p}}_{L_i | B} = \mathbf{J}_{B_i} \cdot \begin{bmatrix} {}^B \mathbf{v}_P | B \\ {}^B \boldsymbol{\omega}_P | B \end{bmatrix} \quad (46)$$

where the jacobian \mathbf{J}_{B_i} is given by:

$$\mathbf{J}_{B_i} = \left(1 - \frac{b_{cm}}{L}\right) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{b_{cm}}{L - b_{cm}} J_{Ci1} & \frac{b_{cm}}{L - b_{cm}} J_{Ci2} & \frac{b_{cm}}{L - b_{cm}} J_{Ci3} + 1 \\ 0 & {}^P p_i | B_z & -{}^P p_i | B_y \\ -{}^P p_i | B_z & 0 & {}^P p_i | B_x \\ {}^P p_i | B_y + \frac{b_{cm}}{L - b_{cm}} J_{Ci4} & -{}^P p_i | B_x + \frac{b_{cm}}{L - b_{cm}} J_{Ci5} & \frac{b_{cm}}{L - b_{cm}} J_{Ci6} \end{bmatrix} \quad (47)$$

being J_{Cij} the elements of line i column j of matrix \mathbf{J}_C .

The linear momentum of each fixed-length link, $\mathbf{Q}_{L_i | B}$, can be represented in frame {B} as:

$$\mathbf{Q}_{L_i|_B} = m_L \cdot \dot{\mathbf{p}}_{L_i|_B} \quad (48)$$

where m_L is fixed-length link mass.

Introducing jacobian \mathbf{J}_{B_i} and transformation \mathbf{T} in the previous equation results into:

$$\mathbf{Q}_{L_i|_B} = m_L \cdot \mathbf{J}_{B_i} \cdot \mathbf{T}^B \dot{\mathbf{x}}_{P|_{B/E}} \quad (49)$$

The inertial component of the force applied to the fixed-length link due to its translation and expressed in {B} can be obtained from the time derivative of equation (49):

$${}^{L_i} \mathbf{f}_{L_i(ine)(tra)|_B} = \dot{\mathbf{Q}}_{L_i|_B} = m_L \cdot \frac{d}{dt} (\mathbf{J}_{B_i} \cdot \mathbf{T})^B \dot{\mathbf{x}}_{P|_{B/E}} + m_L \cdot \mathbf{J}_{B_i} \cdot \mathbf{T}^B \ddot{\mathbf{x}}_{P|_{B/E}} \quad (50)$$

When equation (50) is multiplied by $\mathbf{J}_{B_i}^T$ the inertial component of the force applied to {P} due to each fixed-length link translation is obtained in frame {B}:

$$\begin{aligned} {}^P \mathbf{f}_{L_i(ine)(tra)|_B} &= \mathbf{J}_{B_i}^T \cdot {}^{L_i} \mathbf{f}_{L_i(ine)(tra)|_B} \\ &= m_L \cdot \mathbf{J}_{B_i}^T \cdot \frac{d}{dt} (\mathbf{J}_{B_i} \cdot \mathbf{T})^B \dot{\mathbf{x}}_{P|_{B/E}} + m_L \cdot \mathbf{J}_{B_i}^T \cdot \mathbf{J}_{B_i} \cdot \mathbf{T}^B \ddot{\mathbf{x}}_{P|_{B/E}} \end{aligned} \quad (51)$$

The inertial component of the generalized force applied to {P} due to each fixed-length link translation and expressed using the Euler angles system can be obtained pre-multiplying equation (51) by matrix \mathbf{T}^T :

$$\mathbf{T}^T \cdot {}^P \mathbf{f}_{L_i(ine)(tra)|_B} = m_L \cdot \mathbf{T}^T \cdot \mathbf{J}_{B_i}^T \cdot \frac{d}{dt} (\mathbf{J}_{B_i} \cdot \mathbf{T})^B \dot{\mathbf{x}}_{P|_{B/E}} + m_L \cdot \mathbf{T}^T \cdot \mathbf{J}_{B_i}^T \cdot \mathbf{J}_{B_i} \cdot \mathbf{T}^B \ddot{\mathbf{x}}_{P|_{B/E}} \quad (52)$$

$${}^P \mathbf{f}_{L_i(ine)(tra)|_{B/E}} = \mathbf{T}^T \cdot {}^P \mathbf{f}_{L_i(ine)(tra)|_B} \quad (53)$$

The inertia matrix and the Coriolis and centripetal terms matrix of the translating fixed-length link, expressed in the Euler angles system, may be extracted from equation (52) as:

$$\mathbf{I}_{L_i(tra)|_{E(eq)}} = m_L \cdot \mathbf{T}^T \cdot \mathbf{J}_{B_i}^T \cdot \mathbf{J}_{B_i} \cdot \mathbf{T} \quad (54)$$

$$\mathbf{V}_{L_i(tra)|_{E(eq)}} = m_L \cdot \mathbf{T}^T \cdot \mathbf{J}_{B_i}^T \cdot \frac{d}{dt} (\mathbf{J}_{B_i} \cdot \mathbf{T}) \quad (55)$$

These matrices represent the inertia matrix and the Coriolis and centripetal terms matrix of a virtual mobile platform that is equivalent to each translating fixed-length link.

The angular momentum of each fixed-length link can be represented in frame {B} as:

$$\mathbf{H}_{L_i|_B} = \mathbf{I}_{L_i(rot)|_B} \cdot \dot{\boldsymbol{\omega}}_{L_i|_B} \quad (56)$$

It is convenient to express the inertia matrix of the rotating fixed-length link in a frame fixed to the fixed-length link itself, $\{L_i\} \equiv \{\mathbf{x}_{L_i}, \mathbf{y}_{L_i}, \mathbf{z}_{L_i}\}$. So,

$$\mathbf{I}_{L_i(rot)|_B} = {}^B \mathbf{R}_{L_i} \cdot \mathbf{I}_{L_i(rot)|_{L_i}} \cdot {}^B \mathbf{R}_{L_i}^T \quad (57)$$

where ${}^B \mathbf{R}_{L_i}$ is the orientation matrix of each fixed-length link frame, $\{L_i\}$, relative to the base frame, $\{B\}$.

Fixed-length links frames were chosen in the following way: axis \mathbf{x}_{L_i} coincides with the fixed-length link axis and points towards the fixed-length link to mobile platform connecting point, meaning that it is coincident with vector \mathbf{a}_i ; axis \mathbf{y}_{L_i} is perpendicular to \mathbf{x}_{L_i} and always parallel to the base plane, this condition being possible given the existence of a universal joint in the fixed-length link to actuator connecting point that negates any rotation along its own axis; axis \mathbf{z}_{L_i} completes the frame following the right hand rule and its projection along axis \mathbf{z}_B is always positive. With this choice matrix ${}^B \mathbf{R}_{L_i}$ becomes:

$${}^B \mathbf{R}_{L_i} = [\mathbf{x}_{L_i} \quad \mathbf{y}_{L_i} \quad \mathbf{z}_{L_i}] \quad (58)$$

where

$$\mathbf{x}_{L_i} = \begin{bmatrix} \frac{a_{ix}}{L} & \frac{a_{iy}}{L} & \frac{a_{iz}}{L} \end{bmatrix}^T \quad (59)$$

$$\mathbf{y}_{L_i} = \begin{bmatrix} -\frac{a_{iy}}{\sqrt{a_{ix}^2 + a_{iy}^2}} & \frac{a_{ix}}{\sqrt{a_{ix}^2 + a_{iy}^2}} & 0 \end{bmatrix}^T \quad (60)$$

$$\mathbf{z}_{L_i} = \mathbf{x}_{L_i} \times \mathbf{y}_{L_i} \quad (61)$$

So, the inertia matrices of the fixed-length links can be written as

$$\mathbf{I}_{L_i(rot)|_{L_i}} = \begin{bmatrix} I_{L_{xx}} & 0 & 0 \\ 0 & I_{L_{yy}} & 0 \\ 0 & 0 & I_{L_{zz}} \end{bmatrix} \quad (62)$$

where $I_{L_{xx}}$, $I_{L_{yy}}$ and $I_{L_{zz}}$ are the fixed-length link moments of inertia expressed in its own frame.

The angular velocity of each fixed-length link can be obtained from the linear velocities of two points belonging to it. If these two points are taken as the fixed-length link to actuator and the fixed-length link to mobile platform connecting points, the following expression results:

$${}^B \boldsymbol{\omega}_{L_i|_B} \times \mathbf{a}_i = {}^B \mathbf{v}_{P|_B} + {}^B \boldsymbol{\omega}_{P|_B} \times {}^P \mathbf{p}_{i|_B} - \dot{l}_i \cdot \mathbf{z}_B \quad (63)$$

As the fixed-length link cannot rotate along its own axis the angular velocity along $\mathbf{x}_{L_i} = \hat{\mathbf{a}}_i$ is always zero and so vectors \mathbf{a}_i and ${}^B \boldsymbol{\omega}_{L_i|_B}$ are always perpendicular. This property enables equation (63) to be rewritten as:

$${}^B \boldsymbol{\omega}_{L_i|_B} = \frac{1}{L^2} \cdot \left[\mathbf{a}_i \times \left({}^B \mathbf{v}_{P|_B} + {}^B \boldsymbol{\omega}_{P|_B} \times {}^P \mathbf{p}_{i|_B} - \dot{l}_i \cdot \mathbf{z}_B \right) \right] \quad (64)$$

or,

$${}^B \boldsymbol{\omega}_{L_i|_B} = \mathbf{J}_{D_i} \cdot \begin{bmatrix} {}^B \mathbf{v}_{P|_B} \\ {}^B \boldsymbol{\omega}_{P|_B} \end{bmatrix} \quad (65)$$

where jacobian \mathbf{J}_{D_i} is given by:

$$\mathbf{J}_{D_i} = \frac{1}{L^2} \cdot \begin{bmatrix} -a_{iy} J_{Ci1} & -a_{iy} J_{Ci2} - a_{iz} & a_{iy} (1 - J_{Ci3}) \\ a_{iz} + a_{ix} J_{Ci1} & a_{ix} J_{Ci2} & -a_{iz} (1 - J_{Ci3}) \\ -a_{iy} & a_{ix} & 0 \\ a_{iy} \left({}^P p_{i|By} - J_{Ci4} \right) + a_{iz} {}^P p_{i|Bz} & -a_{iy} \left({}^P p_{i|Bx} + J_{Ci5} \right) & -a_{iy} J_{Ci6} - a_{iz} {}^P p_{i|Bx} \\ -a_{ix} \left({}^P p_{i|By} - J_{Ci4} \right) & a_{iz} {}^P p_{i|Bz} + a_{ix} \left({}^P p_{i|Bx} + J_{Ci5} \right) & -a_{iz} {}^P p_{i|By} + a_{ix} J_{Ci6} \\ -a_{ix} {}^P p_{i|Bz} & -a_{iy} {}^P p_{i|Bz} & a_{ix} {}^P p_{i|Bx} + a_{iy} {}^P p_{i|By} \end{bmatrix} \quad (66)$$

Introducing jacobian \mathbf{J}_{D_i} and transformation \mathbf{T} in equation (56) results into:

$$\mathbf{H}_{L_i|_B} = \mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \cdot {}^B \dot{\mathbf{x}}_{P|_{BE}} \quad (67)$$

The inertial component of the generalized force applied to the fixed-length link due to its rotation and expressed in {B} can be obtained from the time derivative of equation (67):

$$\begin{aligned} {}^{L_i} \mathbf{f}_{L_i(ine)(rot)|_B} &= \dot{\mathbf{H}}_{L_i|_B} \\ &= \frac{d}{dt} \left(\mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \right)^B \dot{\mathbf{x}}_{P|_{BE}} + \mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \cdot {}^B \ddot{\mathbf{x}}_{P|_{BE}} \end{aligned} \quad (68)$$

When equation (68) is pre-multiplied by $\mathbf{J}_{D_i}^T$, the inertial component of the generalized force applied to {P} due to each fixed-length link rotation is obtained in frame {B}:

$$\begin{aligned} {}^P \mathbf{f}_{L_i(ine)(rot)|_B} &= \mathbf{J}_{D_i}^T \cdot {}^{L_i} \mathbf{f}_{L_i(ine)(rot)|_B} \\ &= \mathbf{J}_{D_i}^T \cdot \frac{d}{dt} \left(\mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \right)^B \dot{\mathbf{x}}_{P|_{BE}} + \mathbf{J}_{D_i}^T \cdot \mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \cdot {}^B \ddot{\mathbf{x}}_{P|_{BE}} \end{aligned} \quad (69)$$

The inertial component of the generalized force applied to {P} due to each fixed-length link rotation, ${}^P \mathbf{f}_{L_i(ine)(rot)|_{BE}}$, expressed using the Euler angles system, can be obtained pre-multiplying equation (70) by matrix \mathbf{T}^T :

$$\begin{aligned} \mathbf{T}^T \cdot {}^P \mathbf{f}_{L_i(ine)(rot)|_B} &= \mathbf{T}^T \cdot \mathbf{J}_{D_i}^T \cdot \frac{d}{dt} \left(\mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \right)^B \dot{\mathbf{x}}_{P|_{BE}} + \\ &\quad \mathbf{T}^T \cdot \mathbf{J}_{D_i}^T \cdot \mathbf{I}_{L_i(rot)|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \cdot {}^B \ddot{\mathbf{x}}_{P|_{BE}} \end{aligned} \quad (71)$$

$${}^P \mathbf{f}_{L_i(\text{ine})(\text{rot})|_{B|E}} = \mathbf{T}^T \cdot {}^P \mathbf{f}_{L_i(\text{ine})(\text{rot})|_B} \quad (72)$$

The inertia matrix and the Coriolis and centripetal terms matrix of the rotating fixed-length link, expressed in the Euler angles system, may be extracted from equation (71) as:

$$\mathbf{I}_{L_i(\text{rot})|_{E(\text{eq})}} = \mathbf{T}^T \cdot \mathbf{J}_{D_i}^T \cdot \mathbf{I}_{L_i(\text{rot})|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \quad (73)$$

$$\mathbf{V}_{L_i(\text{rot})|_{E(\text{eq})}} = \mathbf{T}^T \cdot \mathbf{J}_{D_i}^T \cdot \frac{d}{dt} \left(\mathbf{I}_{L_i(\text{rot})|_B} \cdot \mathbf{J}_{D_i} \cdot \mathbf{T} \right) \quad (74)$$

These matrices represent the inertia matrix and the Coriolis and centripetal terms matrix of a virtual mobile platform that is equivalent to each rotating fixed-length link.

It should be noticed that equations (24), (38), (51) and (69) by providing expressions for the inertial component of the generalized force applied to {P} and expressed in {B} enable a clear physical meaning to the moments applied to {P}.

3.4 Gravitational component of the RCID dynamic model

Given a general frame $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$, with $\mathbf{z} \equiv -\hat{\mathbf{g}}$, the potential energy of a rigid body is given by:

$$P_c = m_c \cdot g \cdot z_c \quad (75)$$

where m_c is the body mass, g is the modulus of the gravitational acceleration and z_c the distance, along \mathbf{z} , from the frame origin to the body the centre of mass.

The gravitational components of the generalized forces acting on {P} can be easily obtained from the potential energy of the different bodies that compose the system:

$${}^P \mathbf{f}_{P(\text{gra})|_{B|E}} = \frac{\partial \mathcal{P}_P \left({}^B \mathbf{x}_P |_{B|E} \right)}{\partial {}^B \mathbf{x}_P |_{B|E}} \quad (76)$$

$${}^P \mathbf{f}_{A_i(\text{gra})|_{B|E}} = \frac{\partial \mathcal{P}_{A_i} \left({}^B \mathbf{x}_P |_{B|E} \right)}{\partial {}^B \mathbf{x}_P |_{B|E}} \quad (77)$$

$${}^P \mathbf{f}_{L_i(\text{gra})|_{B|E}} = \frac{\partial \mathcal{P}_{L_i} \left({}^B \mathbf{x}_P |_{B|E} \right)}{\partial {}^B \mathbf{x}_P |_{B|E}} \quad (78)$$

Vectors ${}^P \mathbf{f}_{P(\text{gra})|_{B|E}}$, ${}^P \mathbf{f}_{A_i(\text{gra})|_{B|E}}$ and ${}^P \mathbf{f}_{L_i(\text{gra})|_{B|E}}$ represent the gravitational components of the generalized forces acting on {P}, expressed using the Euler angles system, due to, in that order, the mobile platform, each actuator and each fixed-length link. Therefore, to be added to the inertial force components, these vectors must be transformed, to be expressed in frame {B}. This may be done pre-multiplying the gravitational components force vectors by the following matrix:

$$\begin{bmatrix} \mathfrak{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_A^{-T} \end{bmatrix} \quad (79)$$

4. Computational effort of the RCID dynamic model

The computational effort of the RCID dynamic model obtained through the use of the generalized momentum approach is compared with the one resulting from applying the Lagrange method using the Koditschek representation (Lebret et al., 1993; Koditschek, 1984). As the largest difference between the two methods rests on how the Coriolis and centripetal terms matrices are calculated, the two models are evaluated by the number of arithmetic operations involved in the computation of these matrices. The results were obtained using the symbolic computational software Maple[®] and are presented in Table 2.

	Lagrange			Generalized Momentum		
	Add.	Mult.	Div.	Add.	Mult.	Div.
Mobile platform	310	590	0	94	226	0
Six actuators	3028	4403	30	724	940	18
Translating link	751	1579	6	131	279	4
Rotating link	2180	3711	7	355	664	7
Total operations	20924	36733	108	3734	6824	84

Table 2. Number of arithmetic operations involved in the computation of the Coriolis and centripetal terms matrices of the RCID dynamic model

The dynamic model obtained by the generalized momentum approach is computationally much more efficient and its superiority manifests precisely in the computation of the matrices requiring the largest relative computational effort: the Coriolis and centripetal terms matrices.

The proposed approach was used in the dynamic modelling of a 6-dof parallel manipulator similar to a Stewart platform. Nevertheless, it can be applied to any mechanism.

5. Conclusion

Dynamic modelling of parallel manipulators presents an inherent complexity. Despite the intensive study in this topic of robotics, mostly conducted in the last two decades, additional research still has to be done in this area.

In this paper an approach based on the manipulator generalized momentum is explored and applied to the dynamic modelling of parallel manipulators. The generalized momentum is used to compute the inertial component of the generalized force acting on the mobile platform. Each manipulator rigid body may be considered and analyzed independently. Analytic expressions for the rigid bodies' inertia and Coriolis and centripetal matrices are obtained, which can be added, as they are expressed in the same frame. Having these matrices, the inertial component of the generalized force acting on the mobile platform may be easily computed. This component can be added to the gravitational part of the generalized force, which is obtained through the manipulator potential energy.

The proposed approach is completely general and can be used as a dynamic modelling tool applicable to any mechanism.

The obtained dynamic model was found to be computationally much more efficient than the one resulting from applying the Lagrange method using the Koditschek representation. Its superiority manifesting precisely in the computation of the matrices requiring the largest relative computational effort: the Coriolis and centripetal terms matrices.

6. References

- Bhattacharya, S.; Nenchev, D. & Uchiyama, M. (1998). A recursive formula for the inverse of the inertia matrix of a parallel manipulator. *Mechanism and Machine Theory*, Vol. 33, (957-964)
- Bruzzzone, L.; Molfino, R. & Zoppi, M. (2005) An impedance-controlled parallel robot for high-speed assembly of white goods. *Industrial Robot*, Vol. 32, (226-233)
- Caccavale, F.; Siciliano, B. & Villani, L. (2003). The Tricept Robot: Dynamics and Impedance Control. *IEEE/ASME Transactions on Mechatronics*, Vol. 8, (263-268)
- Callegari M.; Palpacelli, M. & Principi, M. (2006). Dynamics modelling and control of the 3-RCC translational platform. *Mechatronics*, Vol. 16, (589-605)
- Carvalho, J. & Ceccarelli, M. (2001). A Closed-Form Formulation for the Inverse Dynamics of a Cassino Parallel Manipulator. *Multibody System Dynamics*, Vol. 5, (185-210)
- Chablat, D.; Wenger, P.; Majou, F. & Merlet, J. (2004). An Interval Analysis Based Study for the Design and the Comparison of Three-Degrees-of-Freedom Parallel Kinematic Machines. *The International Journal of Robotics Research*, Vol. 23, (615-624)
- Constantinescu, D.; Salcudean, S. & Croft, E. (2005). Haptic Rendering of Rigid Contacts Using Impulsive and Penalty Forces. *IEEE Transactions on Robotics*, Vol. 21, (309-323)
- Dasgupta, B. & Choudhury, P. (1999). A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, Vol. 34, (801-824)
- Dasgupta, B. & Mruthyunjaya, T. (1998). A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator. *Mechanism and Machine Theory*, Vol. 34, (711-725)
- Do, W. & Yang, D. (1988). Inverse Dynamic Analysis and Simulation of a Platform Type of Robot. *Journal of Robotic Systems*, Vol. 5, (209-227)
- Featherstone, R. & Orin, D. (2000). Robot dynamics: equations and algorithms, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 826-834
- Gallardo, J.; Rico, J.; Frisoli, A.; Checcacci, D. & Bergamasco, M. (2003) Dynamics of parallel manipulators by means of screw theory. *Mechanism and Machine Theory*, Vol. 38, (1113-1131)
- Geng, Z.; Haynes, L.; Lee, J. & Carroll, R. (1992). On the dynamic model and kinematics analysis of a class of Stewart platforms. *Robotics and Autonomous Systems*, Vol. 9, (237-254)
- Gregório, R. & Parenti-Castelli, V. (2004). Dynamics of a Class of Parallel Wrists. *Journal of Mechanical Design*, Vol. 126, (436-441)
- Guo, H. & Li, H. (2006). Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 220, (61-72)
- Ji, Z. (1994). Dynamics Decomposition for Stewart Platforms. *ASME Journal of Mechanical Design*, Vol. 116, (67-69)
- Khalil, W. & Guegan, S. (2004). Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. *IEEE Transactions on Robotics*, Vol. 20, (754- 761)
- Khalil, W. & Ibrahim, O. (2007), General Solution for the Dynamic Modelling of Parallel Robots. *Journal of Intelligent and Robot Systems*, Vol. 49, (19-37)
- Kim, J.; Hwang, J.; Kim, J.; Iurascu, C.; Park, F. & Cho, Y. (2002). Eclipse II: A New Parallel Mechanism Enabling Continuous 360-Degree Spinning Plus Three-Axis Translational Motions. *IEEE Transactions on robotics and automation*, Vol. 18, (367-373)
- Koditschek, D. (1984) Natural Motion for Robot Arms, *Proceedings of 23rd Conference on Decision and Control*, pp. 733-735

- Lebret, G.; Liu, K. & Lewis, F. (1993). Dynamic Analysis and Control of a Stewart Platform Manipulator. *Journal of Robotic Systems*, Vol. 10, (629-655)
- Lee, K. & Chirikjian, G. (2005). A New Perspective on $O(n)$ Mass-Matrix Inversion for Serial Revolute Manipulators, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4722-4726
- Li, Y. & Xu, Q. (2005). Kinematics and inverse dynamics analysis for a general 3-PRS spatial parallel mechanism. *Robotica*, Vol. 23, (219-229)
- Lilly, K. (1993). *Efficient Dynamic Simulation of Robotic Mechanisms*, Kluwer Academic Publishers, Dordrecht, Netherlands
- Liu, K.; Lewis, F.; Lebret, G. & Taylor, D. (1993). The Singularities and Dynamics of a Stewart Platform Manipulator. *Journal of Intelligent and Robotic Systems*, Vol. 8, (287-308)
- Lopes, A. & Almeida, F. (1996). Manipulability Optimization of a Parallel Structure Robotic Manipulator, *Proc. of the 2nd Portuguese Conference on Automatic Control*, pp. 243-248
- Mata, V.; Provenzano, S.; Valero, F. & Cuadrado, J. (2002). Serial-robot dynamics algorithms for moderately large numbers of joints. *Mechanism and Machine Theory*, Vol. 37, (739-755)
- Merlet, J. & Gosselin, C. (1991). Nouvelle Architecture pour un Manipulateur Parallèle à Six Degrés de Liberté. *Mechanism and Machine Theory*, Vol. 26, (77-90)
- Merlet, J. (2006). *Parallel robots*, Springer, Dordrecht
- Mouly, N. (1993). Développement d'une Famille de Robots Parallèles à Motorisation Électrique. *Thèse de Doctorat*, École des Mines de Paris
- Naudet, J.; Lefeber, D.; Daerden, F. & Terze Z. (2003). Forward Dynamics of Open-Loop Multibody Mechanisms Using an Efficient Recursive Algorithm Based on Canonical Momenta. *Multibody System Dynamics*, Vol. 10, (45-59)
- Nguyen, C. & Pooran, F. (1989). Dynamic Analysis of a 6 DOF CKCM Robot End-Effector for Dual-Arm Telerobot Systems. *Robotics and Autonomous Systems*, Vol. 5, (377-394)
- Pernette, E.; Henein, S.; Magnani, I. & Clavel, R. (2000). Design of parallel robots in microrobotics. *Robotica*, Vol. 15, (417-420)
- Reboulet, C. & Berthomieu, T. (1991). Dynamic Models of a Six Degree of Freedom Parallel Manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1153-1157
- Riebe, S. & Ulbrich, H. (2003). Modelling and online computation of the dynamics of a parallel kinematic with six degrees-of-freedom. *Archive of Applied Mechanics*, Vol. 72, (817-829)
- Sciavicco, L. & Siciliano, B. (1996). *Modeling and Control of Robot Manipulators*, McGraw-Hill International Editions, New York
- Staicu, S.; Liu, X. & Wang, J. (2007). Inverse dynamics of the HALF parallel manipulator with revolute actuators. *Nonlinear Dynamics*, Vol. 50, (1-12)
- Torby, B. (1984). *Advanced Dynamics for Engineers*, CBS College Publishing, New York
- Tsai, L. (2000). Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work. *Journal of Mechanical Design*, Vol. 122, (3-9)
- Vischer, P. & Clavel, R. (2000). Argos: A Novel 3-DoF Parallel Wrist Mechanism. *The International Journal of Robotics Research*, Vol. 19, (5-11)
- Vukobratovic, M. & Kircanski, M. (1986). *Kinematics and Trajectory Synthesis of Manipulation Robots*, Springer-Verlag, Berlin
- Wang, J. & Gosselin, C. (1998). A New Approach for the Dynamic Analysis of Parallel Manipulators. *Multibody System Dynamics*, Vol. 2, (317-334)
- Zhang, D. & Gosselin, C. (2002). Kinetostatic Analysis and Design Optimization of the Tricept Machine Tool Family. *Journal of Manufacturing Science and Engineering*, Vol. 124, (725-733)

Redundant Actuation of Parallel Manipulators

Andreas Müller

*Institute of Mechatronics at the Chemnitz University of Technology
Germany*

1. Introduction

High stiffness, low inertia, large accelerations, and high precision are desirable properties attributed to parallel kinematics machines (PKM). However, relatively small workspace and the abundance of singularities within the workspace partly annihilate the aforementioned advantages. Redundant actuation and novel redundant kinematics are means to tackle these shortcomings. Redundant parallel kinematics machines are ideal candidates for use in high-precision applications, such as robot-assisted surgery. Their advantageous features promise to deliver the needed accuracy, stiffness, dexterity and reliability. Redundant actuation admits to eliminate singularities, increase the usable workspace, augment the dexterity, and partially control the internal forces. Actuator redundancy is also a means to improve fault tolerance, as redundant actuators can compensate the failure of other actuators. Redundant actuation increases the payload and acceleration, can yield an optimal load distribution among the actuators, or can reduce the power consumption of the individual drives. Actuator redundancy can also improve the force transmission properties and the manipulator stiffness. It can be purposefully exploited for secondary tasks, such as the generation of internal prestress and the generation of a desired compliance of the PKM. The first can be used to avoid backlash, whereas the second admits to homogenize the stiffness properties within the workspace. Kinematically redundant PKM, i.e. systems that possess a higher mobility than required for the task, allow to circumvent singularities as well as obstacles, and to increase the dexterity.

The control of redundantly actuated PKM poses additional challenges, rooted in the resolution of the redundancy within the control schemes. Whereas, model-based control techniques can be directly applied to the control of non-redundantly actuated PKM, redundancy, however, brings up two specific problems, one is the computationally efficient resolution of the actuation redundancy, and the other is the occurrence of unintentional antagonistic actuation due to model uncertainties.

This chapter is devoted to the modeling and control of redundantly actuated PKM. The aim of the chapter is to summarize concepts for dynamic modeling of redundantly actuated PKM, with emphasis on the inverse dynamics and control, and to clarify the terminology used in the context of redundant actuation. Based on a mathematical model, PKM are regarded as non-linear control systems.

The chapter is organized as follows. A short literature review in section 2 is meant to familiarize the reader with current developments and research directions. In order to point out the potential of redundantly actuated PKM, a motivating example is given in section 3.

The PKM motion equations are recalled in section 4 as basis for the subsequent considerations. The associated non-linear control problem is formulated in section 5, and used for the definition of actuation and redundancy in section 6. Section 7 is devoted to the resolution of actuator redundancy. For the important case of simply-redundant actuation a closed form solution to the inverse dynamics problem is given, and actuator redundancy is exploited for secondary tasks. The applicability of standard model based control schemes to redundantly actuated PKM is studied in section 8. The effect of geometric uncertainties is analyzed and shown to lead to interference effects that are peculiar to redundantly actuated PKM. An amended version of the augmented PD and computed torque control schemes is proposed that eliminates these effects. The chapter closes with a conclusion and hints to open problems in section 9.

2. Literature review

Compared to serial manipulators PKM exhibit a much richer phenomenology, and give rise to more types of redundancy. A brief overview of redundancy in PKM can be found in (Merlet, 1996). Redundantly actuated PKM were analyzed with regard to their kinematic and dynamic properties, and in view of singularities in (Alba et al., 2007; Dasgupta & Mruthyunjaya, 1998; Firmani & Podhorodeski, 2004; Gardner et al., 1989; Kim et al., 2001; Kock & Schumacher, 1998; Kurtz & Hayward, 1992; Liao et al., 2004; Mohamed & Gosselin, 1999; Müller, 2005; O'Brien & Wen, 1999; Valasek, 2002; Zhang et al., 2007).

It was shown by a number of authors that redundant actuation is a means to eliminate singularities and so enlarges the usable workspace. Redundant actuation can be achieved in different ways, and there are two directions: the actuation of passive joints, and the inclusion of additional kinematic chains without increasing the PKM DOF. Most authors propose using additional chains, such as planar 3RRR (Alba et al., 2007; Buttolo & Hannaford, 2005; Kock & Schumacher, 1998), planar 4RRR (Valasek, 2002), spherical wrists (Kurtz & Hayward, 1992) and shoulder (Yi et al., 1994), Stewart platforms with one (O'Brien & Wen, 1999) or two (Valasek, 2002) additional struts, or the Eclipse (Kim et al., 2001). The improvement of kinematic manipulability or dexterity via redundant actuation has been investigated in (O'Brien & Wen, 1999). The optimal design of a robotic wrist aiming to maximize manipulability was addressed in (Kurtz & Hayward, 1992). Actuation redundancy was successfully applied to maximize and to homogenize the force output of a haptic force display (Buttolo & Hannaford, 2005). Other redundantly actuated PKM were developed for use as robot hands (Lee et al., 1998). Temporarily redundant actuation was proposed as a way to cope with singularities (Ganovski et al., 2004). The basic idea was to equip the PKM with more drives than needed, and to activate the 'excess drives' whenever the main drives are unable to properly control the machine. Systems with variable topology are also temporarily redundantly. The inverse dynamics of such systems was addressed in (Nahon & Angeles, 1989).

Kinematically redundant PKM possess multiple inverse kinematics solutions that can be applied for various purposes, such as maximizing dexterity or stiffness, avoiding singularities or obstacles, and minimizing drive power or the overall joint motions. A singularity avoiding inverse kinematics algorithm was proposed in (Alba et al., 2007) for a kinematically redundant planar 3RRR positioning PKM, where the concept of feasibility maps for serial manipulators, was adopted for the identification of working modes of PKM, i.e. singularity-free regions in joint space. Kinematic redundancy was further used in

(Mohamed & Gosselin, 1999) as a means to reshape the manipulator's platform. Kinematic calibration and model identification is much more involved as shown in (Jeong et al., 2004) and (Abdellatif et al., 2007).

The force capability of redundantly actuated PKM were investigated in (Nokleby et al., 2005). A peculiarity of redundantly actuated PKM is the ability to generate internal prestress, via antagonistic control of the redundant drives, without generating end-effector forces. This feature was employed in (Chakarov, 2004; Kock & Schumacher, 1998; Kock & Schumacher, 2000; Müller, 2006; Yi et al., 1994) for the generation of a desired (tangential) EE-stiffness. Prestress was further used for the avoidance of joint backlash, which is critical in the presence of joint clearing and DC motor hysteresis (Müller, 2005; Valasek, 2002).

Dynamic modelling is crucial for control of redundantly actuated PKM. Modelling and control were addressed in (Cheng et al, 2003; Liu et al., 2003; Müller, 2005; Nakamura & Ghodoussi, 1989). In (Cheng et al, 2003) model based motion control of redundantly actuated PKM was considered, and it was proposed to adopt the established computed torque and augmented PD control schemes (Murray, et al., 1993). In (Garrido, 2004) these control schemes were extended by allowing for measurement uncertainties, and in (Gourdeau et al., 1999) a computed torque control scheme without velocity measurement was proposed. An important issue for the inverse dynamics of redundantly actuated PKM is a goal-oriented resolution of the redundancy. The resolution is achieved using a weighed pseudoinverse, which is however computationally expensive to evaluate. For simply redundantly actuated PKM a closed form solution for the pseudoinverse was presented in (Müller, 2005).

Another type of redundancy is related to the placement of sensors. Sensor redundancy was shown in (Yiu & Li, 2003) to be beneficial for the solution of the forward kinematics problem.

3. A motivating example

For demonstration purpose consider the $RP/2RPR$ PKM in figure 1. This PKM has the DOF 2, and is controlled by actuation of the three prismatic joints. The PKM could be uniquely positioned using two of the prismatic joints only. Therefore, the $RP/2RPR$ is redundantly full-actuated.

Manipulability/Dexterity/Singularities: An attractive feature of redundantly actuated PKM is the fact that singularities are eliminated that would occur in the non-redundant counterpart. A PKM is in a singularity if the EE-motion can not be determined by the actuated joints, reflected by a drop of its manipulability. Assumed, however, that an additional actuated kinematic chain is suitably attached to the moving platform, it shall be possible to overcome the indeterminacy.

The kinematic capability of robotic manipulators can be quantified by manipulability or dexterity measures, introduced in (Murray, et al., 1993) and (Yoshikawa, 1985), that characterize the velocity and force transformation. The EE-twist is determined by a set of independent joint velocities via $\mathbf{V} = \mathbf{J}_E \dot{\mathbf{q}}_2$, and away from singularities, $\dot{\mathbf{q}}_2 = \mathbf{J}_E^{-1} \mathbf{V}$. For redundant PKM there are more velocities of actuated joints than independent ones. But, there exists a Matrix \mathbf{A} , such that the actuator velocities are $\dot{\mathbf{q}}_a = \mathbf{A} \dot{\mathbf{q}}_2 = \mathbf{A} \mathbf{J}_E^{-1} \mathbf{V}$ (section 4). Hence, the manipulability measures

$$\mu' := \det(\mathbf{J}_E^{-T} \mathbf{A}^T \mathbf{A} \mathbf{J}_E^{-1}), \quad \mu'' := 1/\text{cond}(\mathbf{J}_E^{-T} \mathbf{A}^T \mathbf{A} \mathbf{J}_E^{-1})$$

characterize the velocity transmission from EE to actuators. Figure 2 shows the distribution of the two measures in the main part of the workspace of the $\text{RP}/2\text{RPR}$ PKM. For comparison, the manipulability measures are also shown for a non-redundant RP/RPR PKM. The $\text{RP}/2\text{RPR}$ PKM arises from the non-redundant PKM by addition of another RPR limb. Obviously the manipulability of the $\text{RP}/2\text{RPR}$ PKM is much higher and more homogeneously distributed in the workspace. In particular, the singularities of the RP/RPR PKM, at the bottom of the workspace, are removed (singularities are marked by vanishing manipulability measure).

Actuator loads: Beside eliminating singularities, additional redundant actuators allow to distribute the required work load among the drives. In this way, the individual drive loads can be reduced. The resolution and optimal distribution of control forces among the drives is achieved by a strategic inverse dynamics, as derived in section 7. In the $\text{RP}/2\text{RPR}$ example, the third strut compensates a large part of EE-loads, that cause high control forces in the RP/RPR PKM. Clearly, redundant actuation increases the dynamical capability of the PKM.

Stiffness/Compliance: Under working conditions, the accuracy of PKM is strongly related to its structural stiffness, and a realistic analysis must take into account the link flexibility. Using the same argument as for the load distribution among the drives, it is clear that the overall EE-stiffness increases with the addition of redundant struts. Clearly, the stiffness apparent at the EE depends on the PKM's pose.

Fault tolerance: It is clear from the manipulability analysis of the $\text{RP}/2\text{RPR}$ PKM that the system is manipulable even if one of the actuators fails. For example, if the third actuator fails, then the PKM is still maneuverable as a RP/RPR manipulator, apart from singular postures.

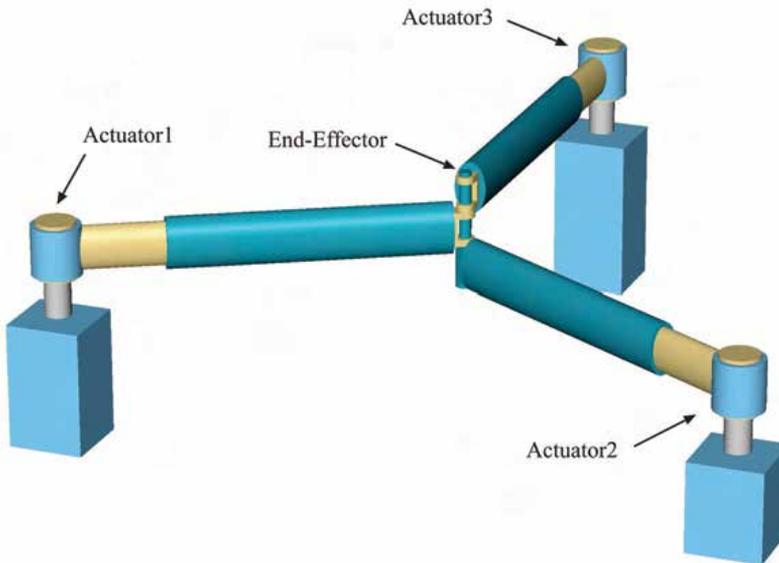


Figure 1. Redundantly full-actuated planar $\text{RP}/2\text{RPR}$ manipulator.

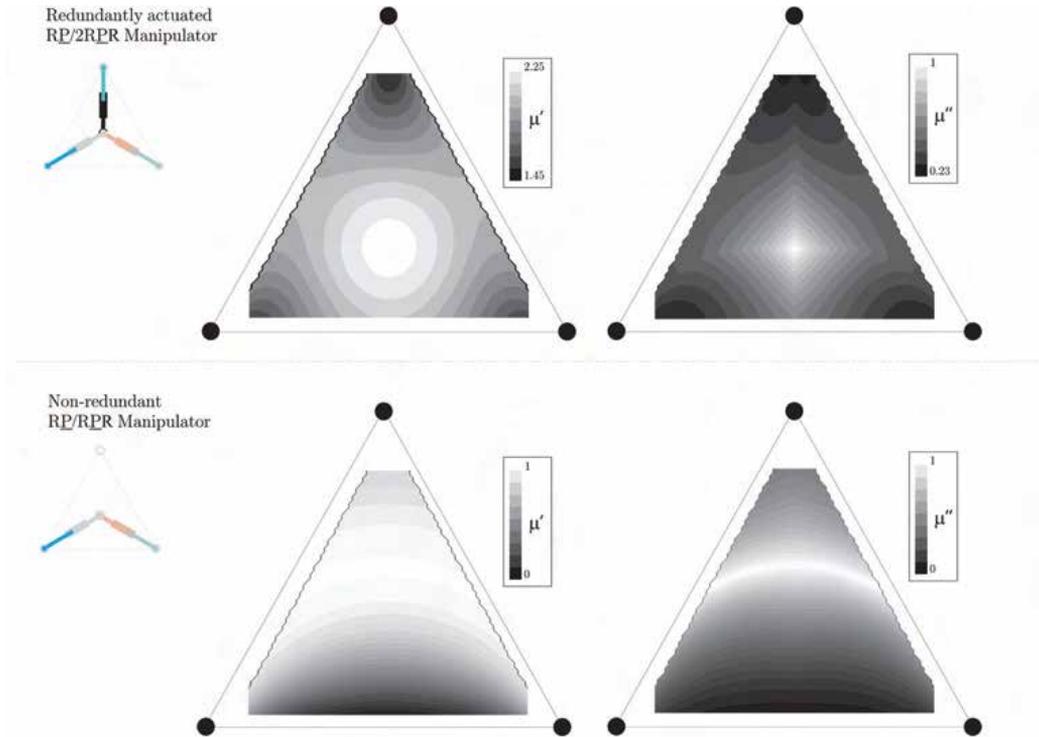


Figure 2. Manipulability distribution for the redundantly actuated $\underline{R}\underline{P}/2\underline{R}\underline{P}\underline{R}$ and the non-redundant $\underline{R}\underline{P}/\underline{R}\underline{P}\underline{R}$ manipulator.

4. Dynamic modeling

A PKM is a controlled, holonomically constrained multibody system (MBS), where the constraints embody the geometric closure conditions of kinematic loops. In applications where the manipulator interacts with its environment, the PKM is subject to additional possibly non-holonomic constraints. The latter will not be taken into account here.

The Lagrangian motion equations of second kind for a PKM can be derived with the standard methods for MBS with kinematic loops (Maisser, 1997; Müller, 2006; Papastavridis, 2002) as it was pursued in (Cheng et al, 2003; Liu et al., 2003; Müller, 2005; Nakamura & Ghodoussi, 1989). This proceeds by transforming the MBS with kinematic loops into an MBS with tree topology, subject to the closure constraint that enforces the loop closure. In each fundamental loop of the topological graph one joint (the cut-joint) is removed, and corresponding cutjoint constraints (closure conditions) are imposed to the resulting MBS with tree topology (Müller, 2006). Figure 3 shows the topological graph of the $\underline{R}\underline{P}/2\underline{R}\underline{P}\underline{R}$ PKM in figure 1. Two fundamental loops can be identified according to the indicated cut-joints. Each loop gives rise to two closure constraints.

Denote with $\mathbf{q} \in \mathbb{V}^n$ the vector of joint variables q^a , $a = 1, \dots, n$ (higher DOF joints are split into one DOF joints) of the tree MBS, where $\mathbb{V}^n := \mathbb{T}^{nR} \times \mathbb{R}^{nP}$ if the PKM comprises nR revolute and nP prismatic/screw joints. $\mathbf{q} \in \mathbb{V}^n$ is called the configuration of the PKM. A

configuration is admissible only if it fulfils the r geometric loop closure conditions. Now, the fundamental loops give rise to a set of r geometric constraint $0 = \mathbf{h}(\mathbf{q})$, $\mathbf{h}(\mathbf{q}) \in \mathbb{R}^r$. In case of the RP/2RPR PKM in figure 3 this is a system of 4 constraints for the $n = 6$ joint variables of the tree system. Time differentiation yields the kinematic constraints

$$\mathbf{0} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad \mathbf{J}(\mathbf{q}) \in \mathbb{R}^{r,n} \quad (1)$$

The geometric constraints define the configuration space of the PKM

$$V := \{\mathbf{q} \in \mathbb{V}^n | \mathbf{h}(\mathbf{q}) = \mathbf{0}\}. \quad (2)$$

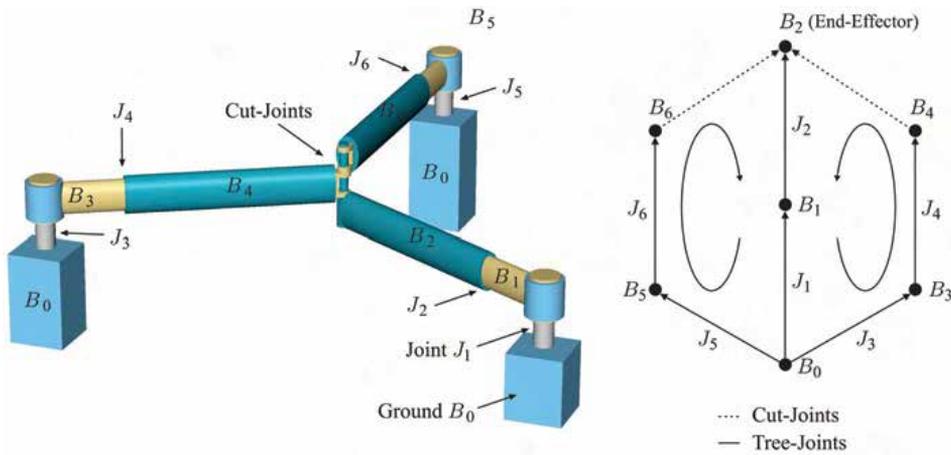


Figure 3. Topological graph, and spanning tree of the RP/2RPR manipulator.

The configuration space is the set of all admissible configurations of the PKM. V is an analytic variety and only locally a smooth manifold. The manifolds are separated by the singular points of V , where the rank of \mathbf{J} changes. The latter are called c-space singularities. Their determination is vital for a reliable operation of the PKM. If the r constraints are locally independent, the local DOF of the PKM is $\delta := n - r$.

The admissible configuration \mathbf{q} is locally determined by $\delta := n - r$ independent generalized coordinates. Denoting the vector of dependent and independent variables respectively with \mathbf{q}_1 and \mathbf{q}_2 , the kinematic constraints are

$$\mathbf{J}_1 \dot{\mathbf{q}}_1 + \mathbf{J}_2 \dot{\mathbf{q}}_2 = \mathbf{0}, \quad (3)$$

where $\mathbf{J} = (\mathbf{J}_1, \mathbf{J}_2)$, with $\mathbf{J}_1(\mathbf{q}) \in \mathbb{R}^{r,r}$, $\mathbf{J}_2(\mathbf{q}) \in \mathbb{R}^{r,\delta}$. The independent coordinates can be chosen so that \mathbf{J}_2 is full rank, and the generalized velocities are

$$\dot{\mathbf{q}} = \mathbf{F} \dot{\mathbf{q}}_2, \quad \text{where } \mathbf{F} := \begin{pmatrix} -\mathbf{J}_1^{-1} \mathbf{J}_2 \\ \mathbf{I}_\delta \end{pmatrix}. \quad (4)$$

\mathbf{F} is an orthogonal complement of \mathbf{J} , i.e. $\mathbf{J}\mathbf{F} \equiv 0$. The accelerations follow with $\ddot{\mathbf{q}} = \mathbf{F}\ddot{\mathbf{q}}_2 + \dot{\mathbf{F}}\dot{\mathbf{q}}_2$. The constituent feature of any PKM is that a moving platform, carrying an end-effector (EE), is connected to the base by several (possibly identical) kinematic chains (limbs, struts, legs) containing actuated joints. The EE is represented by an EE-frame. The configuration of the EE-frame w.r.t. a inertial (world) frame is represented by $\mathbf{C} \in SE(3)$. The **EE-map** $f_E: \mathbb{V}^n \rightarrow SE(3)$, gives the EE-configuration $\mathbf{C} = f_E(\mathbf{q})$ in the configuration \mathbf{q} . The workspace of the PKM is

$$W := \{f_E(\mathbf{q}) \mid \mathbf{q} \in V\} \subset SE(3). \quad (5)$$

The **EE-Jacobian** $\mathbf{J}_E(\mathbf{q}) : T_{\mathbf{q}}\mathbb{V}^n \rightarrow se(3)$ yields the EE-twist $\mathbf{V} = \mathbf{J}_E(\mathbf{q})\dot{\mathbf{q}}$, in terms of the state of the PKM. If $\boldsymbol{\tau} \in se^*(3)$ is an EE-wrench, then $\mathbf{Q} = \mathbf{J}_E^T \boldsymbol{\tau}$ is the corresponding vector of generalized forces.

Now, the dynamics of a force-controlled holonomic constrained MBS with kinematical tree structure is governed by the Lagrangian motion equations

$$\mathbf{G}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{J}_E^T(\mathbf{q})\boldsymbol{\tau} + \mathbf{J}^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{u}, \quad (6)$$

where \mathbf{G} is the generalized mass matrix, $\mathbf{C}\dot{\mathbf{q}}$ represents generalized Coriolis and centrifugal forces, \mathbf{Q} represents all remaining, including generalized potential forces, and \mathbf{u} are the generalized control forces. The Lagrange multipliers $\boldsymbol{\lambda}$ can be identified with the constraint reactions in cut-joints.

For a PKM some of the possible control forces in \mathbf{u} are identically zero, and only m control forces corresponding to active joints are present. Denote with $\mathbf{c} \equiv (c_1, \dots, c_m)$ the vector of generalized control forces in the actuated joints. Let \mathbf{A} be the relevant part of \mathbf{F} so that $\mathbf{F}^T \mathbf{u} = \mathbf{A}^T \mathbf{c}$. Projecting the Lagrangian equations (6) onto the configuration space V , with the help of the orthogonal complement \mathbf{F} and the relation (4), yields the Voronets equations (Maisser, 1997; Papastavridis, 2002)

$$\mathbf{G}(\mathbf{q})\ddot{\mathbf{q}}_2 + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_2 + \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{J}_E^T(\mathbf{q})\boldsymbol{\tau} = \mathbf{A}^T \mathbf{c}, \quad (7)$$

where

$$\mathbf{G} := \mathbf{F}^T \mathbf{G} \mathbf{F}, \quad \mathbf{C} := \mathbf{F}^T (\mathbf{C} \mathbf{F} + \mathbf{G} \dot{\mathbf{F}}), \quad \mathbf{Q} := \mathbf{F}^T \mathbf{Q}, \quad \mathbf{J}_E := \mathbf{J}_E \mathbf{F}.$$

Clearly, only those \mathbf{c} that are not in the kernel of \mathbf{A}^T are effective control forces. The system (7) together with the kinematic constraints in (1) yield n differential equations in $\mathbf{q} \in \mathbb{V}^n$, that completely determine the MBS dynamics.

Note that the motion equations are formulated in terms of minimal coordinates $\mathbf{q} \in \mathbb{V}^n$, for the purpose of deriving an unconstrained control system. One has to be cautious, however, since the removal of cut-joints can lead to dependent closure constraints that have no physical meaning. These artifacts are merely due to the parameterization. Geometrically, V

is only a section of the 'complete' configuration space, including cut-joint variables. This issue is important for model based control, as pointed out in (Liu et al., 2003), since the actual controller is built upon the model (7), i.e. using a certain V . In fact, it may be necessary to switch between PKM models with different cut-joints.

5. The associated non-linear control systems

A PKM is a force-controlled holonomically constrained dynamical system, whose dynamics is governed by (7). The control purpose is to manipulate the EE, which embodies the system's primary output. A PKM can be regarded as a second order control-affine control system on the configuration space V , which can be transformed to the first order control system on the $2n$ -dimensional state space TV

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^m \mathbf{g}_i(\mathbf{x}) c^i \quad (8)$$

$$\mathbf{C} = f_E(\mathbf{x})$$

with state vector $\mathbf{x} := (\mathbf{q}_2, \dot{\mathbf{q}}_2)$. Therein

$$\mathbf{f} := \begin{pmatrix} \dot{\mathbf{q}}_2 \\ -\mathbf{G}^{-1}(\mathbf{C}\dot{\mathbf{q}}_2 + \mathbf{Q} + \mathbf{J}_E^T \boldsymbol{\tau}) \end{pmatrix}, \quad (9)$$

is the drift vector field, and the columns \mathbf{g}_i , $i = 1, \dots, m \leq n$ of

$$\mathbf{g} := \begin{pmatrix} \mathbf{0} \\ \mathbf{G}^{-1} \mathbf{A}^T \end{pmatrix} \quad (10)$$

define the control vector fields, via which the control forces affect the system.

From a control point of view, one is interested in the controllability and observability of the PKM. That is, one is concerned with whether the PKM can be steered between two given configurations (Nijmeijer & van der Schaft).

6. Actuation and redundancy

The terms actuation and redundancy are differently used in the literature. In order clarify this notion a stringent definition is given based on the above control system. The following definitions refer to a regular configuration \mathbf{q} , i.e. a configuration for which the orthogonal complement \mathbf{F} and its submatrix \mathbf{A} has full rank in a neighborhood of \mathbf{q} in V . The dependence on \mathbf{q} is omitted, and δ denotes the DOF.

Definition 1. The rank of the input vector field is called the **degree of actuation** (DOA)

$$\alpha := \text{rank}(\mathbf{g}) = \text{rank}(\mathbf{A}).$$

If $\alpha < \delta$ the PKM is **underactuated** and if $\alpha = \delta_{\text{loc}}$ the PKM is **full-actuated**. The **degree of redundancy** of the actuation is $\rho_\alpha := m - \alpha$. The PKM is called **redundantly actuated** if $\rho_\alpha > 0$ and **nonredundantly actuated** if $\rho_\alpha = 0$.

Actuation refers to the effect that control forces have on the state change of a system. The above definition is in accordance with this notion, though it refers to the ability to influence the PKM's acceleration. This is so because a PKM (as considered here) is a holonomically constrained system, so that prescribing the acceleration also determines the velocity and configuration, with known initial conditions. Actuation is a pointwise property, and the DOA changes in singular configurations. The effect of the actuation on the motion is described by the controllability of the system. This is a local property, i.e. considering the effects over a small time (Nijmeijer & van der Schaft). Redundantly actuated PKM are occasionally termed 'overactuated'. Notwithstanding that redundantly actuated PKM can be underactuated, a full-actuated PKM is completely actuated, and an improvement is impossible. Therefore, the term 'overactuation' makes no sense.

7. Resolution of actuation redundancy

7.1 Inverse dynamics of redundantly full-actuated PKM

The first step in navigating PKM consists in task/motion planning and a subsequent solution of the inverse kinematics, i.e. the determination of required actuator motions. The inverse dynamics problem is to determine the actuator forces required for this motion to take place. The DOA of a full-actuated PKM equals its DOF ($\delta = \alpha$). The number $m = \delta + \rho$ of active drives of a redundantly full-actuated PKM exceeds its DOF by ρ . Without loss of generality, the joint variables can be arranged as $\mathbf{q} \equiv (\mathbf{q}_p, \mathbf{q}_a)$, with $\mathbf{q}_a \equiv (\dots, \mathbf{q}_2)$. Accordingly, the generalized control vector has the form $\mathbf{u} = (\mathbf{0}, \mathbf{c})$, with $\mathbf{c} \equiv (c_1, \dots, c_m)$. The orthogonal complement takes on the form

$$\mathbf{F} = \begin{pmatrix} -\mathbf{J}_1^{-1} \mathbf{J}_2 \\ \mathbf{I}_\delta \end{pmatrix} \equiv \begin{pmatrix} \mathbf{P} \\ \mathbf{A} \end{pmatrix}, \quad (11)$$

$$\mathbf{P}(\mathbf{q}) \in \mathbb{R}^{n-m, \delta}, \quad \mathbf{A}(\mathbf{q}) = \begin{pmatrix} \mathbf{A}_1(\mathbf{q}) \\ \mathbf{I}_\delta \end{pmatrix} \in \mathbb{R}^{m, \delta}, \quad \mathbf{A}_1(\mathbf{q}) \in \mathbb{R}^{\rho, \delta},$$

where \mathbf{P} contains the first $n - m$ and \mathbf{A}_1 the remaining rows of $-\mathbf{J}_1^{-1} \mathbf{J}_2$. \mathbf{A} is full rank δ . The kernel of \mathbf{A}^T is ρ -dimensional, so that (7) can not be uniquely solved for the controls \mathbf{c} . As consequence, 1) the load distribution among the drives is not unique, and 2) one can generate control forces in the null-space of \mathbf{A}^T that have no effect on the motion, so-called prestress.

Let $\mathbf{c}^0 \in \mathbb{R}^m$ be a desired prestress vector, then a solution for the controls \mathbf{c} such that $(\mathbf{c} - \mathbf{c}^0)^T \mathbf{M}(\mathbf{c} - \mathbf{c}^0) \rightarrow \min$ is

$$\mathbf{c} = (\mathbf{A}^T)_M^+ (\overline{\mathbf{G}}(\mathbf{q}) \ddot{\mathbf{q}}_2 + \overline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_2 + \overline{\mathbf{Q}} + \overline{\mathbf{J}}^T(\mathbf{q}) \boldsymbol{\tau}) + \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 \quad (12)$$

where $(\mathbf{A}^T)_M^+ := \mathbf{M}^{-1} \mathbf{A} (\mathbf{A}^T \mathbf{M}^{-1} \mathbf{A})^{-1}$ is the weighed right pseudoinverse, and $\mathbf{N}_{\mathbf{A}^T} := (\mathbf{I}_m - (\mathbf{A}^T)^+ \mathbf{A}^T)$ is a projector to the null-space of \mathbf{A}^T . Complete knowledge of the system and the EE-load $\boldsymbol{\lambda}$ is assumed. \mathbf{M} is a positive definite weighting matrix for the drive forces.

For the important case of simply redundant actuation ($\rho = 1$) a close form solution, with $\mathbf{M} = \mathbf{I}$, was derived in (Müller, 2005). In this case \mathbf{A}_1 is a row vector, and

$$\begin{aligned} (\mathbf{A}^T)^+ &= \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{I}_\delta \end{pmatrix} \left(\mathbf{I}_\delta - \frac{1}{1 + \|\mathbf{A}_1\|^2} \mathbf{A}_1^T \mathbf{A}_1 \right) \\ \mathbf{N}_{\mathbf{A}^T} &= \begin{pmatrix} \mathbf{I}_\rho \\ -\mathbf{A}_1^T \end{pmatrix}. \end{aligned} \quad (13)$$

Note that no matrix inversion is necessary, which is numerically advantageous.

On the basis of a preceding path planing and inverse kinematics solution of redundantly actuated PKM, the inverse dynamics is not unique and can take into account various goals. Actuation redundancy can be used to reduce the load of individual drives by strategically distributing the required control forces. On the other hand, the null-space components of the control forces can be employed for 'secondary' tasks.

7.2 Optimal distribution of control forces

An immediate application of the redundancy is a purposeful allocation of the control forces (Kock & Schumacher, 1998). This is achieved via the weighing matrix \mathbf{M} . Without prestress, i.e. with $\mathbf{c}^0 = 0$, the inverse dynamics solution (12) is such that $\mathbf{c}^T \mathbf{M} \mathbf{c} \rightarrow \min$. Usually \mathbf{M} is a diagonal matrix, and its entries scale the control forces according to their drive performances. The lower the force capability of a drive the higher its weight. E.g., one can think of a lowpowered redundant drive, used to balance and reduce otherwise high force peaks in the main drives.

Note, that these force considerations are essentially static, and do not take into account the PKM dynamics. For highly dynamic applications, the driving power distribution may significantly differ from the force distribution.

7.3 Backlash avoiding control

In (Müller, 2005; Valasek, 2002; Valasek, 2002) it was proposed to use internal prestress \mathbf{c}^0 to avoid actuator backlash, which refers to situations, where the sign of the control forces changes. One practical motivation for this is to eliminate the negative effects of joint clearings, and another is rooted in the observation of DC motor hysteresis. Also, for tendon driven PKM actuator signs must remain constant.

The main idea is to include the generation of internal prestress in the control scheme of the PKM. The condition for backlash free control is that the magnitude of each particular control force c_a remains above a certain level c_a^{\min} and that its sign remains constant during the considered task with a duration T . Denote with $s_a \in \{-1, 1\}$ the required sign of c_a , then the condition

$$s_a c_a(t) \geq c_a^{\min}, \quad t \in [0, T] \quad (14)$$

must be satisfied with $c_a^{\min} > 0$.

In (Müller, 2005) a method for backlash avoiding control of simply redundantly actuated PKM was presented. In this case \mathbf{A}^T has a one-dimensional null-space that can be parameterized by a prestress parameter $\sigma(t)$, so that

$$\mathbf{c} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{I}_\delta \end{pmatrix} \left(\mathbf{I}_\delta - \frac{1}{1+\|\mathbf{A}_1\|^2} \mathbf{A}_1^T \mathbf{A}_1 \right) \boldsymbol{\varphi} + \sigma \begin{pmatrix} \mathbf{I}_\rho \\ -\mathbf{A}_1^T \end{pmatrix} \quad (15)$$

with $\boldsymbol{\varphi} := \bar{\mathbf{G}}(\mathbf{q}) \ddot{\mathbf{q}}_2 + \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_2 + \bar{\mathbf{Q}}$.

Given a prescribed trajectory $\mathbf{q}^d(t)$, the control problem at time instant t_i consists in determining the prestress parameter $\sigma(t_i)$ such that (14) holds and an objective functional $L(\mathbf{q}^d(t_i), \sigma(t_i))$ is minimized. The latter can be the weighed sum of squared control forces or the overall driving power. In summary the one-dimensional optimization problem

$$\left\{ \begin{array}{l} L \rightarrow \min, t \in [0, T] \\ s_a c_a \geq c_a^{\min} \\ s_a c_a \leq c_a^{\max} \end{array} \right\} \quad (16)$$

with $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \sigma)$ in (15), has to be solved at any time step. This can either be solved independently at each time instant, or the σ can be approximated as a function of time, which results in a smoother behavior.

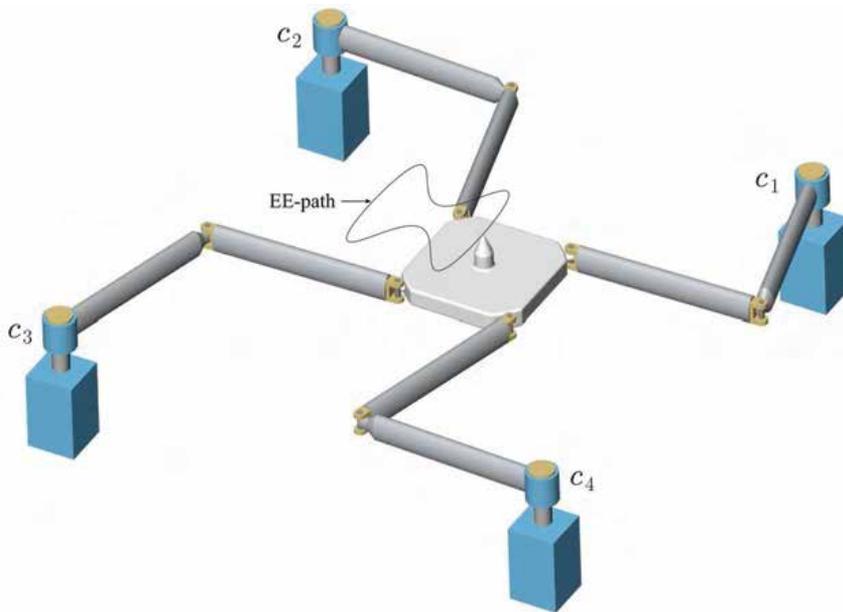


Figure 4. Planar 4RRR manipulator.

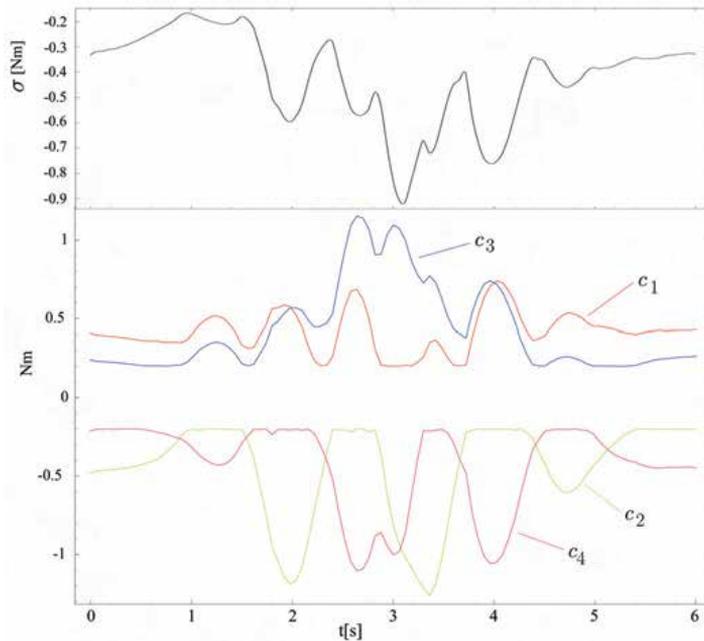


Figure 5. Prestress parameter and control torques for the EE motion in figure 4 with sign vector $s = (-1, 1, -1, 1)$.

For illustration purpose, we recall an example from (Müller, 2005), where the redundantly full-actuated planer PKM in figure 4 is navigated along the shown EE-path with fixed EE orientation. Figure 5 shows the drive torques, where a minimum drive torque of 0.2 Nm was required for prestress, with sign vector $(s_n) = (-1, 1, -1, 1)$.

7.4 Stiffness control

Stiffness or impedance control has long been proposed and developed for serial manipulators (Asada & Slotine, 1986). These concepts can straightforwardly be adopted for non-redundantly actuated PKM, thanks to the identical structure of the motion equations. Essentially, stiffness control (more precisely the control of the tangential stiffness since a PKM is a highly non-linear dynamical system) aims to mimic the force-deflection properties of an elastic medium, so that an applied EE-wrench causes an 'elastic' evasive deflection. This is achieved by generating control forces as reactions to joint motions caused by EE-motions. It is thus the result of a control cycle, which operates in discrete time steps. The actual behavior is therefore only 'elastic' for sufficiently slow effects, due to the latency in the force response to due to a perturbation.

Now, redundantly actuated PKM possesses the potential for another approach that does not suffer from the control latency. Redundant actuation allows for the generation of prestress, using controls in the null-space of \mathbf{A}^T . Since \mathbf{A}^T and thus $\mathbf{N}_{\mathbf{A}^T}$ are configuration dependent, part of the null-space component of a given control vector c becomes effective when the configuration is perturbed. Hence, there is an immediate! response to EE-deflection. In order to exploit this effect, the control forces in the null-space of \mathbf{A}^T must be such that the change of \mathbf{A}^T due to a EE-perturbation yields a desired EE-wrench. This was attempted in (Müller,

2006; Yi et al., 1989). It turned out that, for the considered PKM, a large number of redundant actuators is required for stable control of all stiffness components. This requirement is eased if only some stiffness components are to be controlled. Moreover, so far, stiffness control was not taken into account in the PKM design, and other novel PKM structures may need lower actuator redundancy to control EE stiffness via prestress.

8. PKM control

Upon the motion equations (7), established non-linear control methods can be applied to PKM, and shall exhibit the known stability properties. Model based motion control of redundantly actuated PKM was addressed in (Cheng et al, 2003). It was proposed to adopt the established computed torque and augmented PD control schemes, where perfect knowledge of the PKM model as well as perfect measurement of position, velocity and acceleration was presumed. The assumption of perfect measurements was abandoned in (Garrido, 2004), and a standard PD control in conjunction with a velocity estimator was proposed. In (Gourdeau et al., 1999) a computed torque control scheme without velocity measurement was proposed. Common to the control methods proposed so far, is the assumption of a perfect model. Robust control of redundantly actuated PKM has not yet been attempted.

In the following we briefly recall the standard model bases control schemes and their application to PKM control, and point out problems specific to redundantly actuated PKM that arise in the presence of model uncertainties. For notational simplicity the weighting matrix $\mathbf{M} = \mathbf{I}$ is assumed.

8.1 Model-based control schemes

Two model-based control schemes frequently used for the control of robotic manipulators: the augmented PD and the computed torque control (Asada & Slotine, 1986; Murray, et al., 1993). The unaltered augmented PD control attains the form

$$\mathbf{c} = (\mathbf{A}^T)^+ \left[\overline{\mathbf{G}}\ddot{\mathbf{q}}_2^d + \overline{\mathbf{C}}\dot{\mathbf{q}}_2^d + \overline{\mathbf{Q}} - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e} \right] + \mathbf{N}_{\mathbf{A}^T}\mathbf{c}^0, \quad (17)$$

with the desired nominal path $\mathbf{q}^d(t)$, and the tracking error $\mathbf{e}(t) := \mathbf{q}(t) - \mathbf{q}^d(t)$ (Cheng et al., 2003). The computed torque control law adopted for PKM is

$$\mathbf{c} = (\mathbf{A}^T)^+ \left[\overline{\mathbf{G}}\mathbf{v} + \overline{\mathbf{C}}\dot{\mathbf{q}}_2 + \overline{\mathbf{Q}} \right] + \mathbf{N}_{\mathbf{A}^T}\mathbf{c}^0, \quad (18)$$

setting $\mathbf{v} := \ddot{\mathbf{q}}_2^d - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e}$. Perfect matching of model and plant presumed, both control laws applied to (7) result in exponential trajectory tracking for sufficiently large gains \mathbf{K}_D and \mathbf{K}_P , provided $\overline{\mathbf{G}}$ is regular. The latter assumption fails in configuration space singularities.

8.2 Model uncertainties

The aforementioned control laws yields exponential stability for the nominal system only. Any real-life manipulator will differ from the nominal model used in the control scheme due to inevitable model uncertainties. There is a plethora of methods for the estimation of kinematic and other model parameters of serial manipulators, such as inertia, stiffness, and friction. Adaptations of these algorithms to PKM were proposed in (Valasek, 2002). Friction identification in particular was attempted in (Abdellatif et al., 2007).

A parameter estimation, whatsoever, will not achieve perfect matching of model and plant. To tackle this uncertainties, a number of robust control schemes have been proposed for serial manipulators. Robust control is still a field of active research, and we will not attempt to develop a such for redundant PKM here. The interested reader is referred to (Asada & Slotine, 1986) for the fundamentals. It is nevertheless instructive to investigate the effect of model uncertainties. In contrast to non-redundant manipulators, where model uncertainties cause incorrect positioning, geometric uncertainties of redundantly actuated PKM may cause (possibly high) actuator loads that have no effect on the motion.

Deviations from the nominal geometry alter the geometric constraints and thus the configuration space V . That is, a configuration $\mathbf{q} \in \mathbb{V}^n$ that complies with the nominal constraints will not do so for the uncertain system. If the variations are small, the PKM configuration will still be expressible in terms of the independent coordinates \mathbf{q}_2 . The constraint Jacobian in (1) changes according to

$$\underline{\mathbf{J}} \equiv (\underline{\mathbf{J}}_1, \underline{\mathbf{J}}_2) := \mathbf{J} + \Delta\mathbf{J}, \quad \Delta\mathbf{J} := (\Delta\mathbf{J}_1, \Delta\mathbf{J}_2). \quad (19)$$

Underlines indicate perturbed objects. For $\Delta\mathbf{f}$ small compared to \mathbf{f} , and neglecting second order terms of $\Delta\mathbf{J}$, yields

$$\underline{\mathbf{J}}_1^{-1} \underline{\mathbf{J}}_2 = (\mathbf{I} - \mathbf{J}_1^{-1} \Delta\mathbf{J}_1) \mathbf{J}_1^{-1} \mathbf{J}_2 + \mathbf{J}_1^{-1} \Delta\mathbf{J}_2, \quad (20)$$

The perturbed orthogonal complement is then

$$\underline{\mathbf{F}} := \mathbf{F} + \Delta\mathbf{F}, \quad \Delta\mathbf{F} = \begin{pmatrix} \mathbf{J}_1^{-1} \Delta\mathbf{J}_1 \mathbf{J}_1^{-1} \mathbf{J}_2 - \mathbf{J}_1^{-1} \Delta\mathbf{J}_2 \\ \mathbf{0} \end{pmatrix}. \quad (21)$$

The splitting (11) according to active and passive joints yields

$$\underline{\mathbf{F}} \equiv \begin{pmatrix} \underline{\mathbf{P}} \\ \underline{\mathbf{A}} \end{pmatrix} := \begin{pmatrix} \mathbf{P} + \Delta\mathbf{P} \\ \mathbf{A} + \Delta\mathbf{A} \end{pmatrix}, \quad \Delta\mathbf{A} := \begin{pmatrix} \Delta\mathbf{A}_1 \\ \mathbf{0} \end{pmatrix} \quad (22)$$

where $\Delta\mathbf{A}$ comprises the last $\rho = m - \delta$ rows of $\mathbf{J}_1^{-1} \Delta\mathbf{J}_1 \mathbf{J}_1^{-1} \mathbf{J}_2 - \mathbf{J}_1^{-1} \Delta\mathbf{J}_2$. Thus, the pseudoinverse of $\underline{\mathbf{A}}^T$ is

$$\begin{aligned} (\underline{\mathbf{A}}^T)^+ &= (\mathbf{A}^T)^+ + \mathbf{B}, \\ \mathbf{B} &:= \mathbf{N}_{\mathbf{A}^T} \Delta\mathbf{A}^T (\mathbf{A}^T \mathbf{A})^{-1} - (\mathbf{A}^T)^+ \Delta\mathbf{A}^T (\mathbf{A}^T)^+. \end{aligned} \quad (23)$$

The projector to the null-space of $\underline{\mathbf{A}}^T$ is

$$\begin{aligned}\mathbf{N}_{\underline{\mathbf{A}}^T} &= \mathbf{N}_{\mathbf{A}^T} + \Delta\mathbf{N}_{\mathbf{A}^T} \\ \Delta\mathbf{N}_{\mathbf{A}^T} &:= -(\mathbf{A}^T)^+ \Delta\mathbf{A}^T \mathbf{N}_{\mathbf{A}^T} - \mathbf{N}_{\mathbf{A}^T} \Delta\mathbf{A} \mathbf{A}^+.\end{aligned}\quad (24)$$

The null-space difference makes part of the control forces ineffective causing unintentional prestress, and part of the control forces applied to the uncertain system are annihilated by $\Delta\mathbf{N}_{\mathbf{A}^T}$.

The objects in the motion equations change accordingly,

$$\underline{\mathbf{G}} := \overline{\mathbf{G}} + \Delta\overline{\mathbf{G}}, \quad \underline{\mathbf{C}} := \overline{\mathbf{C}} + \Delta\overline{\mathbf{C}}, \quad \underline{\mathbf{Q}} := \overline{\mathbf{Q}} + \Delta\overline{\mathbf{Q}}, \quad (25)$$

that give rise to the motion equations of the uncertain PKM

$$\underline{\mathbf{G}}\ddot{\mathbf{q}}_2 + \underline{\mathbf{C}}\dot{\mathbf{q}}_2 + \underline{\mathbf{Q}} = \underline{\mathbf{A}}^T \mathbf{c}. \quad (26)$$

Application of the augmented PD controller (17) to (26), results in the error dynamics governed by

$$\begin{aligned}\overline{\mathbf{G}}\ddot{\mathbf{e}} + \overline{\mathbf{C}}\dot{\mathbf{e}} + (\mathbf{I} + \mathbf{S})\mathbf{K}_D\dot{\mathbf{e}} + (\mathbf{I} + \mathbf{S})\mathbf{K}_P\mathbf{e} \\ + \Delta\overline{\mathbf{G}}\ddot{\mathbf{q}}_2 + \Delta\overline{\mathbf{C}}\dot{\mathbf{q}}_2 + \Delta\overline{\mathbf{Q}} \\ - \mathbf{S}(\overline{\mathbf{G}}\dot{\mathbf{q}}_2^d + \overline{\mathbf{C}}\dot{\mathbf{q}}_2 + \overline{\mathbf{Q}}) - \Delta\mathbf{A}^T \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 = \mathbf{0},\end{aligned}\quad (27)$$

with

$$\mathbf{S} := \Delta\mathbf{A}^T (\mathbf{A}^T)^+. \quad (28)$$

It is obvious that, with the perturbation \mathbf{S} of the gain matrices, model uncertainties do not only affect the dynamics of the controlled system but also interfere with the PD feedback. This is a peculiarity of the redundant actuation. The extent of the effect depends on the degree of non-linearity of the geometric constraints. With, usually large gains, the parasitic control forces due to $\mathbf{S}\mathbf{K}_D\dot{\mathbf{e}}$ and $\mathbf{S}\mathbf{K}_P\mathbf{e}$ may be large too. Moreover, the critical point to observe here is that these parasitic forces can never be equilibrated by adjusting the gains. Also observe that in view of $\Delta\mathbf{A}^T \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0$, the control forces, deduced from the nominal model, are partially annihilated, whereas some null-space components (according to a secondary task, e.g. prestress generation) become effective and interfere with the motion control. The latter is due to the mismatch of the null-space of \mathbf{A}^T and $\underline{\mathbf{A}}^T$ that, with (24), can be inferred from $\underline{\mathbf{A}}^T \mathbf{N}_{\mathbf{A}^T} \neq \mathbf{0}$.

8.3 Amended control schemes

Parasitic control forces can be avoided by restricting the linear feedback to the subspace of independent coordinates \mathbf{q}_2 , which are a subset of \mathbf{q}_a , since $\mathbf{q}_a \equiv (\dots, \mathbf{q}_2)$. This gives rise to the following adapted augmented PD control law for redundantly actuated PKM

$$\begin{aligned} \mathbf{c} = & (\mathbf{A}^T)^+ (\overline{\mathbf{G}}(\mathbf{q}) \ddot{\mathbf{q}}_2^d + \overline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_2^d + \overline{\mathbf{Q}}) \\ & + \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 - \begin{pmatrix} \mathbf{0} \\ \mathbf{I}_\delta \end{pmatrix} (\mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}). \end{aligned} \quad (29)$$

The adapted computed torque control law is

$$\begin{aligned} \mathbf{c} = & (\mathbf{A}^T)^+ (\overline{\mathbf{G}}(\mathbf{q}) \ddot{\mathbf{q}}_2^d + \overline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_2^d + \overline{\mathbf{Q}}) \\ & + \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 - \begin{pmatrix} \mathbf{0} \\ \overline{\mathbf{G}} \end{pmatrix} (\mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}). \end{aligned} \quad (30)$$

It is vital that both control schemes work stable for the nominal system. To see this, consider the error dynamics of the closed loop control law (29) that is governed by $\overline{\mathbf{G}}(\mathbf{q}) (\ddot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}) = 0$, and the error dynamics for (30) governed by $\overline{\mathbf{G}}(\mathbf{q}) \ddot{\mathbf{e}} + \overline{\mathbf{G}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} = 0$. Thereupon, with the classical stability results (Murray, et al., 1993) it can be shown that the control laws (29) and (30) applied to the nominal system (7) are exponentially stable.

Having concluded stability for the nominal system, it remains to show the claimed elimination of parasitic control forces. This is obvious from the closed loop dynamics

$$\begin{aligned} & \overline{\mathbf{G}} \ddot{\mathbf{e}} + \overline{\mathbf{C}} \dot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} \\ & + \Delta \overline{\mathbf{G}} \ddot{\mathbf{q}}_2 + \Delta \overline{\mathbf{C}} \dot{\mathbf{q}}_2 + \Delta \overline{\mathbf{Q}} \\ & - \mathbf{S} (\overline{\mathbf{G}} \ddot{\mathbf{q}}_2^d + \overline{\mathbf{C}} \dot{\mathbf{q}}_2^d + \overline{\mathbf{Q}}) - \Delta \mathbf{A}^T \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 = \mathbf{0}, \end{aligned} \quad (31)$$

when (29) is applied to the uncertain system (26), and from

$$\begin{aligned} & \overline{\mathbf{G}} (\ddot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e}) \\ & + \Delta \overline{\mathbf{G}} \ddot{\mathbf{q}}_2 + \Delta \overline{\mathbf{C}} \dot{\mathbf{q}}_2 + \Delta \overline{\mathbf{Q}} \\ & - \mathbf{S} (\overline{\mathbf{G}} \ddot{\mathbf{q}}_2^d + \overline{\mathbf{C}} \dot{\mathbf{q}}_2^d + \overline{\mathbf{Q}}) - \Delta \mathbf{A}^T \mathbf{N}_{\mathbf{A}^T} \mathbf{c}^0 = \mathbf{0}, \end{aligned} \quad (32)$$

when the computed torque controller (30) is applied.

Now the control forces act freely upon the uncertain system, in contrast to (29) and (30). Therewith the uncertainties affect the dynamics of the controlled PKM, but not the way the controls act upon the system. The second and third lines in (31) and (32) embody the uncertain dynamics that is not balanced by the controller.

The proposed adapted control schemes shall motivate the development of tailored model-based robust control concepts for redundantly actuated PKM.

8.4 Example

For illustration purpose the effect of geometric uncertainties of the planar RP/2RPR PKM in figure 6 is analyzed. This is a fully-parallel but not symmetric PKM. There is no moving platform, and the EE is mounted on one of the limbs. The EE is connected to the base by one RP and two RPR chains. The PKM is obtained from a non-redundant RP/RPR by adding one RPR chain.

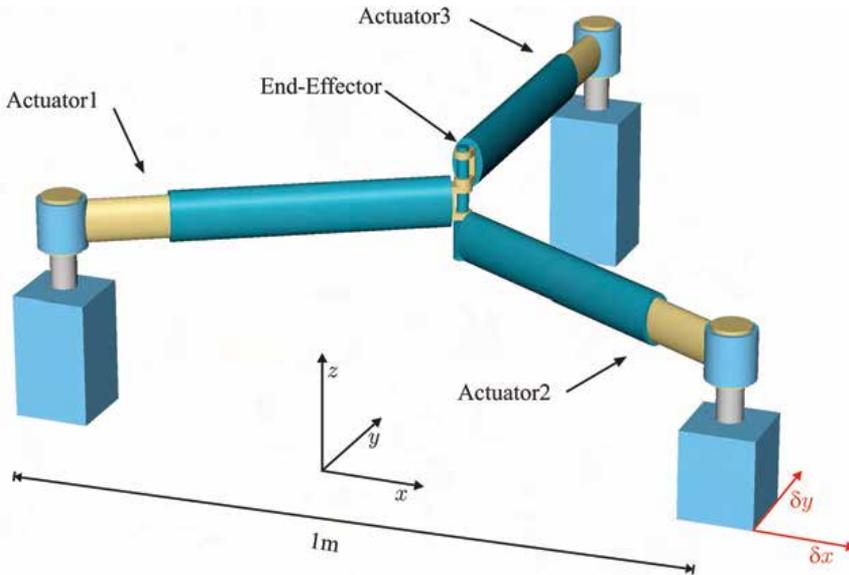


Figure 6. Planar 2RPR/RP PKM with DOF 2.

The drive units are mounted on the base at the corners of an equilateral triangle. A disturbance frequently encountered in setting up a PKM is the misplacement of joints. Now assume that one of the drive units is displaced on the ground plane as indicated in figure 6. This leads to a perturbed plant with input matrix $\underline{\mathbf{A}}^T$. The control forces are deduced from the nominal model with \mathbf{A}^T . Consequently, the inverse dynamics solution (12) applied to the perturbed system (26) can not perfectly reproduce the desired control forces, due to $\underline{\mathbf{A}}^T (\mathbf{A}^T)^+ \neq \mathbf{I}$. This leads to desired forces in the null-space of \mathbf{A}^T becoming effective, due to $\underline{\mathbf{A}}^T \mathbf{N}_{\mathbf{A}^T} \neq 0$. For a quantitative analysis the drive unite has been displaced by 5% of the triangle side length, as shown in figure 7. The perfect model and the perturbed plant are evaluated along the indicated EE path. For this PKM the null-space projector and thus $\underline{\mathbf{A}}^T \mathbf{N}_{\mathbf{A}^T}$ is two dimensional vector (being zero for perfect matching). Figure 7 shows the two components evaluated for the EE positions on the indicated path. It turns out that the matrix $\mathbf{S} = \Delta \mathbf{A}^T (\mathbf{A}^T)^+$ leads to uncontrollable counter action of the drives. For the RP/2RPR PKM this is a 2×2 matrix, which is identically zero for a perfect match of plant and model. The norm of \mathbf{S} is shown in figure 7. It is clear that even for this simple PKM the effect of geometric uncertainties can not be neglected.

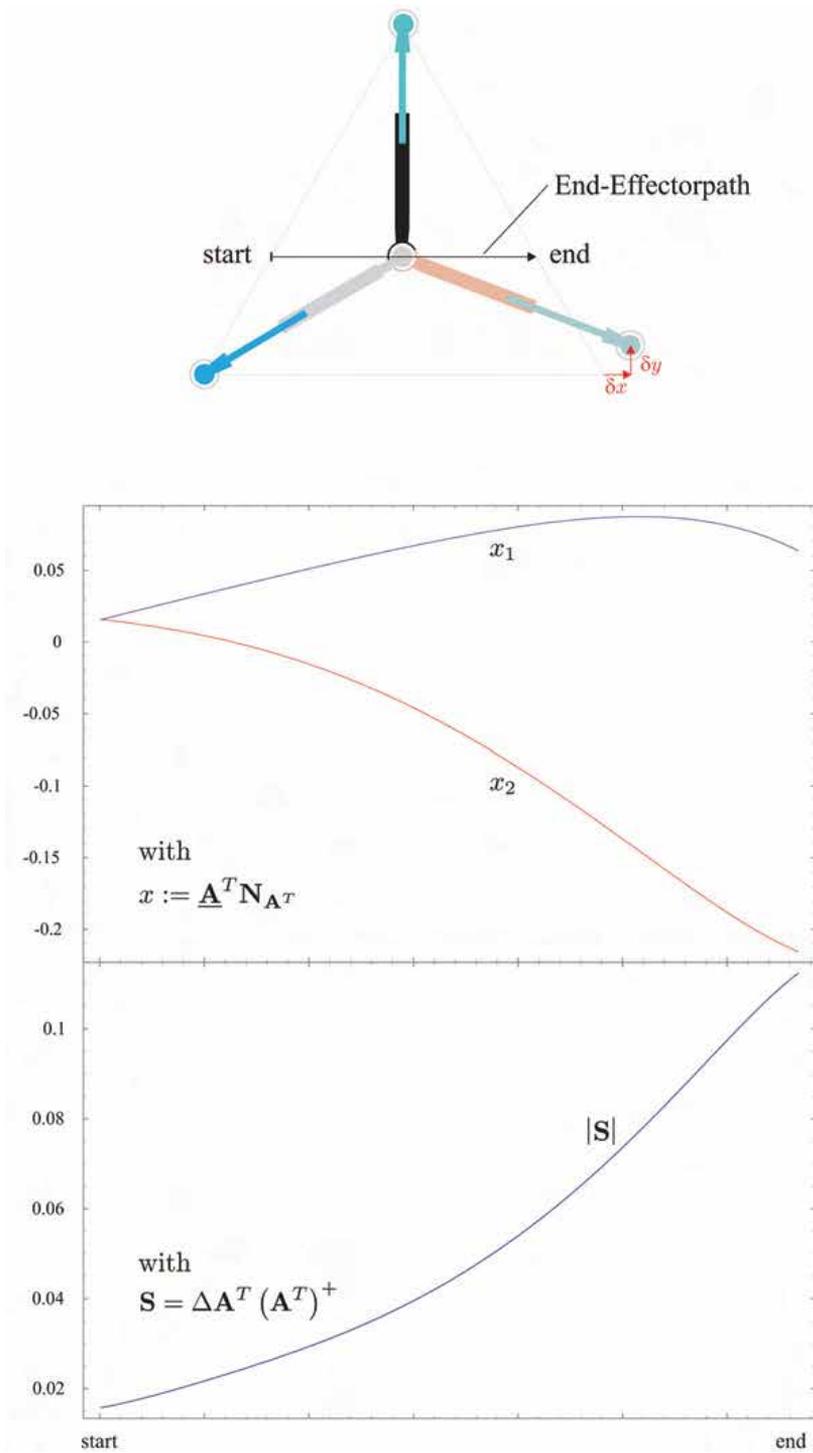


Figure 7. Effect of displacement of a drive unit of the PKM in figure 6.

9. Conclusions and open problems

In this chapter the dynamics modeling of redundantly actuated PKM is reviewed, and the redundancy resolution is addressed. The resolution takes into account different secondary tasks, such as backlash avoidance and stiffness control.

It was aimed to point out the potential of redundant actuation, but also the challenges that need to be addressed. In this contribution the effect of kinematic parameter uncertainties on the control of redundantly actuated PKM is analyzed. It is shown how geometric uncertainties effect the control system. The application of standard model-based control schemes to redundantly actuated PKM is shown not only to change the control system, but also to change the way in which control forces act upon the system. A consequence thereof is that the perturbation forces, due geometric uncertainties, can not be compensated by the actuation. To overcome these effects, an amended augmented PD and computed torque control scheme for redundantly actuated PKM is introduced. This is a first step that at least ensures the applicability of the control schemes. It shall be clear that robust control of uncertain redundantly actuated PKM is a critical issue for redundant PKM.

10. References

- Abdellatif, H. et al. (2007). Independent Identification of Friction Characteristics for Parallel Manipulators, *ASME J. Dyn. Sys., Meas., Cont.*, Vol. 129, 294-302
- Asada, H.; Slotine, J.J.E. (1986). *Robot Analysis and Control*, John Wiley and Sons
- Alba, O.; Pamanes, G.J.A.; Wenger, P. (2007). Trajectory planning of a redundant parallel manipulator changing of working mode, *12th IFToMM World Congress*, Besancon, France
- Borrel, P.; Liegeois, A. (1986). A study of multiple manipulator inverse kinematic solutions with applications to trajectory planning and workspace determination, *Proc. IEEE Int. Conf. Rob. Aut.*, 1180-1185
- Buttolo, P.; Hannaford, B. (2005). Advantages of Actuation Redundancy for the Design of Haptic Displays, *Proc. ASME, Fourth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, San Francisco, 2005, SDCVol. 57-2, 623-630
- Chablat, D.; Wenger, Ph. (1998). Working Modes and Aspects in Fully-Parallel Manipulator, *IEEE Int. Conf. Rob. Aut.*, 1998, 1970-1976
- Chakarov, D. (2004). Study of the antagonistic stiffness of parallel manipulators with actuation redundancy, *Mech. Mach. Theory*, Vol. 39, 2004, 583-601
- Cheng, H.; Yiu, Y.-K.; Li, Z. (2003). Dynamics and Control of Redundantly Actuated Parallel Manipulators, *IEEE/ASME Trans. on Mechatronics*, Vol. 8, No. 4, 2003, 483-491
- Chiu, Y.-J.; Perng, M.-H. (2001). Forward Kinematics of a General Fully Parallel Manipulator with Auxiliary Sensors, *The International Journal of Robotics Research*, Vol. 20, No. 5, 2001, 401-414
- Conkur, E.S.; Buckingham, R. (1997). Clarifying the definition of redundancy as used in robotics, *Robotica*, Vol. 15, 1997, 583-586

- Dasgupta, B.; Mruthyunjaya, T.S. (1998). Force redundancy in parallel maipulators: theoretical and practical issues, *Mech. Mach. Theory*, Vol. 33, No. 6, 1998, 727-742
- Firmani, F.; Podhorodeski, R.P. (2004). Force-unconstrained poses for a redundantly actuated planar parallel manipulator, *Mech. Mach. Theory*, Vol. 39, 2004, 459-476
- Ganovski, L.; Fiset, P.; Samin, J.C. (2004). Piecewise Overactuation of Parallel Mechanisms Following Singular Trajectories: Modelling, Simulation and Control, *Multibody System Dynamics*, Vol. 12, 2004, 317-343
- Gardner, J.F.; Kumar, V.; Ho, J.H. (1989). Kinematics and Control of Redundantly Actuated Closed Chains, *IEEE Conf. on Rob. and Aut.*, 1989, 418-424
- Garrido, R.; Torres-Cruz, D. (2004). On PD control of parallel robots with redundant actuation, *IEEE Int. Conf. on Electrical and Electronics Engineering (ICEEE)*, Acapulco, Mexico, 8 - 10 Sep. 2004, pp. 356-360
- Gosselin, C. M.; Angeles, J. (1990). Singular analysis of closed-loop kinematic chains, *IEEE Trans. Rob. and Aut.*, Vol. 6, No. 3, 1990, 281-290
- Gourdeau, R.; Blouin, S.; Hurteau, R. (1999). Computed Torque Control of Robots Without Joint Velocity Measurements, *IEEE Canadian Conf. on Electrical and Computer Eng.*, Edmonton, Canada, 1999, pp. 1413-1418
- Jeong, J.; Kang, D.; Cho, Y.M.; Kim, J. (2004). Kinematic Calibration for Redundantly Actuated Parallel Mechanisms, *ASME Journal of Mechanical Design*, Vol. 126, No. 2, 2004, 307-318
- Kim J. et al. (2001). Design and Alalysis of a Redundantly Actuated Parallel Mechanism for Rapid Machining, *IEEE Trans. Rob. and Aut.*, Vol. 17, No. 4, 2001, 423-434
- Kock, S.; Schumacher, W. (1998). A parallel x-y manipulator with actuation redundancy for high-speed and active-stiffness applications, *Proc. IEEE Int. Conf. Robotics Automation*, Leuven, 1998, 2295-2300.
- Kock, S.; Schumacher, W. (2000). A mixed elastic and rigid-body dynamic model of an actuation redundant parallel robot with high-reduction gears, *IEEE Int. Conf. Robotics Automation*, San Francisco, 2000
- Kurtz, R.; Hayward, V. (1992). Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy, *IEEE Trans. on Rob. and Aut.*, Vol. 8. No. 5, 1992, 644-651
- Lee, J.H.; Li, B.J.; Suh, H. (1998). Optimal design of a five-bar finger with redundant actuation, *Proc. IEEE Int. Conf. Rob. Aut.*, Leuven, 1998, 2068-2074
- Liao, H.; Li, T.; Tang, Y. (2004). Singularity analysis of redundant parallel manipulators, *IEEE Int. Conf. Syst., Man and Cybernetics*, 2004, 4214-4220
- Liu, G.; Lou, Y.; Li, Z. (2003). Singularities of Parallel Manipulators: A geometric Treatment, *IEEE Trans. Rob. Vol. 19, No. 4, 2003*, pp 579-594
- Maisser, P. (1997). Differential geometric methods in multibody dynamics, *Proc. 2nd World Congress of Nonlinear Analysts*, 1997, 5127-5133
- Merlet, J.P. (1996). Redundant Parallel Manipulators, *J. of Laboratory Rob. and Aut.*, Vol. 8, 1996, 17-24

- Mohamed, M.G.; Gosselin, C.M. (1999). Design and Analysis of Kinematically Redundant Parallel Manipulators With Configurable Platforms, *IEEE Trans. Robotics*, Vol. 21, No. 3, 1999, 277-287
- Müller, A. (2005). Internal Prestress Control of redundantly actuated Parallel Manipulators Its Application to Backlash avoiding Control, *IEEE Trans. on Rob.*, Vol. 21, No. 4, 2005, 668-677
- Müller, A. (2006). A Conservative Elimination Procedure for Redundant Closure Constraints in MBS Models, *Multibody Sys. Dyn.*, Vol. 16, No. 4, 2006, 309-330
- Müller, A. (2006). Stiffness Control of redundantly actuated Parallel Manipulators, *Proc. IEEE Int. Conf. Rob. Aut.*, Orlando, 15.-19. May, 2006, 1153-1158
- Murray, R.M.; Li, Z.; Sastry, S.S. (1993). *A mathematical Introduction to robotic Manipulation*, CRC Press, 1993
- Nahon, M.A.; Angeles, J. (1989). Force optimization in redundantly-actuated closed kinematic chains, *Proc. IEEE Int. Conf. Rob. Aut.*, Scottsdale, 1989, 951-956
- Nakamura, Y.; Ghodoussi, M. (1989). Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators, *IEEE Tran. Rob. and Aut.*, Vol. 5, No. 3, 1989, 294-302
- Nijmeijer, H.; van der Schaft, A.J. (1990). *Nonlinear dynamical control systems*, Springer, 1990
- Nokleby, S.B. et al. (2005). Force capabilities of redundantly-actuated parallel mechanisms, *Mech. Mach. Theory*, Vol. 40, 2005, 578-599
- O'Brien, J.F.; Wen, J.T. (1999). Redundant actuation for improving kinematic manipulability, *Proc. IEEE Int. Conf. Rob. Aut.*, 1999, 1520-1525
- J.G. Papastavridis (2002). *Analytical Mechanics*, Oxford University Press, 2002
- Park, F.C.; Kim, J.W. (1998). Manipulability of closed kinematic chains, *Trans. ASME J. Mech. Des.*, Vol. 120, No. 4, 1998, 542-548
- Valasek, M.; Bauma, V.; Sika, Z.; Vampola, T. (2002). Redundantly actuated parallel structures - principle, examples, advantages, *3rd Parallel Kinematics Seminar Chemnitz*, 2002, 993-1009
- Valasek, M.; Belda, K.; Florian, M. (2002). Control and calibration of redundantly actuated parallel robots, *3rd Parallel Kinematics Seminar Chemnitz, Chemnitz*, 2002, pp. 411-427
- Yi, B.Y.; Freeman, R.A.; Tesar, D. (1989). Open-loop stiffness control of overconstrained mechanisms/robot linkage systems, *Proc. IEEE Int. Conf. Robotics Automation*, Scottsdale, 1989, pp. 91340-1345
- Yi, B.Y.; Freeman, R.A.; Tesar, D. (1994). Force and stiffness transmission in redundantly actuated mechanisms: the case for a spherical shoulder mechanism, *Robotics, Spatial Mechan. and Mech. Sys.*, Vol. 45, 1994, 163-172
- Yiu, Y.-K.; Li, Z.X. (2003). Optimal Forward Kinematics Map for Parallel Manipulator with Sensor Redundancy, *IEEE Int. Symp. on Comp. Intelligence in Rob. and Aut.*, Kobe, Korea, 2003, 354-359
- Yoshikawa, T. (1985). Manipulability of Robotic Mechanisms, *Int. J. of Robotic Research*, Vol. 4, No. 2, 1985, 3-9

- Zhang, Y.; Gong, J.; Gao, F. (2007). Singularity elimination of parallel mechanisms by means of redundant actuation, *12th IFToMM World Congress*, Besancon, France, 2007
- Zlatanov, D.; Fenton, R.G.; Benhabib, B. (1998). Identification and classification of the singular configurations of mechanisms, *Mech. Mach. Theory*, 1998, 743-760

Wrench Capabilities of Planar Parallel Manipulators and their Effects Under Redundancy

Flavio Firmani¹, Scott B. Nokleby², Ronald P. Podhorodeski¹ and Alp Zibil¹

¹University of Victoria, ²University of Ontario Institute of Technology
Canada

1. Introduction

1.1 Instantaneous twist and wrench capabilities

The instantaneous twist and wrench capability analyses are essential for the design and performance evaluation of serial and parallel manipulators. An instantaneous twist is a screw quantity that contains both angular and translational velocities of the end-effector, i.e., $\mathbf{V} = \{\boldsymbol{\omega}^T; \mathbf{v}^T\}^T$. Whereas, a wrench is a screw quantity that contains the forces and moments acting on the end-effector, i.e., $\mathbf{F} = \{\mathbf{f}^T; \mathbf{m}^T\}^T$. For a given pose, the required task of the end-effector is to move with a desired twist and to sustain (or apply) a specific wrench. These kinematic conditions are achieved with corresponding joint velocities ($\dot{\mathbf{q}}$) and joint torques ($\boldsymbol{\tau}$), respectively. The relationship between the task and joint spaces is defined by the well known linear transformations:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \quad (1)$$

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} \quad (2)$$

where \mathbf{J} is referred to as the Jacobian matrix.

In addition, an extended problem can be formulated as the analysis of the maximum twist or wrench that the end-effector can perform in the twist or wrench spaces, respectively. The knowledge of maximum twist and wrench capabilities is an important tool for achieving the optimum design of manipulators. For instance, by being able to graphically visualize the twist and wrench capabilities, comparisons between different design parameters, such as the actuator torque capabilities and the dimensions of the links, can be explored. Also, the performance of an existing manipulator can be improved by identifying the optimal capabilities based on the configuration of the branches and the pose of the end-effector.

This work focuses on the wrench capabilities of planar parallel manipulators (PPMs), the geometric interpretation of their wrench polytopes, the derivation of wrench performance indices, and how the inclusion of redundancy affects the performance of parallel manipulators (PMs). The wrench capability analysis of a manipulator depends on its design, posture, and actuator torque capabilities.

For a 3-degree-of-freedom (DOF) planar manipulator, the wrench \mathbf{F} is a screw quantity that contains the two components of the force on the plane (f_x and f_y) and the moment normal to the plane (m_z). The problem of finding the wrench capabilities of a manipulator involves the forward static force equation ($\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}$) and the actuator output capabilities (τ_i). To date, three different approaches for determining wrench capabilities have been proposed in the literature: constrained optimization, wrench ellipsoid, and wrench polytope.

1.2 Constrained optimization

In general, the constrained optimization approach involves: an objective function that maximizes either the magnitude of the force f or the moment m_z ; one equality constraint ($\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}$); and a set of inequality constraints ($\tau_{i_{\min}} \leq \tau_i \leq \tau_{i_{\max}}$), indicating the actuator output capabilities. Kumar and Waldron (1988) investigated force distribution in redundantly-actuated closed-loop kinematic chains and concluded that there would be zero internal forces using the Moore-Penrose pseudo-inverse solution. Tao and Luh (1989) developed an algorithm that determines the minimum torque required to sustain a common load between two joint redundant cooperating manipulators. Nahon and Angeles (1992) described the problem of a hand grasping an object as a redundantly-actuated kinematic chain. The problem was formulated with both equality and inequality constraints and the torques were found by minimizing the internal forces in the system using Quadratic Programming. Buttolo and Hannaford (1995) analyzed the force capabilities of a redundant planar parallel manipulator. Torques were optimized using the ∞ -norm resulting in higher force capabilities when compared to the pseudo-inverse solution. Nokleby et al. (2005) developed a methodology to optimize the force capabilities of redundantly-actuated planar parallel manipulators using an n -norm, for large values of n , and a scaling factor. Garg et al. (2007) implemented this approach to spatial parallel manipulators.

This approach is usually slow due to the numerical nature of the algorithm and inaccuracies due to the existence of local minima.

1.3 Wrench ellipsoid

The wrench ellipsoid approach is based on bounding the actuator torque vector by a unit sphere $\boldsymbol{\tau}^T \boldsymbol{\tau} \leq 1$. The torques are mapped into the wrench space with Eq.(2), yielding a force ellipsoid $\mathbf{F}^T \mathbf{J} \mathbf{J}^T \mathbf{F} \leq 1$. If Singular Value Decomposition (SVD) is applied to \mathbf{J} i.e., $\mathbf{J} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$, the principal axes of the ellipsoid can be determined as \mathbf{u}_k / σ_k where σ_k is the k^{th} singular value and \mathbf{u}_k is the k^{th} column of matrix \mathbf{U} . These axes can be employed as wrench performance indices of the manipulator. This approach was introduced by Yoshikawa (1985) with the manipulability (twist) ellipsoid and proposed manipulability measurements. Also, Yoshikawa (1990) presented the duality between the twist and wrench ellipsoids concluding that axes of the ellipsoids coincide but their lengths are inversely proportional.

For cooperating manipulators, Chiacchio et al. (1996) presented a complete analysis of wrench ellipsoids for multiple-arm systems, which involves external and internal forces. Lee and Kim (1991) (velocity problem) and Chiacchio et al. (1997) (static force problem) proposed to normalize the joint space variables (joint velocities and joint torques, respectively) when the actuators do not produce the same output. As a result, the resulting ellipsoid is defined as the pre-image of the unit sphere in the scaled joint variable space.

The wrench ellipsoid approach can be implemented easily and the required computation is immediate. However, this approach is an approximation because the joint torques are normalized ($\tau^T \tau \leq 1$) yielding a hypersphere in the torque space. The correct model of the joint torques must be an m -dimensional parallelepiped in the torque space due to the nature of the extreme torque capabilities of each actuator, i.e., $\tau_{i\min}$ or $\tau_{i\max}$.

1.4 Wrench polytope

The wrench polytope approach considers the complete region in which the actuator can operate. A comparison between the ellipsoid approach and the polytope approach is shown in Fig. 1. Assume a manipulator with two actuated revolute joints whose extreme capabilities are $\tau_{i\text{ext}} = \pm 1$ Nm, for $i = 1, 2$. Fig. 1a shows the generation of an ellipse (in general, an ellipsoid) as a result of mapping a circle (in general, a hypersphere). Fig. 1b shows the generation of a polygon (in general, a polytope) as a result of mapping a square (in general, a hypercube). Each plot contains two coordinate systems. The inner circle of Fig. 1a and the inner square of Fig. 1b describe the torque limits in the torque space (bottom and left axes); whereas, the outer ellipse and polygon describe the wrench capabilities in the wrench space (top and right axes). The lines that connect the inner to the outer shapes illustrate the linear transformation. Note how the edges and vertices of the square and polygon correspond in both spaces. The areas comprised by these geometrical shapes represent the feasible capabilities in their corresponding spaces. The square is an exact representation of the torque capabilities; while, the circle is an approximation. For example, the upper-right vertex of the square is $\tau_1 = \tau_2 = 1$ Nm; although this torque combination is feasible, the circle model does not include it. Thus, modeling the torque capabilities as a square is better than as a circle. Fig. 1c shows how the circle and ellipse are inscribed within the square and polygon, respectively. It is important to mention that the principal axes of the ellipse are directed towards the vertices of the polygon.

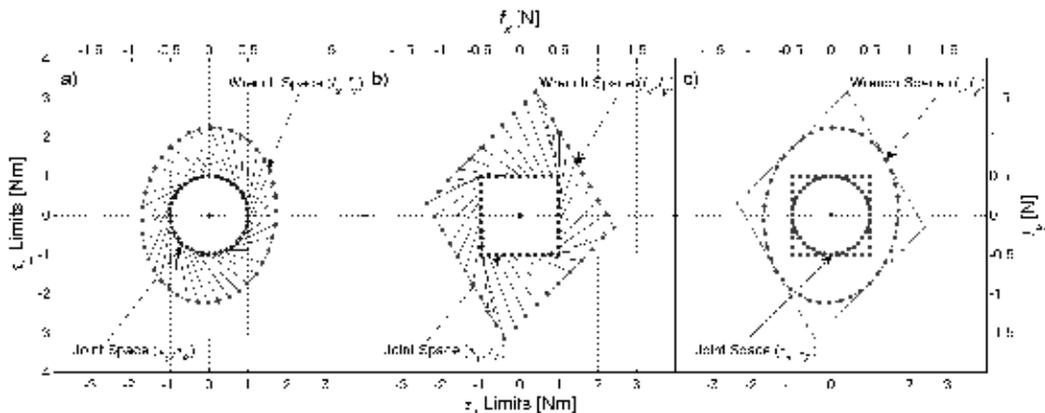


Fig. 1. Mapping of ellipsoids and polytopes from the joint space to the task space.

In general, each actuator torque defines an orthonormal axis in \mathbb{R}^m . The extremes of each torque constrain the torque space with a pair of parallel planes along each axis. The feasible region in which the manipulator can operate is bounded by these pairs of parallel planes yielding an m -dimensional parallelepiped.

A linear transformation, such as the equation of the forward static force, Eq. (2), maps vector $\boldsymbol{\tau}$ from \mathbb{R}^m (joint torque space) to \mathbb{R}^n (wrench space).

Rockafellar (1997) studied the properties of convex polyhedral sets. From his analysis, the following relationship is held through a linear transformation: Let $\boldsymbol{\tau}$ be the m -dimensional parallelepiped (a convex set) and \mathbf{J}^T be the linear transformation from \mathbb{R}^m to \mathbb{R}^n . Then the resulting transformation $\mathbf{J}^T\boldsymbol{\tau}$ leads to another convex polyhedral set (\mathbf{F}) in \mathbb{R}^n and it contains a finite number of facets.

Kokkinis and Paden (1989) introduced the concept of twist and wrench convex polytopes. The analysis was applied to a single serial manipulator and to two cooperating manipulators. Chiacchio et al. (1997) analyzed the wrench polytopes of redundant serial manipulators. Finotello et al. (1998) introduced two sets of indices that can be implemented to twist and wrench polytopes: the maximum isotropic value (MIV) and the maximum available value (MAV). These indices will be discussed in detail in Section 4. For 6-DOF manipulators, Finotello et al. (1998) proposed to analyze these indices with force and moment as separate entities. Gallina et al. (2001) analyzed the manipulability of a 3-DOF wire driven planar haptic device using polytopes. Lee and Shim (2004) expanded the concept to dynamic manipulability of multiple cooperating manipulators resulting in acceleration polytopes. Krut et al. (2004a) analyzed twist ellipsoids and polytopes in redundant parallel manipulators and established performance indices. They showed that there is another ellipsoid, besides the one derived with SVD, which is larger in volume and is fully inscribed within the polytope. Krut et al. (2004b) also studied force performance indices of redundant parallel manipulators and determined the isotropic wrench workspaces of planar wire-driven manipulators with multiple actuated limbs. Firmani et al. (2007a and 2007b) derived a set of wrench performance indices for PPMs.

2. Redundancy

2.1 Types of redundancy

Merlet (1996) described that the inclusion of redundancy may lead to improvements in various analyses such as forward kinematics, singular configurations, optimal force control, and calibration. Lee and Kim (1993) defined a redundant parallel manipulator as one that has an infinite number of choices for either generating motion or resisting external forces. Also, Lee and Kim (1993) presented an analysis of different types of redundancy. Ebrahimi et al. (2007) classified redundancy into two categories: kinematic and actuation redundancy.

2.2 Kinematic redundancy

A manipulator is termed kinematically redundant when at least one of the branches can have self-motion while keeping the mobile platform fixed. Thus, there is an infinite number of possible solutions to the inverse displacement problem. This is the typical case of redundant serial manipulators. For parallel manipulators, this redundancy happens when the number of joints of at least one branch is greater than the number of joints that are required to provide the desired mobility of the mobile platform. This type of redundancy allows self-motion of the redundantly-jointed branch(es) improving the dexterity and workspace of the manipulator. A draw back of this type of redundancy is the increase of mass and/or inertia due to the addition of actuators on the mobile links. Despite the redundancy, there is only one vector force per branch acting on the mobile platform. Thus,

the load capability cannot be optimized, but as an alternative, the direction of the branch forces can be optimized by changing the posture of the redundantly-jointed branch(es). With this type of redundancy, each actuator can be manipulated independently and there are no internal forces that could damage the device. Kinematic redundancy can be employed to reduce or even eliminate singular configurations. Wang and Gosselin (2004) added an extra revolute joint to one branch of the 3-RPR PPM yielding a RRPR-2RPR layout. The singularity conditions were identified and the singularity loci were reduced. Ebrahimi et al. (2007) proposed the 3-PRRR PPM, a layout that contains joint redundancy in every branch. This manipulator can provide singularity free paths and obstacle avoidance by properly manipulating the actuated joints.

2.3 Actuation redundancy

A parallel manipulator is termed redundantly actuated when an infinite number of resultant force combinations can span the system of external forces. Thus, there is an infinite number of solutions to the inverse static force problem. The implementation of this redundancy requires a reliable control system because a small variation in the displacement may cause severe damage to the manipulator. There are two types of actuation redundancy: in-branch redundancy and branch redundancy.

In-Branch Redundancy. Passive joints are replaced by active joints. For every redundant actuator added within branch(es), the number of the forces resisting an external load is augmented by one. This type of redundancy can be easily incorporated into an existing device. Nokleby et al. (2005) developed a methodology to optimize the force capabilities of the 3-RRR PPM using a high norm and a scaling factor. Zibil et al. (2007) determined the force capabilities of the 3-RRR PPM by using an analytical based method. Nokleby et al. (2007a) investigated the force-moment capabilities of different in-branch redundancy architectures. With in-branch redundancy, there is no change in the workspace of the manipulator. However, there is an increase of mass and/or inertia due to the addition of actuators. Firmani & Podhorodeski (2004) eliminated families of singular configurations by adding a redundant actuator to the 3-RRR PPM, yielding a RRR-2RRR layout.

Branch Redundancy. An additional actuated branch is added to the system. For every additional actuated branch incorporated into the system, the number of forces acting on the mobile platform is augmented by one. Buttolo and Hannaford (1995) designed and analyzed the force capabilities of a 2-DOF 3-RRR PPM haptic device, where all three branches are pinned together. Gallina et al. (2001) analyzed the maximum force and moment of a four-wire driven 3-DOF planar haptic device. Krut et al. (2004a) implemented performance indices, previously developed in Krut et al. (2004b) for velocity analysis, to 2-DOF parallel wire-driven manipulators. Different analyses of multi-actuated wires were considered. Nokleby et al. (2007b) investigated the force-moment capabilities of the 4-RRR, 4-RPR, and 4-PRR PPMs. Firmani & Podhorodeski (2005) presented a methodology to identify singular configurations of planar parallel manipulators with redundant branches. The main problem of manipulators with branch redundancy is the reduction of their dexterity and workspace.

3. Wrench polytope analysis

3.1 Joint space parallelepiped

Let n be the DOF of the task space coordinates and m be the number of actuated joints. The i^{th} joint torque variable, which is bounded by $\tau_{i_{\min}}$ and $\tau_{i_{\max}}$, can be represented in the joint

torque space as two parallel planes in \mathbb{R}^m . With m joints, there are $2m$ planes or m pairs of parallel planes. An m -dimensional parallelepiped is formed with the combination of all of these parallel planes yielding the region of joint torque capabilities. If all the torque capabilities were equal, the m -dimensional parallelepiped would result in a hypercube. Also, if the magnitude of the extreme torques were equal, i.e., $|\tau_{i_{\min}}| = |\tau_{i_{\max}}|$, the parallelepiped would be centro-symmetric; otherwise it would be skewed. A vertex of the m -dimensional parallelepiped defines the intersection of m extreme torque planes. Thus, a vertex occurs when all joint torques are at their extreme capabilities, i.e.,

$$v_j = [\tau_{1_{\text{ext}}} \quad \tau_{2_{\text{ext}}} \quad \dots \quad \tau_{m_{\text{ext}}}]^T \quad (3)$$

where $\tau_{i_{\text{ext}}}$ denotes the extreme capabilities of the i^{th} actuator, i.e., $\tau_{i_{\min}}$ or $\tau_{i_{\max}}$. The total number of vertices in the m -dimensional parallelepiped is $v_{T_m} = 2^m$ (Chiacchio et al., 1997).

3.2 Linear transformation

Visvanathan and Milor (1986) investigated the problems in analog integrated circuits while accounting for the tolerance variations of the principal process parameters. The problem involved the mapping of a parallelepiped under a linear transformation. Their mathematical formulation is similar to the one used for analyzing wrench capabilities in this work. Let the coordinates of the vertices of a parallelepiped in \mathbb{R}^m be v_j , for $j = 1, \dots, 2^m$. Through a linear transformation from \mathbb{R}^m to \mathbb{R}^n , such as $\mathbf{F} = \mathbf{J}^T \boldsymbol{\tau}$, the m -dimensional parallelepiped becomes a polytope (Visvanathan and Milor, 1986). A polytope is a convex region, i.e., any two points inside the polytope can be connected by a line that completely fits inside the polytope. An n -dimensional convex polytope is bounded by $(n-1)$ -dimensional facets or hyperplanes, e.g., linear edges in \mathbb{R}^2 bounding a polygon or planar facets in \mathbb{R}^3 bounding a polyhedron.

A polytope P can be completely characterized by mapping all the vertices of the parallelepiped and enclosing them in a convex hull, i.e.,

$$P = \text{convh}\{\mathbf{J}^T v_j, j = 1, \dots, 2^m\} \quad (4)$$

where convh denotes a convex hull operator which encloses all the extreme points forming the feasible region of the torque space in the wrench space. A closed bounded convex set is the convex hull of its extreme points (Rockefeller, 1997).

The total number of vertices in the polytope (v_{T_n}) depends on the dimension of the two spaces.

3.3 Non-redundant planar manipulators

For non-redundant manipulators ($n = m$) the number of vertices in the polytope equals the number of vertices in the m -dimensional parallelepiped, i.e., $v_{T_n} = v_{T_m} = 2^m$, and the vertices

of the polytope are the image of the vertices of the m -dimensional parallelepiped (Chiacchio et al., 1996), i.e.,

$$p_j = \mathbf{J}^T v_j \quad (5)$$

where p_j and v_j are the vertices of the polytope and parallelepiped, respectively.

The linear transformation between the two spaces also makes that the edges and facets of the polytope are the corresponding image of the edges and facets of the m -dimensional parallelepiped.

For a planar parallel manipulator the vertices of the wrench polytope are found as follows:

$$p_j = \mathbf{J}^T v_j \quad (6)$$

$$\begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \gamma_{1,3} \\ \gamma_{2,1} & \gamma_{2,2} & \gamma_{2,3} \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{bmatrix} \begin{bmatrix} \tau_{1\text{ext}} \\ \tau_{2\text{ext}} \\ \tau_{3\text{ext}} \end{bmatrix}$$

where $\gamma_{i,j}$ denotes the elements of \mathbf{J}^T . There are eight vertices (2^3) due to the combination of the extreme torque capabilities, i.e., $\tau_{i\text{ext}}$ can be either $\tau_{i\text{min}}$ or $\tau_{i\text{max}}$.

Fig. 2 illustrates the linear transformation of the torque capabilities of a non-redundant planar parallel manipulator from the torque space to the wrench space. Fig. 2 also shows the corresponding image of the vertices, edges, and facets between the parallelepiped and the polytope.

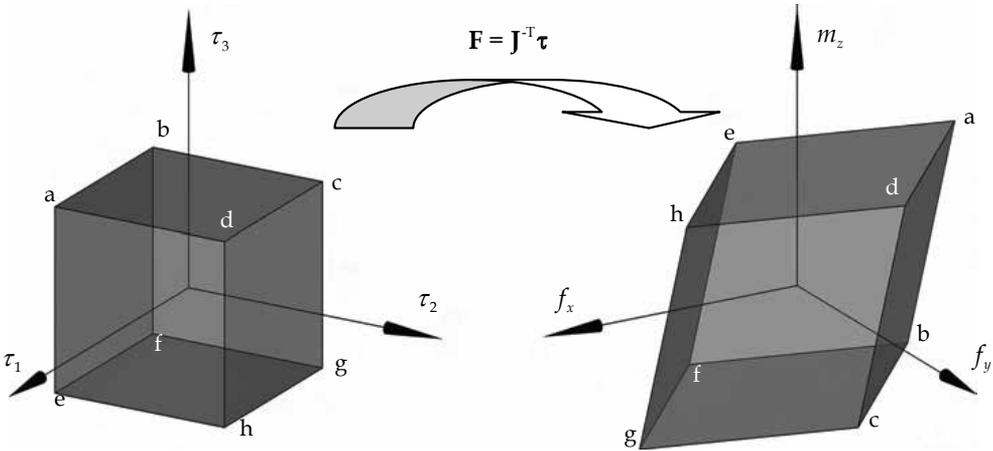


Fig. 2. Linear transformation of a parallelepiped to a polytope of a non-redundant PPM and image projection of vertices, edges, and facets.

The resulting wrench polytope of a non-redundant manipulator has the following characteristics:

- i. Any point outside the polytope is a wrench that cannot be applied or sustained;
- ii. Any point inside the polytope is achieved with actuators that are not working at their extreme capabilities;

- iii. Any point on a facet of the polytope has one actuator working at an extreme capability;
- iv. Any point on an edge of the polytope has two actuators working at their extremes;
- v. Any vertex of the polytope has all three actuators working at their extremes.

3.4 Redundant manipulators

For redundant manipulators ($n < m$) the number of vertices in the polytope is less than the vertices of the m -dimensional parallelepiped, i.e., $v_{T_n} < v_{T_m}$. In this case, the vertices of the polytope are formed with the mapping of some of the vertices of the m -dimensional parallelepiped, i.e.,

$$p_k \subset \mathbf{J}^{-T} v_j \quad (7)$$

with $k < j$. The points that do not form the vertices of the polytope are internal points in \mathcal{P} . Let the potential vertices (p_j) of the polytope be all the projected vertices of the m -dimensional parallelepiped in \mathbb{R}^n . Thus, the potential vertices are determined as follows:

$$p_j = \mathbf{J}^{-T} v_j \quad (8)$$

$$\begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,m} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,m} \\ \gamma_{3,1} & \gamma_{3,2} & \cdots & \gamma_{3,m} \end{bmatrix} \begin{bmatrix} \tau_{1\text{ext}} \\ \tau_{2\text{ext}} \\ \vdots \\ \tau_{m\text{ext}} \end{bmatrix}$$

The number of external vertices may vary. For instance, the projection of a cube on a plane may lead to six external vertices (general projection) or four external vertices (projection normal to a coordinate axis).

The number of vertices of the wrench polytope depends on the pose of the manipulator, which defines the elements of the linear transformation matrix, \mathbf{J}^{-T} .

Finding the external vertices of a polytope can be computationally expensive. Generating a polytope through a convex hull has been studied thoroughly in the field of computational geometry and the goal has been to make a more efficient algorithm. Chand and Kapur (1970) proposed the so-called gift wrapping algorithm, where the facets of a polytope are found by determining the angles between one vertex and the rest of the points. The minimum and maximum angles correspond to the hyperplanes passing through that point. Visvanathan and Milor (1986) proposed an algorithm that searches in the directions that are orthogonal to each of the known hyperplanes. New vertices and hyperplanes are formed and the process is repeated. Bicchi et al. (1995) presented an algorithm that involves slack variables that transform the inequality constraints of the actuator limits into equality constraints. Lee (1997) proposed a method for determining the vertices of twist polytopes using vector algebra. Hwang et al. (2000) developed a recursive algorithm that removes all the internal points when first encountered. Hwang et al. (2000) also showed that even though the number of potential vertices grows exponentially (2^m), the number of external points increases linearly.

The scope of this work is not to develop a new algorithm for determining the external vertices and facets of a polytope, although some of the concepts that will be described in this work may be used to generate an even more efficient algorithm.

The geometrical interpretation of the internal points is illustrated with the following example. Assume a planar manipulator with a redundant joint. Thus, the linear transformation maps the torque capabilities from \mathbb{R}^4 to \mathbb{R}^3 .

Fig. 3 illustrates the resulting polytope as a wire frame. This polytope is formed with the convex hull of the extreme points. The same polytope is repeated in all the sub-plots. Each sub-plot shows the regions in which one of the actuator torques is working at its extreme capabilities. The darker and lighter regions denote the two extremes $\tau_{i_{\min}}$ and $\tau_{i_{\max}}$, respectively. These regions are convex sets themselves, here defined as inner polytopes. The un-shaded region of each plot represents the space in which the actuator works within its capabilities.

The overall polytope was generated with 16 potential vertices, of which 14 are external and 2 are internal. The external vertices are illustrated with dots. Internal vertices lead to particularly low wrenches despite having all torques working at their extreme capabilities.

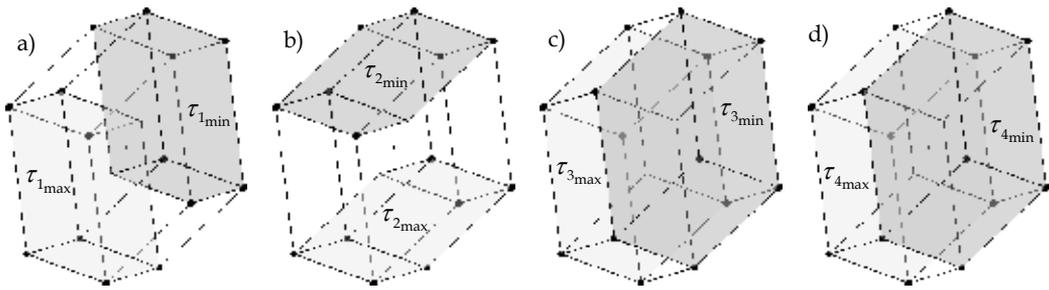


Fig. 3. Polytope of a redundant planar parallel manipulator with shaded regions showing torques at extreme capabilities: a) Extremes of τ_1 , b) Extremes of τ_2 , c) Extremes of τ_3 , and d) Extremes of τ_4 .

Based on these plots, the following conclusions can be made. While for a non-redundant manipulator each facet of the m -dimensional parallelepiped corresponded to a facet of the polytope; for a redundant manipulator this projection leads to volumes in the polytope. Also, each edge of the polytope is defined with the projection of three torques set at their extremes; while, each facet is formed with two torques at their extremes.

Further actuation would result in more complicated polytopes and the number of internal points will increase exponentially (Hwang et al., 2000). In general, the number of geometrical entities in a wrench polytope is determined with the number of combinations involving torque at their extreme capabilities and the associated magnitude, e.g., say $\tau_{i_{\text{ext}}}$ is at an extreme capability which can be of magnitude $\tau_{i_{\min}}$ or $\tau_{i_{\max}}$. The number of geometrical entities is summarized in Table 1.

Geometrical Entity (Internal and External)	Actuators at Extreme	Number of Geometrical Entities
Vertices	m	2^m
Edges	$m-1$	$2^{m-1}m$
Facets	$m-2$	$2^{m-3}(m-1)m$

Table 1. Geometrical entities of a wrench polytope for redundant manipulators.

The resulting wrench polytope of a redundant manipulator has the following characteristics:

- i. Any point outside the polytope is a wrench that cannot be applied or sustained;
- ii. Any point inside the polytope is achieved with actuators that may or may not work at their extreme capabilities;
- iii. Any point on a facet of the polytope has $m-2$ actuators working at their extremes;
- iv. Any point on an edge of the polytope has $m-1$ actuators working at their extremes;
- v. Any vertex of the polytope has all m actuators working at their extremes.

4. Wrench performance indices

4.1 Operational conditions

A wrench polytope represents the region in which the manipulator can apply feasible wrenches. Unfortunately, a major drawback of this approach compared to the ellipsoid approach is the efficiency of the algorithm. Determining the axes of the ellipsoid by applying SVD to \mathbf{J} is definitely more efficient than constructing a polytope. A small eigenvalue indicates that the manipulator requires large actuator torques to sustain an exerted wrench.

Nonetheless, the best representation, from a design perspective, may not be the polytope itself, but rather a set of indices that characterize it. These points may lie on facets, edges, or vertices of the wrench polytope, and represent maximum/minimum values of either moments or forces. Thus, these points, which are referred to as wrench performance indices, allow the determination of either force or moment ranges.

Under operational conditions, the manipulator performance is dictated by the requirements of the application. These requirements establish some parameters of moments and forces acting on the manipulator. This is, the range of forces can be determined based on moment requirements; similarly, the range of moments can be determined based on force requirements. For the force analysis, there are two ranges of forces that can be determined. Finotello et al. (1998) defined these forces as maximum available value (MAV) and maximum isotropic value (MIV). Herein, MAV and MIV are denoted as F_{av} and F_{is} , respectively.

Assume a wrench with a constant moment, thus the polytope is reduced to a polygon, i.e., the polytope is sliced at the constant moment yielding a polygon. The area enclosed by the polygon represents the force capabilities of the manipulator. The maximum available force (F_{av}) is the farthest distance from the center of the force space to the polygon. This force can be only applied in a particular direction and corresponds to a vertex of the polygon. The maximum isotropic force (F_{is}) is the shortest distance from the center of the force space to the polygon. Fig. 4 illustrates the force polygon of a 3-RRR PPM, where the underline denotes the actuated joints and is indicated in the figure with τ_1 , τ_2 , and τ_3 . For an arbitrary

direction α , the distance from the center of the force space to any point within the polygon is proportional to the magnitude of the force that can be applied or sustained. Fig. 4 also shows an arbitrary force vector $\mathbf{f} = [f_x, f_y]^T = [f \cos\alpha, f \sin\alpha]^T$ and the maximum available (F_{av}) and isotropic (F_{is}) forces.

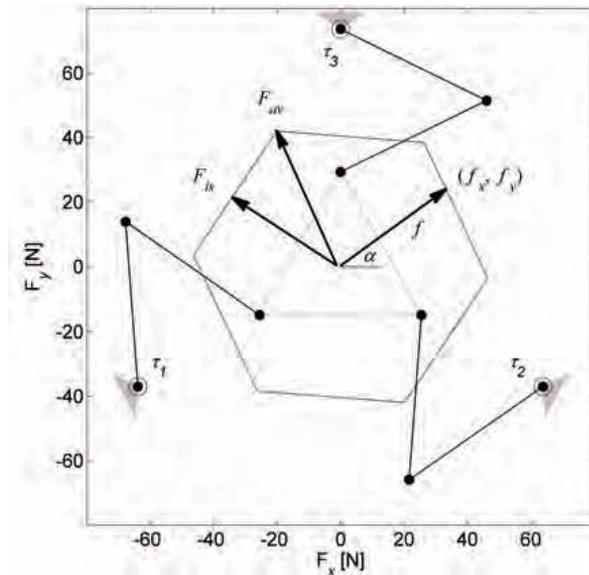


Fig. 4. Force polygon and force capabilities.

In this work, six different scenarios of operational conditions in which the forces and moments interact are presented. Table summarizes the six operational conditions which lead to two force analyses and four moment analyses.

Operational Condition	Analysis
Prescribed Moment	Force Analysis: Find Range of Available and Isotropic Forces
Largest Allowable Force with an Associated Moment	
Prescribed Force (magnitude and direction)	Moment Analysis: Find Range of Moments
Largest Allowable Moment with an Associated Force	
Prescribed Isotropic Force (magnitude)	
Prescribed Available Force (magnitude)	

Table 2. Operational condition and corresponding analyses.

4.2 Explicit analysis

To determine a particular performance index, Eq. (8) is rearranged as a linear system of three equations of the form $\mathbf{Ax} = \mathbf{b}$; where \mathbf{x} is a vector that contains all of the unknown variables, either wrench or torque space coordinates, \mathbf{A} is a coefficient matrix, and \mathbf{b} is a vector that contains the torques that are set to their extreme capabilities.

If the performance index value lies on a vertex of the polytope, all m actuators will be set to their extreme capabilities. There are 2^m possible combinations due to the two extreme magnitudes of the torque outputs ($\tau_{i_{\min}}$ or $\tau_{i_{\max}}$).

If the performance index value lies on an edge of the polytope, $m-1$ actuators are set to their extreme output capabilities, while the remaining actuator torque is working within its output range and is referred to as being in transition (τ_i). Torques that are not at their extreme capabilities are said to be in transition because they transfer from one torque limit to the opposite limit, e.g., from $\tau_{i_{\min}}$ to $\tau_{i_{\max}}$. A torque in transition is an unknown variable in vector \mathbf{x} . There are $2^{m-1}m$ combinations.

If the performance index value lies on a facet of the polytope, $m-2$ actuators are set to their extreme capabilities and two torques are in transition. There are $2^{m-3}(m-1)m$ combinations.

Once all the combinations are evaluated, the performance index can be determined by verifying the maximum f or m_z among all of the combinations. If the problem involves finding a torque in transition, it is important to verify that this torque does not exceed its torque output capabilities.

Table 3 summarizes the operational condition, the number of actuators working at their extreme capabilities, a list of known and unknown variables, and the number of combinations that are required to evaluate. This procedure is equivalent for both non-redundant and redundant planar parallel manipulators.

Operational Condition	Actuators at Extremes	Variables Known Unknown	Number of Combinations
Maximum Force with a Prescribed Moment (${}^{\text{pm}}F_{av}$ and ${}^{\text{pm}}F_{is}$)	$m - 1$	$m_z \mid f_x \text{ and } f_y$	$2^{m-1}m$
Maximum Allowable Force with an Associated Moment (${}^{\text{am}}F_{av}$ and ${}^{\text{am}}F_{is}$)	m	$\mid m_z, f_x \text{ and } f_y$	2^m
Maximum Moment with a Prescribed Force (${}^{\text{pf}}M_z$)	$m - 2$	$f_x \text{ and } f_y \mid m_z$	$2^{m-3}(m-1)m$
Maximum Allowable Moment with an Associated Force (${}^{\text{af}}M_z$)	m	$\mid m_z, f_x \text{ and } f_y$	1
Maximum Moment with a Prescribed Isotropic Force (${}^{\text{pif}}M_z$)	$m - 2$	$f \mid m_z \text{ and } \alpha$	$2^{m-3}(m-1)m$
Maximum Moment with a Prescribed Available Force (${}^{\text{paf}}M_z$)	m $m - 1$	$\mid m_z, f_x \text{ and } f_y$ $f \mid m_z \text{ and } \alpha$	2^m $2^{m-1}m$

Table 3. Wrench performance indices of planar parallel manipulators.

4.2 Force analysis

Maximum Force with a Prescribed Moment. If the moment must be preserved in the requirements of the application, either zero or any other value, the polytope is reduced to a polygon. In Fig 5a, the dark area illustrates the polygon at $m_z=0$, while the other lines show

polygons at different moments. For this problem, m_z must be specified yielding the following set of unknown variables: $\mathbf{x} = [f_x \ f_y \ \tau_t]^T$. Thus, Eq. (8) is rearranged as follows:

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} 1 & 0 & -\gamma_{1,t} \\ 0 & 1 & -\gamma_{2,t} \\ 0 & 0 & -\gamma_{3,t} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ \tau_t \end{bmatrix} = \begin{bmatrix} \gamma_{1,1} & \dots & \gamma_{1,t-1} & \gamma_{1,t+1} & \dots & \gamma_{1,m} \\ \gamma_{2,1} & \dots & \gamma_{2,t-1} & \gamma_{2,t+1} & \dots & \gamma_{2,m} \\ \gamma_{3,1} & \dots & \gamma_{3,t-1} & \gamma_{3,t+1} & \dots & \gamma_{3,m} \end{bmatrix} \begin{bmatrix} \tau_{1\text{ext}} \\ \vdots \\ \tau_{t-1\text{ext}} \\ \tau_{t+1\text{ext}} \\ \vdots \\ \tau_{m\text{ext}} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ m_z \end{bmatrix} \quad (9)$$

The maximum available force (${}^{\text{pm}}F_{av}$) corresponds to the largest value of f that is evaluated with the combinations, where $f = \sqrt{f_x^2 + f_y^2}$. The maximum isotropic force (${}^{\text{pm}}F_{is}$) is determined as the shortest distance from the center of the force space to the polygon. ${}^{\text{pm}}F_{av}$ and ${}^{\text{pm}}F_{is}$ represent a point on an edge and a point on a facet of the polytope.

Maximum Allowable Force with an Associated Moment. If a moment does not affect the requirement of the application, the manipulator can reach the largest available and isotropic forces. To achieve these forces a particular moment must be associated with them. The set of unknown variables of the $\mathbf{Ax} = \mathbf{b}$ problem is $\mathbf{x} = [f_x \ f_y \ m_z]^T$. A force polygon may be generated by projecting the vertices of the polytope on the force plane. Fig 5 b illustrates the projection of the polytope vertices on the force space plane. The available and isotropic forces (${}^{\text{am}}F_{av}$ and ${}^{\text{am}}F_{is}$) are respectively the longest and shortest distances from the center of the force space to the projected polygon.

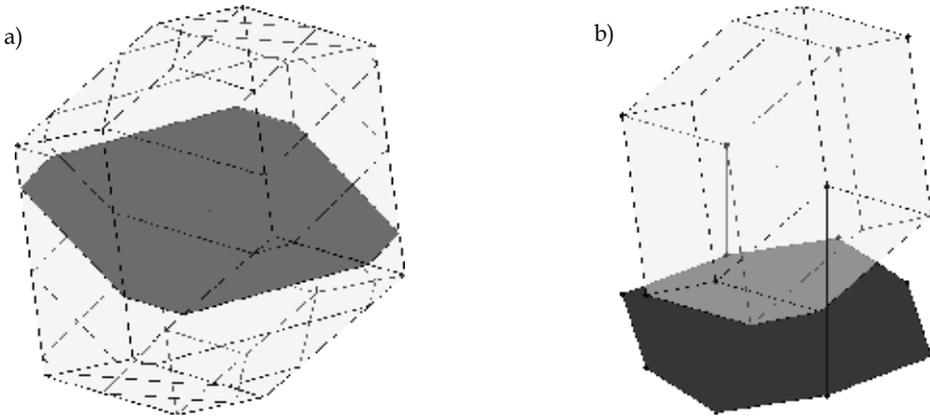


Fig. 5. Force analysis: a) Maximum force with a prescribed moment and b) Maximum force with an associated moment.

4.3 Moment analysis

Maximum Moment with a Prescribed Force. For a fully described force (f and α), the force vector may be drawn within the polytope and the set of moments (${}^{\text{pf}}M_z$) that can be reached with this force can be determined. Eq. (8) is rearranged as an $\mathbf{Ax} = \mathbf{b}$ problem with

$\mathbf{x} = [m_z \ \tau_{i_a} \ \tau_{i_b}]^T$. The largest and smallest m_z that can be obtained while keeping the torques in transition within their capabilities define the range of ${}^{pf}M_z$. Fig. 6a illustrates an arbitrary force and the vertical line represents the range of moments.

Maximum Allowable Moment with an Associated Force. If the force does not affect the application, the maximum range of moments (${}^{af}M_z$) has an associated force, i.e., a specific force must be applied to achieve the largest moment. To find the maximum moment all the actuators are set to their maximum capabilities. To achieve the largest range of ${}^{af}M_z$, the third row of Eq. (8) is arranged to obtain the combination of monomials that yields the maximum and the minimum m_z . Thus, only a single evaluation is required for each extreme value. The highest and lowest vertices of the polytope represent this performance index.

Maximum Moment with a Prescribed Isotropic Force. Assume that the manipulator is required to apply or sustain the same force in all directions, i.e., an isotropic force f_{is} . The region of moments that can attain this force may be seen as a cylinder of radius f_{is} that is fully contained within the polytope, as shown in Fig. 6b. The range of moments (${}^{pif}M_z$) is the height of the cylinder. The cylinder intersects facets of the polytope. This case cannot be solved by simply rearranging Eq. (8). As an alternative, the maximum and minimum ${}^{pif}M_z$ are determined by comparing the resulting isotropic moment associated with every plane of the polytope. Isotropy is ensured with the plane that yields the minimum of the maximum m_z moment. A detailed formulation of this problem is described in Firmani et al. (2007a).

Maximum Moment with a Prescribed Available Force. Assume that the manipulator is required to apply a large force regardless of its direction, i.e., available force f_{av} . This case may be seen as the intersection of a cylinder of radius f_{av} with a point on the polytope which is the farthest away from the $m_z=0$ plane, as shown in Fig 6c. The range of moments (${}^{paf}M_z$) is the height of this cylinder. The cylinder usually intersects an edge of the polytope, but in some particular cases the intersection may happen with a facet or a vertex of the polytope.

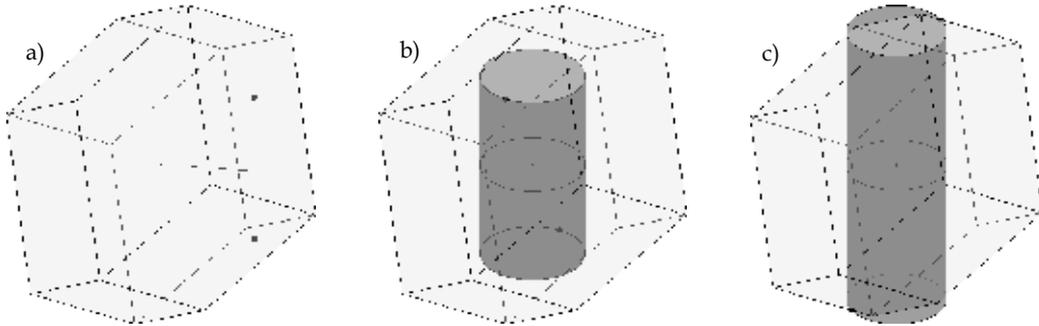


Fig. 6 Moment analysis. Maximum moment with a prescribed a) Force, b) Isotropic force, and c) Available force.

5. Wrench workspaces

5.1 Planar parallel manipulator architectures

For three-branch PPM layouts, there are seven possible architectures: RRR, RPR, PRR, RRP, PPR, RPP, and PRP, where R and P denote a revolute and prismatic joint, respectively. Of these seven architectures, the PRR and the RRP are kinematically equivalent. Likewise, the

PPR and the RPP. Considering only one kinematically equivalent architecture yields only five unique architectures. Eliminating those architectures with two prismatic joints, as they are not convenient for implementation as PPMs, leaves only the RRR, RPR, and PRR architectures to be studied. Based on these architectures, three actuation schemes are analyzed: non-redundant PPMs, in-branch redundant PPMs, and branch redundant PPMs.

Non-Redundant PPMs. By considering the first joints actuated, the inertia of the mechanism is kept low allowing manipulators to be used for high-speed applications. Thus, the actuation layouts $3\text{-}\underline{\text{RRR}}$, $3\text{-}\underline{\text{RPR}}$, and $3\text{-}\underline{\text{PRR}}$ are considered. Fig. 7 shows the schematics of the three non-redundant PPM layouts. The fixed and mobile platforms are similar in every case: the base triangle edge lengths are 0.5 m and the end-effector triangle edge lengths are 0.2 m. For the $3\text{-}\underline{\text{RRR}}$, the lengths of the first and second links of each branch are 0.2 m. The torque limits of the actuators are ± 4.2 Nm. For the $3\text{-}\underline{\text{RPR}}$, the torque limits are ± 4.2 Nm and the prismatic joints' extension limits are 0.0 to 0.4 m. For the $3\text{-}\underline{\text{PRR}}$, the prismatic joints' orientations are 0° , 120° , 240° , the prismatic joints' extension limits are 0.0 to 1.0 m and the force limits are ± 10 N, while the lengths of the second links are 0.23 m.

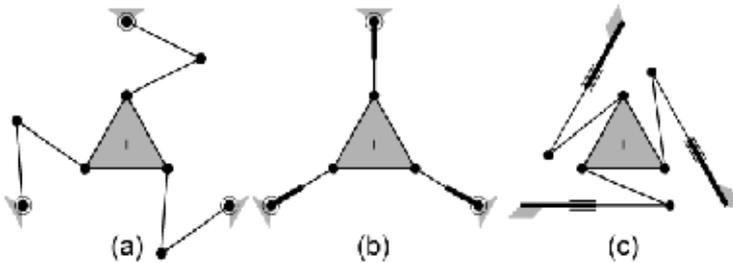


Fig. 7. Non-redundant PPM layouts: a) $3\text{-}\underline{\text{RRR}}$, b) $3\text{-}\underline{\text{RPR}}$, c) $3\text{-}\underline{\text{PRR}}$

In-Branch Redundant PPMs. With this actuation scheme, the second joints of every branch are actuated yielding the $3\text{-}\underline{\text{RRR}}$, $3\text{-}\underline{\text{RPR}}$, and $3\text{-}\underline{\text{PRR}}$ layouts. Fig. 8 shows schematics of the in-branch redundant PPM layouts. These PPMs have the same dimensions and actuator capabilities as the ones used for the non-redundant PPMs. In addition, for the $3\text{-}\underline{\text{RRR}}$ and the $3\text{-}\underline{\text{PRR}}$ the second joint is actuated and the torque limits are ± 2.1 Nm; whereas, for the $3\text{-}\underline{\text{RPR}}$, the force limits of the prismatic joints are ± 10 N.

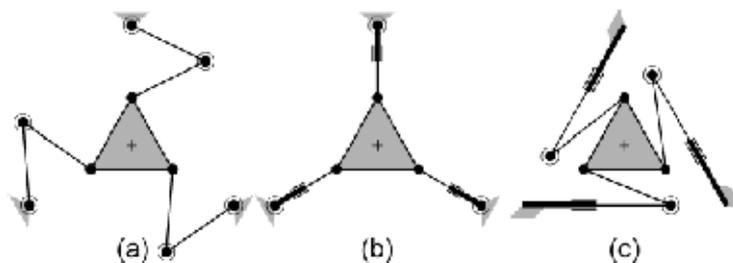


Fig. 8. In-branch redundant PPM layouts: a) $3\text{-}\underline{\text{RRR}}$, b) $3\text{-}\underline{\text{RPR}}$, c) $3\text{-}\underline{\text{PRR}}$

Branch Redundant PPMs. An additional branch is added to the non-redundant PPMs yielding the $4\text{-}\underline{\text{RRR}}$, $4\text{-}\underline{\text{RPR}}$, and $4\text{-}\underline{\text{PRR}}$ layouts. The same dimensions and actuator

capabilities used for the non-redundant PPMs are employed for this analysis. For the 4-PRR, the prismatic joints' orientations are shown in Fig. 9c.

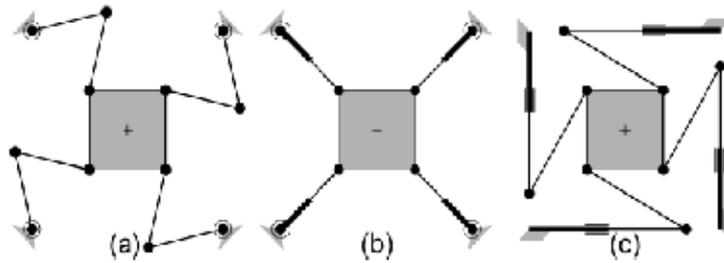


Fig. 9. Branch redundant PPM layouts: a) 4-RRR, b) 4-RPR, c) 4-PRR.

5.2 Wrench workspaces of planar parallel manipulators

In this work, the platform orientations are held constant at 0° and the assembly modes for the RRR and PRR architectures are shown in Figs. 7 to 9. Four analyses are considered:

- a) Maximum force with a prescribed moment at $m_z = 0$, i.e., pure force analysis,
- b) Maximum allowable force with an associated moment, i.e., absolute force analysis,
- c) Maximum moment with a prescribed force at $f = 0$, i.e., pure moment analysis,
- d) Maximum allowable moment with an associated Force, i.e., absolute moment analysis.

The third analysis ($f = 0$) is a special case that also involves the operational condition of prescribed isotropic and available forces. Examples of prescribed isotropic and available forces workspaces, where $f \neq 0$, can be found in Firmani et al. (2007b).

Figs. 10 to 18 show the wrench capabilities of the previously described manipulators. These capabilities are presented in two dimensional plots whose axes indicate the location of the mobile platform throughout the workspace [m]. At each location, either the maximum force or maximum moment capability is determined and illustrated with a dot using a grayscale gradient. Nonetheless, some of the magnitudes were very large compared to the rest of the results in the workspace. Large values are caused by the proximity of the manipulator to a singular configuration and this spoils the overall grayscale gradient.

Parallel manipulators are affected by inverse and direct singularities. An inverse singularity configuration usually occurs at the boundaries of the workspace. For the RRR architecture, a branch is either fully extended or folded back. For the RPR architecture, the displacement of a prismatic joint is zero. For the PRR architecture, the second link of a branch is perpendicular to the prismatic joint. Under these configurations, the manipulator cannot apply any force along one direction, but in theory it can sustain an infinite load. Similarly, the manipulator cannot apply any moment but in theory it can sustain an infinite moment, if an associated force goes to infinity. Since infinite wrench magnitudes would destroy the grayscale gradient, hence, the authors opted to cap these values.

A direct singularity occurs when the branch resultant forces together do not span an external wrench, i.e., the branch resultant forces intersect at a common point (planar pencil). If the branch forces intersect at infinity, an external force, normal to the branch forces, cannot be sustained, i.e., $f_{is} = 0$. If the intersection occurs somewhere else, an external moment applied to the mobile platform cannot be balanced by the actuators, i.e., $m_z = 0$.

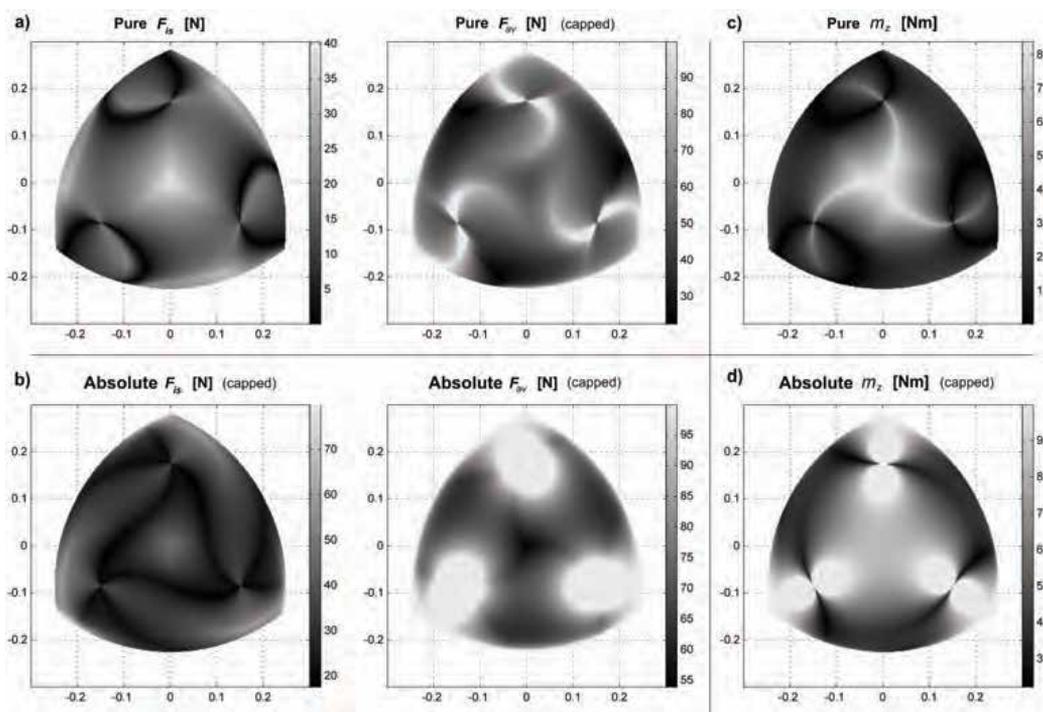


Fig. 10. Wrench workspaces for the 3-RRR PPM.

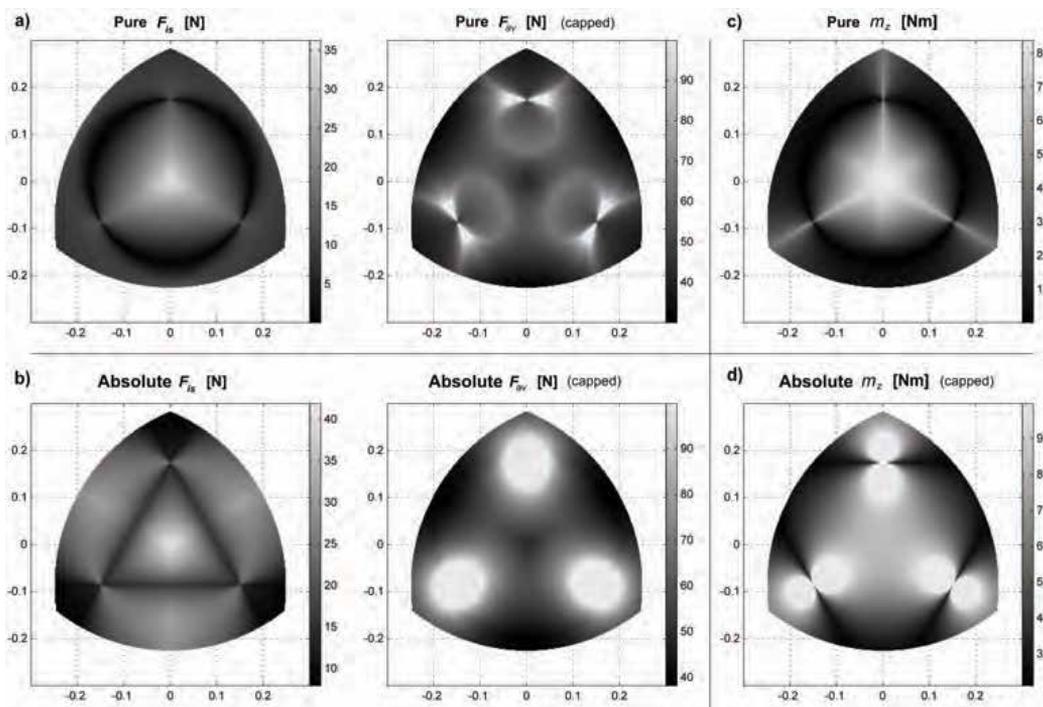


Fig. 11. Wrench workspaces for the 3-RPR PPM.

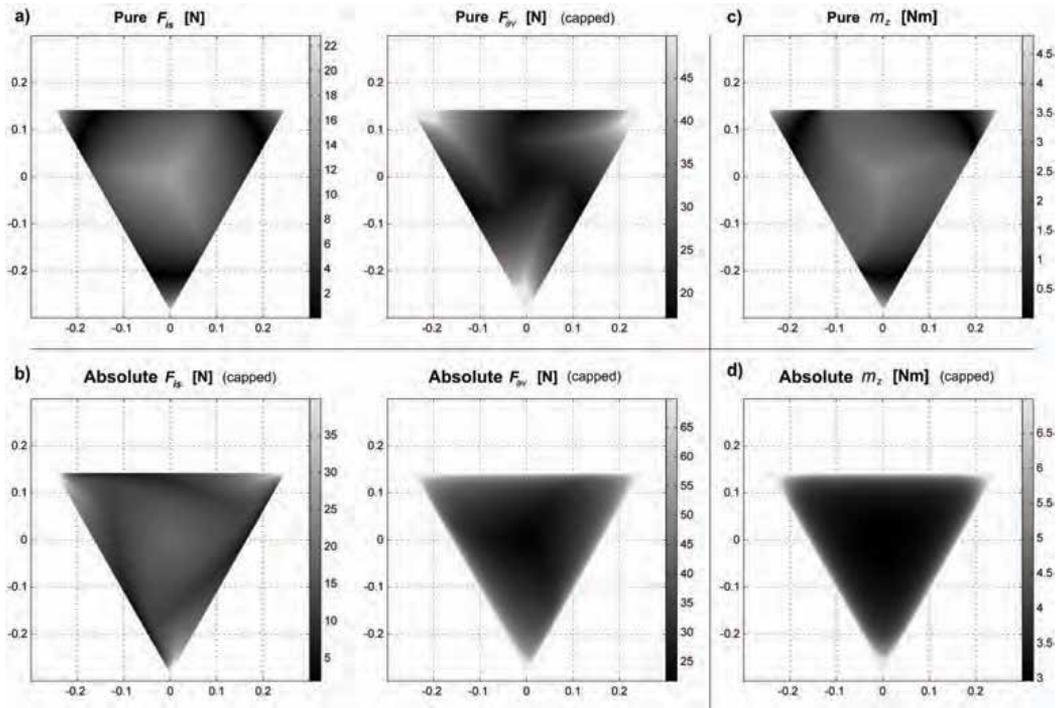


Fig. 12. Wrench workspaces for the 3-PRR PPM.

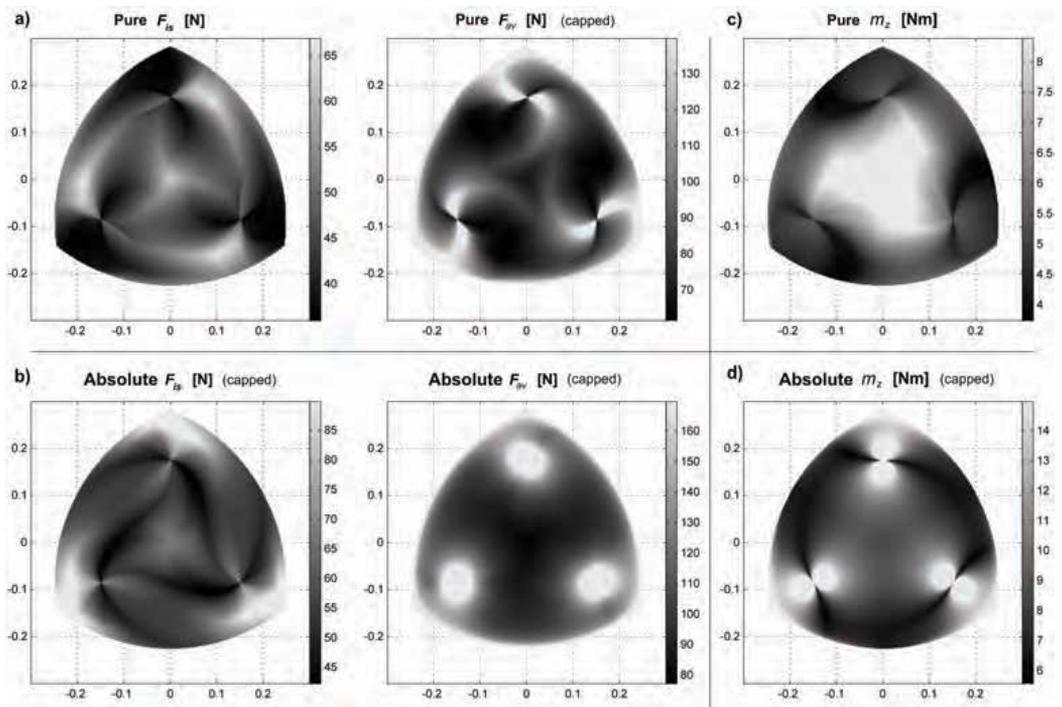


Fig. 13. Wrench workspaces for the 3-RRR PPM.

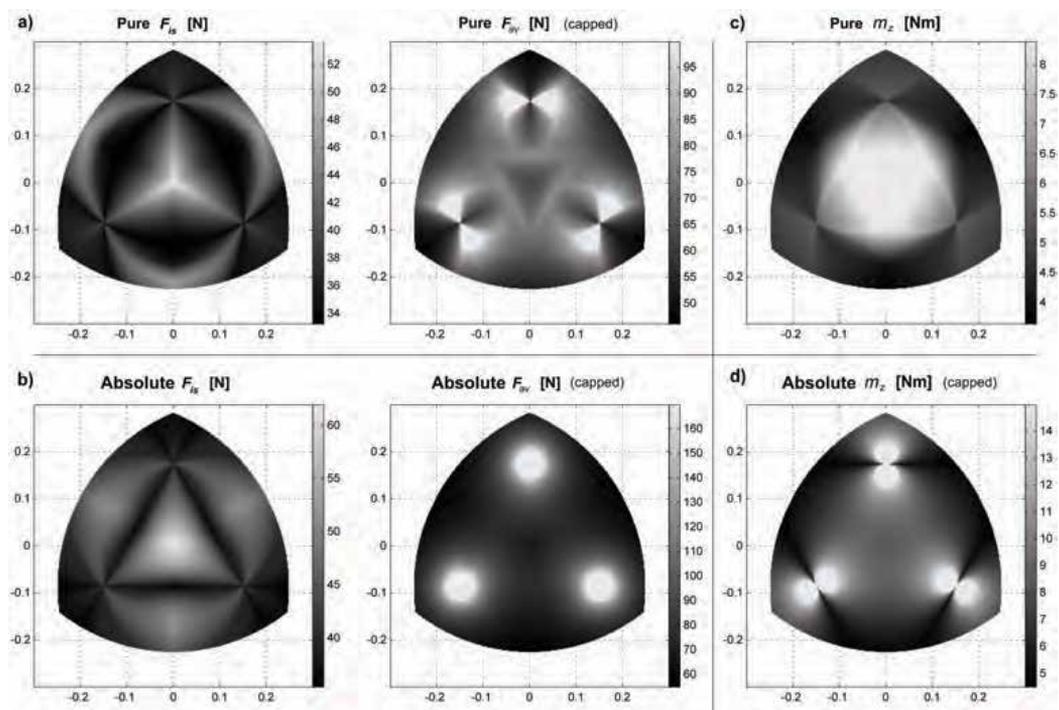


Fig. 14. Wrench workspaces for the 3-RPR PPM.

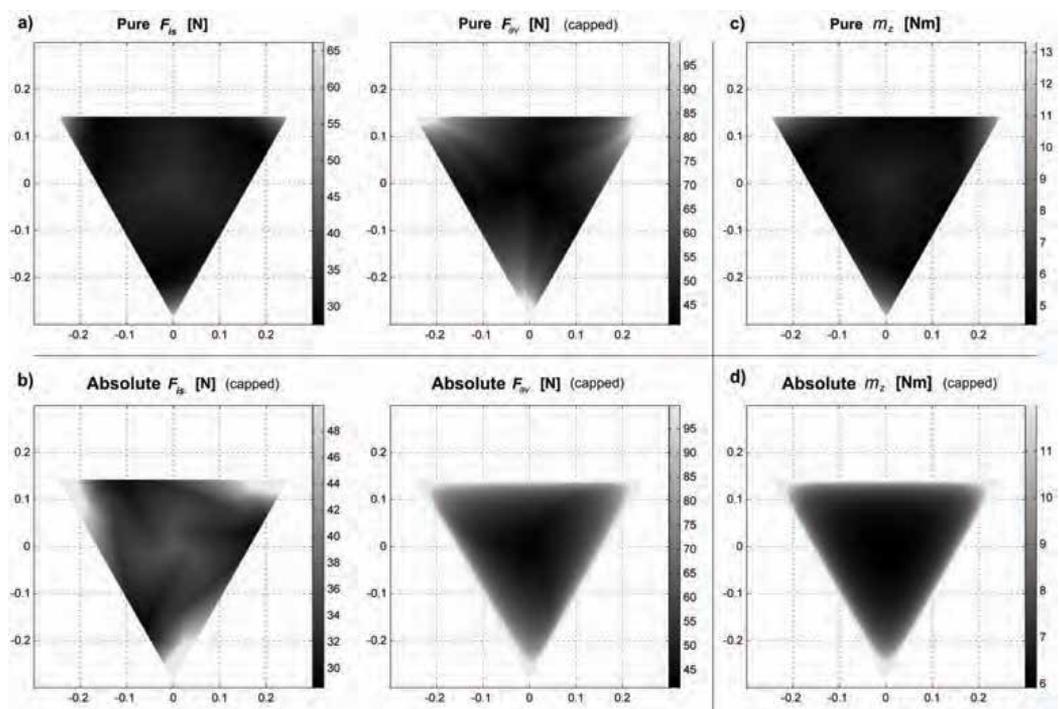


Fig. 15. Wrench workspaces for the 3-PRR PPM.

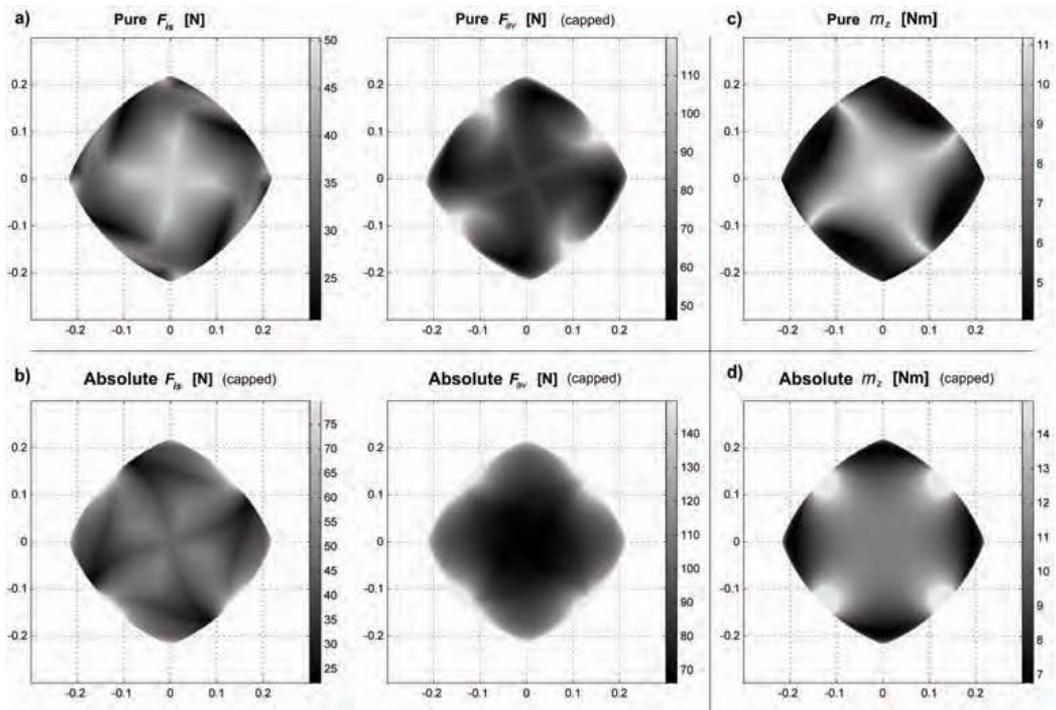


Fig. 16. Wrench workspaces for the 4-RRR PPM.

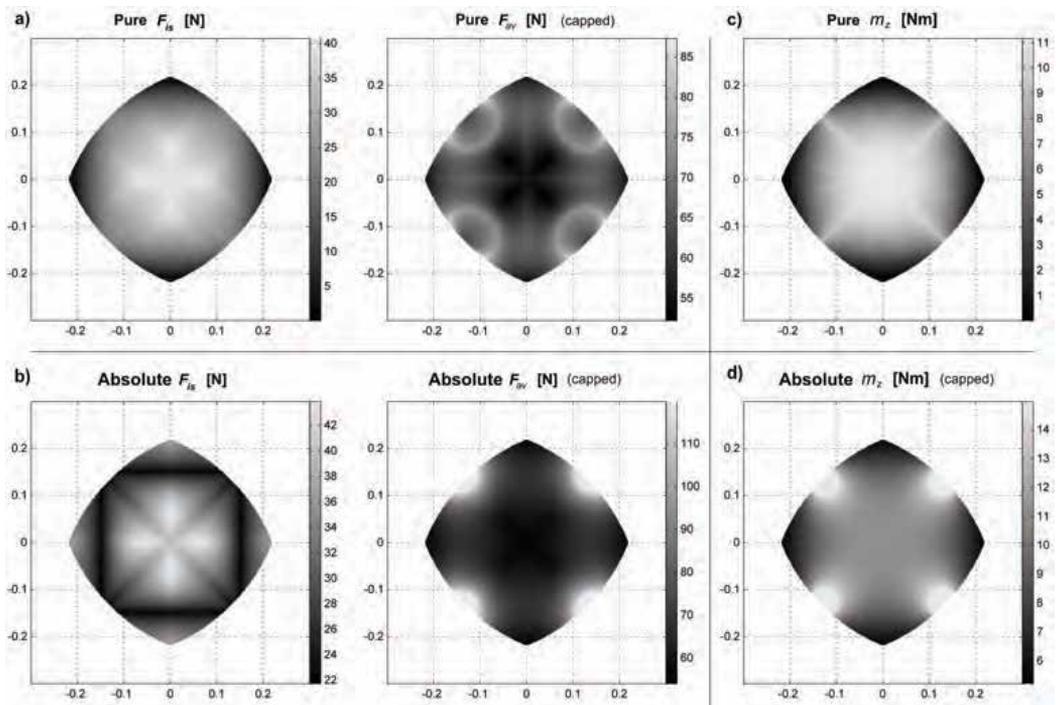


Fig. 17. Wrench workspaces for the 4-RPR PPM.

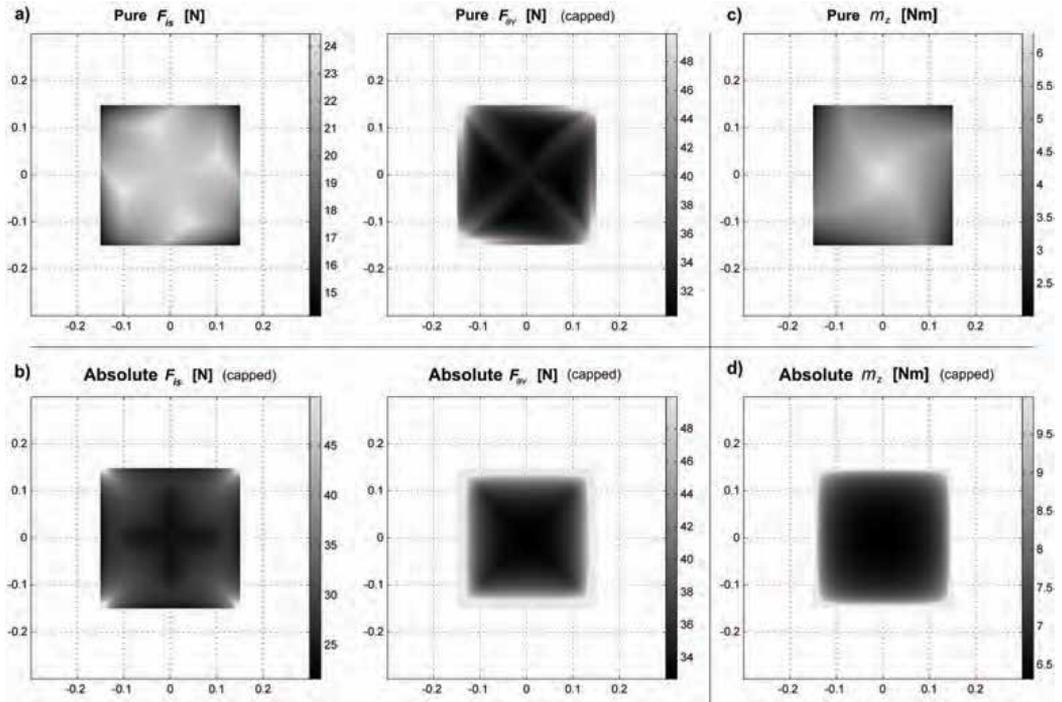


Fig. 18. Wrench workspaces for the 4-PRR PPM.

5.3 Discussion of the results

The plots show regions where larger forces or moments can be applied/sustained. Table 4 shows a numerical comparison of the different actuation layouts. Minimum and median values are adopted as indices of comparison and are denoted as $\min(*)$ and $\tilde{*}$, respectively. These indices are preferred over maximum or mean values because they are not affected by infinite or very large results caused by singularities.

PPM Layout	Pure F_{is} [N]		Pure F_{av} [N]		Abs. F_{is} [N]		Abs. F_{av} [N]		Pure m_z [Nm]		Abs. m_z [Nm]	
	$\min(F_{is})$	\tilde{F}_{is}	$\min(F_{av})$	\tilde{F}_{av}	$\min(F_{is})$	\tilde{F}_{is}	$\min(F_{av})$	\tilde{F}_{av}	$\min(m_z)$	\tilde{m}_z	$\min(m_z)$	\tilde{m}_z
3-RRR	0.00	17.13	22.33	57.11	17.41	31.22	53.80	79.76	0.00	2.00	2.21	6.87
3-RPR	0.00	9.57	29.90	53.42	7.88	22.98	38.08	63.23	0.00	1.62	2.10	6.43
3-PRR	0.00	8.57	17.06	25.06	1.83	13.95	21.59	34.37	0.00	1.90	2.95	3.58
3-RRR	35.92	46.92	61.40	87.25	42.17	56.29	77.05	114.06	3.73	5.51	5.55	8.98
3-RPR	33.19	37.22	45.97	70.25	35.34	43.05	54.59	77.43	3.63	5.03	4.47	8.27
3-PRR	27.48	31.66	40.86	49.18	28.49	34.05	41.20	55.75	4.40	5.07	5.91	7.19
4-RRR	20.61	34.47	46.30	66.24	22.02	46.14	66.05	92.14	4.06	7.01	6.75	11.06
4-RPR	0.01	26.81	52.17	65.58	21.69	30.92	53.89	67.76	0.00	7.71	5.19	11.03
4-PRR	14.14	22.32	30.27	33.29	21.55	27.84	32.66	38.63	2.00	4.56	6.31	7.10

Table 4. Wrench performance indices of planar parallel manipulators.

The results clearly show that the addition of redundancy, whether in-branch or branch redundancy, has a dramatic improvement on the wrench capabilities for PPMs when compared to non-redundant PPMs. In particular, in-branch redundancy provides greater forces; whereas, branch redundancy offers greater moments.

6. Future research

A number of possible avenues for future investigation exist. For this work, the orientation of the platform was kept constant. Investigating the effects on wrench capabilities for PPMs by varying the orientation of the platform would provide a better understanding of the full range of wrench capabilities of a given architecture. Another avenue of investigation would be determining the effects that modifying the geometric parameters of a PPM architecture has on its wrench capabilities. Also, exploring the effects on the wrench capabilities of changing the assembly modes for a given architecture could be conducted. Application of the proposed indices to spatial PMs would be a further area to explore.

7. Conclusions

This work presents a method to understand and quantify wrench capabilities of PPMs. Wrench capabilities are determined by projecting the actuator torque capabilities into the wrench space. This projection is a linear transformation that leads to a convex set, i.e., a wrench polytope. To numerically evaluate wrench polytopes, six wrench performance indices are derived. Each index is associated to a particular operational condition of the manipulator. These indices are plotted throughout the workspace of the manipulator. As a design tool, the wrench workspaces allow for easy visualization of the differences in wrench capabilities between different PPM architectures. For an existing manipulator, this visual representation can be employed to improve path planning.

The wrench capability analysis is implemented to three non-redundant layouts: 3-RRR, 3-RPR, and 3-PRR. In addition, the effects of including in-branch redundancy (3-RRR, 3-RPR, and 3-PRR) and branch redundancy (4-RRR, 4-RPR, and 4-PRR) are presented. It is concluded that in-branch redundancy yields greater forces; whereas, branch redundancy offers greater moments.

8. References

- Bicchi, A., Melchiorri, C., & Balluchi, D. (1995) On the Mobility and Manipulability of General Multiple Limb Robotic Systems, *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 2, (Apr. 1995), pp. 215-228.
- Buttolo, P. & Hannaford, B. (1995) Advantages of Actuation Redundancy for the Design of Haptic Displays, *Proceedings of the 1995 ASME International Mechanical Engineering Congress and Exposition – Part 2*, pp. 623-630, San Francisco, CA, USA, Nov. 1995.
- Chand, D. R. & Kapur, S. S. (1970) An Algorithm for Convex Polytopes, *Journal of the Association for Computing Machinery*, Vol. 17, No. 1, (Jan. 1970), pp. 78-86.
- Chiacchio, P., Bouffard-Vercelli, Y., & Pierrot, F. (1996) Evaluation of Force Capabilities for Redundant Manipulators, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3520-3525, Minneapolis, MN, USA, Apr. 1996.

- Chiacchio, P., Bouffard-Vercelli, Y., & Pierrot, F. (1997) Force polytope and force ellipsoid for redundant manipulators, *Journal of Robotic Systems*, Vol. 14, No. 8, Aug. 1997, pp. 613-620.
- Ebrahimi, I., Carretero, J. A., & Boudreau, R. (2007) 3-PRRR Redundant Planar Parallel Manipulator: Inverse Displacement, Workspace and Singularity Analyses, *Mechanism and Machine Theory*, Vol. 42, No. 8, (Aug. 2007), pp. 1007-1016.
- Finotello, R., Grasso, T., Rossi, G., & Terribile, A. (1998) Computation of Kinetostatic Performances of Robot Manipulators with Polytopes, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3241-3246, Leuven, Belgium, May 1998.
- Firmani, F. & Podhorodeski, R. P. (2004) Force-Unconstrained Poses for a Redundantly-Actuated Planar Parallel Manipulator, *Mechanism and Machine Theory*, Vol. 39, No. 5, (May 2004), pp. 459-476.
- Firmani, F. & Podhorodeski, R. P. (2005), Force-Unconstrained Poses for Parallel Manipulators with Redundant Actuated Branches, *Transactions of the CSME*, Vol. 29, No. 3, (Sept. 2005), pp. 343-356.
- Firmani, F., Zibil, A., Nokleby, S. B., & Podhorodeski, R. P. (2007a) Wrench Capabilities of Planar Parallel Manipulators- Part I: Wrench Polytopes and Performance Indices, *accepted for publication in Robotica*.
- Firmani, F., Zibil, A., Nokleby, S. B., & Podhorodeski, R. P. (2007b) Wrench Capabilities of Planar Parallel Manipulators- Part II: Redundancy and Wrench Workspace Analysis, *accepted for publication in Robotica*.
- Gallina, P. Rosati, G., & Rossi, A. (2001) 3-d.o.f. Wire Driven Planar Haptic Interface, *Journal of Intelligent and Robotic Systems*, Vol. 32, No. 1, (Sept. 2001), pp. 23 - 36.
- Garg, V., Carretero, J. A., & Nokleby, S. B. (2007) Determining the Force and Moment Workspace Volumes of Redundantly-Actuated Spatial Parallel Manipulators, *Proceedings of the 2007 ASME Design Engineering Technical Conference*, ASME, Las Vegas, NV, USA, Sept. 2007.
- Hwang, Y. S., Lee, J., & Hsia, T. C. (2000) A Recursive Dimension-Growing Method for Computing Robotic Manipulability Polytope, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2569-2574, San Francisco, CA, USA, Apr. 2000.
- Kokkinis, T. & Paden, B. (1989). Kinetostatic Performance Limits of Cooperating Robot Manipulators Using Force-Velocity Polytopes, *Proceedings of the ASME Winter Annual Meeting*, pp. 151-155, San Francisco, CA, USA, Dec. 1989.
- Krut, S., Company, O., & Pierrot, F. (2004a) Velocity Performance Indices for Parallel Mechanisms with Actuation Redundancy, *Robotica*, Vol. 22, No. 2, (Mar. 2004), pp. 129-139.
- Krut, S., Company, O., & Pierrot, F. (2004b) Force Performance Indexes for Parallel Mechanisms with Actuation Redundancy, Especially for Parallel Wire-Driven Manipulators, *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 4, pp. 3936-3941, Sendai, Japan, Sept.-Oct. 2004.
- Kumar, V. & Waldron, K. J. (1988) "Force Distribution in Closed Kinematic Chains," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pp. 114-119, Philadelphia, PA, USA, Apr. 1988.

- Lee, S. & Kim, S. (1991) A Self-Reconfigurable Dual-Arm System, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 164-169, Sacramento, CA, USA, Apr. 1991.
- Lee S. & Kim, S. (1993) Kinematic Analysis of Generalized Parallel Manipulator Systems, *Proceedings of the 32nd IEEE Conference on Decision and Control*, Vol. 2, pp. 1097-1102, San Antonio, TX, USA, Dec. 1993.
- Lee, J. (1997) A Study on the Manipulability Measures for Robot Manipulators, *Proceedings of 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 1458-1465, Grenoble, France, Sept. 1997.
- Lee, J. & Shim, H. (2004) On the Dynamic Manipulability of Cooperating Multiple Arm Robot Systems, *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 2087 - 2092, Sendai, Japan, Sept.-Oct. 2004.
- Merlet, J. P. (1996) Redundant Parallel Manipulators, *Journal of Laboratory Robotics and Automation*, Vol. 8, No. 1, (Feb. 1996)., pp. 17-24
- Nahon, M. A. & Angeles, J. (1992) Real-Time Force Optimization in Parallel Kinematic Chains Under Inequality Constraints, *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 4, (Aug. 1992), pp. 439-450.
- Nokleby, S. B., Fisher, R., Podhorodeski, R. P., & Firmani, F. (2005). Force Capabilities of Redundantly-Actuated Parallel Manipulators, *Mechanism and Machine Theory*, Vol. 40, No. 5, (May 2005), pp. 578-599.
- Nokleby, S. B., Firmani, F., Zibil, A., & Podhorodeski, R. P. (2007a) Force-Moment Capabilities of Redundantly-Actuated Planar-Parallel Architectures, *Proceedings of the IFToMM 2007 World Congress*, 6 pages, Besançon, France, June 2007.
- Nokleby, S. B., Firmani, F., Zibil, A., & Podhorodeski, R. P. (2007b) An Analysis of the Force-Moment Capabilities of Branch-Redundant Planar-Parallel Manipulators, *Proceedings of the 2007 ASME Design Engineering Technical Conference*, ASME,), 8 pages, Las Vegas, NV, USA, Sept. 2007.
- Rockafellar, R. T. (1997, first printed 1970). *Convex Analysis*, Princeton University Press, Princeton, NJ, USA.
- Tao, J.M. & Luh, J. Y. S. (1989) Coordination of Two Redundant Manipulators, *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pp. 425-430, Scottsdale, AZ, USA, May 1989.
- Visvanathan, V. & Milor, L. S. (1986) An Efficient Algorithm to Determine the Image of a Parallelepiped Under a Linear Transformation, *Proceedings of 2nd Annual Symposium on Computational Geometry*, pp. 207-215, Yorktown Heights, NY, USA, June 1986.
- Wang, J., & Gosselin, C. M. (2004) Kinematic Analysis and Design of Kinematically Redundant Parallel Mechanisms, *Transactions of the ASME, Journal of Mechanical Design*, Vol. 126, No. 1, (Jan. 2004), pp. 109-118.
- Yoshikawa, T. (1985) Manipulability of Robotic Mechanisms, *International Journal of Robotics Research*, Vol. 4, No. 2, (Mar. 1985), pp. 3-9.
- Yoshikawa, T. (1990). *Foundations of Robotics: Analysis and Control*, The MIT Press, Cambridge, MA, USA.
- Zibil, A., Firmani, F., Nokleby, S. B., & Podhorodeski, R. P. (2007) An Explicit Method for Determining the Force-Moment Capabilities of Redundantly-Actuated Planar Parallel Manipulators, *Transactions of the ASME, Journal of Mechanical Design*, Vol. 129, No. 10, (Oct. 2007), pp. 1046-1055.

Robust, Fast and Accurate Solution of the Direct Position Analysis of Parallel Manipulators by Using Extra-Sensors

Rocco Vertechy and Vincenzo Parenti-Castelli
*University of Bologna
 Italy*

1. Introduction

Parallel manipulators (PMs) are closed kinematic chains with one or more loops where only some pairs are actuated while the remaining are passive. In particular, they feature a fixed link (base) and an output moving link (platform) interconnected by at least two independent kinematic chains (legs) to form one loop. The most well known and commonly employed PMs (hereafter called UPS-PMs) feature n variable-length legs of type UPS (where U, P and S are for universal, spherical and prismatic pairs respectively). Equivalently, a revolute pair R could be used instead of the prismatic pair P in order to make the leg length variable (in this case the leg would be of type URS). These leg topologies provide the platform with six degrees of freedom with respect to the base.

Although the definition of UPS-PMs requires $n \geq 2$, in practice, neglecting overconstrained and redundantly-actuated manipulators, performance issues recommend $3 \leq n \leq 6$. Indeed, UPS-PMs with only two UPS legs might exhibit a low stiffness against torques acting along the line joining the centers of the two spherical pairs, and their control would require the in-series placement of at least three actuators/sensors (one of them placed to control/measure at least one out of the three degrees of freedom of the spherical pairs) which reduces the overall manipulator dynamic and accuracy capabilities. On the other side, the use of more than six legs reduces the exploitable manipulator workspace for the increase of leg interference.

Different sub-classes of manipulator architectures can be obtained according to the location of the centers of the U and S pairs in the base and in the platform respectively (Innocenti & Parenti-Castelli, 1994; Faugere & Lazard, 1995). General UPS-PM architectures feature distinct joint centers. Special architectures can be devised by setting some of the joint centers to be coincident.

A schematic of a 6-DOF UPS-PM having six legs ($n = 6$) and general architecture is shown in Fig. 1. In the figure, the U pairs (connecting the legs to the base) and S pairs (connecting the legs to the platform) are depicted as grey and white dots respectively. Points B_i and P_i represent the centers of the U and S pairs of the i -th leg on the base and on the platform respectively. The six legs of type UPS are represented by the telescopic rods B_iP_i ($i = 1, \dots, 6$). Accordingly, the length of the i -th leg is defined as the distance $l_i = |P_i - B_i|$.

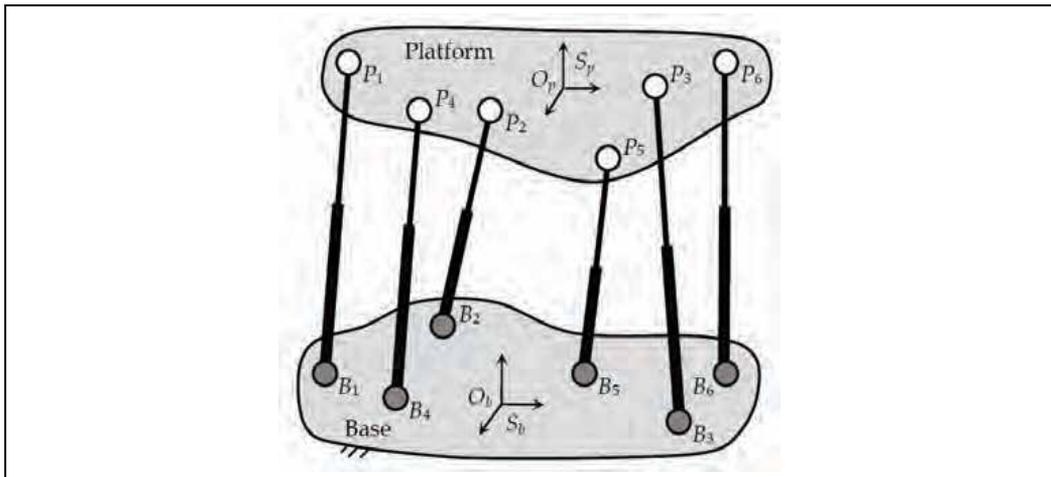


Fig. 1. Parallel manipulator with six legs of type UPS

Manipulators with less than six DOF can be obtained from UPS-PMs by suitably eliminating or locking some of the leg kinematic pairs. For instance, considering a 6-DOF UPS-PM having six legs, elimination of four P pairs yields a 2-DOF PM having two legs of type UPS and four legs of type US.

Well-known examples of UPS-PMs are as follows: 1) the 6-DOF UPS-PMs (Gough & Whitehall, 1962; Stewart, 1965; Cappel, 1967); 2) the 3-DOF spherical PMs (Innocenti & Parenti-Castelli, 1993); 3) the 2-DOF spherical PMs (Vertechy & Parenti-Castelli, 2006); and 4) the 1-DOF helicoidal PMs (Jacobsen, 1975). Because of their parallel architecture, UPS-PMs exhibit large payload-to-weight ratio, high accuracy, high structural rigidity and high dynamic capabilities, which make them excel as: a) fast and high precision robots in vehicle simulators (Gough & Whitehall, 1962; Stewart, 1965; Cappel, 1967), machine tools (Charles, 1995) and positioning systems (Schmidt-Kaler, 1992); b) passive Cartesian input devices in joysticks, master-slave teleoperation systems (Daniel et al., 1993) and other tracking devices (Geng & Haynes, 1994); c) force/torque sensors and generators in multi-axis sensors and motors (Gaillet & Reboulet, 1983; Nguyen et al., 1991; Lewis et al., 2002); d) mechanical transmissions in motion converters (Jacobsen, 1975); and e) orthopedic devices in fixations systems (Taylor & Taylor, 2000; Di Gregorio & Parenti-Castelli, 2002).

Practical use of UPS-PMs requires solving the manipulator direct position analysis (DPA) robustly, quickly and accurately. By definition, the DPA of PMs consists in finding the relative pose (position and orientation) of platform and base when the readouts of an adequate number of joint-sensors (hereafter also referred to as "input variables"), which equip some of the leg kinematic pairs, are given. Usually, this problem involves the solution of a system of kinematic constraint equations (SKCE) that are implicit and non-linear. That is, in general, the DPA of UPS-PMs is very complicated and admits multiple real solutions, each corresponding to a different mode of assembly of the manipulator. The existing methods for the solution of the DPA of UPS-PMs fall into three categories: 1) echelon-form approaches (Griffis & Duffy, 1989; Innocenti & Parenti-Castelli, 1990; Nanua et al., 1990; Merlet, 1992; Innocenti, 2001; Lee & Shim, 2001); 2) iterative approaches (McCallion & Truong, 1979; Reboulet, 1988; Innocenti & Parenti-Castelli, 1991; Merlet, 1993a; Parenti-

Castelli & Di Gregorio, 1995; McAree & Daniel, 1996); and 3) extra-sensor approaches. Both echelon-form methods and iterative methods are based on the use of a number of input variables (that is the joint-sensor number) which equals the number of manipulator DOFs. They differ, however, in the way the SKCE is solved. In particular, in echelon-form approaches, the SKCE is possibly reduced to one univariate polynomial equation, from which all the possible modes of assembly of the manipulator are determined by means of standard root finding techniques. Though of great theoretical significance, echelon-form methods are not suited for real-time applications where the fast and unambiguous identification of the actual pose of the platform is sought for. In iterative approaches, the SKCE is solved monolithically by iterative techniques, mostly based on the Newton-Raphson method. These approaches require a guess solution and aim at determining the actual pose of the platform in real-time. Unfortunately, iterative approaches require both the UPS-PM to be sufficiently far away from a singular configuration and a good initial guess of the actual pose of the platform, two conditions which cannot always be satisfied and can seriously affect the robustness of these approaches. Unlike the first two methods, extra-sensor approaches use a number of input variables which is greater than the number of manipulator DOFs. The extra-sensors are added for at least one of the following reasons: 1) to render the SKCE an explicit problem, which makes it possible to find closed form solutions of the DPA; 2) to render the SKCE a linear problem, which makes it possible to find the actual pose of the platform unambiguously; 3) to speed-up the computation of the DPA solution; 4) to make the method robust against UPS-PM special configurations (i.e. platform poses for which the DPA problem becomes undetermined); and 5) to improve the accuracy of the solution by reducing the influence of the errors affecting the joint-sensor readouts on the errors affecting the computed actual pose of the platform.

A proper choice of the number, type and location of the sensors makes it possible to devise extra-sensor methods possessing all the abovementioned features. The possibility of determining the actual configuration of the UPS-PM (i.e. the actual platform pose) unambiguously, robustly, quickly and accurately makes such extra-sensor approaches superior to the echelon-form and the iterative ones in practical real-time applications.

In this chapter, a detailed overview of the extra-sensor approaches, presented in the literature, is first provided. Then a novel very robust, fast and accurate general method based on extra-sensors is presented which makes it possible to unambiguously find the actual pose of the platform of UPS-PMs having general architecture. The method readily applies also to the DPA of both UPS-PMs with special geometry and PMs with less than six DOF that can be obtained from the 6-DOF UPS-PMs by suitably eliminating or locking some of the leg kinematic pairs. Finally, discussions are reported to highlight the advantages of the presented method.

2. Measurement of the input variables for the DPA of UPS-PMs

The manipulator DPA requires the knowledge of a number of input variables at least equal to the number of manipulator DOF. The manipulator variables which are frequently chosen as input for the solution of the DPA of UPS-PMs are presented in this section along with the possible methods for their measurement.

Considering UPS-PMs having n legs, possible choice (which practically the most used) of the input variables are the followings:

- the joint variables of the n existing legs of the manipulator;
- the distance between points of suitably chosen links.

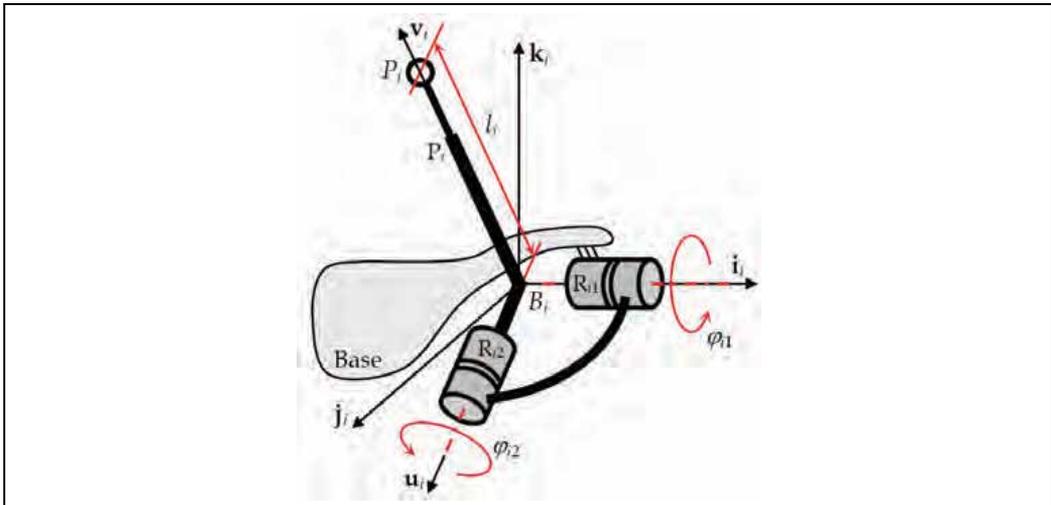


Fig. 2. Leg of type UPS

In the first case, sensors are located on the leg kinematic pairs. For instance, with reference to Fig. 2, the sensors can measure the leg joint variables, i.e. the angles φ_{i1} and φ_{i2} ($i = 1, \dots, n$) and the lengths $l_i = |P_i - B_i|$ of the U and P pairs. Conversely, the spherical pairs are normally not instrumented since, unless they are manufactured as three revolute pairs with intersecting axes, the installation of rotary sensors may be impractical. Moreover, as a matter of fact, because of their own bulk, weight, vulnerability and cabling, sensors should be placed as close as possible to the base in order to not decrease manipulator performance, ruggedness and reliability.

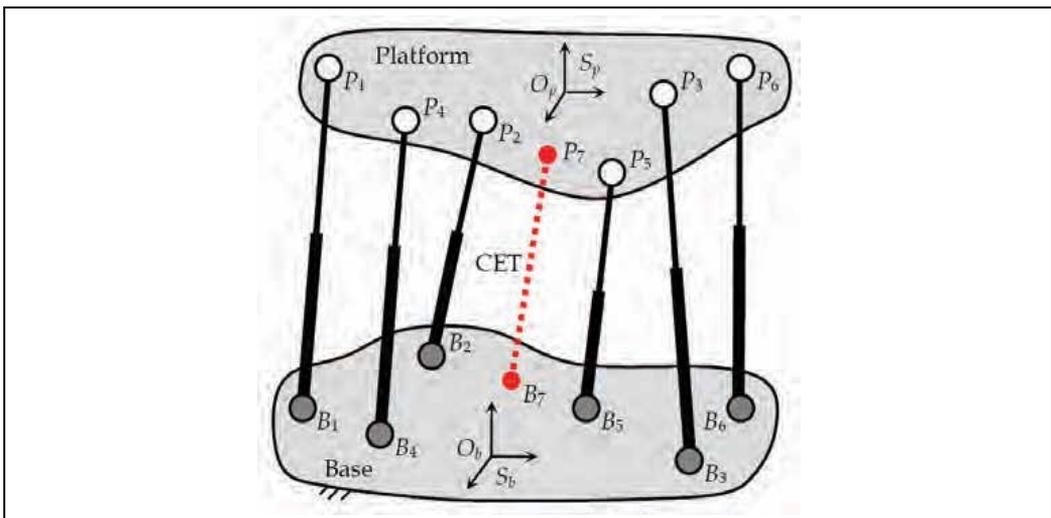


Fig. 3. Parallel manipulator with six legs of type UPS and one string pot

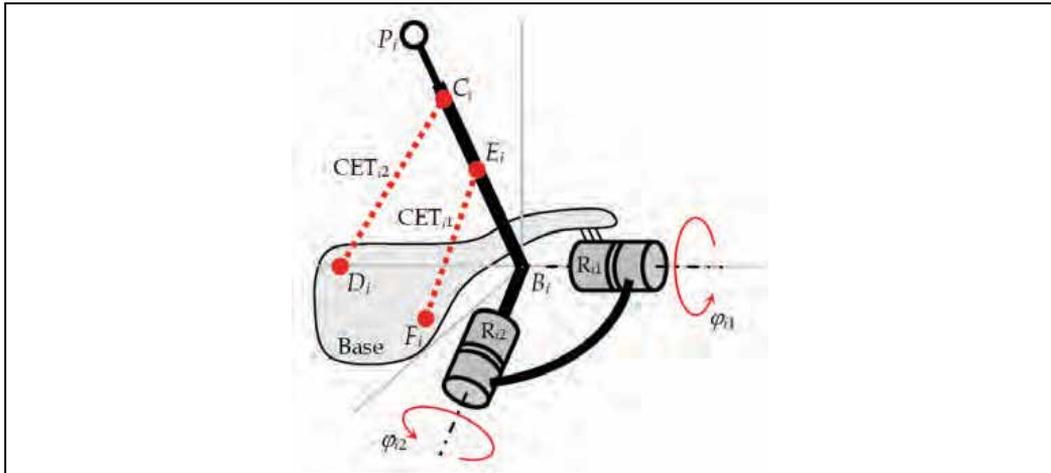


Fig. 4. Leg of type UPS instrumented with string pots

In the second case, additional external sensors are used. The most common way is to use:

- a) cable extension transducers (CET, also known as “string pots”);
- b) passive chains of type UPS with a sensor embedded in the P pair.

By means of these sensors, the distance between points of the base and the platform can be measured (see Fig. 3, points B_7 and P_7) or also the distance of points of suitably chosen links of the UPS legs can be measured, which may provide additional information on the joint leg variables. For instance, (see Fig. 4) the measure of $|C_i - D_i|$ and $|E_i - F_i|$, with D_i and F_i points of the platform, and C_i and E_i points of the second movable link of the UPS leg, indirectly provides the values of the joint angles φ_2 and φ_1 . It is worth noting, however, that the direct measuring of angles φ_1 and φ_2 by rotary sensors (placed locally on the revolute pairs) is normally preferable since it would lead to a unique position of point P_i , while the use of the lengths of the segments C_iD_i and E_iF_i would provide two positions for P_i (two symmetric positions with respect to the plane defined by points B_i , D_i and F_i).

The choice of the UPS joint variables is, in general, the most suitable. Indeed, the addition of CETs or additional UPS measurement legs can both reduce the exploitable manipulator workspace (because of increased possibility of leg interference) and slow-down the manipulator dynamic performance (due to the inertia of the additional UPS legs and to the limited mechanical response of CETs). Moreover, CET sensor accuracy is poor for many practical applications and the implementation of accurate extra UPS measurement legs is rather expensive.

An overview of extra-sensor based methods that have been proposed in the literature for the DPA of 6-DOF UPS-PMs having general architecture is presented in the following section. Of course, all these methods readily apply to the DPA of both UPS-PMs with special geometry and PMs with less than six DOF that can be obtained from the 6-DOF UPS-PMs by suitably eliminating or locking some of the leg kinematic pairs.

3. Literature overview of extra-sensor based methods for the DPA of UPS-PMs

This section provides an overview of extra-sensor based methods that are available for the solution of the DPA of 6-DOF UPS-PMs. The methods are sorted in chronological order

(according to the publication date of the author's most relevant work). For each method, the employed sensor layouts are first described, and the major features and drawbacks of the resulting DPA methods are then highlighted. To describe the sensor layout of each leg of type UPS, the sequence RRP is used to indicate the cascade of joints which are serially connected from base to platform (referring to Fig. 2, RR indicates the two revolute pairs with intersecting axes the U pair is featured by; the spherical pair S is ignored since it is not supposed to be instrumented) and the underline is used to highlight the joint whose position is measured. For instance, the leg sensor layout \underline{RRP} indicates that 1 rotary position sensor and 1 linear position sensor are installed on the leg. The sensor layout of a given manipulator is described by a list (set) of sensor layouts of the legs belonging to the manipulator. That is, the set $\{2-\underline{RRP}, \underline{RRP}, 4-\underline{RRP}\}$ indicates that 8 sensors are mounted on the manipulator; in particular, it features 2 legs each having 1 rotary position sensor, 1 leg having 1 rotary position sensor and 1 linear position sensor, and 4 legs each having 1 linear position sensor.

The first DPA solution of UPS-PMs via extra-sensors was firstly proposed in 1991, when, following the studies on the pose and twist estimation from three collinear measured points (Fenton & Shi, 1989), Shi and Fenton (Shi & Fenton, 1991) employed the set $\{3-\underline{RRP}\}$ to devise a method that reduces the DPA of UPS-PMs having general base and platform to an explicit problem which can be readily solved in real-time. Irrespective of the manipulator configuration, the method always makes it possible to find the actual platform pose. However, the method does not account for the measurement errors, which in practice always affect the sensor readouts. As a matter of fact, the proposed method is rather inaccurate when measurement errors are present.

Several sensor layouts are studied in (Stoughton & Arai, 1991) in order to devise fast and accurate methods for the solution of the DPA of the UPS-PM with general base and platform. Note that results similar to those presented by Stoughton and Arai have also been reported lately in (Hesselbach et al., 2005). In particular: 1) using the set $\{3-\underline{RRP}\}$ the DPA is reduced to an explicit problem readily yielding the actual manipulator configuration; 2) using both the set $\{2-\underline{RRP}, \underline{RRP}\}$ and the set $\{2-\underline{RRP}, \underline{RRP}\}$ the DPA is reduced to the solution of a system of 2 uni-variate quadratic equations in the same unknown usually yielding the actual manipulator configuration; 3) using both the set $\{2-\underline{RRP}, \underline{RRP}\}$ and the set $\{2-\underline{RRP}, \underline{RRP}\}$ the DPA is reduced to the solution of a system of 2 quadratic and 1 linear 3-variate equations in the same 3 unknowns usually yielding 2 possible manipulator configurations from which the actual platform pose cannot be detected; 4) using one of the sets $\{\underline{RRP}, 2-\underline{RRP}\}$, $\{\underline{RRP}, 2-\underline{RRP}\}$ or $\{\underline{RRP}, \underline{RRP}, \underline{RRP}\}$, the DPA is reduced to the solution of 2 uni-variate quadratic equations in 2 different unknowns usually yielding four possible manipulator configurations (although it is not stated in the paper, the actual manipulator configuration may be detected among those 4 possibilities by checking the satisfaction of a further constraint equation); and 5) using the set $\{\underline{RRP}, \underline{RRP}, \underline{RRP}\}$ the DPA is reduced to the sequential solution of a system of 2 quadratic and 1 linear 3-variate equations in the same 3 unknowns, and of a uni-variate quadratic equation in a different unknown usually yielding 4 possible manipulator configurations among which the actual platform pose cannot be detected. All the aforementioned solutions can be computed in real-time. Only the method based on the set $\{3-\underline{RRP}\}$ guarantees that the actual manipulator configuration can always be calculated (manipulator configurations may exist for which the methods based on the other sensor layouts cannot find a unique DPA solution). The paper also addresses accuracy

issues. In particular, the ratios between the magnitudes of the errors affecting the computed manipulator configuration and the measurement errors affecting the joint-sensor readouts are determined for all the abovementioned sensor layouts. This makes it possible to select the required sensor precision which provides the desired accuracy of the calculated platform pose. Moreover, it is shown that the solution of the DPA based on the set $\{3\text{-RRP}\}$ is less sensitive to the measurement errors affecting the joint-sensors than the solution which can readily be computed if the measurement of the 6 joints parameters of one single leg are available (in this latter case the leg sensor layout would be RRP plus 3 additional rotary position sensors measuring the rotations allowed by the S pair of the same leg).

Two sensor layouts are proposed in (Cheok et al, 1992) to devise methods that make it possible to find the actual solution of the DPA of the UPS-PM with general base and platform in real-time. In particular: 1) using the set $\{3\text{-RRP}\}$ the DPA is reduced to an explicit problem readily yielding the actual manipulator configuration; and 2) using the set $\{6\text{-RRP}, \text{RRP}\}$ the DPA is reduced to the solution of a system of 6 linear 6-variate equations in the same 6 unknowns usually yielding the actual manipulator configuration. Only the method based on the set $\{3\text{-RRP}\}$ guarantees that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the 6 linear equations to be solved in method (2) are not linearly independent. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

Two sensor layouts are proposed in (Merlet, 1993b) to devise methods that make it possible to find the solution of the DPA of UPS-PMs in real-time. In particular: 1) the set $\{2\text{-RRP}, 2\text{-RRP}\}$ is used to reduce the DPA of the UPS-PM with general base and platform to the solution of a system of 2 uni-variate quadratic equations in the same unknown usually yielding the actual manipulator configuration; 2) the set $\{\text{RRP}, \text{RRP}, 2\text{-RRP}\}$ is used to reduce the DPA of the UPS-PM with general base and platform to the sequential solution of a system of 2 uni-variate quadratic equations in the same unknown and of a uni-variate quadratic equation in a further different unknown usually yielding 2 possible manipulator configurations from which the actual platform pose cannot be detected; and 3) the set $\{\text{RRP}, 6\text{-RRP}\}$ is used to reduce the DPA of the UPS-PM with planar base and platform to the solution of a system of 9 9-variate linear equations in the same 9 unknowns usually yielding the actual manipulator configuration. Note that the proposed methods do not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which either the two pairs of solutions of the two quadratic equations to be solved in method (1) are identical or the 9 linear equations to be solved in method (3) are not linearly independent. Accuracy issues related to the $\{2\text{-RRP}, 4\text{-RRP}\}$ sensor layout are addressed in a later paper (Tancredi and Merlet, 1994) in which the pose dependent ratios between the magnitudes of the errors affecting the computed manipulator configuration and the errors affecting the joint-sensor readouts are evaluated and mapped.

Two sensor layouts are proposed in (Nair & Maddocks, 1994) to devise methods that make it possible to reduce the solution of the DPA of UPS-PMs to an explicit problem which can be solved in real-time. In particular: 1) the set $\{16\text{-RRP}\}$ is used to reduce the DPA of manipulators with general base and platform to the solution of a system of 16 16-variate linear equations in the same 16 unknowns usually yielding the actual manipulator configuration; and 2) the set $\{9\text{-RRP}\}$ is used to reduce the DPA of manipulators with planar base/platform to the solution of a system of 9 9-variate linear equations in the same 9

unknowns usually yielding the actual manipulator configuration. None of the proposed methods guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which either the 16 linear equations to be solved in method (1) or the 9 linear equations to be solved in method (2) are not linearly independent. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A method is proposed in (Jin, 1994), which uses the set {4-RRP, 2-RRP}, to reduce the DPA of the UPS-PM with planar base and platform to the sequential solution of a system of 2 linear 2-variate equations in the same 2 unknowns and of a system of 5 5-variate linear equations in a further 5 unknowns. The problem can be solved in real-time and usually admits one solution corresponding to the actual manipulator configuration. The proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which either the 2 equations belonging to the first system to be solved or the 5 equations belonging to the second system to be solved are not linearly independent. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A study for the determination of the maximum number of possible DPA solutions for UPS-PMs having different sensor layouts was accomplished in (Tancredi et al., 1995). It turned out that: 1) the DPA of UPS-PMs with general base and platform admits up to 35 possible solutions when the set {5-RRP, RRP} is used; 2) the DPA of UPS-PMs with general base and platform admits up to 8 possible solutions when the set {3-RRP, 3-RRP} is used; 3) the DPA of UPS-PMs with planar base and platform admits up to 6 possible solutions when the set {RRP, 5-RRP} is used; 4) the DPA of UPS-PMs with planar base and platform admits up to 4 possible solutions when the set {6-RRP} is used; and 5) the DPA of UPS-PMs with general base and platform admits up to 8 possible solutions (however, only two solutions are more likely) when the set {5-RRP, RRP} is used.

A method is proposed in (Etemadi-Zanganeh & Angeles, 1995), which uses the set {5-RRP, RRP}, to reduce the DPA of the UPS-PM with general base and platform to the solution of 5 eigenproblems of 6×6 matrices usually admitting a unique solution which can be computed in real-time. Note that the proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the condition number of the aforementioned 6×6 matrices is close to infinity (i.e. it is very large). The paper addresses accuracy issues too. Using the redundant information provided by the extra-sensors, the proposed method is able to reduce the influence of the errors affecting joint-sensor readouts on the errors affecting the computed manipulator configuration.

A method is proposed in (Han et al., 1996), which uses the set {5-RRP, RRP}, to reduce the DPA of the UPS-PM with planar base and platform to the solution of a system of 5 linear 6-variate equations and one quadratic 3-variate equation in the same unknowns. The problem can be solved in real-time and admits 2 possible solutions, among which the actual manipulator configuration can usually be determined by (a-posteriori) checking the satisfaction of a further two quadratic constraint equations. Note that the proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the two possible solutions of

the system of equations both satisfy the additional constraint equations. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A method is proposed in (Jin & Hai-Rong, 1996), which uses the set $\{5\text{-RRP}, \underline{\text{RRP}}\}$, to reduce the DPA of the UPS-PM with planar base and platform to the sequential solution of two systems of equations, the first one of 20 linear 20-variate equations in the same 20 unknowns and the second one of 3 3-variate linear equations in another 3 different unknowns, and then to the solution of a quadratic equation in a further unknown. The problem can be solved in real-time and usually admits two solutions (that are symmetric with respect to the planar manipulator base) one of which corresponds to the actual manipulator configuration. Note that the proposed method does not guarantee that the two aforementioned solutions (and, thus, the actual manipulator configuration) can always be calculated. Indeed, special manipulator configurations may exist for which the 20 equations belonging to the first system to be solved are not linearly independent. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A method based on either the set $\{7\text{-RRP}\}$ or the set $\{5\text{-RRP}, \underline{\text{RRP}}\}$ is proposed in (Innocenti, 1998), which reduces the DPA of the UPS-PM with general base and platform to the solution of a system of 146 146-variate linear equations in the same 146 unknowns usually yielding the actual manipulator configuration. Note that the proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the 146 equations to be solved are not linearly independent. Due to the large number of equations, the solution of the system of equations requires a rather large computational burden. However, since the system of 146 equations has a sparse coefficient matrix, rather efficient sparse solvers may be used to find the solution in real-time. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

Two sensor layouts are used in (Parenti Castelli & Di Gregorio, 1998) to devise methods which make it possible to reduce the DPA of UPS-PMs to an explicit problem that can be solved in real-time. In particular: 1) the set $\{4\text{-RRP}, \underline{\text{RRP}}\}$ is used to reduce the DPA of manipulators with general base and platform to the solution of a system of 15 15-variate linear equations in the same 15 unknowns usually yielding the actual manipulator configuration; and 2) the set $\{5\text{-RRP}, \underline{\text{RRP}}\}$ is used to reduce the DPA of manipulators with general base and platform to the solution of a system of two 6-degree polynomial univariate equations in the same unknown usually yielding the actual manipulator configuration. Note that the proposed methods do not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which either the 15 equations to be solved in method (1) are not linearly independent or the two 6-degree polynomials involved in method (2) have more than one common root. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A method based on the set $\{5\text{-RRP}, \underline{\text{RRP}}\}$ is used in (Parenti Castelli & Di Gregorio, 1999) to reduce the DPA of UPS-PMs with general base and platform to the solution of two 48-degree uni-variate polynomial equations in the same unknown usually having a unique common root, corresponding to the actual manipulator configuration. Note that the

proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the two 48-degree polynomials have more than one common root. The solution of the reduced problem requires a large computational burden and, thus, cannot be computed in real-time. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

A method based on the set {9-RRP} is used in (Bonev & Ryu, 1999) to reduce the DPA of UPS-PMs with general base and planar platform to the solution of two sets of three quadratic 3-variate equations in the same 3 unknowns usually having a unique common solution, corresponding to the actual manipulator configuration. The proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which the two sets of quadratic equations have more than one common solution. The calculations involved in the determination of manipulator configuration require a large computational burden and, thus, cannot be computed in real time. The paper addresses accuracy issues. In particular it is shown that the errors in the calculated platform pose are of the same magnitude of the measurement errors affecting the sensor readouts.

A method based on the set {4-RRP, RRP} is proposed in (Parenti Castelli & Di Gregorio, 2000) to reduce the DPA of manipulators with general base and platform to the sequential solution of a 6-degree uni-variate polynomial equation and of a system of two linear bi-variate equations in two further unknowns. The problem can be solved in real-time and admits up to six possible solutions, among which the actual manipulator configuration can usually be determined by (a-posteriori) checking the satisfaction of a further additional quadratic constraint equation. Note that the proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which more than one solution (among the abovementioned six possible solutions) satisfy the additional quadratic constraint equation. The paper does not address accuracy issues. As a matter of fact, the proposed method is rather inaccurate when the joint-sensors are affected by measurement errors.

As a result of several investigations (Angeles, 1990; Baron & Angeles, 1994; Baron & Angeles, 1995) a very general method based on at least nine measurements, obtained from the sensors placed on n legs according to the following sensor layouts RRP, RRP and RRP, is proposed in (Baron & Angeles, 2000a; Baron & Angeles, 2000b) which reduces the DPA of UPS-PMs with general base and platform to the evaluation of the orthogonal polar factor of a 3×3 matrix whose components are obtained from the least-square-solution of a system of $3n$ 9-variate linear equations in the same nine unknowns. The reduced problem can be solved in real-time and usually admits a unique solution, corresponding to the actual manipulator configuration. However, in general, the uniqueness of the solution is not guaranteed. Indeed, special manipulator configurations may exist for which 9 linearly independent equations cannot be found among the $3n$ equations cited above. The method accounts for the measurement errors, which always affect the joint-sensor readouts. In particular, the redundant information provided by the extra-sensors is also used to reduce the influence of the measurement errors on the errors affecting the computed platform pose (that is, the computed manipulator configuration is the solution which most closely satisfies all the aforementioned $3n$ equations). Among all the possible sets of leg sensor layouts, the sets { n -RRP} (for $n \geq 3$) are shown to be very effective since they guarantee that both a unique (the actual) DPA solution can always be found and the matrix from which to extract

the orthogonal polar factor is simply obtained by the matrix multiplication of two matrices having dimensions $3 \times n$ and $n \times 3$. In practice, the set {3-RRP} is very interesting since it provides a very fast and accurate unique solution of the DPA by using the minimum number of sensors (among the sensor layouts this method is based on). As compared to other methods (Shi & Fenton, 1991; Stoughton & Arai, 1991; Cheok et al, 1992) using the set {3-RRP}, the method proposed by Baron and Angeles is the most accurate and only slightly more expensive in terms of computational cost.

A method based on the set {9-RRP} is proposed in (Bonev et al., 2001) to reduce the DPA of the UPS-PM with planar base and platform to the solution of a system of six linear 6-variate equations in the same 6 unknowns usually admitting a unique solution, corresponding to the actual manipulator configuration, which can be computed in real time. Note that the proposed method does not guarantee that the actual manipulator configuration can always be found. Indeed, special manipulator configurations may exist for which the 6 equations to be solved are not linearly independent. The paper addresses accuracy issues too. In particular a procedure is proposed for the determination of the optimal extra-sensor location, which makes it possible to minimize (throughout the desired manipulator workspace) the ratio between the magnitudes of the errors affecting the computed manipulator configuration and of the errors affecting the joint-sensor readouts.

A method based on the set {6-RRP, RRP} is proposed in (Chiu & Perng, 2001) to reduce the DPA of the UPS-PM with general base and platform to the solution of two quadratic univariate equations in two different unknowns. The problem can be solved in real-time and admits four possible solutions, among which the actual manipulator configuration can usually be determined by (a-posteriori) checking the satisfaction of a further three quadratic constraint equations. The proposed method does not guarantee that the actual manipulator configuration can always be calculated. Indeed, special manipulator configurations may exist for which more than one solution (among the four possible solutions cited above) satisfies the three additional quadratic constraint equations. The paper addresses accuracy issues too. In particular a procedure is proposed for the determination of the optimal extra-sensor location, which makes it possible to minimize (throughout the desired manipulator workspace) the ratio between the magnitudes of the errors affecting the computed manipulator configuration and of the errors affecting the joint-sensor readouts.

Focusing on the popular measurement set {3-RRP}, which is the only one guaranteeing that a unique DPA solution can always be found irrespective of the manipulator configuration, and accounting for the measurement errors, which always affect the sensor readouts, a method is proposed in (Vertechy & Parenti Caselli, 2007; Vertechy et al., 2002) which, following an approach similar to that of Baron and Angeles (Baron & Angeles, 2000a; Baron & Angeles, 2000b), reduces the DPA of the UPS-PM with general base and platform to the solution of one simple trigonometric equation in a single unknown. The method always provides the actual platform pose in real-time, it is insensitive to singular configurations, it has the same accuracy as the method by Baron and Angeles (Baron & Angeles, 2000a; Baron & Angeles, 2000b) but it requires a reduced computational burden (it is three times more efficient).

4. A robust, fast and accurate novel method for the DPA of UPS-PMs by using extra-sensors

In this section, a novel extra-sensor-based method for the solution of the DPA of 6-DOF UPS-PMs having general geometry is presented (the method readily applies also to the DPA of both UPS-PMs with special geometry and PMs with less than six DOF). The method is

based on the sensor layout $\{n\text{-RRP}\}$ ($n \geq 3$) and is: robust since it always provide the actual platform pose; fast since the calculation of the actual platform pose can be performed in real-time; and accurate since the redundant information provided by the extra-sensors is used to reduce the influence of the measurement errors on the errors affecting the computed platform pose. The method is based on the DPA algorithms developed in (Baron & Angeles, 2000a; Baron & Angeles, 2000b) but it improves both the accuracy and the computational efficiency.

In the following, in sub-section 4.1 the fundamentals of the method are introduced. In sub-section 4.2 a general method is presented which makes it possible to solve the DPA of UPS-PMs having general architecture, general sensor layout and noisy sensors, but which cannot guarantee the uniqueness of the DPA solution. In section 4.3 the novel method is presented. Finally, in sub-section 4.4 results are reported which show that the novel method is more accurate and computationally more efficient than other methods available in the literature.

4.1 Fundamentals of the method: general sensor layout without measurement errors

For a UPS-PM two reference frames S_b , centered at O_b , and S_p , centered at O_p , are attached to the manipulator base and platform respectively. With reference to Fig. 1, the platform pose is described by the vector $c = (O_p - O_b)$, which gives the origin of S_p with respect to S_b , and by the proper orthogonal matrix \mathbf{R} (i.e. $\det(\mathbf{R}) = +1$, $\mathbf{R}^T \mathbf{R} = \mathbf{1}$ where $\mathbf{1}$ is the 3×3 identity matrix) which describes the orientation of S_p with respect to S_b . In some applications, \mathbf{R} is defined equivalently as $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]^T$, where the \mathbf{r}_i 's ($i = 1, \dots, 3$) are the 3×1 orthonormal vectors (i.e. $\mathbf{r}_i \cdot \mathbf{r}_j = 0$ if $i \neq j$ and $\mathbf{r}_i \cdot \mathbf{r}_i = 1$ if $i = j$) indicating the components of the unit vectors of the frame S_b in the frame S_p . With reference to Fig. 2, consider the leg variables φ_{1i} , φ_{2i} and l_i which define the position of points P_i with respect to S_b (without losing in generality, in the following it is assumed that the leg geometry is such that the leg unit vector \mathbf{v}_i , $\mathbf{v}_i = B_i P_i / |B_i P_i|$, is orthogonal to the axis \mathbf{u}_i of the revolute pair R_{2i} and that the unit vector \mathbf{u}_i is orthogonal to the axis \mathbf{i}_i of the revolute pair R_{1i} ; thus, φ_{1i} indicates the angle between axes \mathbf{u}_i and \mathbf{j}_i , φ_{2i} indicates the angle between the vector $P_i B_i$ and the axis \mathbf{i}_i , and l_i indicates the distance between points P_i and B_i). By definition, the DPA of 6-DOF UPS-PMs having n legs consists in finding c and \mathbf{R} once the magnitude of at least 6 leg variables (among the $3n$ possible variables φ_{1i} , φ_{2i} and l_i , for $i = 1, \dots, n$) are known by measurement. In practice, c and \mathbf{R} are found as the solution of a system of kinematic constraint equations (SKCE) of the type

$$f_i(c, \mathbf{R}; \varphi_{1i}, \varphi_{2i}, l_i) = 0, \quad i = 1, \dots, n. \quad (1)$$

For the class of manipulators under study, the kinematic constraint equations (1) can be derived by considering the analytical expressions of vectors $B_i P_i$ ($i = 1, \dots, n$). Indeed, by referring to Fig. 1, the position vector $\mathbf{q}_i = (P_i - B_i)_b$ expressed in S_b can be written as

$$\mathbf{q}_i = c + \mathbf{R} \mathbf{p}_i - \mathbf{b}_i, \quad (2)$$

where $\mathbf{p}_i = (P_i - C)_p$ and $\mathbf{b}_i = (B_i - O)_b$ are known (at the outset) position vectors expressed in S_p and S_b respectively. Besides, with reference to Fig. 2, the position vector \mathbf{q}_i can also be written as

$$\mathbf{q}_i = l_i \mathbf{v}_i, \quad (3.1)$$

$$\mathbf{v}_i = \mathbf{i}_i \cos \varphi_{i2} + \mathbf{u}_i \times \mathbf{i}_i \sin \varphi_{i2} , \quad (3.2)$$

$$\mathbf{u}_i = \mathbf{j}_i \cos \varphi_{i1} - \mathbf{k}_i \sin \varphi_{i1} , \quad (3.3)$$

where, of course, in Eqs. (3) vectors \mathbf{i}_i , \mathbf{j}_i , \mathbf{k}_i , \mathbf{u}_i and \mathbf{v}_i are assumed to be expressed in S_b . Starting from Eqs. (2) and (3), different sets of rather simple linear kinematic constraint equations (KCE) can be derived for each of the sensor layouts RRP, RRP and RRP. Indeed, if the i -th leg is equipped with one sensor according to the layout RRP, then the angle φ_{i1} (and the vector \mathbf{u}_i) are fully known. Therefore, from equations (2), (3.1) and (3.2) the following KCE can be written:

$$\mathbf{u}_i \mathbf{u}_i^T (\mathbf{c} + \mathbf{R} \mathbf{p}_i - \mathbf{b}_i) = \mathbf{0} , \quad (4)$$

which indicates that the distance of the platform point P_i from the plane passing through B_i and having the measured vector \mathbf{u}_i as normal (i.e. the plane defined by \mathbf{i}_i and \mathbf{v}_i) is zero. Note that Eq. (4) consists of three equations among which only one is independent of the others. If the leg is equipped with two sensors according to the layout RRP, then the angles φ_{i1} and φ_{i2} (and the vector \mathbf{v}_i) are fully known. Therefore, from equations (2) and (3.1) the following KCE can be written:

$$(\mathbf{1} - \mathbf{v}_i \mathbf{v}_i^T) (\mathbf{c} + \mathbf{R} \mathbf{p}_i - \mathbf{b}_i) = \mathbf{0} , \quad (5)$$

which indicates that the distance of the platform point P_i from the line passing through B_i and directed along the measured vector \mathbf{v}_i is zero. Note that Eq. (5) consists of three equations among which only two are independent of the others. If the leg is equipped with three sensors according to the layout RRP, then the angles φ_{i1} and φ_{i2} , and the length l_i (and the vector \mathbf{q}_i) are fully known. Therefore, from equations (2) and (3.1) the following KCE can be written:

$$(\mathbf{c} + \mathbf{R} \mathbf{p}_i - \mathbf{b}_i) - l_i \mathbf{v}_i = \mathbf{0} , \quad (6)$$

which indicates that the distance of the platform point P_i from the corresponding measured point lying on the leg is zero. Note that Eq. (6) consists of three independent equations. Equations (4)-(6) are of the type described by Eq. (1). Considering all the instrumented legs of the manipulator and by resorting to a unified formulation, the SKCE of Eq. (1) can be written as

$$\mathbf{W}_i (\mathbf{c} + \mathbf{R} \mathbf{p}_i - \mathbf{b}_i) - \delta_i \mathbf{v}_i = \mathbf{0} , i=1, \dots, n \quad (7)$$

where $\mathbf{W}_i = \mathbf{u}_i \mathbf{u}_i^T$ and $\delta_i = 0$, $\mathbf{W}_i = \mathbf{1} - \mathbf{v}_i \mathbf{v}_i^T$ and $\delta_i = 0$, or $\mathbf{W}_i = \mathbf{1}$ and $\delta_i = l_i$ if the i -th leg is instrumented according to the sensor layout RRP, RRP or RRP respectively. The SKCE of Eq. (7) consists of $3n$ equations. If the manipulator is equipped with at least nine sensors, then nine linearly independent equations can usually be extracted from Eq. (7) to find the actual manipulator configuration. Indeed, such nine equations can be used to determine the three components of \mathbf{c} and six of the nine components of \mathbf{R} (for instance the components of the orthonormal vectors \mathbf{r}_1 and \mathbf{r}_2); the remaining three components of \mathbf{R} (the components of the orthonormal vector \mathbf{r}_3) can be determined afterwards by using a further three linear

equations coming from the proper orthogonality conditions (the three equations $\mathbf{r}_1 \cdot \mathbf{r}_3 = 0$, $\mathbf{r}_2 \cdot \mathbf{r}_3 = 0$ and $\det(\mathbf{R}) = +1$). Among all the possible sensor layouts, the sets $\{n\text{-RRP}\}$ ($n \geq 3$) guarantee that a unique DPA solution can always be found. For other sensor layouts, manipulator configurations may exist for which the set of measurement data is singular and, thus, nine linearly independent equations cannot be extracted from Eq. (7).

4.2 The general method: general sensor layout with measurement errors

The equalities described by Eq. (7) hold in ideal situations only. Indeed, whenever finite precision arithmetic is used to perform the required calculation and whenever joint-sensor readouts are affected by measurement errors, the following relations

$$\mathbf{W}_i (\mathbf{c} + \mathbf{R}\mathbf{p}_i - \mathbf{b}_i) - \delta_i \mathbf{v}_i = \mathbf{e}_i, \quad i=1, \dots, n, \quad (8)$$

hold instead of Eqs. (7), where the \mathbf{e}_i 's are error vectors whose magnitude should be as small as possible. In such real situations, the DPA can be recast to the solution of the following constrained least-squares (CLS) problem

$$\min_{\mathbf{c}, \mathbf{R}} \sum_{i=1}^n [\mathbf{W}_i (\mathbf{c} + \mathbf{R}\mathbf{p}_i - \mathbf{b}_i) - \delta_i \mathbf{v}_i]^2, \quad (9)$$

subject to $\mathbf{R}^T \mathbf{R} = \mathbf{1}$ and $\det(\mathbf{R}) = +1$.

By observing the quadratic nature of the function to be minimized, the solution of Eq. (9) is reduced to first solving the following CLS problem in \mathbf{R} only

$$\min_{\mathbf{R}} \sum_{i=1}^n \left[\mathbf{W}_i \left[\left(\mathbf{R}\mathbf{p}_i - \underline{\mathbf{W}}^{-1} \sum_{j=1}^n \mathbf{W}_j \mathbf{R}\mathbf{p}_j \right) - \mathbf{b}'_i \right] - \mathbf{v}'_i \right]^2, \quad (10.1)$$

subject to $\mathbf{R}^T \mathbf{R} = \mathbf{1}$ and $\det(\mathbf{R}) = +1$,

and then to computing \mathbf{c} as

$$\mathbf{c} = -\underline{\mathbf{W}}^{-1} \sum_{j=1}^n [(\mathbf{W}_j \mathbf{R}\mathbf{p}_j - \mathbf{W}_j \mathbf{b}_j) - \delta_j \mathbf{v}_j], \quad (10.2)$$

where the 3×3 matrix $\underline{\mathbf{W}}$, and the 3×1 vectors \mathbf{b}'_i and \mathbf{v}'_i are

$$\underline{\mathbf{W}} = \sum_{j=1}^n \mathbf{W}_j, \quad (10.3)$$

$$\mathbf{b}'_i = \mathbf{b}_i - \underline{\mathbf{W}}^{-1} \sum_{j=1}^n \mathbf{W}_j \mathbf{b}_j, \quad (10.4)$$

$$\mathbf{v}'_i = \delta_i \mathbf{v}_i - \underline{\mathbf{W}}^{-1} \sum_{j=1}^n \delta_j \mathbf{v}_j, \quad (10.5)$$

and depend on the given manipulator geometry and on the measured joint variables. In general, the closed-form solution of the CLS problem described by Eq. (10.1) is difficult to compute. In practice, an acceptable minimizer \mathbf{R} of Eq. (10.1) can be obtained by evaluating the orthogonal polar factor (OPF) of the solution of the corresponding unconstrained least-square (ULS) problem, which is given in the following

$$\min_{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3} \left\| \mathbf{P}_W \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} - \mathbf{b}_W - \mathbf{v}_W \right\|, \quad (11.1)$$

$$\mathbf{b}_W = \begin{bmatrix} \mathbf{W}_1 \mathbf{b}'_1 \\ \dots \\ \mathbf{W}_n \mathbf{b}'_n \end{bmatrix}, \quad (11.2)$$

$$\mathbf{v}_W = \begin{bmatrix} \mathbf{v}'_1 \\ \dots \\ \mathbf{v}'_n \end{bmatrix}, \quad (11.3)$$

$$\mathbf{P}_W = \begin{bmatrix} \mathbf{W}_1 \left(\mathbf{P}_1 - \mathbf{W}^{-1} \sum_{j=1}^n \mathbf{W}_j \mathbf{P}_j \right) \\ \dots \\ \mathbf{W}_n \left(\mathbf{P}_n - \mathbf{W}^{-1} \sum_{j=1}^n \mathbf{W}_j \mathbf{P}_j \right) \end{bmatrix}, \quad (11.4)$$

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{p}_i^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{p}_i^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{p}_i^T \end{bmatrix}, \quad (11.5)$$

where \mathbf{P}_W is a $3n \times 9$ matrix, \mathbf{P}_i ($i=1, \dots, n$) is a 3×9 matrix, and \mathbf{b}_W and \mathbf{v}_W are $3n \times 1$ vectors. Hence, an acceptable minimizer of Eq. (10.1) is

$$\mathbf{R} = \text{OPF}(\hat{\mathbf{R}}), \quad (12.1)$$

$$\hat{\mathbf{R}} = [\hat{\mathbf{r}}_1 \quad \hat{\mathbf{r}}_2 \quad \hat{\mathbf{r}}_3]^T, \quad (12.2)$$

$$\begin{bmatrix} \hat{\mathbf{r}}_1 \\ \hat{\mathbf{r}}_2 \\ \hat{\mathbf{r}}_3 \end{bmatrix} = \left(\mathbf{P}_W^T \mathbf{P}_W \right)^{-1} \mathbf{P}_W^T (\mathbf{b}_W + \mathbf{v}_W), \quad (12.3)$$

where the vectors $\hat{\mathbf{r}}_1$, $\hat{\mathbf{r}}_2$ and $\hat{\mathbf{r}}_3$ are estimates of the orthonormal vectors \mathbf{r}_1 , \mathbf{r}_2 and \mathbf{r}_3 . Regarding the meaning of the orthogonal polar factor, note that given a 3×3 matrix \mathbf{A} whose polar decomposition is $\mathbf{A} = \mathbf{Q}\mathbf{M}$, where \mathbf{Q} is an orthogonal 3×3 matrix and \mathbf{M} is a symmetric and positive definite 3×3 matrix, then $\text{OPF}(\mathbf{A}) = \mathbf{Q}$. Providing that matrix $\mathbf{P}_W^T \mathbf{P}_W$ is well conditioned (i.e. if $\text{rank}(\mathbf{P}_W) = 9$), then Eqs. (12) admit a unique solution corresponding to the actual orientation of the manipulator platform.

4.2.1 Uniqueness of the solution and computational issues

According to Eqs. (12), the actual platform orientation can be found if $\text{rank}(\mathbf{P}_W) = 9$. In order for \mathbf{P}_W to have full rank, a minimum of nine leg variables need to be measured. However, this may not be sufficient. Indeed, due to matrices \mathbf{W}_i and \mathbf{P}_i ($i = 1, \dots, 6$), matrix \mathbf{P}_W is dependent on the given manipulator geometry and on the configuration (which is known by measurements). As a matter of fact, special manipulator configurations may exist for which $\text{rank}(\mathbf{P}_W) < 9$. In practice, for given manipulator geometry and for selected sensor layout, a-priori study of the rank of \mathbf{P}_W is required in order to prevent the method to fail. In cases where the drop of rank (which may be caused not only by special configurations and a special manipulator geometry, but also by the availability of less than nine joint-sensor measurements) is not too drastic, a number of remedies that rely on the mutual dependency of the components of \mathbf{R} exist, which make it possible to find the actual manipulator orientation. A first trick (trick 1) consists in circumventing the rank deficiency by solving Eqs. (11) for a reduced number of unknowns only (whose number cannot be greater than the rank of \mathbf{P}_W) and by calculating the remaining ones via the proper orthogonality conditions. As an example, note that the solution of Eqs. (11) for the components of $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$ only, and the a-posteriori evaluation of the components of $\hat{\mathbf{r}}_3$ via the three linear equations $\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{r}}_3 = 0$, $\hat{\mathbf{r}}_2 \cdot \hat{\mathbf{r}}_3 = 0$ and $\det(\hat{\mathbf{R}}) = +1$, requires $\text{rank}(\mathbf{P}_W) \geq 6$ only. A second trick (trick 2) consists in restoring the rank of \mathbf{P}_W by considering, in addition to the points P_i ($i = 1, \dots, n$) of the instrumented legs, additional virtual points P_k ($k > n$) depending on the P_i 's themselves such that $\mathbf{p}_k = \mathbf{p}_i \times \mathbf{p}_j$ and $(\mathbf{b}'_k + \mathbf{v}'_k) = (\mathbf{b}'_i + \mathbf{v}'_i) \times (\mathbf{b}'_j + \mathbf{v}'_j)$, ($i \neq j$; for $i, j = 1, \dots, n$). As an example note that whenever the third components of the vectors \mathbf{p}'_i are zero for all points P_i ($i = 1, \dots, n$), then $\text{rank}(\mathbf{P}_W) \leq 6$. In this case, the rank of \mathbf{P}_W can be restored to 9 by adding an appropriate number of virtual points as defined above. A third last trick (trick 3) consists in circumventing the rank drop of \mathbf{P}_W by solving the rank deficient least-squares problem given by Eqs. (11) via a method based on the singular value decomposition (SVD) of \mathbf{P}_W (Golub & Van Loan, 1983). Among the three remedies, trick (3) is the most general (it does not require a-priori knowledge of the structure of \mathbf{P}_W), rather accurate, but it is also the most computationally intensive; trick (2) is quite general (it requires some a-priori knowledge of the structure of \mathbf{P}_W) and quite computationally efficient, but it is the most inaccurate; trick (1) is the less general (it requires a-priori knowledge of the full structure of \mathbf{P}_W), it is quite accurate and quite computationally efficient.

4.3 A novel method for the manipulator actual configuration determination

As described in sub-section 4.2.1, the effectiveness of the general method relies upon the good conditioning of \mathbf{P}_W . A very practical sensor layout which both guarantees that the rank of \mathbf{P}_W is independent of manipulator configuration and greatly simplifies the solution of the DPA is the set $\{n\text{-RRP}\}$ ($n \geq 3$). With this sensor layout, the DPA problem described by Eqs. (10) is reduced to

$$\min_{\mathbf{R}} \|\mathbf{R}\mathbf{P} - \mathbf{B} - \mathbf{V}\|_{\text{F}}, \quad (13.1)$$

subject to $\mathbf{R}^T\mathbf{R} = \mathbf{1}$ and $\det(\mathbf{R}) = +1$,

and

$$\mathbf{c} = \underline{\mathbf{b}} + \underline{\mathbf{v}} - \mathbf{R}\underline{\mathbf{p}}, \quad (13.2)$$

where $\underline{\mathbf{p}}$, $\underline{\mathbf{b}}$ and $\underline{\mathbf{v}}$ are the following 3×1 mean vectors

$$\underline{\mathbf{p}} = \frac{1}{n} \sum_{j=1}^n \mathbf{p}_j, \quad (13.3)$$

$$\underline{\mathbf{b}} = \frac{1}{n} \sum_{j=1}^n \mathbf{b}_j, \quad (13.4)$$

$$\underline{\mathbf{v}} = \frac{1}{n} \sum_{j=1}^n l_j \mathbf{v}_j, \quad (13.5)$$

and \mathbf{P} , \mathbf{B} and \mathbf{V} are the following $3 \times n$ matrices

$$\mathbf{P} = [\mathbf{p}'_1 \quad \dots \quad \mathbf{p}'_n], \quad (13.6)$$

$$\mathbf{B} = [\mathbf{b}'_1 \quad \dots \quad \mathbf{b}'_n], \quad (13.7)$$

$$\mathbf{V} = [\mathbf{v}'_1 \quad \dots \quad \mathbf{v}'_n], \quad (13.8)$$

which are formed, respectively, by the 3×1 vectors $\mathbf{p}'_i = (\mathbf{p}_i - \underline{\mathbf{p}})$, $\mathbf{b}'_i = (\mathbf{b}_i - \underline{\mathbf{b}})$ and $\mathbf{v}'_i = (\mathbf{v}_i - \underline{\mathbf{v}})$. It is worth highlighting that the quantities $\underline{\mathbf{p}}$, $\underline{\mathbf{b}}$, \mathbf{P} and \mathbf{B} depend only on the manipulator geometry, while $\underline{\mathbf{v}}$ and \mathbf{V} depend also on the manipulator configuration. As usual, the notation $\|\mathbf{A}\|_{\text{F}}$ appearing in Eq. (13.1) is used to indicate the Frobenius norm of matrix \mathbf{A} . Equations (13) show that if the center O_p of the mobile frame S_p is chosen as the centroid of points P_i ($i = 1, \dots, n$), i.e. $\underline{\mathbf{p}} = \mathbf{0}$, then the orientation and the position problems are decoupled, i.e. $\mathbf{c} = (\underline{\mathbf{b}} + \underline{\mathbf{v}})$.

Following the procedure based on the ULS estimate which was described in section 4.2, an acceptable minimizer \mathbf{R} of the CLS problem described by Eq. (13.1) is

$$\mathbf{R} = \text{OPF}(\hat{\mathbf{R}}), \quad (14.1)$$

$$\hat{\mathbf{R}} = (\mathbf{B} + \mathbf{V})\mathbf{P}^T (\mathbf{P}\mathbf{P}^T)^{-1}. \quad (14.2)$$

However, for the set $\{n\text{-RRP}\}$ ($n \geq 3$), the optimal solution of Eq. (13.1) can be found in closed-form. Indeed, the CLS problem described in Eq. (13.1) is well known in computer vision (Umeyama, 1991) and admits the following solution

$$\mathbf{R} = \mathbf{U} \left[\text{diag}(1, 1, \det(\mathbf{US})) \right] \mathbf{S}^T, \quad (15.1)$$

where \mathbf{U} and \mathbf{V} are the 3×3 matrices coming from the SVD of the cross-covariance matrix

$$\mathbf{C} = (\mathbf{B} + \mathbf{V})\mathbf{P}^T. \quad (15.2)$$

That is, $\mathbf{C} = \mathbf{UDS}^T$ ($\mathbf{UU}^T = \mathbf{SS}^T = \mathbf{1}$ and $\mathbf{D} = \text{diag}(d_1, d_2, d_3)$, $d_1 \geq d_2 \geq d_3 \geq 0$). The unique solution given by Eq. (15) does not require the full rank of \mathbf{C} (Umeyama, 1991). As a matter of fact, the actual platform orientation can be computed whenever $\text{rank}(\mathbf{C}) \geq 2$.

The solution given in Eq. (15) is different from that proposed in (Baron and Angeles, 2000)

$$\mathbf{R} = \text{OPF}(\mathbf{C}), \quad (16)$$

which is the solution of the orthogonal Procrustes problem (Golub & Van Loan, 1983) obtained from the CLS problem of Eq. (13.1) by relaxing the constraint $\det(\mathbf{R}) = +1$.

4.4 Comparison of different DPA methods in terms of accuracy and computational efficiency

Among the different solution methods represented by equations (14), (15) and (16), only Eqs. (15) always provides the exact minimum of the CLS problem given by Eq. (13). Thus, only the solution given by Eqs. (15) always corresponds to the actual platform orientation and is the most accurate. Indeed, the solutions given by Eqs. (14) and Eq. (16) do not guarantee the proper orthogonality ($\det(\mathbf{R}) = +1$) of matrix \mathbf{R} . This is rather risky since Eqs. (14) and Eq. (16) may fail to give the correct rotation matrix (corresponding to the actual manipulator configuration) and may give a reflection instead when the sensor readouts are affected by measurement errors (this drawback is more severe the larger the measurement errors are). Between the solutions given by Eqs. (14) and Eq. (16), the former is the least accurate. Indeed, Eqs. (14) do not even minimize Eq. (13.1) (Eqs. (14) can be a viable good estimate of the solution in cases where measurement errors are rather small only). Moreover, due to the matrix inversion operation, note that Eqs. (14.2) requires matrix \mathbf{P} to have full rank. This is not the case whenever points P_i 's ($i = 1, \dots, n$) are coplanar. In such instances, as already described in section 4.2.1, to obtain the solution of Eq. (14.2) it is necessary to resort to either trick (2), which however leads to a rather inaccurate solution, or trick (3), which however implies a large computational effort.

In terms of computational efficiency, it is worth highlighting that the solution represented by Eqs. (15) requires the calculation of the SVD of a 3×3 matrix, while the solutions represented by equations (14) and (16) require the calculation of the polar decomposition (PD) of a 3×3 matrix. In general the algorithms available for the computation of the PD are more efficient than those available for the computation of the SVD. However, when 3×3 matrices are of concern, fast and robust solutions of the SVD exist which require fewer calculations than those required by the PD of 3×3 matrices. As a matter of fact, the SVD of a 3×3 matrix can be obtained via non-iterative algorithms. As an example, an improved version of the algorithm presented in (Vertechy & Parenti-Castelli, 2004), which is based on the analytical solution of the cubic equation, requires only 150 multiplications/divisions, 88 sums/subtractions, 5 square root evaluations and 4 trigonometric evaluations to obtain the full SVD. Conversely, the algorithms available for the PD are iterative. In particular, considering the most well known and adopted algorithms, the PD of 3×3 matrices via the routine proposed in (Dubrulle, 1999) requires $(87 + k_D \cdot 78)$ multiplications/divisions,

$(47 + k_D \cdot 39)$ sums/subtractions and $(4 + k_D \cdot 3)$ square root evaluations, where k_D is the number of iterations required by the Dubrulle's routine to converge; and the PD of 3×3 matrices via the routine proposed in (Higham, 1986) requires $(48 + k_H \cdot 63)$ multiplications/divisions, $(38 + k_H \cdot 62)$ sums/subtractions and $(k_H \cdot 3)$ square root evaluations, where k_H is the number of iterations required by Higham's routine to converge. In practice, simulations of the DPA solution of UPS-PMs employing both Dubrulle's and Higham's routines show that $k_D > 3$ and $k_H > 2$ when solving Eq. (14.1), and that $k_D > 5$ and $k_H > 5$ when solving Eq. (16). Note that the solution of Eq. (16) requires more iterations than those of Eq. (14.1) since matrix $\hat{\mathbf{R}}$ is closer to orthogonality than matrix \mathbf{C} .

Finally, it is worth mentioning that both Dubrulle's and Higham's routines involve the matrix inversion operation of either $\hat{\mathbf{R}}$ or \mathbf{C} and, thus, both Eq. (14.1) and Eq. (16) require such matrices to have full rank. Again, this is not the case whenever points P_i 's ($i = 1, \dots, n$) are coplanar, and this requires resorting to either trick (2), which leads to a rather inaccurate solution, or trick (3). In this latter case, once the SVD of either \mathbf{C} or $\hat{\mathbf{R}}$ is calculated (i.e. either $\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ or $\hat{\mathbf{R}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$), the solution of Eq. (14.1) and Eq. (16) is found as $\mathbf{R} = \mathbf{U}\mathbf{V}^T$. Hence, generally, in order to find a unique and accurate solution of the DPA, the computation of the SVD of either \mathbf{C} or $\hat{\mathbf{R}}$ is anyway required.

5. Conclusions

This chapter addresses the solution of the direct position analysis (DPA) of parallel manipulators. More specifically, it focuses on the determination of the actual configuration of parallel manipulators, which have legs of type UPS (where U, S and P are for universal, spherical and prismatic pairs respectively), by using extra-sensor data, that is a number of sensor data which is greater than the number of manipulator degrees of freedom. First, an extensive overview of the extra-sensor approaches that are available in the literature for the solution of the manipulator direct position analysis is provided. Second, a general method is described which makes it possible to solve accurately and in real-time the DPA of manipulators having general architecture, general sensor layouts and sensor data affected by measurement errors. The method, however, may suffer from singularities of the set of sensor data. Third, a novel method is presented which, by exploiting a suitable sensor layout, makes it possible to solve robustly, accurately and in real-time the direct position analysis of manipulators having general architecture and sensor data affected by measurement errors. A comparison with other methods based on mathematical proofs is provided that shows the accuracy and the computational efficiency of the proposed novel method.

6. References

- Angeles, J. (1990). Rigid-body pose and twist estimation in the presence of noisy redundant measurements, *Proc. Eighth CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*, pp. 69-78, Cracow, July 2-6 1990
- Baron, L. & Angeles, J. (1994). The measurement subspaces of parallel manipulators under sensor redundancy, *ASME Design Automation Conf.*, pp. 467-474, Minneapolis, 11-14 September 1994
- Baron, L. & Angeles, J. (1995). The isotropic decoupling of the direct kinematic of parallel manipulators under sensor redundancy, *IEEE Int. Conf. on Robotics and Automation*, pp. 1541-1546, Nagoya, 25-27 May 1995

- Baron, L. & Angeles, J. (2000a). The direct kinematics of parallel manipulators under joint-sensor redundancy. *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 1, 12-19
- Baron, L. & Angeles, J. (2000b). The kinematic decoupling of parallel manipulators using joint-sensor data. *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 6, 644-651
- Bonev, I.A. & Ryu, J. (2000). A new method for solving the direct kinematics of general 6-6 Stewart platforms using three linear extra sensors. *Mechanism and Machine Theory*, Vol. 35, No. 3, 423-436
- Bonev, I.A.; Ryu, J.; Kim, S.-G. & Lee, S.-K. (2001). A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 2, 148-156
- Cappel, K.L (1967). Motion simulator. *US Patent #3295224*
- Charles, P.A.-S. (1995). Octahedral machine tool frame. *US Patent #5392663*
- Cheok, K.C.; Overholt, J.L. & Beck, R.R. (1993). Exact Method for Determining the Kinematics of a Stewart Platform Using Additional Displacement Sensors. *Journal of Robotic Systems*, Vol. 10, No. 5, 689-707
- Chiu, Y.J. & Perng, M.-H. (2001). Forward kinematics of a general fully parallel manipulator with auxiliary sensors. *Int. J. of Robotics Research*, Vol. 20, No. 5, 401-414
- Daniel, R.W.; Fischer, P.J. & Hunter, B. (1993). A High Performance Parallel Input Device. *Proc. SPIE Vol. 2057, Telemanipulator Technology and Space Telerobotics*, pp. 272-281, Boston, December 1993
- Di Gregorio, R. & Parenti-Castelli, V. (2002). Fixation devices for long bone fracture reduction: an overview and new suggestions. *Journal of Intelligent and Robotic Systems*, Vol. 34, No. 3, 265-278
- Dubrulle, A.A. (1999). An optimum Iteration for the Matrix Polar Decomposition. *Electronic Transactions on Numerical Analysis*, Vol. 8, 21-25
- Etemadi-Zanganeh, K. & Angeles, J. (1995). Real time direct kinematics of general six-degree-of-freedom parallel manipulators with minimum sensor data. *Journal of Robotics Systems*, Vol. 12, No. 12, 833-844
- Faugere, J.C. & Lazard, D. (1995). The combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, Vol. 30, No. 6, 765-776
- Fenton, R.G. & Shi, X. (1989). Comparison of methods for determining screw parameters of finite rigid body motions from initial positions and final position data, in *Advances in Design Automation*, Vol. 3, 433-439
- Gaillet, A. & Reboulet, C. (1983). An Isostatic Six Component Force and Torque Sensor. *Proc. 13th Int. Symposium on Industrial Robotics*, pp. 102-111, Chicago, 18-21 April 1983
- Geng, Z. & Haynes, L.S. (1994). A 3-2-1 kinematic configuration of a Stewart platform and its application to six degree of freedom pose measurements. *Robotics & Computer-Integrated Manufacturing*, Vol. 11, No. 1, 23-34
- Golub, G.H. & Van Loan, C.F. (1983). *Matrix Computations*, The Johns Hopkins University Press, ISBN 0-946536-00-7, Baltimore
- Gough, V.E. & Whitehall, S.G. (1962). Universal Tire Test Machine. *Proceedings 9th Int. Technical Congress F.I.S.I.T.A*, Vol. 117, pp. 117-135, London, 30 April - 5 May 1962
- Griffis, M. & Duffy, J. (1989). A Forward Displacement Analysis of a Class of Stewart Platform. *Journal of Robotics Systems*, Vol. 6, No. 6, 703-720
- Han, H.; Chung, W. & Youm, Y. (1996). New Resolution Scheme of the Forward Kinematics of Parallel Manipulators Using Extra Sensors. *ASME Journal of Mechanical Design*, Vol. 118, No. 2, 214-219
- Hesselbach, J.; Bier, C.; Pietsch, I.; Plitea, N.; Büttgenbach, S.; Wogersien, A. & Güttler, J. (2005). Passive-joint sensors for parallel robots. *Mechatronics*, Vol. 15, 43-65

- Higham, N.J. (1986). Computing the Polar Decomposition - with Applications. *SIAM Sci. Stat. Comput.*, Vol. 7, No. 4, 1160-1174
- Innocenti, C. & Parenti-Castelli, V. (1990). Direct Position Analysis of the Stewart Platform Mechanism. *Mechanism and Machine Theory*, Vol. 25, No. 6, 611-621
- Innocenti, C. & Parenti-Castelli, V. (1991). A Novel Numerical Approach to the Closure of the 6-6 Stewart Platform Mechanism. *ICAR'91, Fifth Int. Conf. on Advance Robotics*, pp. 851-855, Pisa, 19-22 June
- Innocenti, C. & 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, 553-561.
- Innocenti, C. & Parenti-Castelli, V. (1994). Exhaustive Enumeration of Fully Parallel Kinematic Chains, *ASME International Winter Annual Meeting DSC-55-2*, pp. 1135-1141, Chicago, November 1994.
- Innocenti, C. (1998). Closed-Form Determination of the Location of a Rigid Body by Seven In-Parallel Linear Transducers. *ASME Journal of Mechanical Design*, Vol. 120, 293-298
- Innocenti, C. (2001). Forward Kinematics in Polynomial Form of the General Stewart Platform. *ASME Journal of Mechanical Design*, Vol. 123, 254-260
- Jacobsen, S.C. (1975). Rotary-to-Linear and Linear-to-Rotary Motion Converters. *US Patent #3864983*
- Jin Y. (1994). Exact solution for the forward kinematics of the general stewart platform using two additional displacement sensors. *Proc. of the 23rd ASME Mechanism Conference, DE-Vol. 72*, pp. 491-495, Minneapolis, 11-14 September 1994
- Jin, Y. & Hai-rong, F. (1996). Explicit Solution for the Forward Displacement Analysis of the Stewart Platform Manipulator. *Proc. ASME DETC 1996*, Irvine, 18-22 August 1996
- Lee, T.-Y. & Shim, J.-K. (2001). Forward kinematics of the general 6-6 Stewart platform using algebraic elimination. *Mechanism and Machine Theory*, Vol. 36, No. 9, 1073-1085
- Lewis, J.L.; Carroll, M.B.; Morales, R.H. & Le, T.D. (2002). Androgynous, reconfigurable closed loop feedback controlled low impact docking system with load sensing electromagnetic capture ring. *US Patent #6354540*
- McAree, P.R. & Daniel, R.W. (1996). A Fast, Robust Solution to the Stewart Platform Forward Kinematics. *Journal of Robotics Systems*, Vol. 13, No. 7, 407-427
- McCallion, H. & Truong, P.D. (1979). The Analysis of a Six-Degree-of-Freedom Work Station for Mechanised Assembly. *Proceedings of the Fifth World Congress on Theory of Machines and Mechanisms*, 611-617, Montreal, July 1979.
- Merlet, J-P. (1992). Direct Kinematics and Assembly Modes of Parallel Manipulators. *The International Journal of Robotics Research*, Vol. 11, No. 2, 150-162
- Merlet, J-P. (1993a). Direct Kinematics of Parallel Manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, 842-845
- Merlet, J-P. (1993b). Closed-Form Resolution of the Direct Kinematics of Parallel Manipulators Using Extra Sensors Data. *Proc. IEEE Int. Robotics and Automation Conf.*, pp. 200-204, Atlanta, 2-7 May 1993.
- Nair, R. & Maddocks, J.H. (1994). On the Forward Kinematics of Parallel Manipulators. *The Int. Journal of Robotics Research*, Vol. 13, No. 2, 171-188
- Nanua, P.; Waldron, K.J. & Murty, V. (1990). Direct Solution of a Stewart Platform. *IEEE Transaction on Robotics and Automation*, Vol. 6, No. 4, 438-443
- Nguyen, C.C.; Antrazi, S.S. & Zhou, Z.L. (1991). Analysis and Implementation of a 6 DOF Stewart Platform-Based Force Sensor for Passive Compliant Robotic Assembly. *IEEE Proc. of the Southeast Conf'91*, pp. 880-884, Williamsburg, 7-10 April 1991
- Parenti-Castelli, V. & Di Gregorio, R. (1995). A Three Equations Numerical Method for the Direct Kinematics of the Generalized Gough-Stewart Platform. *9th World Congress*

- on the Theory of Machines and Mechanisms*, pp. 837-841, Milan, 30 August - 2 September 1995.
- Parenti-Castelli, V. & Di Gregorio, R. (1998). Real-Time Computation of the Actual Posture of the General Geometry 6-6 Fully-Parallel Mechanism using Two Extra Rotary Sensors. *Journal of Mechanical Design*, Vol. 120, No. 4, 549-554
- Parenti-Castelli, V. & Di Gregorio, R. (1999). Determination of the Actual Configuration of the General Stewart Platform Using Only One Additional Sensor. *Journal of Mechanical Design*, Vol. 121, No. 1. 21-25
- Parenti-Castelli, V. & Di Gregorio, R. (2000). A new algorithm based on two extra sensors for real-time computation of the actual configuration of the generalized Stewart-Gough manipulator. *ASME J. of Mechanical Design*, Vol. 122, No. 3, 294-298
- Reboulet, C. (1988). Robot parallèles. *Technique de la Robotique*, Hermes (Ed.), Paris
- Schmidt-Kaler, T. (1992). The Hexapod Telescope: A New Way to Very Large Telescopes. *Progress in Telescope and Instrumentation Technologies, ESO Conference and Workshop Proceedings, ESO Conference on Progress in Telescope and Instrumentation Technologies*, p. 117, European Southern Observatory (ESO), Garching, 27-30 April 1992.
- Shi, X. & Fenton, R.G. (1991). Forward Kinematic Solution of a General 6 DOF Stewart Platform Based on Three Point Position Data. *Eight World Cong. on the Theory of Machines and Mechanism*, 1015-1018, Prague, 26-31 August 1991.
- Stewart, D. (1965). A Platform with Six Degree of Freedom. *Proc. of the Institution of Mechanical Engineers*, vol. 180, No. 15, 371-386
- Stoughton, R. & Arai, T. (1991). Optimal sensor placement for forward kinematics evaluation of a 6-dof parallel link manipulator. *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'91*, pp. 785-790, Osaka, 3-5 November 1991.
- Taylor, H.S. & Taylor, J.C. (2000). Six axis external fixator strut. *US Patent#6030386*
- Tancredi, L. & Merlet, J.-P. (1994). Evaluation of the errors when solving the direct kinematics of parallel manipulators with extra sensors, In: *Advances in Robot Kinematics and Computational Geometry*, Lenarcic J. and Ravani B., (Ed), 439-448, Springer, ISBN:978-0-7923-2983-1
- Tancredi, L.; Teillaud, M. & Merlet, J.-P. (1995). Extra sensors data for solving the forward kinematics problem of parallel manipulators. *9th IFToMM World Congress on the Theory of Machines and Mechanisms*, pp. 2122-2126, Milan, 30 August-2 September 1995
- Umeyama, S. (1991). Least-Squares Estimation of Transformation Parameters Between Two point Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 4, 376-380
- Vertechy, R.; Dunlop, G.R. & Parenti-Castelli, V. (2002). An accurate algorithm for the real-time solution of the direct kinematics of 6-3 Stewart platform manipulators, In: *Advances in Robot Kinematics: Theory and Applications*, Lenarcic J. & Thomas F., (Ed.), 369-378, Springer, ISBN: 978-1-4020-0696-8
- Vertechy, R. & Parenti-Castelli, V. (2004). A fast and Accurate Method for the Singular Value Decomposition of 3x3 Matrices, In: *On Advances in Robot Kinematics*, Lenarcic J. and Galletti C., (Ed.), 3-12, Springer, ISBN: 978-1-4020-2248-7
- Vertechy, R. & Parenti-Castelli, V. (2006). Synthesis of 2-DOF Spherical US Parallel Mechanisms, In: *Advances in Robots Kinematics: Mechanisms and Motion*, Lenarcic J. and Roth B., (Ed.), 385-394, Springer, ISBN: 978-1-4020-4940-8
- Vertechy, R. & Parenti-Castelli, V. (2007). Accurate and Fast Body Pose Estimation by Three Point Position Data. *Mechanism and Machine Theory*, Vol. 42, No. 9, 1170-1183

Kinematic Modeling, Linearization and First-Order Error Analysis

Andreas Pott† and Manfred Hiller‡

† *Fraunhofer Institute for Manufacturing Engineering and Automation, Stuttgart*

‡ *Chair of Mechatronics, University of Duisburg-Essen, Germany*

1. Introduction

The kinematic analysis of parallel kinematic machines (PKM) is a challenging field, since PKM are complex multi-body systems involving a couple of closed kinematic loops. It is well-known that the forward kinematic function has in general no closed-form solution, and that up to 40 different real solutions may exist for general geometry (Husty, 1996; Dietmaier, 1998). Therefore, an efficient and handy method is needed in practise, e.g. for design, simulation, control, and calibration.

The analysis of manufacturing and assembly errors of manipulators is a topic that is highly relevant for practical applications because the magnitude of these errors is directly coupled to the total cost of production of the manipulator. In this setting, there exist intensive studies on how to estimate the error of certain moving points, e.g. the tool center point, in terms of the errors in the components of the mechanism (Brisan et al., 2002; Jelenkovic & Budin, 2002; Kim & Choi, 2000; Song et al., 1999; Zhao et al., 2002), as well as how to allocate cost-optimal tolerances to a mechanism (Chase et al., 1990; Ji et al., 2000). In this paper, an approach to estimate the first-order influence of geometric errors on target quantities is suggested in which linearization is performed by considering the force transmission of the manipulator. This enables one to obtain a comprehensive model of linearized geometric sensitivities at a low computational cost.

Error analysis for serial manipulators is relatively easy because one can establish an analytical expression for the forward kinematics which maps the generalized joint and link coordinates to the spatial displacements of the end-effector. There are numerous methods to parameterize the forward kinematics, where the approach of Denavit and Hartenberg (1955) is the most popular one. Once one has a closed-form expression for the forward kinematics, one can take derivatives of it (with respect to the geometric parameters one is interested in) and use these as sensitivity coefficients. In general, one introduces the sensitivity parameters in such a way that they vanish at the nominal configuration. This is always possible by introducing corresponding constant offsets where necessary.

For example, consider a robot involving a universal joint, and assume that the sensitivity to errors in the fulfilment of the intersection property of the axes is to be analyzed. This can be done by adding a parameter for the normal distance between the joint axes which is zero in the nominal design, and with respect to which the partial derivative will yield the sought sensitivity. However, such a method for sensitivity analysis results in a model with a

significant overhead. Examples of such models for joints are presented (Brisan et al., 2002; Song et al., 1999). Some force-based methods for clearance analysis were introduced, which are similar to the approach in this paper (Innocenti, 1999; Innocenti, 2002; Parenti-Castelli & Venanzi, 2002; Parenti-Castelli & Venanzi, 2005).

A linearization method for complex mechanisms using the kinetostatic dualism and the concept of *kinematical differentials* to efficiently set up the equations of motion of multi-body systems has been proposed (Kecskeméthy & Hiller, 1994). Using this method, all required partial derivatives can be described solely by using the kinematic transmission functions for position and velocity, as well as the force transmission function of the system. Based on these transmission functions, an algorithm is formulated for generating the Jacobian matrix and the equations of motion through multiple evaluations of the kinematic transmission functions for certain pseudo input velocities and accelerations. The corresponding algorithms are denoted as *kinematical differentials* for the case of the pure kinematic transmission function (Hiller & Kecskeméthy, 1989) and *kinetostatic approach* for the case of use of force transmission (Kecskeméthy, 1994). Later, Lenord et al. (2003) showed that kinematical differentials may be applied also to more general interdisciplinary systems which also involve hydraulic components by using an exact linearization through the kinematical differentials for the determination of the velocity linearization and numerical differentiation for the calculation of the stiffness matrix of the hybrid system. Other authors studied the determination of the stiffness matrix for complex multi-body systems using explicit symbolic derivatives (El-Khasawneh & Ferreira, 1998; Rebeck & Zhang, 1999), taking into account the stiffness of the actuators and the stiffness of special components. These approaches however require numerous computational steps when many sensitivity parameters are involved.

2. Kinematic modeling of parallel kinematic machines

2.1 Kinematic delimitation and geometry

In order to study a wide range of machine types, a generic approach for the modeling of PKM is proposed (Pott, 2007). Since PKMs tend to be symmetric and different types of PKM have similar components a modular design is used. In a first step the machine is divided into three types of components: frames, platforms and legs (Fig. 1), which form the modules of the kinetostatic code.

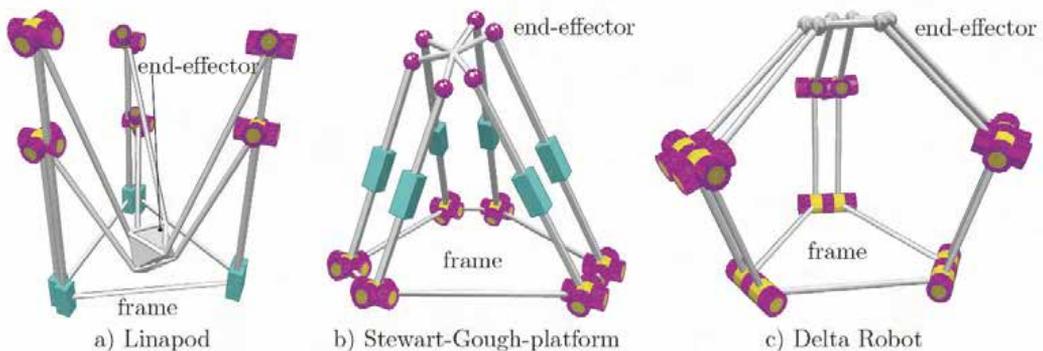


Fig. 1. Platform, legs, and machine frame modules of a generic six-degree-of-freedom parallel kinematic machine

The machine frame defines the position and orientation of six pivot points A_i . The mobile platform introduces the position of six pivot point B_i . Furthermore, the platform defines the parameterization of the six-degrees-of-freedom (dof) of spatial motion at the tool center point (TCP). Finally, different types of legs are introduced which mainly determine the kinematic behaviour. The most common legs for PKMs are PUS, UPS and RUS structures each consisting of an actuated prismatic (P) or revolute (R) joint as well as a pair of a universal (U) and a spherical (S) joints. Each of these structures can be described by one scalar constraint, as it is shown hereafter.

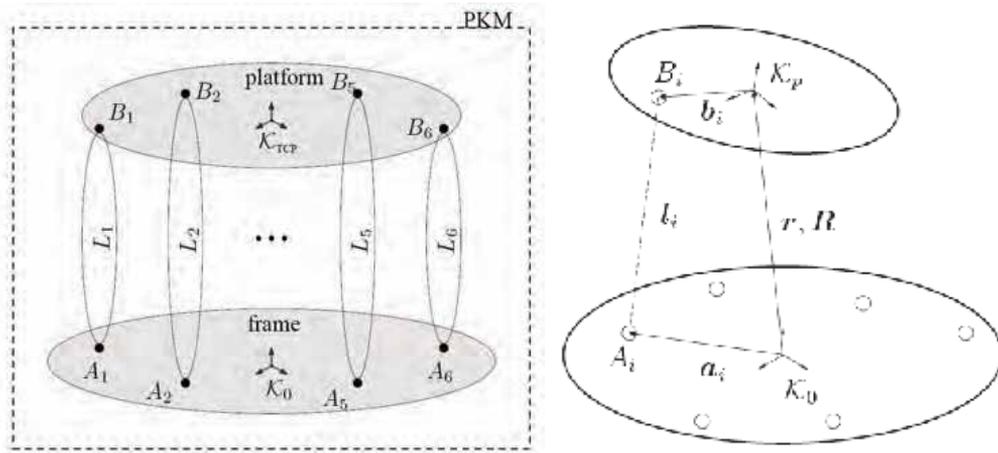


Fig. 2. Generic model of a spatial six-degree-of-freedom parallel kinematic machine

Each legs considered in this paper possess a pair of joints formed by a universal joint and a spherical joint. For the analysis of the closed-kinematic chains, these joints are known as characteristic pair of joints (Woernle, 1988). One can remove both of these joints and replace this partial chain by one nonlinear scalar constraint. This constraint describes the geometrical distance between the center of the universal joint and the center of the spherical joint for the i -th leg as

$$\mathbf{a}_i + \mathbf{l}_i = \mathbf{r} + \mathbf{R}\mathbf{b}_i, \quad (1)$$

where \mathbf{a}_i denotes the position vector of the pivot point on the base and \mathbf{b}_i is the relative position of the pivot point with respect to the coordinate system fixed to the platform. The Cartesian position and orientation of the platform frame K_{TCP} is given by the vector \mathbf{r} and the orthogonal matrix \mathbf{R} , respectively. The vector \mathbf{l}_i denotes the length of the leg. Solving Eq. (1) for the magnitude l_i^2 of the vector \mathbf{l}_i yields the system of six nonlinear constraints

$$(\mathbf{a}_i - \mathbf{r} - \mathbf{R}\mathbf{b}_i)^2 - l_i^2 = 0 \quad i=1, \dots, 6. \quad (2)$$

The world coordinates \mathbf{y} consist of a parameterization of the position \mathbf{r} and the orientation matrix \mathbf{R} . The geometry of the machine is expressed by the vectors \mathbf{a}_i , \mathbf{b}_i and \mathbf{l}_i . In the following sections the definition of these vectors is introduced depending on the generalized coordinate q_i of the six actuators and the kinematic structure of the basic types of legs for parallel kinematic machines.

The UPS legs are used in the Stewart-Gough-platforms which are often applied for motion simulators of cars and aircrafts. The prismatic joint is actuated as linear actuator, e.g. by a

linear direct drive, ball bearing screw, hydraulic/pneumatic cylinder. For these mechanisms the pivot points A_i on the base

$$\mathbf{a}_i = \mathbf{c}_i \tag{3}$$

are determined by the machine frame and fixed to a given position \mathbf{c}_i . The length of the strut can be controlled by the drive through

$$l_i = q_i + q_0 \tag{4}$$

where q_0 is a constant offset.

The PUS leg results from changing the order of the joints within the UPS leg. The universal-spherical pair encloses a leg of constant length while the proximal pivot point is actuated along a line. PUS legs are the basic leg components for Hexaglide, Linaglide and Linapod PKM. They are described by the position vector \mathbf{c}_i and the direction \mathbf{u}_i . Thus, the position of the proximal pivot point A_i is defined as

$$\mathbf{a}_i = \mathbf{c}_i + q_i \mathbf{u}_i, \tag{5}$$

where q_i is the generalized coordinate of the drive. The length $l_i = \text{const}$ of the strut is given by design.

Finally, the kinematics of the RUS leg is considered which is the basis of the Delta-robot (Clavel, 1988). In contrast to the PUS leg, the proximal pivot points A_i of RUS legs move a circle defined by its center \mathbf{c}_i , an axis of rotation \mathbf{u}_i , and a lever \mathbf{v}_i which is given in its initial position. Thus, it holds for the point A_i

$$\mathbf{a}_i = \mathbf{c}_i + \mathbf{T}(\mathbf{u}_i, q_i) \mathbf{v}_i \tag{6}$$

where $\mathbf{T}(\mathbf{u}, q)$ is the rotation matrix for the axis \mathbf{u} and the angle q as it can be calculated by Rodrigues formula. Again, for RUS legs the length of the strut $l_i = \text{const}$ is given by design.

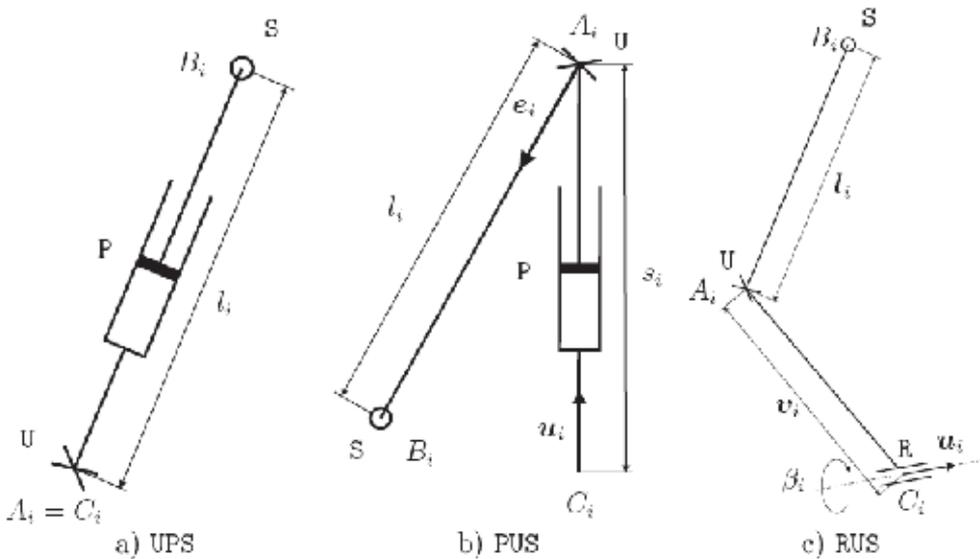


Fig. 3. Geometric parameters of the leg-types under consideration

2.2 Review of the kinetostatic method

In the sequel, a short introduction to the kinetostatic method (Kecskeméthy, 1993; Kecskeméthy, 1994) is given as a basis for the herewith presented linearization procedure. Below, the basic equations of the kinetostatic transmission element are reviewed for better reference in this paper. Details can be found in the cited papers. In the kinetostatic formalism, mechanical components are modeled as transmission elements (Fig. 4) that map the kinematic state \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ at the input to the kinematic state \mathbf{q}' , $\dot{\mathbf{q}}'$, $\ddot{\mathbf{q}}'$ at the output and the associated generalized forces \mathbf{Q}' at the output to generalized forces \mathbf{Q} at the input. The kinetostatic state is composed of position, velocity, acceleration, and force. A mechanism is divided into joints and links which transmit the state from one set of coordinate frames \mathcal{K}_i and scalar variables β_i to another set $\{\mathcal{K}_i', \beta_i'\}$. This concept allows one to model serial manipulators as chains of transmission elements.

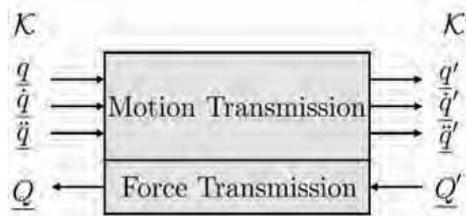


Fig. 4. General kinetostatic transmission element

Interestingly, the motion of closed kinematic loops can also be represented with the help of transmission functions. For simple kinematic loops with one dof, this may be done in an explicit form. In general, however, one has to employ iterative methods to solve the loops. Nevertheless, in both cases one is able to compute the transmission function for position, velocity, acceleration, and force.

Assuming that the position transmission function is given as $\mathbf{y}=\boldsymbol{\varphi}(\mathbf{q})$, where $\mathbf{q}=[q_1, \dots, q_n]^T$ is a set of independent joint variables of the mechanism, the velocity transmission takes the form

$$\dot{\mathbf{y}} = \frac{\partial \boldsymbol{\varphi}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_q(\mathbf{q}) \dot{\mathbf{q}} \quad (7)$$

For a given set of joint coordinates \mathbf{q} , the twist $\dot{\mathbf{y}}$ of the end-effector frame (EEF) is a linear combination of the joint rates $\dot{\mathbf{q}}$ with the columns of the Jacobian \mathbf{J}_q acting as coefficient vectors. Assuming ideal transmission behaviour within the transmission element, power is neither generated nor consumed. Thus, virtual work at the input and the output can be set equal, and one obtains

$$\delta \mathbf{q}^T \mathbf{Q} = \delta \mathbf{q}'^T \mathbf{Q}' \quad (8)$$

where the virtual displacements fulfil $\delta \mathbf{q}' = \mathbf{J}_q \delta \mathbf{q}$. This yields $\delta \mathbf{q}'^T \mathbf{Q}' = \delta \mathbf{q}^T \mathbf{J}_q^T \mathbf{Q}'$ and since this holds for any $\delta \mathbf{q}$, the force transmission takes the form

$$\mathbf{Q} = \mathbf{J}_q^T \mathbf{Q}' \quad (9)$$

The relations (7) and (9) for velocity transmission and force transmission is called *kinetostatic duality*. The basic idea of the *kinematical differentials* is to evaluate the velocity transmission function $\dot{\phi}$ for pseudo unit velocities $\hat{q}_i = [0, \dots, 1, \dots, 0]^T$ with zeros everywhere up to the i -th component in order to identify the i -th column of the Jacobian J_q . Thus, the Jacobian J_q can be determined with n passes of velocity transmission. This method is called *velocity-based Jacobian algorithm*. By exploiting kinetostatic duality, this algorithm can be analogously applied to force transmission yielding the *force-based Jacobian algorithm*. Here the Jacobian J_q is computed row-wise by setting unit pseudo forces $\hat{Q}_i = [0, \dots, 1, \dots, 0]^T$ with zeros everywhere besides the i -th component, performing the force mapping (9), and storing the resulting vector of generalized forces \hat{Q}_i at the input as the i -th row of J_q . Thus, the Jacobian can be computed row-wise by six force transmissions for a general manipulator independently of the number of input parameters. This Jacobian evaluation procedure shall be further exploited here.

The complete kinetostatic formalism is implemented in the object-oriented programming library *Mobile* that uses the C++ programming language (Kecskeméthy, 1994), and a differential geometric interpretation of the kinetostatics has been given in (Kecskeméthy, 1993).

2.3 Modular modeling of parallel kinematic machines

As already mentioned before, the PKM is subdivided into the modules platform, frame, and legs. These components are the foundation of a modular kinetostatic model, which automatically assembles and solves the system of nonlinear constraints. The expressions introduced for the different legs in section 2.1 can be used to calculate the relative kinematics of the different types of PKM. By means of the kinetostatic method, C++ classes for the elementary components of multi-body systems, i.e. prismatic joints, revolute joints, and rigid bodies as well as the constraint solvers for kinematic loops are used for the modeling. These elements defined the required transmission functions for position, velocity, acceleration, and force. Thus, if the machine can be automatically assembled from these classes, one receives a comprehensive tool for the kinematic analysis of parallel robots.

A class called generic machine assembles a kinetostatic model from the components introduced above. Firstly, the legs is attached to the platform and to the frame. For the forward kinematics the legs provide constraints that characterize which motions can be transmitted. The generic machine assembly module collects the constraints from the legs and the generalized coordinates from the platform in order to combine them to a nonlinear system of equations. Then a numerical procedure like a Newton-Raphson-algorithm is applied to solve the forward kinematics. Once the position of the platform is determined one can use local methods from the legs to calculate the complementary variables of the passive joints in each kinematic loop.

The module frame defines the geometry of the base of the PKM by providing the position and orientation of the coordinate frame K_{C_i} , $i=1, \dots, 6$. These coordinate frames are connected to the world coordinate system by rigid links. On the other hand, the coordinate frames K_{C_i} are the interface for the legs to be attached to it. The module platform firstly defines the end-effectors frame K_P by means of the world coordinates $y=[x, y, z, \psi, \theta, \xi]^T$ with respect to the world coordinate frame K_0 , where x, y, z define the Cartesian position of the end-effector and

ψ, θ, ξ are a parameterization of the special orthogonal group $SO(3)$, e.g. by use of Bryant angles. The position of the platform pivot points K_{Bi} , $i=1, \dots, 6$ can be defined with respect to the frame K_P giving the geometry of the moveable platform. This presents a kinetostatic transmission function $\mu(\mathbf{y})$ mapping the world coordinates \mathbf{y} to the pivot points K_{Bi} . The modules for the legs present the governing properties for the kinematic transmission of the PKM, i.e. by presenting the kinetostatic transmission elements for the joints and the rigid links. The frames K_{Bi} and K_{Ci} act as interfaces to attach the legs to the platform and the base. To solve the forward kinematics each type of leg presents a specific constraint $v_i(K_{Bi}, K_{Ci}, q_i)$ which will be used by a central solver for forward kinematics. The constraint v_i for the different types of legs basically implements equations (3) – (6). Finally, the legs implement functions to solve for the angles in the passive joints, i.e. computes the orientation of the universal and spherical joints. This can be done in an explicit way by projection techniques that are well-known from solving four-link bar mechanisms.

For the forward kinematic problem one has to determine the platform world coordinates \mathbf{y} from given generalized coordinates \mathbf{q} . Based on the aforementioned modules the following algorithm can be used for all parallel robots treated in this work:

1. The module frame calculates the pivot points K_{Ci} .
2. A central constraint solver collects the constraint $v_i(K_{Bi}, K_{Ci}, q_i)$ from each leg module. Furthermore, the constraint solver receives the function $\mu(\mathbf{y})$ from the module platform. The constraint solver uses these equations to set up the nonlinear system $\Gamma(\mathbf{q}, \mathbf{y})=0$.
3. The constraint solver calculates the solution \mathbf{y}^* for the system $\Gamma(\mathbf{q}, \mathbf{y})=0$ with a Newton-Raphson-algorithm.
4. The platform update the K_{Bi} with $\mu(\mathbf{y}^*)$.
5. Each leg determines the dependent angles of the passive joints from the known values of (K_{Bi}, K_{Ci}, q_i) .

Thus, a comprehensive algorithm for forward kinematics for the Stewart-Gough-platform, the Delta-robot, and Linapod like machines is presented. Note, that by using the kinetostatic methods one also receives these relations in terms of velocity, acceleration, and force. The resulting kinetostatic model can be used for a wide range of functions for kinematic analysis e.g. forward kinematics, calculation of the Jacobian matrix, and dexterity indexes, and equations of motion. The discussion of all these algorithms is out of the scope of the paper. In the following section, the determination of a geometrical linearization will be highlighted.

2.4 Linearization and sensitivity analysis

In this study, the function of the forward kinematics of a multi-body system is denoted by $\boldsymbol{\varphi}(\mathbf{q}, \mathbf{g})$, where \mathbf{q} are the generalized independent joint coordinates, and \mathbf{g} collects all geometric parameters of the manipulator. The evaluation of the forward kinematics yields the world coordinates \mathbf{y} of a particular point of the end-effector of a manipulator together with the orientation of the end-effector, which shall be denoted here as end-effector frame (EEF). For most of the non-serial mechanisms, the function of the forward kinematics $\boldsymbol{\varphi}$ is not unique, since there may be multiple positions for the EEF that correspond to a given set of generalized joint coordinates \mathbf{q} due to different assembly modes. Here, it is assumed that it is possible to choose the solution that corresponds to the actual assembly mode, e.g. by

giving appropriate initial conditions. The linearization of the mechanism is formally achieved by taking the derivative with respect to the variables \mathbf{q} and \mathbf{g} , respectively, i.e. as

$$\delta\mathbf{y} = \frac{\partial\varphi(\mathbf{q},\mathbf{g})}{\partial(\mathbf{q},\mathbf{g})} \delta(\mathbf{q},\mathbf{g}) = \frac{\partial\varphi(\mathbf{q},\mathbf{g})}{\partial\mathbf{q}} \delta\mathbf{q} + \frac{\partial\varphi(\mathbf{q},\mathbf{g})}{\partial\mathbf{g}} \delta\mathbf{g} = \mathbf{J}_q \delta\mathbf{q} + \mathbf{J}_g \delta\mathbf{g}. \quad (10)$$

Here, quantities $\delta\mathbf{y}$, $\delta\mathbf{q}$, $\delta\mathbf{g}$ denote infinitesimal variations of the aforementioned coordinates, while \mathbf{J}_q denotes the well-known Jacobian matrix that is related to the kinematic dexterity of the manipulator. The matrix \mathbf{J}_g is also a Jacobian matrix which characterizes the sensitivity of the position \mathbf{y} of the EEF with respect to small changes, e.g. errors, in geometric parameters and which is used for sensitivity analysis.

For serial manipulators with n dof, the function φ can be written analytically in terms of the Denavit-Hartenberg-parameters $(\alpha_i, \theta_i, d_i, a_i)$ (Denavit & Hartenberg, 1955) and the vector of the geometric parameters becomes

$$\mathbf{g} = [\alpha_1, \theta_1, d_1, a_1, \dots, \alpha_n, \theta_n, d_n, a_n]^T. \quad (11)$$

Thus, the Jacobian matrices \mathbf{J}_q and \mathbf{J}_g can be calculated symbolically for serial manipulators. For nontrivial robots, however, the expressions are usually so extensive that they only can be handled by means of computer algebra.

Complex manipulator systems are characterized by the occurrence of closed kinematic loops. Such mechanisms have more joints than degrees-of-freedom, and the joint coordinates are coupled through closure conditions. This implies that the expressions for φ are either complicated, or that φ can only be computed point-wise by the iterative solution of an implicit system of nonlinear constraints the latter being the general case which occurs especially for parallel kinematic mechanisms that involve multiple coupled loops. Closed kinematic loops also occur in transmission mechanisms that can be found for instance in hydraulically driven manipulators like excavators or large scale manipulators, since they support the force transmission.

To overcome the lack of an analytical forward kinematic function for complex manipulators, the loop closure conditions $\mathbf{f}(\mathbf{y},\mathbf{g})=\mathbf{0}$ can be utilized for sensitivity analysis; by applying implicit differentiation (see e.g. Wittwer et al., 2004), one obtains

$$\frac{\partial\mathbf{f}(\mathbf{y},\mathbf{g})}{\partial\mathbf{y}} \delta\mathbf{y} + \frac{\partial\mathbf{f}(\mathbf{y},\mathbf{g})}{\partial\mathbf{g}} \delta\mathbf{g} = \mathbf{A}\delta\mathbf{y} + \mathbf{B}\delta\mathbf{g} = \mathbf{0}. \quad (12)$$

where $\mathbf{y}=[\mathbf{x}^T, \boldsymbol{\theta}^T]^T$ are the world coordinates of the end-effector frame, e.g. in form of a position vector \mathbf{x} in \mathbb{R}^3 and the orientation $\boldsymbol{\theta}$ holding e.g. Bryant angles, and \mathbf{g} are the geometric parameters. Then, one immediately obtains for the variation of the EEF world coordinates $\delta\mathbf{y}=\mathbf{A}^{-1}\mathbf{B}\delta\mathbf{g}$, where the matrix $\mathbf{A}^{-1}\mathbf{B}$ maps the errors $\delta\mathbf{g}$ in the components to the displacements $\delta\mathbf{y}$ of the EEF (Wittwer et al., 2004).

There are certain drawbacks to this approach: First, for mechanisms with more than three dof, an analytical form of matrix \mathbf{A}^{-1} can hardly be handled due to the length of the corresponding expressions. Second, if sensitivity analysis is established on the closure condition, one cannot access geometric parameters that are canceled ad-hoc through the formulation of the closure conditions. For example, the normal distance between the joint axes in universal joints is often eliminated because the number of closure constraints can be significantly reduced by assuming it to be exactly zero.

2.5 Linearization of manipulator systems

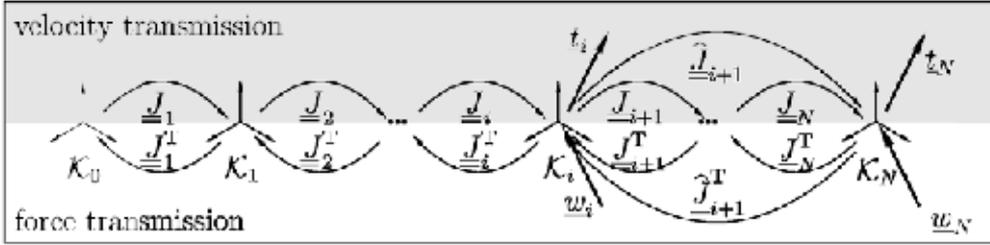


Fig. 5. Velocity and force transmission in a chain of kinetostatic transmission elements

Applying the kinetostatic formalism provides a procedure to calculate position, velocity, acceleration, and force transmission for an arbitrary manipulator. In general, this results in a chain of transmission elements as depicted in Fig. 5, where the individual mapping can also correspond to closed kinematic loops since such loops are also represented by transmission elements. In this figure, a twist $\mathbf{t}_i^T = [\boldsymbol{\omega}_i^T, \mathbf{v}_i^T]$ denotes the combination of an angular velocity $\boldsymbol{\omega}_i$ of a frame K_i and its corresponding velocity \mathbf{v}_i of its origin, both decomposed in some frame. A wrench $\mathbf{w}_i^T = [\mathbf{m}_i^T, \mathbf{f}_i^T]$ is composed of an applied moment \mathbf{m}_i at the frame K_i and an applied force \mathbf{f}_i at its origin, again decomposed in some frame. Given a certain set of joint coordinates \mathbf{q} , one can introduce a virtual twist displacement $\delta \mathbf{t}_i^T = [\delta \boldsymbol{\phi}_i^T, \delta \mathbf{r}_i^T]$ at the frame K_i , where $\delta \mathbf{r}_i$ is a virtual translational displacement and $\delta \boldsymbol{\phi}_i$ is an infinitesimal rotational increment in the space of rigid-body rotations, and study the corresponding virtual twist displacement $\delta \mathbf{t}_N$ at the EEF K_N . This linear relation is given by

$$\delta \mathbf{t}_N = (\mathbf{J}_{i+1} \mathbf{J}_{i+2} \dots \mathbf{J}_N) \delta \mathbf{t}_i = \hat{\mathbf{J}}_{i+1} \delta \mathbf{t}_i, \quad (13)$$

where \mathbf{J}_i denotes the Jacobian of the velocity transmission from frame K_{i-1} to the frame K_i . Using kinematical differentials (Sec. 2.2) one calculates the Jacobian $\hat{\mathbf{J}}_{i+1}$, which contains the sensitivity of the frame K_N with respect to displacements in K_i , and then concatenates the matrices $\hat{\mathbf{J}}_i^T$ for the sought matrix $\mathbf{J}_g^T = [\hat{\mathbf{J}}_1^T, \hat{\mathbf{J}}_2^T, \dots, \hat{\mathbf{J}}_N^T]$. Thus, for a comprehensive linearization, one needs six passes of the velocity transmission function for each $\hat{\mathbf{J}}_i^T$ and hence $6N$ passes for the whole manipulator.

In contrast, one can evaluate the force transmission function, relating the wrench \mathbf{w}_N at the EEF to the internal wrenches $\mathbf{w}_i^T = [\mathbf{m}_i^T, \mathbf{f}_i^T]$ at the intermediate frames K_i , where \mathbf{m}_i represents the moment being applied to the frame K_i from the base-distal subchain to the base-proximal subchain, and \mathbf{f}_i is the corresponding force with respect to the origin. Due to kinetostatic duality, one obtains

$$\mathbf{w}_i = (\mathbf{J}_{i+1} \mathbf{J}_{i+2} \dots \mathbf{J}_N)^T \mathbf{w}_N = \hat{\mathbf{J}}_{i+1}^T \mathbf{w}_N. \quad (14)$$

The force transmission presents the major advantage that one can use \mathbf{w}_{i+1} to determine $\hat{\mathbf{J}}_i$. Therefore, only 6 passes of the force transmission are needed to calculate the complete

Jacobian \mathbf{J}_g . This leads to the following simple algorithm to determine the sensitivity Jacobian \mathbf{J}_g :

1. Solve the forward kinematics for the desired set of joint coordinates \mathbf{q} of the manipulator
2. Choose unit forces $\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z$ and unit torques $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z$ with $|\mathbf{F}_x| = |\mathbf{F}_y| = \dots = |\mathbf{M}_z| = 1$ along the axes x, y, z of the EEF, respectively.
3. For each of the unit loads described above, perform the following steps:
 - a) Apply the load to the EEF and compute the internal forces/torques \mathbf{w}_i for each internal frame \mathbf{K}_i one is interested in.
 - b) Store the internal forces/torques \mathbf{w}_i in the respective row of the Jacobian \mathbf{J}_g .

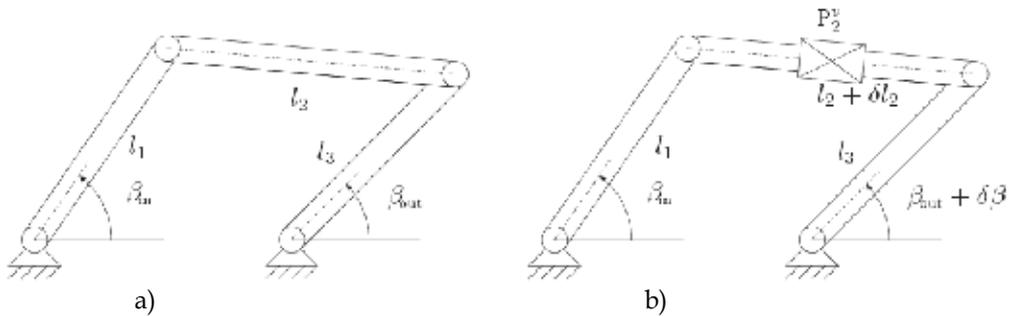


Fig. 6. Planar four-bar mechanism a) with nominal geometry b) with virtual error joint for changes δl_2 in length l_2 of the coupler

2.6 Virtual error joints

The relations in the previous section can be intuitively illustrated by *virtual error joints* which were introduced by Woernle (1988) and extended by Pott and Hiller (2004). The basic idea is to insert additional independent joints which allow motion in the direction of the expected errors. Fig. 6a presents a mechanism that involves a kinematic loop. For this planar four-bar mechanism, one might wish to investigate the influence of changes in length of the coupler. One introduces an additional prismatic joint in which the variation of the length of the coupler is embodied (Fig. 6b) and the influence of changes in length of the coupler can be calculated by using the velocity transmission. The algorithm in Sec. 2.5 can also be derived using this model (Pott & Hiller, 2004), where the virtual error joints are used to measure the back-propagated forces.

Based on the linearization algorithm described above, a number of applications can be investigated, as described next.

3. Applications

3.1 Manufacturing error analysis

The Jacobian matrix \mathbf{J}_g permits to study the influence of geometric errors on the accuracy of the EEF. These errors may arise from the manufacturing and assembly process of the manipulator. They cannot be avoided, but may be controlled through more precise, but at the same time more expensive manufacturing techniques. Consequently, a comprehensive analysis of the influence of changes in parameters is useful for optimal system design. This analysis is basically done evaluating the Jacobian \mathbf{J}_g mapping parameter variations $\delta \mathbf{g}$ to the

displacement $\delta\mathbf{y}$ of the EEF. The magnitude of geometric errors $\delta\mathbf{g}_i$ is normally small compared to the nominal parameter $\delta\mathbf{g}_i$. Therefore, the error estimated by the linear model is very close to the error calculated with the generally nonlinear model (Sec. 4.2).

For different manipulators of the same type, the actual kinematic parameters may vary within the tolerance intervals defined by the manufacturing and assembly process. Here, it is assumed that the actual errors are Gaussian variables with a standard deviation proportional to the tolerance. The changes in parameters are assumed to be small and independent. Thus, the square of the total error is equal to the sum of squares of the single errors obtained by propagation of the manufacturing, clearance, and assembly errors.

For the addition of two Gaussian variables, the standard variation of the sum becomes $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$, where σ_1, σ_2 denote the standard deviations of each summand. In the case of a three-dimensional vector, the total error becomes $\Delta e^2 = e_x^2 + e_y^2 + e_z^2$, where e_x, e_y, e_z are the errors in the three components. For the columns of the Jacobian, one has to mix rotational and translational components. This requires the introduction of metric coefficients that relate rotations to translations and vice versa. Such metric coefficients can be regarded as virtual levers that map rotations at one end to translations at the other and thus generate sensitivities such as “long and slender” (orientations over-emphasized) or “short and thick” (rotations under-emphasized). Assuming that standard deviations σ_i are known for all geometric parameters, the error of the EEF becomes

$$|\Delta\mathbf{e}| = \sqrt{\sum_i \sum_k (\rho_k [\mathbf{J}_g]_{ik} \sigma_i)^2} . \quad (15)$$

where $[\mathbf{J}_g]_{ik}$ denote the entries of the Jacobian and ρ_k are the aforementioned metric coefficients. For the case of an identical standard deviation σ for all components σ_i one obtains

$$|\Delta\mathbf{e}| = \sigma \sqrt{\sum_i \sum_k [\mathbf{J}_g]_{ik}^2} = \sigma \hat{\sigma} . \quad (16)$$

Here, $\hat{\sigma}$ is referred to as the *overall error amplification index* since it estimates the sensitivity of the whole manipulator at a given configuration with respect to geometric errors. This index is similar to the statistical approach to error analysis of Wittwer et al. (2004).

If a certain accuracy is required for a specific task, the maximal error Δe_{\max} of the EEF is known and one can estimate the average standard deviation $\sigma = \Delta e_{\max} / \hat{\sigma}$ that is needed for the geometric parameters. This is illustrated in example Sec. 4.2.

3.2 Stiffness analysis

The linearization of a manipulator with respect to its geometric parameters provides a linear mapping between infinitesimal changes in the geometry and infinitesimal variations of the EEF. Assume \mathbf{J}_g to be the Jacobian transmitting infinitesimal twists $\delta\mathbf{t}_i$ at each of the frames \mathbf{K}_i to infinitesimal twists $\delta\mathbf{t}_{\text{EEF}}$ at the end-effector frame \mathbf{K}_{EEF} . Moreover, denote by $\delta\mathbf{w}_{\text{EEF}}$ a small wrench being applied to the end-effector frame \mathbf{K}_{EEF} and by $\delta\mathbf{w}_i$ the corresponding wrenches at the frames \mathbf{K}_i ensuring static equilibrium. The Jacobian \mathbf{J}_g can be used to set up the stiffness matrix of the mechanism as follows. As pointed out in section 2.2, by equivalence of virtual work it holds

$$\delta\mathbf{w}_g^T \delta\mathbf{t}_g = \delta\mathbf{w}_{\text{EEF}}^T \delta\mathbf{t}_{\text{EEF}} . \quad (17)$$

where $\delta \mathbf{t}_g = [\delta t_{1,1}, \delta t_{1,2}, \dots, \delta t_{N,6}]^T$ collects all virtual variations of the geometric parameters and $\delta \mathbf{w}_g = [\delta w_{1,1}, \delta w_{1,2}, \dots, \delta w_{N,6}]^T$ are the respective internal forces. Here, each $\delta \mathbf{t}_i$ is decomposed in its six elementary components $\delta t_{i,1}, \dots, \delta t_{i,6}$, where $\delta t_{i,1}, \delta t_{i,2}, \delta t_{i,3}$ are elementary infinitesimal rotations and $\delta t_{i,4}, \delta t_{i,5}, \delta t_{i,6}$ are elementary translations with respect to the coordinate frame axis (Fig. 7). Similarly, the wrench $\delta \mathbf{w}_i = [\delta w_{i,1}, \dots, \delta w_{i,6}]^T$ is set up. Assuming that each elementary infinitesimal twist component $\delta t_{i,j}$ produces a corresponding infinitesimal wrench component $\delta w_{i,j}$ by means of an associated linear spring with stiffness coefficient $k_{i,j}$, one obtains

$$\mathbf{K}_g^{-1} \delta \mathbf{w}_g^T = \delta \mathbf{t}_g \quad \text{with} \quad \mathbf{K}_g^{-1} = \text{diag} \left\{ \frac{1}{k_{1,1}}, \frac{1}{k_{1,2}}, \dots, \frac{1}{k_{N,6}} \right\}. \quad (18)$$

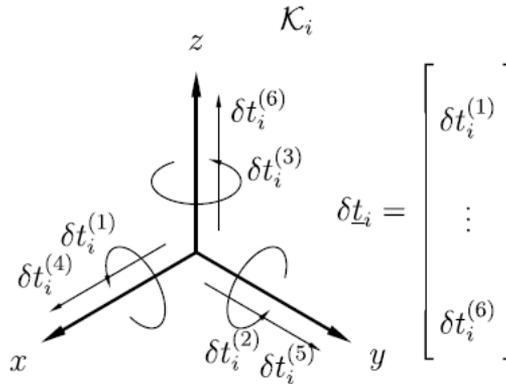


Fig. 7. Decomposition of the unit twist $\delta \mathbf{t}_i$ at frame \mathbf{K}_i

Note that this assumption is a simplification of the structural properties of a general mechanical component that applies to many technical applications (slender bars, joints, etc.). The generalization to a full generic model is accomplished by a stiffness matrix in which all coefficients may be non-zero and which may be obtained from finite element analysis. This generalization is not further pursued here as it bears no new insight and is not required for the examples treated in this paper. A generalization is conceivable as a later step. On the other hand, it holds

$$\mathbf{K}_{\text{EEF}}^{-1} \delta \mathbf{w}_{\text{EEF}} = \delta \mathbf{t}_{\text{EEF}}. \quad (19)$$

where \mathbf{K}_{EEF} is the sought stiffness matrix at the EEF. Substituting Eq. (18) and Eq. (19) into Eq. (17) and using the global force transmission $\delta \mathbf{w}_g = \mathbf{J}_g^T \delta \mathbf{w}_{\text{EEF}}$ gives

$$\delta \mathbf{w}_{\text{EEF}}^T \mathbf{J}_g \mathbf{K}_g^{-1} \mathbf{J}_g^T \delta \mathbf{w}_{\text{EEF}} = \delta \mathbf{w}_{\text{EEF}}^T \mathbf{K}_{\text{EEF}}^{-1} \delta \mathbf{w}_{\text{EEF}}. \quad (20)$$

Since this equation holds for any $\delta \mathbf{w}_{\text{EEF}}$, it follows

$$\mathbf{K}_{\text{EEF}}^{-1} = \mathbf{J}_g \mathbf{K}_g^{-1} \mathbf{J}_g^T. \quad (21)$$

Thus, the Jacobian \mathbf{J}_g can be used to transform the stiffness coefficients $k_{i,j}$ of the geometric parameters contained in the stiffness matrix \mathbf{K}_g to the global stiffness matrix \mathbf{K}_{EEF} .

4. Examples

In this section, the proposed linearization technique is applied to analyze a six-dof parallel kinematic machine where no closed-form solution for the forward kinematics is possible.

4.1 Error analysis for a parallel kinematic manipulator

This example considers the six-dof parallel kinematic machine tool Linapod (Pritschow et al. 2004; Wurst, 1998) installed at the Institute for Control Engineering of Machine Tools and Manufacturing Units at the University Stuttgart (Germany), see Fig. 8. Six rigid links connect the mobile platform to the fixed frame with spherical/universal joints. The pivot points on the frame are actuated by linear drives moving parallel to the z-axis. The nominal position lies in the center of the workspace. Errors in the length of every leg are assumed to be small. Applying the algorithm from section 2.5, the sensitivity matrix \mathbf{J}_g for errors in the length of the bar can be established. To this end, the forward kinematic problem is solved to obtain the position \mathbf{r}_i and the direction \mathbf{u}_i of each of the six legs with respect to the EEF (Fig. 3b). The calculation of the internal forces in the bars from force equilibrium conditions is carried out by applying unit forces and torques (Fig. 8) to the EEF resulting in the matrix equation

$$\underbrace{\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_6 \\ \chi_1 & \dots & \chi_6 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_6 \end{bmatrix}}_{\mathbf{H}} = \underbrace{\begin{bmatrix} \mathbf{F}_x & \mathbf{F}_y & \mathbf{F}_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{M}_x & \mathbf{M}_y & \mathbf{M}_z \end{bmatrix}}_{\mathbf{I}_6}. \quad (22)$$

where $\mathbf{f}_i = f_i \mathbf{u}_i$ are the leg forces, respectively, and $\chi_i = \mathbf{u}_i \times \mathbf{r}_i$. The resulting Jacobian becomes $\mathbf{J}_g^T = \mathbf{A}^{-1}$. With the geometric parameters of Linapod (see Tab.1) the Jacobian \mathbf{J}_g becomes

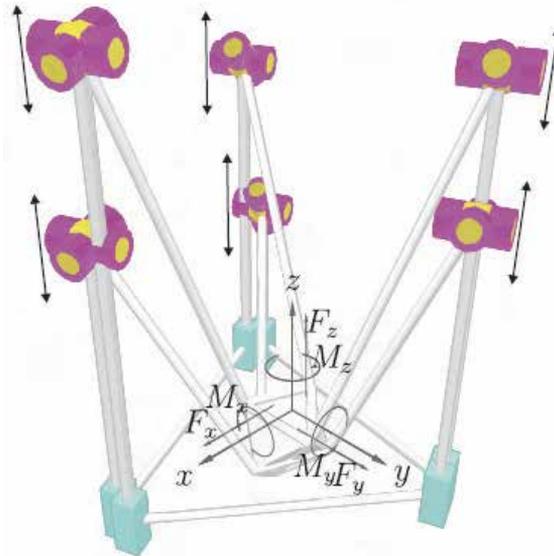


Fig. 8. Six-dof parallel kinematic machine Linapod with fixed length legs. Unit forces and torques are applied to the platform.

$$J_g = \begin{pmatrix} -0.058 & 0.617 & -0.558 & 0.010 & 0.557 & -0.567 \\ -0.678 & 0.289 & 0.390 & -0.649 & 0.333 & 0.316 \\ -0.154 & -0.154 & -0.154 & -0.230 & -0.230 & -0.230 \\ 0.905 & -2.130 & 1.220 & -0.103 & 2.520 & -2.410 \\ 1.930 & -0.181 & -1.750 & -2.840 & 1.330 & 1.510 \\ -2.230 & -2.230 & -2.230 & 2.020 & 2.020 & 2.020 \end{pmatrix}. \quad (23)$$

leg i	\mathbf{a}_i	\mathbf{b}_i	\mathbf{u}_i	l_i	q_i
1	[0.250, 0.886, 0.0]	[-0.126, 0.180, 0.2]	[0,0,1]	1.25	1.221
2	[-0.780, -0.421, 0.0]	[-0.093, -0.199, 0.2]	[0,0,1]	1.25	1.221
3	[0.755, -0.465, 0.0]	[0.219, 0.019, 0.2]	[0,0,1]	1.25	1.221
4	[-0.250, 0.886, 0.0]	[0.115, 0.164, 0.4]	[0,0,1]	1.70	1.933
5	[-0.755, -0.465, 0.0]	[-0.199, 0.017, 0.4]	[0,0,1]	1.70	1.933
6	[0.780, -0.421, 0.0]	[0.085, -0.181, 0.4]	[0,0,1]	1.70	1.933

Table 1. Geometrical Parameters for the PKM Linapod at its home position.

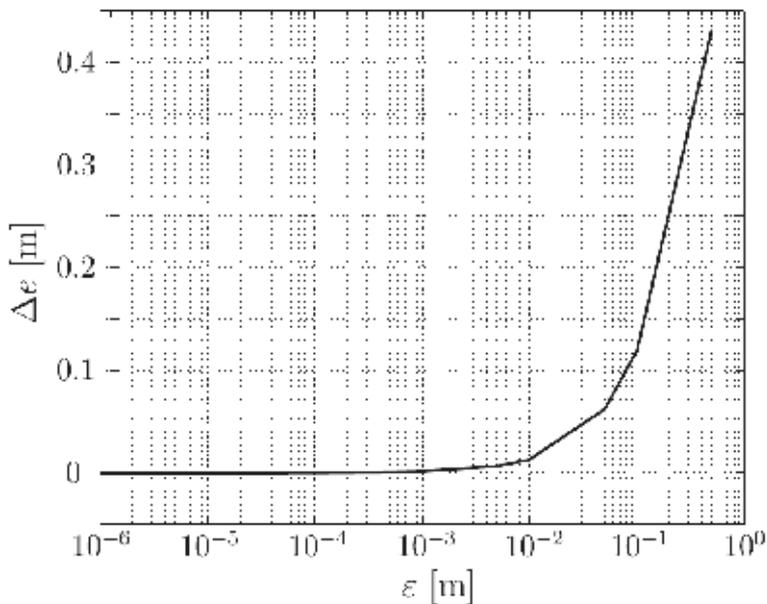


Fig. 9. Difference between discrete error calculation (exact) and linearization.

Assuming that the length error for all bars is $\Delta \mathbf{e} = \varepsilon [1, 1, 1, 1, 1, 1]^T$ with $\varepsilon = 10 \mu\text{m}$, the total position error is $|\Delta \mathbf{e}_{\text{EEF}}| = 11.528 \mu\text{m}$. This matches the exact solution using the nonlinear forward kinematics up to nine digits. In Fig. 9, the effect of variations of the scaling factor on

the difference Δe between linearized and exact model is illustrated. As it can be verified, the approximation is accurate up to a geometric error of about $\varepsilon = 1\text{mm}$. Still, for $\varepsilon = 10\text{mm}$ the relative error is only about 1%, which is still enough for most applications. This shows that the linearization procedure described in this paper is sufficient for most practical applications.

4.2 Accuracy of the Linapod

In this section, the geometric accuracy of the PKM Linapod is analyzed with the force-based method. Assuming errors in every component of the mechanism, the sensitivity matrix J_g contains 126 columns corresponding to the individual geometric parameters. Orientation errors are ignored as these errors are negligible with respect to the translational errors. In Fig. 10 the overall error amplification index according to Eq. (16) is plotted over the workspace. It is recognized from the diagram that the error amplification has its minimum in the center of the workspace, and that the error distribution is roughly circular. It is interesting to observe that changes in the overall error amplification are relatively small from about $\hat{\sigma} = 4.485$ in the center to $\hat{\sigma} = 5$ on the border.

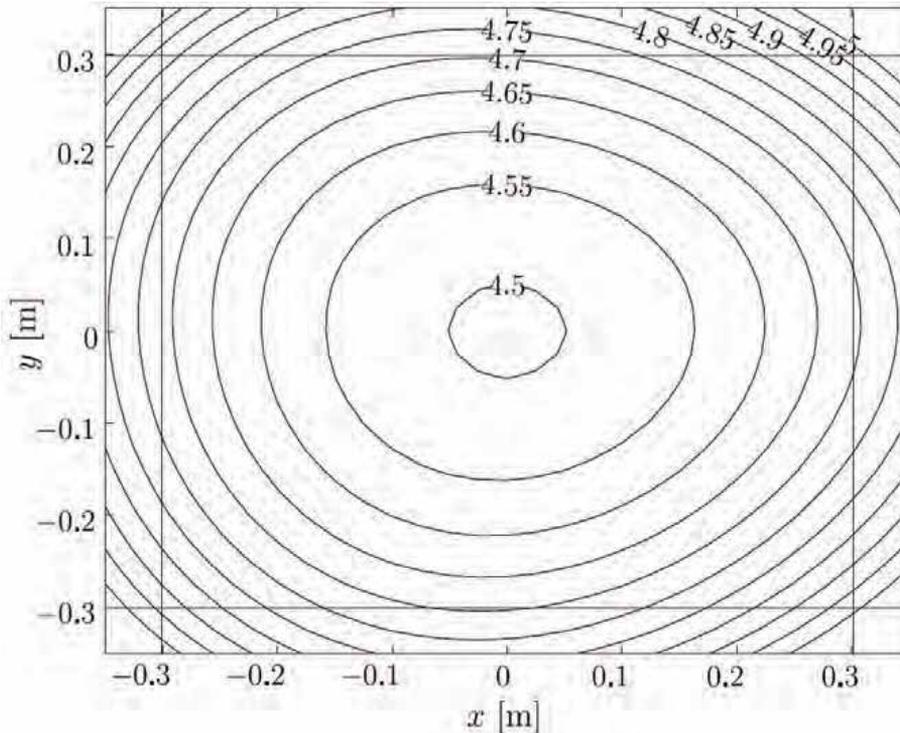


Fig. 10. Overall error-amplification $\hat{\sigma}$ for equally distributed errors in all components. The lines mark the used workspace for Linapod.

Presuming a required accuracy of $\Delta e_{\max} = 10\mu\text{m}$ which is typical for machine tools, this results in an average standard deviation of $\bar{\sigma} = 2\mu\text{m}$ which is essential to reach the given accuracy. One can conclude that it is not possible to manufacture and assemble the machine

with state-of-the-art techniques and reasonable effort at this tolerance level. Therefore, additional steps like calibration are required to ensure the fulfilment of manufacturing requirements.

4.3 Calculation of the stiffness matrix of the parallel robot Linapod

As shown in Sec. 3.2, the Jacobian \mathbf{J}_g can be used for the calculation of the geometric error stiffness matrix. The stiffness coefficients related to elementary geometric variations of a frame are set as $k_l = 8.8e7 \text{ Nm}^{-1}$ for the lower and $k_u = 6.0e7 \text{ Nm}^{-1}$ for the upper leg. Furthermore, elasticity in the linear drives is taken into account with a spring constant $k_d = 8.13e8 \text{ Nm}^{-1}$. For the calculations, only the translational part of the stiffness matrix is taken into account in order to avoid mixing translational and rotational parts. The resulting stiffness behavior of the Linapod is depicted in Fig. 11 by plotting the minimal eigenvalue of the stiffness matrix over the workspace. As it can be seen, the maximum stiffness property is achieved at the home configuration, with softer values farther away of the home configuration.

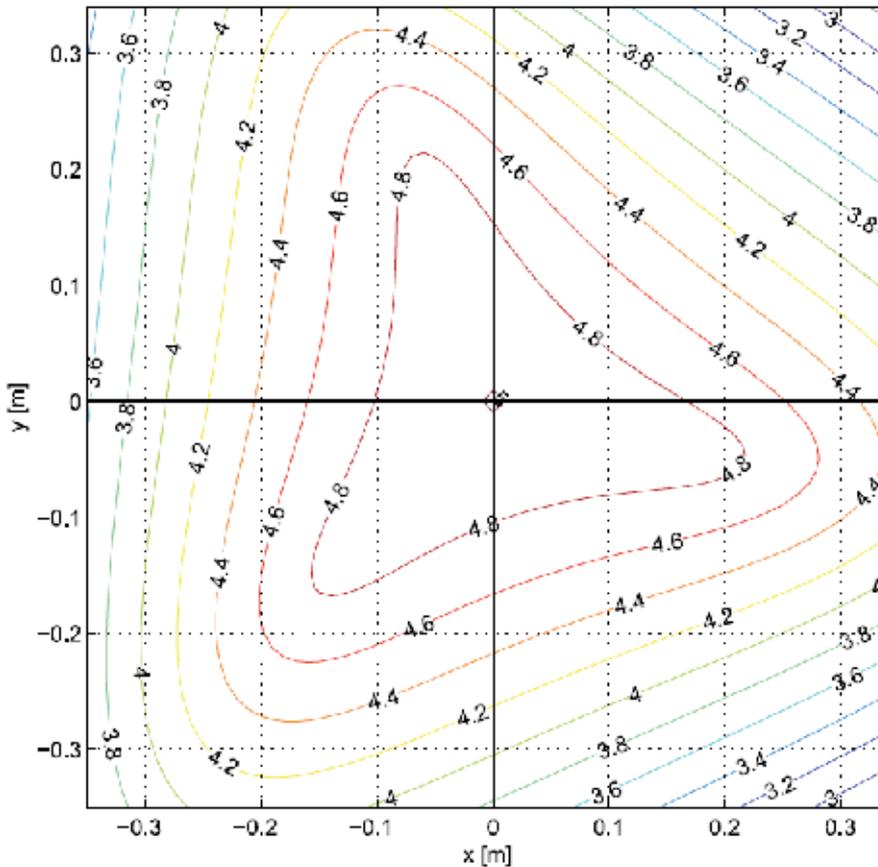


Fig. 11. Minimal eigenvalue λ_{\min} [10^7 Nm^{-1}] of the stiffness matrix of the Linapod.

Algorithm	All parameters		Optimized for Linapod	
	Time (ms)	Relative Time	Time (ms)	Relative Time
numerical differentiation	163.92	68.87	12.31	5.17
velocity-based Jacobian	52.13	21.90	4.33	1.82
force-based Jacobian	2.38	1.00	2.38	1.00

Table 2. Performance of different algorithms implemented in Mobile on an AMD Athlon 1GHz for the error analysis of all 252 parameters for *Linapod*. Relative times compared to force-based Jacobian.

4.4 Computational considerations

In this section, the computational effort of different algorithms to calculate the sensitivity matrix \mathbf{J}_g is compared. The total cost of an algorithm for the error analysis depends on the number of kinematic evaluations, while the administrative overhead e.g. copying and storing the results can be neglected. For the numerical differentiation approach, one needs one evaluation to solve the nominal forward kinematics and one evaluation of the position forward kinematics for each geometric parameter that is considered. The total numerical effort depends on the number of targeted geometric parameters. The velocity-based method (Pott et al., 2007) needs one evaluation of the velocity forward kinematics for each parameter. The force-based approach needs six evaluations of the force transmission. In Tab.2 the computational times of Mobile (Kecskeméthy, 1994) are listed. It can be seen that the numerical differentiation approach needs more time than the velocity-based method, although both need the same number of forward kinematic evaluations. The force-based method needs even less time than the velocity-based method.

5. Conclusions

The contribution describes a general method for kinematic modeling of many wide-spread parallel kinematic machines, i.e. for the Stewart-Gough-platform, the Delta-robot, and Linaglide machines. The kinetostatic method is applied for a comprehensive kinematic analysis of these machines. Based on that model, a general method is proposed to compute the linearization of the transmission behaviour from geometric parameters to the end-effector motion of these machines. By applying the force transmission method, one can perform a linearization with respect to all geometric parameters, for parallel kinematic machines. Especially in cases where no closed-form solution for the forward kinematics is available, the force-based approach provides an efficient procedure for obtaining the linear equations. The method can be directly applied to the presented kinetostatic models of the manipulator and permits also to study parameters that are canceled in the closure conditions. The linear model is used for error analysis and calculation of the stiffness matrix. The algorithm provides a good numerical performance and can be applied to practical examples.

6. Acknowledgment

This work was partly funded by the German Research Foundation (Deutsche Forschungsgemeinschaft) under HI370/19-2 and HI370/19-3 as part of the priority program SPP1099 Parallel Kinematic Machine Tools.

7. References

- Brisan, C.; Franitz, D.; Hiller, M. (2002). Modeling and Analysis of Errors for Parallel Robots, In: *Proceedings of the Kolloquium of SFB 562*, 83-96, Braunschweig, Germany.
- Chase, K. W.; Greenwood, W. H.; Loosli, B. G.; Hauglund, L. F. (1990). Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection. *Manufacturing Review*, Vol. 3, No. 1, 49-59
- Clavel, R. (1988). Delta, a Fast Robot with Parallel Geometry, In: *18th Int. Symp. on Industrial Robot*, 91-100
- Denavit, J.; Hartenberg, R. (1955). A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *Mechanisms Based on Matrices*, Vol. 77
- Dietmaier, P. (1998). The Stewart-Gough Platform of General Geometry can have 40 Real Postures, In: *Advances in Robot Kinematics*, 7-16, Kluwer Academic Publishers, Dordrecht
- El-Khasawneh, B. S.; Ferreira, P. M. (1994). Computation of Stiffness and Stiffness bounds for Parallel Link Manipulators. *International Journal of Machine Tools and Manufacture*, Vol. 39, 321-342
- Hiller, M.; Kecskeméthy, A. (1989). Equations of Motion of Complex Multibody Systems Using Kinematical Differentials, *Transactions of the Canadian Society of Mechanical Engineering*, Vol. 13, No. 4, 113-121
- Husty, M. L. (1996). An Algorithm for Solving the Direct Kinematic of Stewart-Gough-type Platforms. *Mechanism and Machine Theory*, Vol. 31, No. 4, 365-380
- Innocenti, C. (1999). A Static-Based Method to Evaluate the Effect of Joint Clearances on the Positioning Errors of Planar Mechanisms, In: *Tenth World Congress on the Theory of Machines and Mechanisms*, Oulu, Finland
- Innocenti, C. (2002). Kinematic Clearance Sensitivity Analysis of Spatial Structures with Revolute Joints. *Transactions of the ASME*, Vol. 124, 52-57
- Jelenkovic, L. & Budin, L. (2002). Error Analysis of a Stewart Platform based Manipulators, In: *6th International Conference on Intelligent Engineering Systems*, 83-96, Opatija
- Ji, S.; Li, X.; Ma, Y.; Cai, H. (2000). Optimal Tolerance Allocation Based on Fuzzy Comprehensive Evaluation and Genetic Algorithm. *International Journal of Advanced Manufacturing Technology*, Vol. 16, 461-468
- Kecskeméthy, A. (1993). *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen*, Fortschritt-Berichte VDI, Reihe 20, Nr. 88, VDI Verlag, Düsseldorf
- Kecskeméthy, A. (1994). *Mobile - User's Guide and Reference Manual*, Fachgebiet Mechatronik, University Duisburg-Essen

- Kecskeméthy, A.; Hiller, M. (1994). An Object-Oriented Approach For An Effective Formulation of Multibody Dynamics. *Computer Methods in Applied Mechanics and Engineering*, Vol. 115, 287-314
- Kim, H. S. & Choi Y. J. (2000). The Kinematic Error Bound Analysis of the Stewart Platform. *Journal of Robotic Systems*, Vol. 17, No. 1, 63-73
- Lenord, O.; Fang, S.; Franitza, D.; Hiller, M. (2003). Numerical Linearisation Method to Efficiently Optimize the Oscillation Damping of an Interdisciplinary System Model. *Multibody System Dynamics*, Vol. 10, 201-217
- Parenti-Castelli, V.; Venanzi, S. (2002). A New Deterministic Method for Clearance Influence Analysis in Spatial Mechanisms, In: *Proceedings of ASME International Mechanical Engineering Congress*, New Orleans, Louisiana
- Parenti-Castelli, V.; Venanzi, S. (2005). Clearance Influence Analysis on Mechanisms. *Mechanism and Machine Theory*, Vol. 40, No. 12, 1316-1329
- Pott, A.; Hiller, M. (2004). A Force Based Approach to Error Analysis of Parallel Kinematic Mechanisms, In: *Advances in Robot Kinematics*, 293-302, Kluwer Academic Publishers, Dordrecht
- Pott, A. (2007). *Analyse und Synthese von Werkzeugmaschinen mit paralleler Kinematik*, Fortschritt-Berichte VDI, Reihe 20, Nr. 409, VDI Verlag, Düsseldorf
- Pott, A.; Kecskeméthy, A.; Hiller, M. (2007). A Simplified Force-Based Method for the Linearization and Sensitivity Analysis of Complex Manipulation Systems. *Mechanism and Machine Theory*, Vol. 42, No. 11, 1445-1461
- Pritschow, G.; Boye, T.; Franitza, T. (2004). Potentials and Limitations of the Linapod's Basic Kinematic Model, *Proceedings of the 4th Chemnitz Parallel Kinematics Seminar*, 331-345, Verlag Wissenschaftliche Scripten, Chemnitz
- Rebeck, E. & Zhang, G. (1999). A method for evaluating the stiffness of a hexapod machine tool support structure. *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 7, 149-165
- Song, J.; Mou, J.-I.; King, C. (1999). Error Modeling and Compensation for Parallel Kinematic Machines, In: *Parallel Kinematic Machines*, 172-187, Springer-Verlag, London
- Wittwer, J. W.; Chase, K. W.; Howell, L. L. (2004). The Direct Linearization Method Applied to Position Error in Kinematic Linkages. *Mechanism and Machine Theory*, Vol. 39, No. 7, 681-693
- Woernle, C. (1988). *Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter*, Fortschritt-Berichte VDI, Reihe 18, Nr. 18, VDI Verlag, Düsseldorf
- Wurst, K.-H. (1998). Linapod - Machine Tools as Parallel Link System in a Modular Design, *Proceedings of the 1st European-American Forum on Parallel Kinematic Machines*, Milan, Italy

Zhao, J.-W.; Fan, K.-C.; Chang, T.-H.; Li, Z. (2002). Error Analysis of a Serial-Parallel Type Machine Tool. *International Journal of Advanced Manufacturing Technology*, Vol. 19, 174-179

Certified Solving and Synthesis on Modeling of the Kinematics. Problems of Gough-Type Parallel Manipulators with an Exact Algebraic Method

Luc Rolland
*University of Central Lancashire
 United Kingdom**

1. Introduction

The significant advantages of parallel robots over serial manipulators are now well known. However, they still pose a serious challenge when considering their kinematics. This paper covers the state-of-the-art on modeling issues and certified solving of kinematics problems. Parallel manipulator architectures can be divided into two categories: planar and spatial. Firstly, the typical planar parallel manipulator contains three kinematics chains lying on one plane where the resulting end-effector displacements are restricted. The majority of these mechanisms fall into the category of the **3-RPR** generic planar manipulator, [Gosselin 1994, Rolland 2006]. Secondly, the typical spatial parallel manipulator is an hexapod constituted by six kinematics chains and a sensor number corresponding to the actuator number, namely the **6-6** general manipulator, fig. 1.

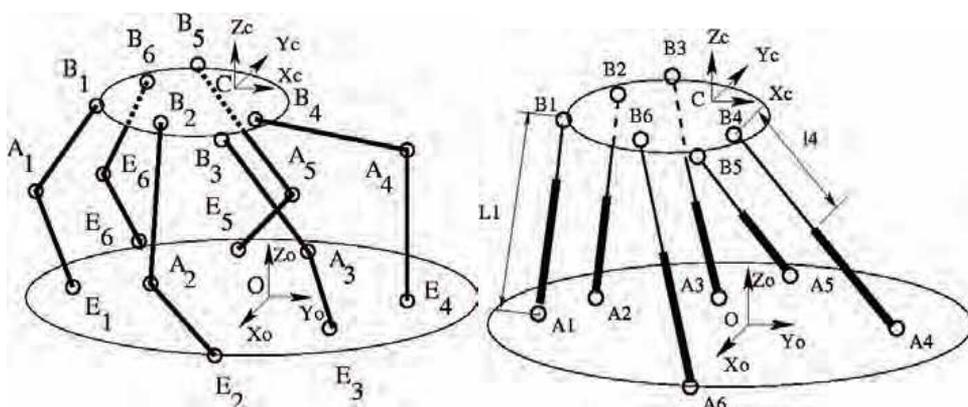


Fig. 1. The general 6-6 hexapod manipulators

Solving the **FKP** of general parallel manipulators was identified as finding the real roots of a system of non-linear equations with a finite number of complex roots. For the **3-RPR**, 8 assembly modes were first counted, [Primerose and Freudenstein 1969]. Hunt geometrically demonstrated that the **3-RPR** could yield 6 assembly modes, [Hunt 1983]. The numeric

iteration methods such as the very popular Newton one were first implemented, [Dieudonne 1972, Merlet 1987, Sugimoto 1987]. They only converge on one real root and the method can even fail to compute it. To compute all the solutions, polynomial equations were justified, [Gosselin and Angeles 1988]. Ronga, Lazard and Mourrain have established that the general **6-6** hexapod **FKP** has 40 complex solutions using respectively Gröbner bases, Chern classes of vector bundles and explicit elimination techniques, [Ronga and Vust 1992, Lazard 1993, Mourrain 1993a]. The continuation method was then applied to find the solutions, [Raghavan 1993], however, it will be explained why they are prone to miss some solutions, [Rolland 2003]. Computer algebra was then selected in order to manipulate exact intermediate results and solve the issue of numeric instabilities related to round-off errors so common with purely numerical methods. Using variable elimination, for the **3-RPR**, 6 complex solutions were calculated [Gosselin 1994] and, for the **6-6**, Husty and Wampler applied resultants to solve the **FKP** with success, [Husty 1996, Wampler 96]. However, resultant or dialytic elimination can add spurious solutions, [Rolland 2003] and it will be demonstrated how these can be hidden in the polynomial leading coefficients. Inasmuch, a sole univariate polynomial cannot be proven equivalent to a complete system of several polynomials. Interval analyses were also implemented with the Newton method to certify results, [Didrit et al. 1998, Merlet 2004]. However, these methods are often plagued by the usual Jacobian inversion problems and thus cannot guarantee to find solutions in all non-singular instances. The geometric iterative method has shown promises, [Petuya et al. 2005], but, as for any other iterative methods, it needs a proper initial guess.

Hence, this justified the implementation of an exact method based on proven variable elimination leading to an equivalent system preserving original system properties. The proposed method uses Gröbner bases and the rational univariate representation, [Faugère 1999, Rouillier 1999, Rouillier and Zimmermann 2001], implementing specific techniques in the specific context of the **FKP**, [Rolland 2005]. Three journal articles have been covering this question for the general planar and spatial manipulators [Rolland 2005, Rolland 2006, Rolland 2007]. This algebraic method will be fully detailed in this chapter.

This document is divided into 3 main topics distributed into five sections. The first part describes the kinematics fundamentals and definitions upon which the exact models are built. The second section details the two models for the inverse kinematics problem, addresses the issue of the kinematics modeling aimed at its adequate algebraic resolution. The third section describes the ten formulations for the forward kinematics problem. They are classified into two families: the displacement based models and position based ones. The fourth section gives a brief description of the theoretical information about the selected exact algebraic method. The method implements proven variable elimination and the algorithms compute two important mathematical objects which shall be described: a Gröbner Basis and the Rational Univariate Representation including a univariate equation. In the fifth section, one **FKP** typical example shall be solved implementing the ten identified kinematics models. Comparing the results, three kinematics models shall be retained. The selected manipulator is a generic **6-6** in a realistic configuration, measured on a real parallel robot prototype constructed from a theoretically singularity-free design. Further computation trials shall be performed on the effective **6-6** and theoretical one to improve response times and result files sizes. Consequently, the effective configuration does not feature the geometric properties specified on the theoretical design. Hence, the **FKP** of theoretical designs shall be studied and their kinematics results compared and analyzed. Moreover, the posture analysis or assembly mode issue shall be covered.

2. Kinematics of parallel manipulator

2.1 Kinematics notations and hypotheses

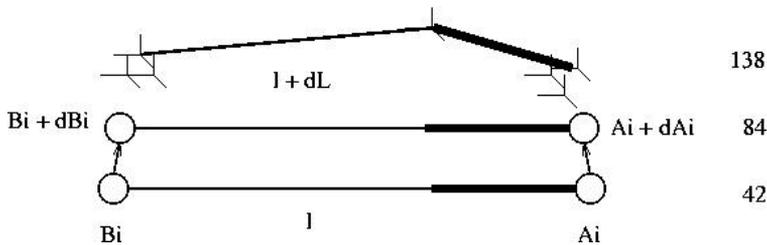


Fig. 2. Typical kinematics chains

The parallel Gough platform, namely **6-6**, is constituted by six kinematics chains, fig. 2. It is characterized by its mechanical configuration parameters and the joint variables. The configuration parameters are thus \mathbf{OA}_{Rf} as the base geometry and \mathbf{CB}_{Rm} as the mobile platform geometry. The joint variables are described as ρ the joint actuator positions (angular or linear). Lets assume rigid kinematics chains, a rigid mobile platform, a rigid base and frictionless ball joints between platforms and kinematics chains.

2.2 Hexapod exact modeling

Stringent applications such as milling or surgery require kinematics models as close as possible to exactness. Realistically, any effective configuration always comprises small but significant manufacturing errors, [Vischer 1996, Patel & Ehmann 1997]. Hence, any constructed parallel manipulator never corresponds to the theoretical one where specific geometric properties may have been chosen, for example, to alleviate singularities or to simplify kinematics solving. Two prismatic actuator axes may be neither collinear nor parallel and may not even intersect. Whilst knowing joints prone to many imperfections, then rotation axes are not intersecting and the angles between them are never perpendicular. Moreover, real ball joints differ from a perfectly circular shape and friction induces unforeseeable joint shape modification, which results into unknown axis changes. However, the joint axis angles stay almost perpendicular and any rotation combination shall be feasible. In a similar fashion, the Cardan joint axes are not perpendicular and may be separated by a small offset. Finally, the articulation center is not crossed by any axis.

Identified the **hexapode 138**, the exact geometric model is then characterized by 138 configuration parameters. Each kinematics chain is described by 23 parameters, as shown on fig. 2 and defined hereafter:

- the 3 parameters of each base joint A_i with their error vector δA_i ,
- the 3 joint A_i inter-axis distances e_{1^a} , e_{2^a} and e_{3^a}
- each prismatic joint measured position l_i with its error coordinate δL_i ,
- the 3 parameters of the minimum distance between the two prismatic actuator axes: \vec{d}_r ,
- the angular deviation between the two prismatic actuator axes: φ ,
- the 3 parameters of the platform joint B_i with their error vector δB_i ,
- the 3 joint B_i inter-axis distances and e_{1^b} , e_{2^b} and e_{3^b}

To solve this model includes the determination of parameters which cannot be measured neither determined. Moreover, the model includes more variables than equations and therefore, its resolution would then only be possible through optimization methods. Relying on a calibration procedure would only determine configuration parameters by specifying an error margin consisting of a radius around joint positions and would not indicate the direction of the error vector. Hence, only an error ball becomes applicable to the model. In practice, the δA_i and δB_i joint error vectors shall reposition the respective kinematics chains by adding an offset to the joint centers. Thus, a random function shall compute the δA_i and δB_i vectors with the maximum being the error ball radius. Finally, the selected model, namely the *hexapod 84*, is effectively based on the *hexapod 42* model with errors added to the configuration data and joint variables.

2.3 Kinematics problems

Definition 2.1 *The kinematics model is an implicit relation between the configuration parameters and the posture variables, $F(\vec{X}, \Gamma, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})=0$ where $\Gamma = \{\rho_1, \rho_2, \dots, \rho_6\}$.*

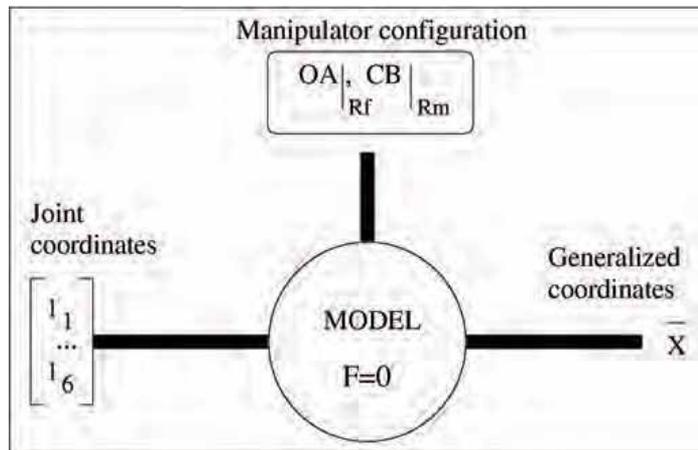


Fig. 3. Kinematics model

Three problems can be derived from the above relation: the forward kinematics problem (FKP), the inverse kinematics problem (IKP) and the kinematics calibration problem, fig. 3. The two first problems shall be covered in this article. The inverse kinematics problem (IKP) is defined as:

Definition 2.2 *Given the generalized coordinates of the manipulator end-effector, find the joint positions.*

The 6-6 IKP yields explicit solutions from vector $\Gamma = G(\vec{X}, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})$ and is used to prepare the FKP which is defined as:

Definition 2.3 *Given the joint positions Γ , find the generalized coordinates \vec{X} of the manipulator end-effector.*

The 6-6 FKP is a difficult problem, [Merlet 1994, Raghavan and Roth 1995] and explicit solutions $\vec{X} = G(\Gamma, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})$ have not yet been established. The difficulties in solving the FKP have hampered the application of parallel robot in the milling industry.

2.4 Vectorial formulation of the basic kinematics model

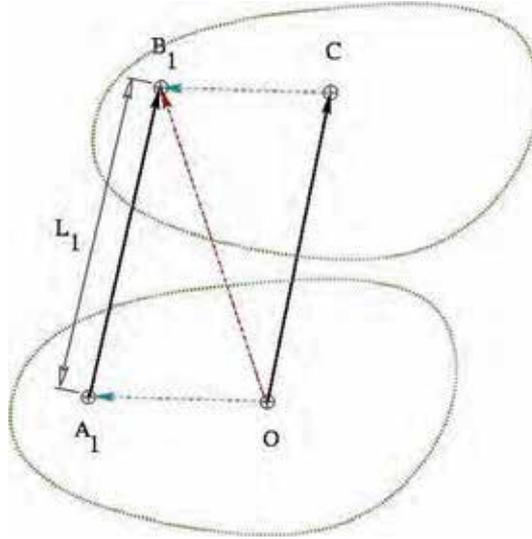


Fig 4. The vectorial formulation

The vectorial formulation produces an equation system which contains the same number of equations as the number of variables, fig. (4), [Dieudonne et al. 1972]. A closed vector cycle is constituted between the manipulator characteristic points: A_i and B_i , kinematics chain attachment points, O the fixed base reference frame and C the mobile platform reference frame. For each kinematics chain, a function between points A_i and B_i expresses the generalized coordinates X , such as $\overline{A_i B_i} = U_1(X)$. Inasmuch, vector $\overline{A_i B_i}$ is determined with the joint coordinates Γ and X giving a function $U_2(X, \Gamma)$. Finally, the following equality has to be solved: $U_1(X) = U_2(X, \Gamma)$.

3. The inverse kinematics problem

For each kinematics chain, $i = 1, \dots, 6$, each platform point $\overline{OB_{i|R_f}}$ can be expressed in terms of the distance constraint, [Merlet 1997]:

$$l_i^2 = \|\overline{A_i B_i}\|^2, i = 1 \dots 6 \quad (1)$$

Using the vectorial formulation, two equation families can be derived: displacement-based and position-based equations.

3.1 Displacement based equations

Any mobile platform position $\overline{OB_{i|R_f}}$ which meets constraints 1 has a rotation matrix \mathfrak{R} such that:

$$\overline{OB_{i|R_f}} = \overline{OC_{i|R_f}} + \mathfrak{R} \cdot \overline{CB_{i|R_m}}, i = 1 \dots 6 \quad (2)$$

Substituting 2 in 1, we obtain:

$$l_i^2 = \left\| \overline{OC}_{|R_f} + \mathfrak{R} \cdot \overline{CB}_{i|R_m} - \overline{OA}_{i|R_f} \right\|^2, i = 1 \dots 6 \quad (3)$$

This last equation system can be developed and simplified, leading to the **IKP** :

$$l_i^2 = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathfrak{R} \cdot \overline{CB}_{i|R_m} + \overline{CB}_i^2 \quad (4)$$

3.2 Position based equations

In 3D space, any rigid body can be positioned by 3 of its distinct non-colinear points, [Fischer and Daniel 1992, Lazard 1992b]. The 3 mobile platform distinct points are usually selected as the 3 joint centers B_1, B_2, B_3 , fig. 5. The 6 variables are set as: $\overline{OB}_{i|R_f} = [x_i, y_i, z_i]$ for $i = 1 \dots 3$. The $\overline{OB}_{i|R_f}$ parameters define the reference frame R_{b1} relative to the mobile platform and B_1 is chosen as its center. The frame axes u_1, u_2 and u_3 are determined by the 3 platform points:

$$u_1 = \frac{\overline{B_1 B_2}}{\|\overline{B_1 B_2}\|}, u_2 = \frac{\overline{B_1 B_3}}{\|\overline{B_1 B_3}\|}, u_3 = u_1 \wedge u_2 \quad (5)$$

Any platform point M can be expressed by $\overline{B_1 M} = a_M u_1 + b_M u_2 + c_M u_3$ where a_M, b_M, c_M are constants in terms of these three points. Hence, in the case of the **IKP**, the constants are noted $a_{B_i}, b_{B_i}, c_{B_i}$, $i = 1 \dots 6$ and can explicitly be deduced from $\overline{CB}_{i|R_m}$ by solving the following linear system of equations:

$$\overline{B_1 B}_{i|R_{b1}} = a_{B_i} u_1 + b_{B_i} u_2 + c_{B_i} u_3, i = 1 \dots 6 \quad (6)$$

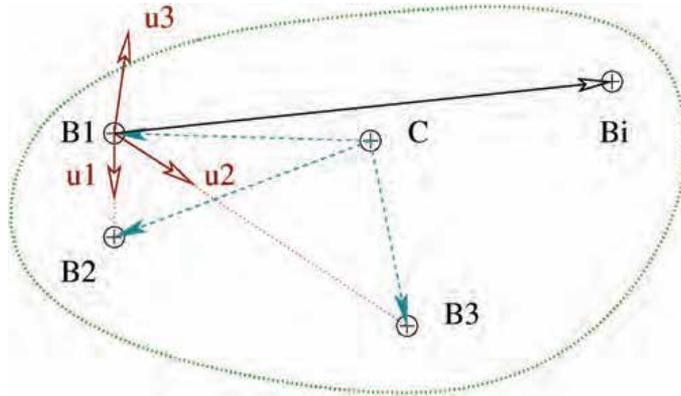


Fig. 5. The platform three point coordinate system

Substituting relations 6 in the distance equations $l_i^2 = \|\overline{A_i B_i}|_{R_f}\|, i = 1 \dots 6$, the system can be expressed with respect to the variables $x_i, y_i, z_i, i = 1, 2, 3$. Thus, for $i = 1 \dots 6$, the **IKP** is obtained by isolating the ρ_i or l_i linear actuator variables in the six following equations:

$$l_i^2 = (x_i - \overline{OA_{ix}})^2 + (y_i - \overline{OA_{iy}})^2, i = 1 \dots 3 \quad (7)$$

$$l_i^2 = \|\overline{B_{k|R_{b_1}} - \overline{OA_{k|R_f}}}\|^2, i = 4 \dots 6 \quad (8)$$

4. The forward kinematics problem

4.1 Displacement based equations

There exist various formulations of the displacement based equation models.

4.1.1 AFD1 - formulation with the position and the trigonometry identity

The AFD1 formulation is obtained by replacing each trigonometric function of the **IKP** rotation matrix, 2, by one distinct variable, [Merlet 1987], for $j = 1, 2, 3$, then $c_j = \cos(\theta_j)$, $s_j = \sin(\theta_j)$. The end-effector position variables are retained. The 9 unknowns are then: $\{x_c, y_c, z_c, c_1, c_2, c_3, s_1, s_2, s_3\}$. The orientation variables can either be any Euler angles or the navigation ones (pitch, yaw and roll). The orientation variables are linked by the 3 trigonometric identities, for $j = 1 \dots 3$, then $c_j^2 + s_j^2 = 1$ which complete the equation system:

$$F_i = (\overline{OC}|_{R_f} - \overline{OA_{i|R_f}})^2 + (\overline{OC}|_{R_f} - \overline{OA_{i|R_f}}) \Re \cdot \overline{CB_{i|R_m}} - L_i^2, i = 1, \dots, 6 \quad (9)$$

$$F_j = c_j^2 + s_j^2 - 1, j = 1, 2, 3 \quad (10)$$

The system is constituted of 9 equations with 6 polynomials of degree 6 and 3 quadratics. The model is simply build by variable substitution without any computation. Thus, the coefficients remain unchanged. The number of variables is not minimal.

4.1.2 AFD2 - formulation with the position and the trigonometric function change

The end-effector position variables are retained. Rotation variable changes can apply the following trigonometric relations, [Griffis & Duffy 1989, Parenti-Castelli & C. Innocenti 1990, Lazard 1993]. For $i = 1, 2, 3$:

$$t_i = \tan\left(\frac{\theta_i}{2}\right), \sin(\theta_i) = \frac{2t_i}{(1+t_i^2)}, \cos(\theta_i) = \frac{(1-t_i^2)}{(1+t_i^2)} \quad (11)$$

The 6 variables become $\{x_c, y_c, z_c, t_1, t_2, t_3\}$. The **IKP** equations (2) are rewritten to obtain the 6 following equations:

$$F_i = (\overline{OC}|_{R_f} - \overline{OA_{i|R_f}})^2 + (\overline{OC}|_{R_f} - \overline{OA_{i|R_f}}) \Re \cdot \overline{CB_{i|R_m}} - L_i^2, i = 1, \dots, 6 \quad (12)$$

The final equation system comprises 6 equations of order 8 with the high degree monomial being $x_i^2 x_j^2 x_k^2 x_l^2$. This model has a minimal variable number. The polynomials coefficients

are expanding due to variable change computation. Moreover, this representation is not intuitive.

4.1.3 AFD3 - formulation with the translation and rotation matrix

The intuitive way to set an algebraic equation system from the **IKP** equations 2 is to straightforwardly use all the rotation matrix parameters and the vector $\overline{OC}_{|R_f}$ coordinates as unknowns, [Lazard 1993, Sreenivasan et al. 1994, Bruyninckx and DeSchutter 1996]. The variables are then $\{X_c, Y_c, Z_c, r_{ij}, j=1\dots 3, i=1\dots 3\}$. Since \mathcal{R} is a rotation matrix, the following relations hold: $\mathcal{R}^T \mathcal{R} = Id$ or $det(\mathcal{R}) = 1$. These relations are redundant since $\mathcal{R}^T \mathcal{R}$ is symmetrical and they generate the 7 following equations:

$$\begin{cases} 1 = r_{11}^2 + r_{12}^2 + r_{13}^2, 1 = r_{21}^2 + r_{22}^2 + r_{23}^2, 1 = r_{31}^2 + r_{32}^2 + r_{33}^2 \\ 0 = r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23}, 0 = r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33}, 0 = r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} \\ 1 = r_{11}r_{22}r_{33} - r_{11}r_{23}r_{32} - r_{21}r_{12}r_{33} + r_{21}r_{13}r_{32} + r_{31}r_{12}r_{23} - r_{31}r_{13}r_{22} \end{cases} \quad (13)$$

Six rotation matrix constraints are then selected and preferably with the lowest degree polynomials. This leads to an algebraic system with 12 polynomial equations (13 and 1) in 12 unknowns.

$$F_i = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathcal{R} \cdot \overline{CB}_{i|R_m} - L_i^2, i = 1, \dots, 6 \quad (14)$$

$$F_7 = r_{11}^2 + r_{12}^2 + r_{13}^2 - 1 \quad (15)$$

$$F_8 = r_{21}^2 + r_{22}^2 + r_{23}^2 - 1 \quad (16)$$

$$F_9 = r_{31}^2 + r_{32}^2 + r_{33}^2 - 1 \quad (17)$$

$$F_{10} = r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23} \quad (18)$$

$$F_{11} = r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33} \quad (19)$$

$$F_{12} = r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} \quad (20)$$

Finally, the model polynomials are quadratic and minimal. They are obtained by substitution and no computations are required. The coefficients are then unchanged. There is a very large number of variables.

4.1.4 AFD4 - formulation with the translation and Gröbner Basis on the rotation matrix

The rotation matrix constraints are not depending on the end-effector position variables. Hence, if one *Gröbner Basis* is computed from the rotation constraints, the *Gröbner Basis* is also independent of the position variables and thus constant for any **FKP** pose. Therefore, one preliminary *Gröbner Basis* can be calculated and saved into a file for later reuse.

Hence, the rotation matrix constraints in the system 20 can be replaced by their *Gröbner Basis* comprising 24 equations where the coefficients are only unity. Thus, the algebraic system involves 30 equations and 12 variables.

4.1.5 AFD5 - translation and quaternion algebraic model

Based on equation (2), quaternions can express mobile platform rotation, [Lazard 1993, Mourrain 1993b, Egner 1996, Murray et al. 1997]. The quaternion representation includes 4 variables $\{q_0; q_1; q_2; q_3\}$ where the vector $\bar{q} = q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$ defines the platform specific rotation axis and $q_0 = \cos(\alpha/2)$ determines the coordinate expressing the rotation α along that axis. Thus, the rotation matrix \mathfrak{R} used in equations 4 may then be expressed in terms of the quaternion coordinates and with $\Delta^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$, we can write:

$$\mathfrak{R} = \Delta^{-2} \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (21)$$

The end-effector position variables are retained. Moreover, one may implement a unitary quaternion: $\Delta^2 = 1$. Rewriting the IKP equations 4, we obtain 7 polynomial equations in the 7 unknowns $\{X_c; Y_c; Z_c; q_0; q_1; q_2; q_3\}$:

$$F_i = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathfrak{R} \cdot \overline{CB}_{i|R_m} - L_i^2, i = 1, \dots, 6 \quad (22)$$

$$F_7 = q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 \quad (23)$$

The system contains 6 polynomials of degree 6 and 1 quadratic. The highest degree monomial is $x_i^2 x_j^2$. The quaternion has intrinsic coordinate redundancy which allows avoiding typical mathematical singularities seen in other representations. The number of variable is almost minimal. The rotation matrix system must be recomputed leading to larger resulting polynomial coefficients.

4.1.6 AFD6 - translation and dual quaternion algebraic model

Not only orientations can be formulated using quaternions, but also positions, [Husty 1996, Wampler 96]. The \mathfrak{R} rotation matrix is then expressed in terms of the first quaternion $\Phi = \{c_0; c_1; c_2; c_3\}$. In a sense, the second $\Psi = \{g_1; g_2; g_3; g_4\}$ represents the end-effector position. Moreover, one relation can be written between the two quaternions: $\Phi = \overline{OC} \Psi$. This relation unfolds in the following equations from which two constraint equations, noted $FC_1 = 0$ and $FC_2 = 0$, are selected. Lets $s_i = \mathbf{OA}_{|R_f}$ and $t_i = \mathbf{CB}_{|R_m}$, then:

$$\begin{aligned} \bar{c}^t \bar{c} &= 1 \\ \bar{g}^t \bar{c} &= 0 \\ \bar{g}^t \bar{g} - l_1 \bar{c}^t \bar{c} &= 0 \quad \text{For } i = 2, \dots, 6 \\ \bar{c}^t s_i \bar{c} + 2 \bar{g}^t t_i \bar{c} &= 0 \end{aligned} \quad (24)$$

The dual quaternion system is thus constituted by the 8 following equations, for $i = 1 \dots 6$:

$$F_i = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathfrak{R} \cdot \overline{CB}_{i|R_m} - L_i^2 \quad (25)$$

$$F_7 = FC_1 \quad (26)$$

$$F_8 = FC_2 \quad (27)$$

The system comprises 6 polynomials of degree 4 and 2 quadratics. The highest degree monomials are either x_i^4 , $x_i^3 x_j$ or $x_i^2 x_j^2$. One more variable is added over the former quaternion model. The variable choice is not intuitive.

4.2 Position based equations

We shall examine four formulations derived from the position based equations. Every variable has the same units and their range is equivalent.

4.2.1 AFP1 - three point model with platform dimensional constraints

The 3 platform distinct points are usually selected as the three joint centers B_1 , B_2 and B_3 , fig. 5. The 6 variables are set as: $\overline{OB}_{i|R_f} = [x_i, y_i, z_i]$ for $i = 1 \dots 3$.

Using the relations 6, the constraint equations $L_i^2 = \|\overline{A_i B_{i|R_f}}\|^2$, $i = 1, \dots, 6$ can be expressed with respect to the variables x_i, y_i, z_i , $i = 1, 2, 3$. Together with equations 30, they define an algebraic system with 9 equations in 9 unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. The resulting kinematics chain system becomes:

$$F_i = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (z_i - OA_{iz})^2 - L_i^2, i = 1 \dots 3 \quad (28)$$

$$F_j = \|\overline{B_{j|R_m}} - \overline{OA_{j|R_f}}\|^2 - L_j^2, j = 4 \dots 6 \quad (29)$$

The mobile platform geometry yields the following three distance equations:

$$\begin{aligned} F_7 &= \|\overline{B_2 B_{1|R_f}}\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \|\overline{B_2 B_{1|R_m}}\|^2 \\ F_8 &= \|\overline{B_3 B_{1|R_f}}\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \|\overline{B_3 B_{1|R_m}}\|^2 \\ F_9 &= \|\overline{B_3 B_{2|R_f}}\|^2 - (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 = \|\overline{B_3 B_{2|R_m}}\|^2 \end{aligned} \quad (30)$$

Together with equations 30, they produce an algebraic system with 9 equations with 9 unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. In all instances, it can be easily proven that this **6-6 FKP** formulation yields 9 quadratic polynomials.

The system variable choice is relatively intuitive. Each equation polynomial is always quadratic. However, the b_1 reference frame and the platform points B_i in the b_1 frame require computations, which usually result into coefficient size explosion. The variable number is not minimal.

4.2.2 AFP2 - the three point model with platform constraints

The former system can be slightly modified by replacing the last mobile platform constraint with a platform normal vector one. Hence, let's take the two mobile platform vectors $\overline{B_1 B_2}$ and $\overline{B_1 B_3}$, then the last constraint is calculated from these two vector multiplication:

$$\begin{aligned}
 F_7 &= \left\| \overrightarrow{B_2 B_{1|R_f}} \right\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|^2 \\
 F_8 &= \left\| \overrightarrow{B_3 B_{1|R_f}} \right\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2 \\
 F_9 &= (x_3 - x_1) * (x_2 - x_1) + (y_3 - y_1) * (y_2 - y_1) + (z_3 - z_1) * (z_2 - z_1) - \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|
 \end{aligned} \tag{31}$$

The result is still an algebraic system with nine equations in the former nine unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. The **6-6 FKP** formulation using this three point model is constituted by nine quadratic polynomials.

4.2.3 AFP3 - the three point model with constraints and function recombination

By rewriting the **IKP** as functions, the algebraic system comprises three equations and three functions in terms of the nine variables: $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$, equation (29).

$$F_i = (x_i - OA_x)^2 + (y_i - OA_y)^2 - l_i^2, i = 1 \dots 3 \tag{32}$$

$$C_i = \left\| \overrightarrow{B_{k|R_{b_1}}} - \overrightarrow{OA_{k|R_f}} \right\|^2 - l_i^2, i = 4 \dots 6 \tag{33}$$

Hence, three constraints are derived from the following three functions, [Faugère and Lazard 1995]. Two functions can be written using two characteristic platform vector norms between the B_1, B_2 distinct points and the B_1, B_3 ones. The last function comes from these vector multiplication.

$$\begin{aligned}
 F_7 &= \left\| \overrightarrow{B_2 B_{1|R_f}} \right\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|^2 \\
 F_8 &= \left\| \overrightarrow{B_3 B_{1|R_f}} \right\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2 \\
 F_9 &= (x_3 - x_1) * (x_2 - x_1) + (y_3 - y_1) * (y_2 - y_1) + (z_3 - z_1) * (z_2 - z_1) - \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|
 \end{aligned} \tag{34}$$

Furthermore, the three last equations (F_7, F_8, F_9) are computed by the following function sequential combinations:

$$\begin{aligned}
 F_7 &= -C_7 + F_1 + F_2 \\
 F_8 &= -C_8 + F_1 + F_3
 \end{aligned} \tag{35}$$

$$F_9 = 2 * C_9 + F_7 + F_8 - 2 * F_1$$

The formulation is completed with other function combinations obtained by the following algorithm leading to three middle equations (F_4, F_5, F_6). Let $d_7 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|$, $d_8 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2$ and $d_9 = \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|$, then for $i = 4, 5, 6$, we compute:

$$\begin{aligned}
 C_i &= C_i - a_{B_i}^2 * C_7 - b_{B_i}^2 * C_8 - c_{B_i}^2 * (C_7 * C_8 - C_9^2) - a_{B_i} * b_{B_i} * (2 * C_9) \\
 C_i &= 2 * C_i - a_{B_i} * (F_7 - 2 * F_1) - b_{B_i} * (F_1 + F_2 - F_7) \\
 F_i &= C_i - 2 * c_{B_i}^2 * d_7 * (F_1 + F_2 - F_8) - 2 * c_{B_i}^2 * d_8 * (F_1 + F_2 - F_7) \\
 &\quad - 2 * c_{B_i}^2 * d_9 * (F_7 - F_9 + F_8 - 2 * F_1) + 2 * (a_{B_i} + b_{B_i} - 1) * F_1 - F_7 * a_{B_i} - F_8 * b_{B_i}
 \end{aligned} \tag{36}$$

The result is an algebraic system with nine equations with the nine unknowns. The **6-6 FKP** formulation using this modified three point model includes six quadratic and three quartic polynomials. The system includes polynomials of higher degree than for the former two position based models. Computations cause to coefficient expansion.

4.2.4 AFP4 - the six point model

The six mobile platform B_i joints can be used in defining 18 variables, [Rolland 2003]. Taking the **IKP** equations (8), a position based variation is obtained:

$$l_i^2 = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (z_i - OA_{iz})^2, i = 1 \dots 6 \quad (37)$$

The system is completed with 12 distance constraint equations selected among the distinct B_i passive platform joints. Here are some examples:

$$\begin{aligned} \left\| \overrightarrow{B_i B_{1R_f}} \right\|^2 &= (x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 = \left\| \overrightarrow{B_i B_{1R_m}} \right\|^2, i = 1 \dots 6 \\ \left\| \overrightarrow{B_j B_{2R_f}} \right\|^2 &= (x_j - x_2)^2 + (y_j - y_2)^2 + (z_j - z_2)^2 = \left\| \overrightarrow{B_j B_{2R_m}} \right\|^2, j = 3 \dots 6 \quad (38) \\ \left\| \overrightarrow{B_k B_{3R_f}} \right\|^2 &= (x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2 = \left\| \overrightarrow{B_k B_{3R_m}} \right\|^2, k = 4 \dots 6 \end{aligned}$$

The formulation results in 18 polynomials in the 18 unknowns:

$\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_5, y_5, z_5, x_6, y_6, z_6\}$. The system is then constituted of quadratic polynomials. This variable choice is intuitive and the system yields minimal degree. Finally, the number of variables is maximal.

5. Solving polynomial systems using exact computation

5.1 Mathematical system solving

Kinematics problems contain systems of several equations containing non-linear functions with various variable numbers. These systems can be difficult to solve, especially in the general **6-6** cases and response times actually makes them inappropriate for implementations in design, simulation or control. In some instances, the results may appear to be faulty bringing doubts to the reliability of the methods.

If left without any reliable and performing methods, the tendency, in engineering practice, would be to convert the difficult models into simpler linearized ones. In material handling, this proposal might suffice, however, in high speed milling where the accuracy requirements are more severe, any simplification can have a dramatic impact, whereby result certification becomes an important issue.

However, with proper polynomial formulation, algebraic methods can lead to at least certified and even exact results, whereas numeric methods, unless they implement proper interval analysis, cannot actually obtain certified results since they are prone to numeric instabilities or matrix inversion problems. Therefore, although time consuming, algebraic methods are preferred since they handle integer, rational and symbolic values as such without any truncation or approximation, even when manipulating intermediate results. Hence, there will be no loss of information.

Solving non-linear equation systems will usually result in several complex solutions, out of which a certain subset are real solutions. However, only the real solutions bear practical significance, since they correspond to effective manipulator poses.

5.2 Calculation accuracies

The calculation accuracies are depending upon the type of arithmetic, the behavior of the calculation methods and the quality of the implemented algorithms.

Definition 5.1 An exact calculation is defined as a calculation which always produces the same exact result to the same specific mathematical problem.

The result does not contain any error. Its representation is also exact.

Definition 5.2 A reliable computation is defined as one which will always give the same result from the same initial input data presented in the same format.

Definition 5.3 A certified calculation is defined as a reliable computation giving a result distant from the true solution by a certain maximum known accuracy.

Hence, such a calculation may not be exact. However, the result contains some exact digits. Hence, we shall try to apply a method that computes certified results and if possible exact ones.

For example, lets take the univariate function $f_1(x) = x_2 - 4/25$. Computing $f_1 = 0$, we obtain the exact response: $\{-2/5, 2/5\}$. The closed-form resolution calculates exact results with rational numbers. Therefore, the result is certified without any error.

Lets consider $f_2(x) = x_2 - 5$. Solving $f_2 = 0$, the result will be two irrational numbers which can only be represented by truncation. However an interval can be certified to contain the exact result: $\{[2, 5/2], [-5/2, 2]\}$. Wherefore, exact computations keep intermediate results in symbolic format whenever possible and only revert to rational or floating boundary numbers for display purposes.

Therefore, any real number can be coded by an interval which width corresponds to the required accuracy. However, the difficulty lies in insuring that the interval contains the exact result which is not known a priori.

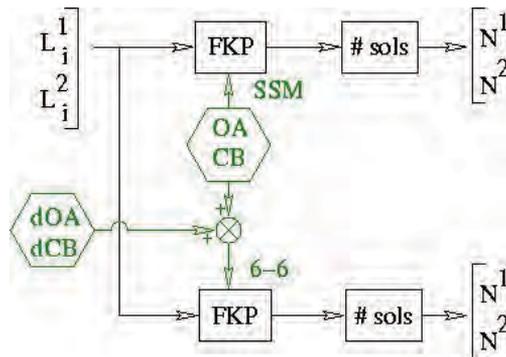


Fig. 6. Bloc Diagram of the Continuation Method

5.3 Solving a non-linear system

Two method groups have been advocated to find all solutions of the FKP, namely: continuation methods and variable elimination ones, [Raghavan and Roth 1995]. The first approach is usually realized in a numeric environment and the later algebraic.

5.4 Continuation method with homothopy

In order to compute several solutions, the continuation method can be implemented with a homothopy process. The Continuation approach implements a numerical iterative method which is successively repeated in order to progressively transfer from an original equation system which solutions are predetermined to another system relatively close to the former, Fig. 6.

Let a system of equation be $F(X) = 0$ with variables $X = \{x_1, \dots, x_n\}$; we wish to find the solution to this equation system. Let $G(X) = 0$ be a similar equation system which roots are already known, namely the variety $V_{\mathbf{r}}(I)^G$; then, we set the continuation process as $H(X, \lambda) = G(X) + \lambda (F(X) - G(X))$ and commence with $\lambda = 0$. It provides for a mechanism to convert an original equation system into a final one through several steps. At each step, $H(X, \lambda)$ is successively computed with a new value of λ which is increased by a small arbitrary increment $\delta\lambda$ such that $\lambda \in \{0, \dots, 1\}$. The homothopy principle assumes connectivity between solutions of each system computed with the various λ . More generally, if the system $H(X, \lambda)$ with λ as a variable would be solved, it would result in paths as solutions.

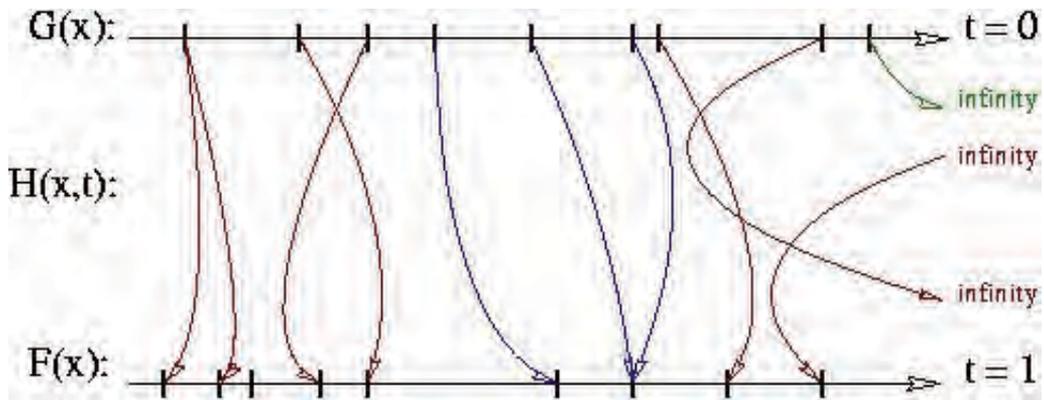


Fig. 7. Examples of path following with the Continuation Method

The continuation method cannot solve any equation system as such and, at each step, when λ is instantiated, the $H(X, \lambda)$ system roots are computed by a typical iterative method, either the Newton one or the new *Geometric Iterative Method* which is a potentially good alternative, [Petuya et al. 2005].

This method was first applied to classical robotics kinematics, [Tsai & Morgan 1984] and then applied to solve the parallel manipulator **FKP**, [Kholi et al. 1992, Sreenivasan and P. Nanua 1992].

Advocating that a little change on parameters of one system shall cause only a small change on solutions, the continuation method could be used to find the 40 solutions on some **6-6 FKP** cases, [Raghavan 1993].

It is feasible to construct an efficient and reliable method; however, the method is still unproven. Moreover, continuation does not alleviate the problems related to the application of a numeric iterative method.

This method can be somewhat delicate to implement. There exist several scenarios which might pose significant problems depending on how the solution paths evolve from $\lambda = 0$ to $\lambda = 1$, see fig. 7, where solutions:

- go to or come from infinity,
- merge or split,
- start complex and become real,
- start real and become complex.

Therefore, proper implementation would require a priori continuation process verification which is still an open question, since this would require solving a one-dimensional system $H(X, \lambda)$ where λ is left as a variable and this is an even more difficult problem. Inasmuch, in many instances, finding a nearby equation system with known roots may not be always be feasible. Then, there is also an issue on what constitutes a sufficiently similar system.

However, it is very difficult to determine precisely what the meaning of sufficiently similar is.

5.5 Variable elimination

5.5.1 Introduction

Most algebraic methods which were implemented to solve the parallel manipulator **FKP** apply one form of variable elimination. Let an algebraic system $F(X) = 0$ be a system of polynomial functions $f_i(X)$, $i = 1, \dots, m$ with variables $X = \{x_1, \dots, x_n\}$, the variable elimination approach consists in the transformation of the original system $F(X) = 0$ into another system $H(Y) = 0$ with functions $g_j(Y)$, $j = 1, \dots, p$ with variables $Y = \{y_1, \dots, y_r\}$ where $r < n$. Ultimately, the goal is to find a method which allows to compute an equation system $H(Y)$ in either triangular format or preferably in univariate form which would be the easiest to solve.

Most variable elimination methods are usually divided into four steps:

- Step 1: Variable elimination.
- Step 2: Solving the univariate equation.
- Step 3: Return or extension to original system variables.

We will examine the variable elimination methods which were successfully applied to solve the **FKP** from which two can be identified:

- method based on resultant calculation including the so-called dialytic elimination,
- method based on Gröbner basis calculation.

5.5.2 Resultant method

Variable elimination can be implemented through a recursive method based on resultants. As input, we give a system of equations with rational coefficients. The output will be one univariate polynomial equation in terms of one of the original variables. Each elimination step involves two polynomial equations which results in one equation with the number of variable reduced by one.

Definition 5.4 [Cox et al. 1992] Let a system be $F(X) = \{f_1, \dots, f_n\} \in Q[x_1, \dots, x_n]$; let $P = f_i$ and $R = f_j$ where $f_i = a_p x^p + \dots + a_0$ and $f_j = b_q x^q + \dots + b_0$ with $i, j \in \{1, 2, \dots, n\}$ and $a_i, b_i \in Q[x_2, \dots, x_n]$, let $p = \deg(P)$ et $r = \deg(R)$ knowing that $p, r \in \mathbb{N}^*$; suppose that $a_0 \neq 0$ and $b_0 \neq 0$; then $\text{Res}(f, g, x_1) = \det(M)$ which is the resultant of P and R in terms of x_1 where M is the identified as the Sylvester matrix.

Then, the Sylvester matrix can be expressed in terms of the polynomial coefficients:

$$M = \begin{pmatrix} a_0 & 0 & \cdots & 0 & b_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \cdots & 0 & b_1 & b_0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ a_l & & \cdots & a_1 & b_m & & \cdots & b_1 \\ 0 & a_l & \cdots & a_2 & 0 & b_m & \cdots & b_2 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & & \cdots & a_l & 0 & & \cdots & b_l \end{pmatrix} \quad (39)$$

Inasmuch, we can write: $\text{Res}(P, R, x_1) = \det(\text{Sylv}(P, R, x_1))$. If we examine the Sylvester matrix, we can observe that part with the a_i parameters contains m columns and the one with b_i n columns. The following proposition holds and its proof is described in [Cox et al. 1992]: *The resultant $\text{Res}(f, g, x)$ is the first ideal of elimination $\langle f, g \rangle \cap k[x^2, \dots, x_n]$; moreover, $\text{Res}(f, g, x) = 0$ iff f, g have a common factor in $k[x_1, \dots, x_n]$ which has a positive degree in x .* The nature of this factor has to be determined and we wish to establish if it is only one root of functions f and g . To answer that question, the following corollary will be employed: *If $f, g \in C[X]$ then $\text{Res}(f, g, x) = 0$ iff f and g contain a common root in C .* This common root is determined by computing $\det(\text{Sylv}(P, R, x_1)) = 0$. The nature of this root has to be determined, notably if it is a partial one and the answer will come from the following proposition, [Cox et al. 1992]: *Knowing that $f, g \in C[X]$, let $a_0, b_0 \neq 0$ and $a_0, b_0 \in C[x_2, \dots, x_n]$, if $\text{Res}(f, g, x) \in C[x_2, \dots, x_n]$ cancels at (c_2, \dots, c_n) , then we obtain that either $a_0 b_0 = 0$ at (c_2, \dots, c_n) or either $\exists c_1 \in C$ such as f and g cancel.*

In certain instances, the head terms of the polynomials can cancel which will result in the cancellation of the determinant and the process consequently adds one extraneous root.

In order to obtain the univariate equation, a recursive algorithm will be applied. Firstly, we calculate $n - 1$ resultants $h_k = \text{Res}(f_{k+1}, f_1, x_1)$ on variable x_1 for $k = 1, \dots, n - 1$. Secondly, we compute $n - 2$ resultants $h^{(2)}_j = \text{Res}(h_{j+1}, h_1, x_2)$ on variable x_2 for $j = 1, \dots, n - 2$ and we continue in the same fashion until the univariate equation is determined. An almost triangular equation system is constructed.

$$\begin{pmatrix} f_1(X) = 0, \dots, f_{n-2}(X) = 0, f_{n-1}(X) = 0, f_n(X) = 0 \\ h_1(x_2, \dots, x_n), \dots, h_{n-2}(x_2, \dots, x_n), h_{n-1}(x_2, \dots, x_n) \\ h^{(2)}_1(x_3, \dots, x_n), \dots, h^{(2)}_{n-2}(x_3, \dots, x_n) \\ \vdots \\ h^{n-2}_1(x_{n-1}, x_n), h^{(n-2)}_2(x_{n-1}, x_n) \\ H(x_n) \end{pmatrix} \quad (40)$$

The last $H(x_n)$ is the targeted univariate equation. However, this equation cannot be considered equivalent to the initial algebraic system because the head terms can cancel.

The return step to original variables is performed by substituting back through the triangular system. The equation is solved $H(x_n) = 0$ and we obtain a series of w roots $\{x_n\}$. We take the w roots, one by one, which is introduced in one of the equations $h_1^{(n-2)}(x_{n-1}) = 0$ or $h_2^{(n-2)}(x_{n-1}) = 0$ and obtain the w roots $\{x_{n-1}\}$. We continue until x_1 is isolated.

The **6-6 FKP** has been solved applying resultants, [Husty 1996], in a computer algebra environment to avoid intermediate result truncation, since, in a sense, parameter truncation can be envisioned as changing the manipulator configuration.

A variation to the resultant method is called the dialytic elimination. Let the variable set be $X = \{x_1, \dots, x_n\}$ of the algebraic system $F(X) = 0$; then select any variable x_i and set it as the hidden variable, then a monomial vector is constructed around x_i for the system $F(X)$ which is expressed as $\bar{W} = (1, x_i, x_i^2, \dots)$. The FKP is rewritten as a linear system in terms of \bar{W} :

$$A\bar{W} = 0 \tag{41}$$

where : $\bar{W} \neq 0$

Being a generalization of $Res(P, Q, x_1) = det(Sylv(P, Q, x_1))$, it is subjected to the same risks of root addition through the head term cancellation. Dialytic elimination has been implemented to solve the FKP of the 3-RS or MSSM parallel manipulators, [Griffis and Duffy 1989, Dedieu and Norton 1990, Innocenti and Parenti-Castelli 1990]. Satisfactory results were produced on simple parallel manipulators, [Raghavan and Roth 1995].

5.6 Gröbner Bases

Lets denote by $Q[x_1, \dots, x_n]$ the ring of polynomials with rational coefficients. For any n-uple $\mu = (\mu_1, \dots, \mu_n) \in N^n$, lets denote by X^μ the monomial $X_1^{\mu_1} \dots X_n^{\mu_n}$. If $<$ is an admissible monomial ordering and $P = \sum_{i=0}^r a_i X^{\mu^{(i)}}$ any polynomial in $Q[X_1, \dots, X_n]$, the following polynomial notations are necessary :

- $LM(P, <) = \max_{i=0 \dots r}, < X^{\mu^{(i)}}$ is the leading monomial of P for the order $<$,
- $LC(P, <) = a_i$ with i such that $LT(P) = X^{\mu^{(i)}}$ is the leading coefficient of P for $<$,
- $LT(P, <) = LC(P, <) \cdot LM(P, <)$ is the leading term of P for $<$.

Lets denote by x_1, \dots, x_n the unknowns and $S = \{P_1, \dots, P_s\}$ any polynomial system as a subset of $Q[x_1, \dots, x_n]$. A point $\alpha \in C^n$ is a zero of S if $P_i(\alpha) = 0 \forall i = 1 \dots s$. Actually, any large polynomial equation system cannot be directly or explicitly solved. Thus, it is necessary to revert to mathematical objects containing sufficient information for resolution. Any polynomial system is then described by an ideal:

Definition 5.5 [Cox et al. 1992] *An ideal I is defined as the set of all polynomial $P(X)$ that can be constructed by multiplying and adding all polynomials in the ring of polynomials with the original polynomials in the set S .*

A Gröbner Basis G is then as a computable polynomial generator set of a selected polynomial set $S = \{P_1, \dots, P_s\}$ with good algorithmic properties and defined with respect to a monomial ordering. This basis is a mathematical object including the ideal I information. The *lexicographic* and *degree reverse lexicographic (DRL)* orders are usually implemented, [Cox et al. 1992, Geddes et al. 1994]. Given any admissible monomial ordering, the classical Euclidean division can be extended to reduce a polynomial by another one in $Q[X_1, \dots, X_n]$. This polynomial reduction can be generalized to the reduction by a polynomial list. The reduction output depends on the monomial ordering $<$ and the polynomial order.

Definition 5.6 *Given any admissible monomial ordering, $<$, a Gröbner Basis G with respect to $<$ of an ideal $I \subset Q[X_1, \dots, X_n]$ is a finite subset of I such that: $\forall f \in I, \exists g \in G$ such that $LM(g, <)$ divides $LM(f, <)$.*

Some useful Gröbner Basis properties are described in the following theorem:

Theorem 5.1 *Let G be a Gröbner Basis G of an ideal $I \subset Q[X_1, \dots, X_n]$ for any $<$ monomial ordering, then a polynomial $p \in Q[X_1, \dots, X_n]$ belongs to I if and only if the reduction algorithm $Reduce(p, G, <) = 0$; the reduction does not depend on the order of the polynomials in the list of G ; it can be used as a simplification function.*

The classical method for computing a *Gröbner Basis* is based on Buchberger's algorithm [Buchberger et al. 1982, Buchberger 1985]. Recently, Faugère proposed more powerful algorithms, namely *accel* and *F4* implemented in the *FGb* software, [Faugère 1999].

5.7 Gröbner Bases and zero dimensional systems

Definition 5.7 [Lazard 1992a] *A zero-dimensional system is defined as a mathematical equation system with a variety (solution set) constituted by a finite number of complex points.*

From a mathematical point of view, any variety is a valid result. From an engineering one, we seek a variety which represents an exploitable result. For any parallel manipulator **FKP**, the *zero-dimensional* systems allow pose analysis, since each *real* solution corresponds to each manipulator pose. Otherwise the problems fall in the category of rarely usable singular configurations. Detection of *zero-dimensional* systems is performed by implementing the following theorem:

Theorem 5.2 *Let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis for any ordering $<$ of any system $S = \{P_1, \dots, P_s\} \in Q[X_1, \dots, X_n]^s$. The two following properties are equivalent:*

- *For all index i , $i = 1..n$, there exists a polynomial $g_j \in G$ and a positive integer n_j such that $X_i^{n_j} = LM(g_j, <)$;*
- *The system $\{P_1 = 0, \dots, P_s = 0\}$ has a finite number of solutions in C^n .*

Hence, one can determine if the **FKP** resolution yields a finite or infinite number of solutions. This is obviously an important issue, since an algebraic system with an infinite number of solutions (variety of degree one or higher) cannot be directly exploited by a control system and failure is usually the outcome.

5.8 The lexicographic Gröbner Basis

The choice of the monomial ordering is critical for the *Gröbner Basis* computation efficiency, which is evaluated by the computation times, the size and the shape of the result. A *lexicographic Gröbner Basis* with $X_1 < X_2 < \dots < X_n$ as variable order has always the following general shape:

$$\begin{cases} f_1(X_1) = 0, f_2(X_1, X_2) = 0, \dots \\ f_{k_2}(X_1, X_2) = 0, f_{k_2+1}(X_1, X_2, X_3) = 0, \dots \\ f_{k_{n-1}+1}(X_1, \dots, X_n) = 0, \dots \\ f_{k_n}(X_1, \dots, X_n) = 0 \end{cases} \quad (42)$$

Whilst computing a *lexicographic Gröbner Basis* using Buchberger's algorithm yields doubly exponential complexity in the worst case, an order change will save computation time. Therefore, one *DRL Gröbner Basis* may first be computed in d^n polynomial time where d is the maximum total degree of the equations. Then, the *DRL Gröbner Basis* can be converted into a *lexicographic* one in $O(nD^3)$ arithmetic operations, [Faugère et al. 1991]. In the case of any *zero-dimensional* system, this can be done using exclusively linear algebra techniques, [Faugère et al. 1991].

Definition 5.8 *Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$, without square factors. Let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis computed for any ordering $<$ on the system S . If the univariate polynomial $f(x_1)$ is without any multiple factors, then the equation is square-free, the first coordinates of all solutions are distinct and the solutions yield no multiple root. Then, the system is considered in **Shape Position**.*

If the polynomial system is in **Shape Position**, the Gröbner Basis has the following simplified univariate format:

$$f_1(X_1) = 0, X_2 = f_2(X_1) = 0, \dots, X_n = f_n(X_1) \tag{43}$$

In this format, the *lexicographic Gröbner Basis* can be computed directly using the F4 algorithm, [Faugère 1999].

5.9 The rational univariate representation

Another root representation is given by the *Rational Univariate Representation*, hereafter identified *Rational Univariate Representation*, [Rouillier 1999]. In the particular case of systems in **Shape Position**, the univariate variable is the first one in the original system ones, so that the *Rational Univariate Representation* can be written in the following format:

$$\{h(X_k) = 0, X_2 = h_2(X_k)/h_0(X_k), \dots, X_n = h_n(X_k)/h_0(X_k)\} \tag{44}$$

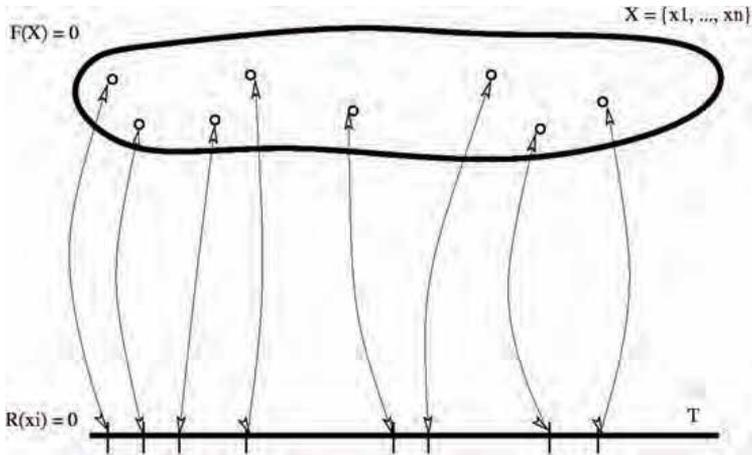


Fig. 8. The correspondence between the *Rational Univariate Representation* and original system varieties.

$h(X_k)$ where $1 \leq k \leq n$ denotes the univariate equation, h_i $i = 0 \dots n$ the univariate return functions. These polynomials are elements of $Q[X_k]$. Inasmuch, h and h_0 are coprime. Moreover, if the *Rational Univariate Representation* system is in **Shape Position** and if $h(X_k)$ is square-free, the *Rational Univariate Representation* can be converted into a *lexicographic Gröbner Basis* and the *Rational Univariate Representation* expression can be simply computed from this basis. This situation is generally the case in most **FKP** instantiations:

$$h(X_k) = f(X_k), g(X_k) = f'(X_k) \tag{45}$$

Moreover, since f and f_0 are coprimes, f_0 is invertible modulo f ; this leads to: $h_i = f_i \cdot (f')^{-1}$ modulo f , $i = 2 \dots n$.

Definition 5.9 Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$ and let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis for any ordering $<$ of the system S . Then, the **RR** form is defined as the system $R = \{p_1, p_1^{-1}p_2 \text{ mod } G, p_1^{-1}p_3 \text{ mod } G, \dots, p_1^{-1}p_k \text{ mod } G\} \in Q[x_1, \dots, x_n]$ where p_1^{-1} is calculated such that $(p_1 p_1^{-1} \text{ modulo } G) = 1$.

The rationale behind the *RR* form computation is to reduce the polynomial system S . The computed *Gröbner Basis* G on any ordering $<$ of the system S is used for the system reduction. Then, the reduced system R resolution shall be less difficult.

Definition 5.10 Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$, if the system S is without square factors and in **Shape Position** and if a lexicographic *Gröbner Basis* is computable, then the system *RR* form R' can be deduced.

Since the S system reduction by a lexicographic *Gröbner Basis* leads to a further reduced system R , one can conjecture faster **FKP** resolution. However, in some instances, p_1^{-1} can hardly be computed and the *RR* form cannot be deduced. Hence, the **FKP** is solved by first computing a *Gröbner Basis* on a **DRL** order and then we rely on the aforementioned order change to obtain a lexicographic *Gröbner Basis*. In the rare instances where the **FKP** system is not in **Shape Position** and a lexicographic *Gröbner Basis* is not computable either by the *RR* form or by order change, the *Rational Univariate Representation* can be always be computed by reverting to the more robust and lengthy algorithm taking as input any monomial ordering *Gröbner Basis* and calculating the multiplication tables and matrices, [Rouillier 1999].

In practice, the *Rational Univariate Representation* coefficients are smaller than the lexicographic *Gröbner Basis* ones, [Alonso et al. 1996]. This is the rationale behind the preference of the *Rational Univariate Representation* expression which can be computed using the algorithm described in [Rouillier 1999]. Anyhow, it should be deduced from a lexicographic *Gröbner Basis* if the system is in **Shape Position**. Hence, in any case, the determination of a lexicographic *Gröbner Basis* is first tried and then, a *Rational Univariate Representation* is computed. In fact, for any parallel manipulator **FKP**, the computation of the *RR* form is preferred. Finally, the method insures that the *Rational Univariate Representation* system is equivalent to the former polynomial system. Each *Rational Univariate Representation* system solution corresponds to one and only one original system solution and the properties are preserved, fig. 8.

5.10 Real roots isolation

Once the *Rational Univariate Representation* is computed, then the real roots of the univariate equation $h(X_k)$ roots can be isolated either using rapid numeric methods or exact algebraic methods. As far as algebra is concerned, such computations can be performed by a method such as the Uspensky one based on Descartes' sign rule and Sturm's theorem. An improved algorithm is introduced in [Rouillier and Zimmermann 2001] applying one Vincent theorem. In computer algebra and in particular in the proposed method, a natural way for coding a real algebraic number α is given by either a square-free polynomial f with rational coefficients such that $f(\alpha) = 0$ or an interval $[l, r]$ with rational bounds containing α . Inasmuch, if $\beta \neq \alpha$ and $f(\beta) = 0$ then $\beta \notin [l, r]$.

Let α be a root of $h(T)$ represented by $(\bar{h}, [l, r])$, then the resolution of $h(T) = 0$ produces a solution list $\{\alpha_1, \dots, \alpha_r\} = [(\bar{h}, [l_i, r_i]), i = 1 \dots r]$ where \bar{h} is the square-free part of h . Moreover, it is very easy to refine the intervals $[l_i, r_i]$ by dichotomy since \bar{h} is square-free which induce that $sign(\bar{h}(l_i)) \neq sign(\bar{h}(r_i))$ if $l_i \neq r_i$.

Since h and h_0 are coprime by definition of the *Rational Univariate Representation*, one can refine $[l, r]$ so that it doesn't contain any real root of G . One can then simply use interval arithmetic to compute $[l^{(i)}, r^{(i)}] = h_i/h_0([l, r]); i = 1 \dots n$. The complete method allows one solution of the equation $h(T) = 0$ to correspond to each solution of the system $F(X) = 0$ since the bijection is guaranteed and the properties are preserved, fig. 8.

Then, using the return functions h_i/h_0 , for each α , we compute an interval product $\prod_{i=0}^n [l^{(i)}, r^{(i)}]$ containing $[h_1/h_0(\alpha), \dots, h_n/h_0(\alpha)]$. For refining $\prod_{i=0}^n [l^{(i)}, r^{(i)}]$, it is sufficient to refine and introduce again $[l, r]$ in the coordinate functions h_i/h_0 ; $i = 1 \dots n$.

The *Rational Univariate Representation* system real root isolation requires a second step where the original system roots are determined in terms of the original variables using the return functions provided by the *Rational Univariate Representation*.

By computing a *Rational Univariate Representation* and then isolating the real roots as described above, we obtain an exact method for isolating all the real roots of any polynomial system with a finite number of complex roots verified during the computations using theorem 5.2. The *Rational Univariate Representation* has rational coefficients and the isolating intervals have rational bounds. For each solution $\alpha = (\alpha_1, \dots, \alpha_n)$, the method produces :

- a box $B_\alpha \prod_{i=0}^n [l^{(i)}, r^{(i)}] \in \mathbb{R}^n$ such that $\alpha \in B_\alpha$, $(l_i, r_i) \in \mathbb{Q}^2$ with exact (rational) values and if $\beta \neq \alpha$ is a root of the system, then $\beta \notin B_\alpha$;
- a function for refining B_α up to any precision $\epsilon > 0$ ($|r_i - l_i| < \epsilon, \forall i = 1 \dots n$).

5.11 The certified method description

Assembling the aforementioned algorithms, we obtain a complete certified method since it handles the intermediate results with exact representations. Taking into account the fact that most parallel manipulator FKP are in **Shape Position**, the method has been simplified, fig. 9.

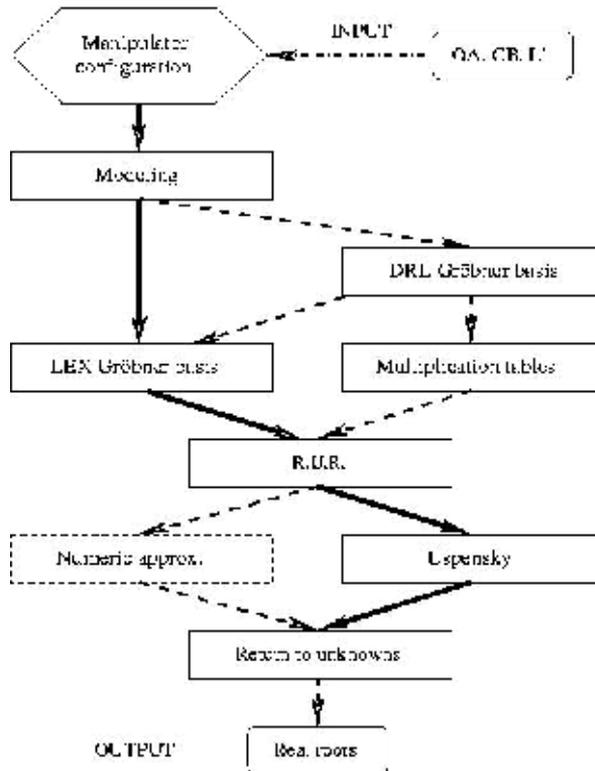


Fig. 9. Bloc diagram of the exact method

6. Solving the forward kinematics problem

6.1 Solving the general 6-6

The selected manipulator is a generic 6-6 in a realistic configuration, measured on a real parallel robot prototype, trying to reproduce a singularity-free theoretical SSM design. This example is a typical one among several successful trials which were performed to test the method without any failures. Even when the 6-6 FKP was in a singular pose, the method returned that the solution is a variety of degree one (infinite number of solutions), thus in singularity.

Computations were done on a personal computer equipped with a 400 MHz *Pentium II* microprocessor including 128 bytes of random access memory. The operating system was *Red Hat LINUX*. Thus, this conservative approach allows the user to expect better performance.

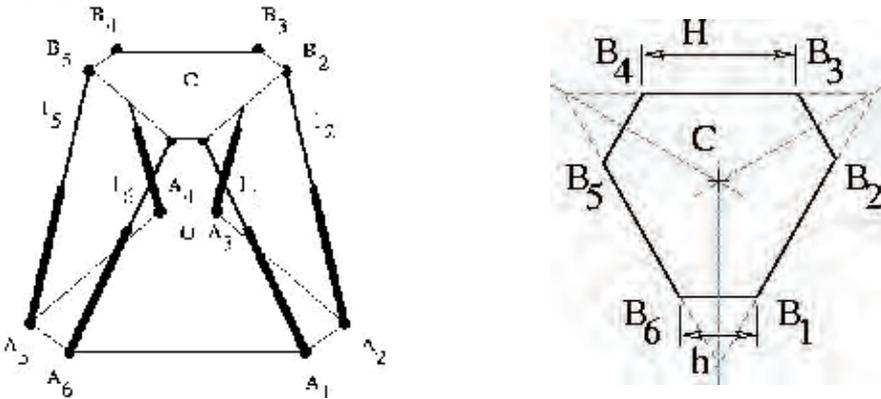


Fig. 10. The selected 6-6 hexapod manipulator

6.1.1 Modeling the 6-6 using the three point model

The parallel manipulator corresponding to the configuration data is shown in fig. 10. Lets take a typical 6-6 configuration example where construction realism introduce manipulator configuration deformations. It is then written in a configuration text file which includes the manipulator essential parameters: the coordinates of the joint center positions $OA_{i|R_f}$ in the fixed base reference frame R_f and the coordinates of the joint center positions $CB_{i|R_m}$ in the mobile platform reference frame R_m . In the computations, we use their simplified format, respectively identified as A and B. The unit is the millimeter:

```

B := [ [ 68410/1000, 393588/1000, 236459/1000 ],
       [ 375094/1000, -137623/1000, 236456/1000 ],
       [ 306664/1000, -256012/1000, 236461/1000 ],
       [ -306664/1000, -255912/1000, 236342/1000 ],
       [ -375057/1000, -137509/1000, 236464/1000 ],
       [ -68228/1000, 393620/1000, 236400/1000 ]]:
A := [ [ 464141/1000, 389512/1000, -178804/1000 ],
       [ 569471/1000, 207131/1000, -178791/1000 ],
       [ 1052905/10000, -597151/1000, -178741/1000 ],
       [ -1052905/10000, -597200/1000, -178601/1000 ],
       [ -569744/1000, 206972/1000, -178460/1000 ],
       [ -464454/1000, 389384/1000, -178441/1000 ]]:
Li := [(1250^2)\$i=1..6]:

```

6.2 The ten formulation comparison

Solving the ten **FKP** proposed formulations has been tried on the aforementioned example. Performance results are displayed on table 1 where the total real root computation times along with model code and name are given. This table specifies if the method terminated or was interrupted by the user if computations lasted more than one hour.

Five **FKP** formulations led to computation termination. However, only the AFD4, AFD5, AFP2 and AFP3 models brought response times within an hour. For the AFD1 formulation, only the *DRL Gröbner Basis* was successfully computed. Therefore, only the three following models were retained:

- AFD4 - Formulation with the translation and *Gröbner Basis* of the rotation matrix,
- AFD5 - Formulation with Translation and Quaternion variables,
- AFP3 - The three point model with constraints and function recombination.

Code	Method	Resp. time seconds	Ending	Comment
AFD1	trigo. ident.	148	FGb yes	stopped
AFD2	trigo relations	3600+	no	stopped
AFD3	\mathcal{H}	3600	yes	
AFD4	GB(\mathcal{H})	4,8	yes	
AFD5	quaternion	59,6	yes	
AFD6	somas	3600	no	stopped
AFP1	3-points+dist	3600+	no	stopped
AFP2	3pts+dist+normal	125,0	yes	
AFP3	3pts+modified	10,9	yes	
AFP4	6-points	3600+	no	stopped

Table 1. Response times for each formulations

Table 2 shows information which can be deduced from system observation. It gives the variable number, equation number and the maximum degree with respect to one variable, file size (*ASCII* size in bytes) and the maximum number of digits in one coefficient.

Code	Method	Variable number	Equations number	Size bytes
AFD4	GB on \mathcal{H}	12	30	7100
AFD5	Quaternion	7	7	6100
AFP3	3 pts & recomb	9	9	4500

Table 2. The selected formulation characteristics

6.3 System model analysis and selection

Solving more examples was carried using the three selected formulations in order to evaluate computations. Performance is studied on the *Gröbner Basis* being the largest computed mathematical object. We compare the resulting file sizes and the computation response times. The computations are performed on the preceding theoretical **SSM**, identified *theo*, and on the corresponding real general **6-6**, identified *eff*.

Robot + model	Strategy tgrob (s)	Accel (s)	F4 (s)	Gröbner Basis Size (Kbytes)	Complex Solution	Comment
theo_AFP3.fgb	29,06	6,53	0,6	130	36	
theo_AFD5.fgb	124,29	36,24	2,45	150	72	
theo_AFD4.fgb	28,22	3600	0,31	85	36	stopped
eff_AFP3.fgb	NA	1200	33,0	800	40	
eff_AFD5.fgb	NA	NA	31,9	790	80	
eff_AFD4.fgb	NA	NA	38,9	768	40	

Table 3. *DRL Gröbner basis* computation times and memory usage

As input, the manipulator configuration data include coefficients which are rational numbers with a six digit numerator and the number 1000 as denominator. Thus, the manipulator is measured at the micron accuracy. Let's note that computations are proceeding with the *RR* form.

6.4 DRL Gröbner Basis computations

Three algorithms shall be applied, namely *tgrob*, *Accel* and *F4* respectively the Buchberger, the *Gb* and the *Fgb* implementations, [Faugère 1999]). Table 3 presents the computations times in seconds and memory space in kilobytes for the various techniques to compute a *DRL Gröbner Basis* for the three selected formulations.

The response time discrepancy between the **SSM** and **6-6** is significant. Practically, a small even infinitesimal coefficient difference such as platform non-planarity leads to response times which can be either dramatically longer (15, 55 or 100 times). The *Accel* algorithm is more sensible to formulation than *F4*. The *Accel* algorithm produces lengthy computations using the *AFD4* model. The quaternion formulation produces contradicting results, since it leads to significantly longer response times for the **SSM**, but slightly faster ones for the **6-6**. For the **6-6**, the response times and file sizes are almost equivalent.

Applying the fast *F4* algorithm, the rotation matrix model is preferred since one part of the system equations (13) does not depend on the studied configuration.

6.5 Rational univariate representation computations

The computed *DRL Gröbner Basis* is provided as the input to the *Rational Univariate Representation* computation algorithms. Knowing that the *Rational Univariate Representation* computation behaviour from the three formulation Gröbner bases cannot be predicted and lead to comparable *Gröbner Basis* sizes, the three *Rational Univariate Representation* computations have been performed. The following figures come from the direct calculations of the *RR* form using Faugère's *F4* algorithm for the lexicographic order followed by a division leading to a *Rational Univariate Representation* as the end-result. This strategy is compared with the original one which computes the *Rational Univariate Representation* by an order change (*F4+FGLM*). Table 4 gives the response times in seconds, memory space usage in kbytes, the number of complex and real roots.

Name	F4+FGLM (s)	RR (s)	File (Kbytes)	Complex #	Real #
theo_AFP3.rur	NA	2,95	210	36	8
theo_AFD5.rur	NA	NA	NA	72	16
theo_AFD4.rur	NA	3,31	270	36	8
eff_AFP3.rur	36,0	10,9	320	40	8
eff_AFD5.rur	3600	59,6	290	80	16
eff_AFD4.rur	17,9	4,8	300	40	8

Table 4. Rational Univariate Representation response times from the RR form

Problems were encountered with the quaternion based formulation to solve the **SSM**. This proves that even simpler hexapod configurations can be challenging. For the real 6-6 manipulator, the RR form calculation is always faster than the computation with an order change. Inasmuch, our results indicate that the RR Rational Univariate Representation calculation using the AFD4 rotation matrix formulation is the most effective one. For the quaternion formulation, two solutions express the same manipulator posture, this is confirmed by the obtained univariate polynomial which has a total degree 80 and the number of real solutions also doubled. This behavior is well known in practice since the redundant nature of quaternions leads to two results for the same pose. Finally, the direct computation of a *lexicographic Gröbner Basis* is the right choice when using recent algorithms such as F4. Many instantiations were tested on various manipulator configurations and all FKP systems were in **Shape Position** and the RR form could be computed.

6.6 FKP of typical hexapod manipulators

6.6.1 Hexapod architecture descriptions

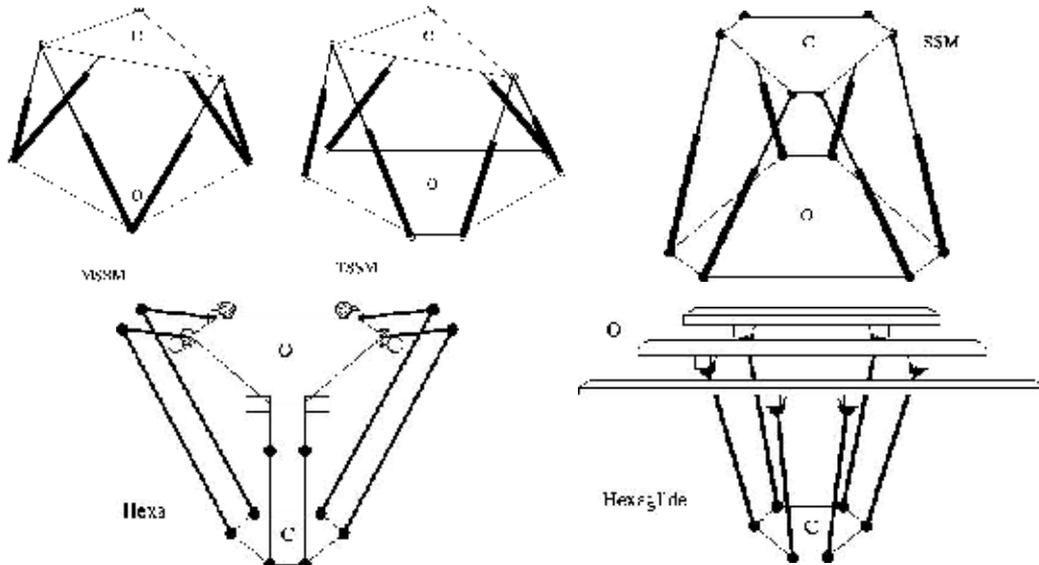


Fig. 11. The various hexapod alternatives

Although the method has been shown to solve the **6-6** and **SSM** manipulator **FKP**, it is relevant to compare results for various hexapods which feature specific and related geometric properties.

Several well-known parallel manipulators feature theoretical architectures, fig. 11. In order to clearly examine the configuration change impact, we have selected the first five hexapods as the slightly modified versions of each other, from the theoretical **MSSM** until the realistic **6-6**. The **6-6p** design comprises the joint positions lying on the same plane for either the fixed base or mobile platform. The **6-6pp** design is characterized by the two platforms being planar. The **SSM** design repeats the **6-6pp** along with planar hexagonal platforms which are made by equally truncating an equilateral triangular platform, [Nanua et al. 1990]. The **TSSM** manipulators are designed by merging the ball or Cardan joints in pairs to obtain a truly triangular mobile platform. The **MSSM** is merging the ball or Cardan joints of the fixed base by pairs to obtain an octahedron. These theoretical design **FKP** are computed and their kinematics results compared with the general **6-6** case. For each manipulator, the selected configuration and joint variables of each manipulator are selected to correspond.

6.6.2 FKP solving results

On table 5, the results are including the manipulator type, the manipulator morphology, the base and platform morphology where *T*, *S*, *P* and *NP* stands respectively for triangular, truncated triangular, planar and not planar, the complete real root computation time, the number of complex roots (the univariate polynomial degree), the number of real roots and the *Gröbner Basis* size. Only the *Gröbner Basis* sizes are compared, since the *Gröbner Basis* is the largest mathematical object calculated by method.

Robot type	Mechanism config	Base	Platform	Time (s)	Complex sols #	Real sols #	Gröbner (kbytes)
MSSM	octahedron	T,P	T,P	0,07	16	16	60
TSSM	hexapod	S,P	T,P	0,08	16	16	76
SSM	(6)	S,P	S,P	0,67	36	16	238
6-6pp	(6)	P	P	1,1	40	16	390
6-6p	(6)	NP	P	1,8	40	16	308
6-6	(6)	NP	NP	10,4	40	16	402
DIET	(6)	NP	NP	9,9	40	40	392
Hexa	6R-6R-6R	P	S,P	2,0	40	8	346
Hexaglide	6T-6R-6R	P	S,P	0,5	36	8	180

Table 5. Hexapod **FKP** overall results and performances

6.6.3 Discussion on the results

The gradual passage to geometrically simpler parallel structures leads to significantly shorter response times. In the **6-6** case, the introduction of one planar platform reduces computation times by 20. Inasmuch, the passage from a theoretical **SSM** to a realistic **6-6** leads to computation times which are 40 times longer.

With model simplification, it is notable that the real solution number is maintained. One could conjecture homothopy in those cases. The number of complex roots varies when the manipulator type changes; the **MSSM** and **TSSM** only have 16 solutions whereas the **SSM** has 36. This last result raises an important classification issue which was not formerly identified, [Faugère and Lazard 1995, Merlet 1997]. The **SSM** manipulator does not go in the class of **6-6** manipulators and becomes a class by itself.

The table second section comprises **FKP** solving results about the hexapod obtained with the Dietmaier's method, [Dietmaier 1998], the **Hexa** and the **Hexaglide** with a **SSM** type mobile platform. In all instances, the **Hexa**, [Pierrot et al. 1991], and **Hexaglide**, [Hebsacker 1998] can feature either 36 or 40 complex solutions, this only depends on the mobile platform configuration. In fact, when the truncated equilateral triangle is used, then the **FKP** yield 36 solutions.

Knowing that the **6-6pp** and **6-6p** lead to 40 complex solutions, the new **SSM** class is characterized not only by its complex solution number but by the mobile platform and fixed base peculiar type.

Therefore, the manipulators which fall into the **SSM** category are the ones which feature one specific truncated platform.

Although Dietmaier's **6-6 FKP** yields the largest number of real solutions, it does not necessarily lead to the longest computation times and largest result files.

Finally, various tests were performed where leg lengths were changed such as moving the robot on a straight line or circles with the same **6-6** manipulator configuration. The real root numbers have all been an even number in the set {4,8,12,16} depending on the location inside the workspace. This could be considered a conjecture. The only case where only one real solution has been found is when a theoretical **6-6pp** has its actuator values bringing the manipulator mobile platform to lie on the fixed base plane. This solution corresponds to two real coincident roots and this case can be identified as a singularity with the loss of three *DOF* since the manipulator can then only move in a plane.

6.7 Assembly mode analyses

The exact method allows addressing the question of assembly modes. This problem is also referred to as posture analysis. Assembly modes are defined as follows:

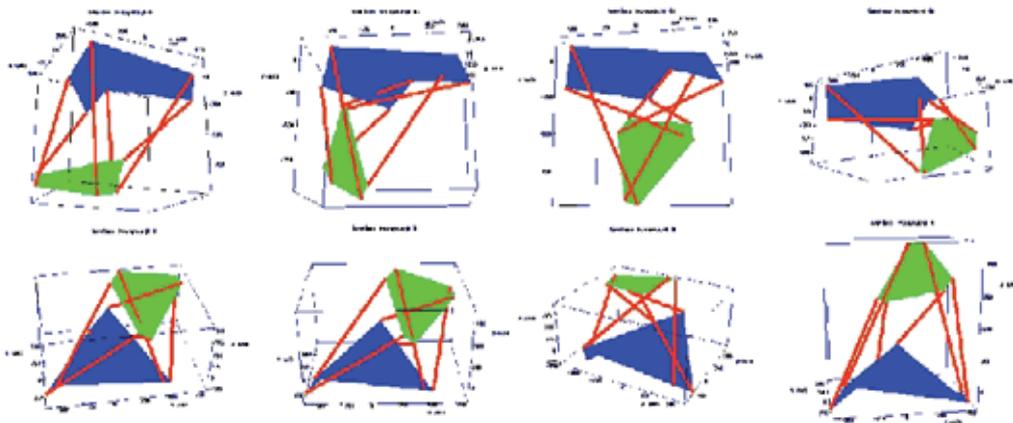


Fig. 12. Posture analysis for the general **6-6** manipulator

Definition 6.1 *Given a manipulator configuration where the fixed base, the mobile platform and kinematics chain lengths are specified, for a set of active joint positions, determine all the possible geometric assemblies for the selected manipulator.*

For the selected example, eight real solutions were obtained which geometrically represent the only possible assembly modes. These modes are then drafted using the *XMuPAD* environment, fig. 12. The position based models lead to root results which are directly usable to draw the effective postures. This exemplifies that posture analysis is feasible for any manipulator which can be modelled as a general 6-6 hexapod.

7. Conclusion

In this chapter, one complete exact method to solve the parallel manipulator **FKP** has been explained. The method was applied to the 6-6 general parallel manipulator, which is recognized to yield the most difficult problem, and also to various other manipulators such as the **SSM**, **TSSM**, **MSSM**, **Hexa** and **Hexaglide**.

Moreover, the modeling of the **FKP** was investigated. Six displacement based models and two position-based models were derived for the 6-6 general parallel manipulator.

One complete algebraic method to solve the Forward Kinematics Problem was applied. Although many methods can find solutions to some of these **FKP** systems, the proposed algebraic exact method insures the exactness of the real solution results, since it is based on one *Gröbner Basis*, which completely describes the ideal related to the original system. From this basis, one computes the *Rational Univariate Representation* including one univariate equation for root isolation. The selected algorithms always succeeded to solve any parallel robot **FKP** in all tested non-singular instances.

The selected manipulator was a typical 6-6 hexapod known as the Gough platform in a calibrated configuration, measured on a real parallel robot prototype constructed from a theoretically singularity free **SSM** design.

The 8 polynomial formulations were implemented and compared. We identified three models that allowed for computation termination, out of which two were retained since their computations occur with acceptable performances: a displacement based formulation with the rotation matrix *Gröbner Basis* with end-effector position and rotation matrix parameters as variables and a formulation with three points on a platform.

Solving typical posture examples, the *Rational Univariate Representation* comprised a univariate equation of degree 40 and 8 to 12 real solutions were computed depending on the position in the workspace. The total computation averaged 130 seconds on a relatively old computer. On a faster computer, the response time falls to less than one second. Hence, this method can be suitable for small-scale trajectory pursuit applications.

This result is very important since any *Gröbner Basis* completely describes the mathematical object related to the original system. From this basis, one can try to build an exact equivalent system with the original one including one univariate equation. Up to author's knowledge, this is actually the only known method that is certified to establish a truly equivalent system that preserves the properties.

Further testing led to favor the first formulation, since it yields slightly faster computations and it gives directly the end-effector position. The quaternion-based models can lead to difficulties in the case of simpler configurations and thus longer computation times, since it is doubling the solution number, which is explained by the *Rational Univariate Representation*

equations having a degree twice as large as the others. Moreover, one final advantage is that the displacement-based equations can be applied on any manipulator mobile platform.

8. Acknowledgment

I would like to thank my wife Clotilde for the time spent on rewriting and correcting the book chapter in Word.

9. References

- Alonso, M.-E.; Becker, E.; Roy M.F. & Woermann T. (1996) Multiplicities and idempotents for zerodimensional systems. In *Algorithms in Algebraic Geometry and Applications*, Vol. 143, Progress in Mathematics, pages 1--20.
- Buchberger, B. & Loos, R. (1982) Algebraic Simplification. In *Computer Algebra-Symbolic and Algebraic Computation*. SpringerVerlag, Vienna.
- Buchberger B. (1985) Gröbner bases: An Algorithmic Method in Polynomial Ideal Theory. In *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems*, N.K. Bose (e.d.) Reidel Publishing Company, Dordrecht, pp.184-232.
- Bruyninckx, H. & DeSchutter, J. (1996) A class of fully parallel manipulators with closed-form forward position kinematics. In *Advances in Robot Kinematics*, pages 411--420.
- Cox, D.; Little, J. & O'Shea D. (1992) *Ideals, varieties, and algorithms an introduction to computational algebraic geometry and commutative algebra*. Undergraduate texts in mathematics. SpringerVerlag, New York.
- Dedieu, J.P. & Norton, G.H. (1990) Stewart varieties: a direct algebraic method for stewart platforms. In *Proceedings of SigSam*, volume 244, pages 42--59.
- P. Dietmaier. (1998) The Stewart-Gough platform of general geometry can have 40 real postures. In *Advances in Robot Kinematics*, pages 7--16.
- Dieudonné, E.; Parrish, R. & Bardusch, R. (1972) An actuator extension transformation for a motion simulator and an inverse transformation applying Newton-Raphson's method. Technical report D7067, NASA, Washington.
- Didrit, O.; Petitot, M. & Walter, E. (1998) Guaranteed solution of direct kinematics problems for general configurations of parallel manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pages 259--265.
- Egner, S. (1996) Semi-numerical solution to 6/6-stewart-platform kinematics based on symmetry. In *Applicable Algebra in Engineering, Communication and Computing*, Vol. 7, No. 6, pages 449--468.
- Faugère, J.-C.; Gianni, P.; Lazard, D. & Mora, T. (1991) Efficient computation of zero-dimensional Gröbner basis by change of ordering. *Journal of Symbolic Computation*, Vol. 16, No. 4, pages 329--344.
- Faugère, J.C. & D. Lazard. (1995) The combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, Vol. 30, No. 6, pages 765--776.
- Faugère, J.C. (1999) A new efficient algorithm for computing Gröbner bases (f4). *J. of Pure and Applied Algebra*, Vol. 139, No. 13, pages 61--88.
- Fischer, P.J. & Daniel, R.W. (1992) Real time kinematics for a 6 dof telerobotic joystick. In *Proceedings of RoManSy 9*, Udine, pages 292--300.

- Geddes, K.; Czapor, S. & Labahn, G. (1994) *Algorithms for computer algebra*. Kluwer Academic Publishers, Nonwell.
- Gosselin, C. & Angeles, J. (1988) The optimum kinematic design of a planar three dof parallel manipulator. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 110, pages 35--41.
- Gosselin, C.; Sefrioui, J. & Richard, M.J. (1994) On the direct kinematics of spherical three dof parallel manipulators with coplanar platform. *J. of Mechanical Design*, Vol. 116, pages 587--593, June 1994.
- Griffis, M. & Duffy, J. (1989) A forward displacement analysis of a class of Stewart platform. *J. of Robotic Systems*, Vol. 6, No. 6, pages 703--720.
- Hebsacker, M. (1998) Parallel werkzeugmaschinenkinematik. In *Proceedings of IPK 98, Internationales ParallelkinematikKolloquium*, Zürich, pages 21--32.
- Hunt, K.H. (1983) Structural kinematics of inparallelactuated robotarms. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 105, pages 705--712.
- Husty, M. (1996) An algorithm for solving the direct kinematic of Stewart-Gough type platforms. *J. of Mechanism and Machine Theory*, Vol. 31, No. 4, pages 365--379, 1996.
- Innocenti, C. & ParentiCastelli, V. (1990) Direct position analysis of the Stewart platform mechanism. *Mechanism and Machine Theory*, Vol. 25, No. 6, pages 611-- 621.
- Kohli, D.; Dhingra, A. & Xu, Y.X. (1992) Direct kinematics of general Stewart platforms. In *Proceedings of ASME Conference on Robotics, Spatial Mechanisms and Mechanical Systems*, Vol. 45, pages 107--112.
- Lazard, D. (1992) Solving zerodimensional algebraic systems. *J. of Symbolic Computation*, Vol. 13, pages 117--131.
- Lazard, D. (1992) Stewart platforms and Gröbner basis. In *Proceedings of Advances in Robotics Kinematics*, pages 136--142, Ferrare, September 1992.
- Lazard, D. (1993) On the representation of rigidbody motions and its application to generalized platform manipulators. *J. of Computational Kinematics*, Vol. 1, No. 1, pages 175--182.
- Merlet, J.-P. (1987) Parallel manipulators, part1: Theory; design, kinematics, dynamics and control. Technical report 646, INRIA, SophiaAntipolis.
- Merlet, J.-P. (1994) Parallel manipulators: state of the art and perspectives. *J. of Advanced Robotics*, Vol. 8, No. 6, pages 589--596, 1994.
- Merlet, J.-P. (1997) *Les Robots parallèles*. Série Robotique. Hermès, Paris, second edition, traité des nouvelles technologies edition, 1997.
- Merlet, J.-P. (2004) Solving the forward kinematics of a Gough type parallel manipulator with interval analysis. *The International Journal of Robotics Research*, Vol. 23, No. 3, pages 221--235.
- Mourrain, B. (1993) The 40 generic positions of a parallel robot. In *proceedings of ISSAC'93*, Kiev, pages 173--182.
- Mourrain, B. (1993) About the rational map associated to a parallel robot. Technical report 2141, INRIA, SophiaAntipolis, November 1993.
- Murray, P.; et al. (1997) A planar quaternion approach to the kinematics synthesis of a parallel manipulator. *Robotica*, Vol. 15, pages 360--365.
- Nanua, P.; Waldron, K. T & Murthy, V. (1990) Direct kinematic solution of a Stewart platform. In *IEEE transactions on Robotics and Automation*, Vol. 6, No.4, pages 438-444.

- ParentiCastelli, V. & Innocenti, C. (1990) Forward displacement analysis of parallel mechanisms: closedform solution of PRR3s and PPR3s structures. In *Proceedings of the ASME 21th Biennial Mechanisms Conf.*, Chicago, pages 263--269.
- Patel, A. & Ehmann, K. (1997) Volumetric error analysis of a Stewart platform based machine tool. In *Annals of the CIRP*, Vol. 46, pages 287--290.
- Petuya, V.; Alonso, A.; Altazurra, O. & Hernandez, A. (2005) Resolution of the direct position problem of the parallel kinematic platforms using the geometric iterative method. In *EEE Intern. Conf. on Robotics and Automation*, Barcelona, pages 3255--3260.
- Pierrot, F.; Dauchez, F. & Fournier, A. (1991) Hexa: a fast six dof fully parallel robot. In *Proceedings of the ICAR Conference*, Pisa, pages 1159--1163.
- Primrose, E.J.F. & Freudenstein, F. (1969) Spatial motions. part 1: Point paths of mechanisms with four or fewer links. *ASME J. of engineering for industry*, Vol. 91, No. 1, pages 103-- 114.
- Raghavan, M. (1993) The stewart platform of general geometry has 40 configurations. *ASME Trans. of Mech. Design*, Vol. 115, No. 2, pages 277--282.
- Raghavan, M. & Roth, B. (1995) Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Transactions of the ASME*, Vol. 117, pages 71--79.
- Rolland, L. (2003) Outils algébriques pour la résolution de problèmes géométriques et l'analyse de trajectoire de robots parallèles prévus pour des applications à haute cadence et grande précision. PhD thesis, Université Henri Poincaré, Nancy 1, December 2003.
- Rolland, L. (2005) Certified solving of the forward kinematics problem with an exact method for the general parallel manipulator. *Advanced Robotics*, Vol. 19, No. 9, pages 995--1025.
- Rolland, L. (2006) Synthesis on the forward kinematics problem algebraic modeling for the planar parallel manipulator. Displacement-based equation systems. *Advanced Robotics*, Vol. 20, No. 9, pages 1035--1065.
- Rolland, L. (2007) Synthesis on the forward kinematics problem algebraic modeling for the spatial parallel manipulator. Displacement-based equation systems. *Advanced Robotics*, Vol. 21, No. 9, 32 pages 1071--1092.
- Ronga, F. & Vust, T. (1992) Stewart platforms without computer ? In *Proc. of the Intern. Conf. of real, analytic and algebraic Geometry*, Trento, pages 197--212.
- Rouillier, F. (1999) Solving zerodimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, Vol. 9, NO. 5, pages 433--461.
- Rouillier, F. & Zimmermann, P. (2001) Efficient isolation of a polynomial real roots. Technical report RR4113, INRIA.
- Sreenivasan, S.V. & Nanua, P. (1992) Solution of the direct position kinematics problem of the general stewart platform using advanced polynomial continuation. In *22nd Biennial Mechanisms Conf.*, Scottsdale, pages 99--106.
- Sreenivasan, S.V.; Waldron, K.J. & Nanua, P. (1994) Direct displacement analysis of a 6-6 stewart platform. *Mechanism and Machine Theory*, Vol. 29, No. 6, pages 855--864.

- Sugimoto, K. (1987) Kinematic and dynamic analysis of parallel manipulators by means of motor algebra. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 109: pages 3--7, 1987.
- Tsai, L.W. & Morgan, A.P. (1984) Solving the kinematics of the most general 6 and 5 dof manipulators by continuation methods. *ASME J. of Mechanisms, Transmissions and Automation in Design*, Vol. 107, pages 189--200.
- Vischer, P. (1996) Improving the accuracy of parallel robots. PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Wampler, C.W. (1996) Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mechanism and Machine Theory*, Vol. 31, NO. 3, pages 33--337.

Advanced Synthesis of the DELTA Parallel Robot for a Specified Workspace

M.A. Laribi¹, L. Romdhane^{1*} and S. Zegloul²

Laboratoire de Génie Mécanique, LAB-MA-05

Ecole Nationale d'Ingénieurs de Sousse, Sousse 4003¹,

Laboratoire de Mécanique des Solides, UMR 6610

Bd Pierre et Marie Curie, BP 30179, Futuroscope 86962 Chasseneuil²

Tunisia¹,

France²

1. Introduction

Parallel manipulators have numerous advantages in comparison with serial manipulators: Higher stiffness, and connected with that a lower mass of links, the possibility of transporting heavier loads, and higher accuracy. The main drawback is, however, a smaller workspace. Hence, there exists an interest for the research concerning the workspace of manipulators.

Parallel architectures have the end-effector (platform) connected to the frame (base) through a number of kinematic chains (legs). Their kinematic analysis is often difficult to address.

The analysis of this type of mechanisms has been the focus of much recent research. Stewart presented his platform in 1965 [1]. Since then, several authors [2],[3] have proposed a large variety of designs.

The interest for parallel manipulators (PM) arises from the fact that they exhibit high stiffness in nearly all configurations and a high dynamic performance. Recently, there is a growing tendency to focus on parallel manipulators with 3 translational DOF [4, 5, 8, 9, 10, 11, 12, 13,]. In the case of the three translational parallel manipulators, the mobile platform can only translate with respect to the base. The DELTA robot (see figure 1) is one of the most famous translational parallel manipulators [5,6,7]. However, as most of the authors mentioned above have pointed out, the major drawback of parallel manipulators is their limited workspace. Gosselin [14], separated the workspace, which is a six dimensional space, in two parts : positioning and orientation workspace. He studied only the positioning workspace, *i.e.*, the region of the three dimensional Cartesian space that can be attained by a point on the top platform when its orientation is given. A number of authors have described the workspace of a parallel mechanism by discretizing the Cartesian workspace. Concerning the orientation workspace, Romdhane [15] was the first to address the problem of its determination. In the case of 3-Translational DOF manipulators, the workspace is limited to

* Corresponding author. email :lotfi.romdhane@enim.rnu.tn

a region of the three dimensional Cartesian space that can be attained by a point on the mobile platform.

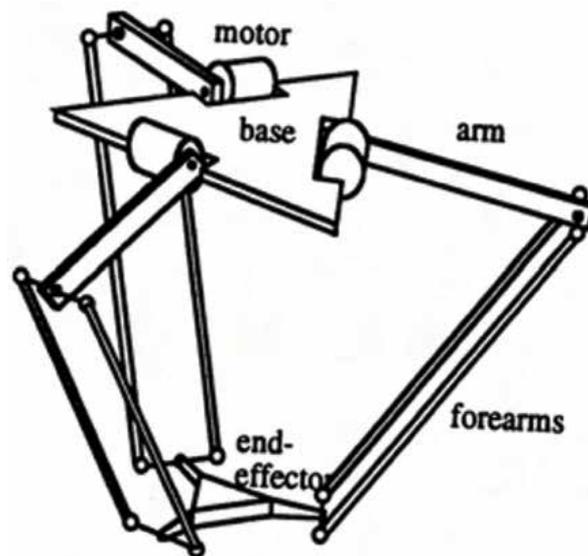


Fig. 1: DELTA Robot (Clavel R. 1986)

A more challenging problem is designing a parallel manipulator for a given workspace. This problem has been addressed by Boudreau and Gosselin [16,17], an algorithm has been worked out, allowing the determination of some parameters of the parallel manipulators using a genetic algorithm method in order to obtain a workspace as close as possible to a prescribed one. Kosinska *et al.* [18] presented a method for the determination of the parameters of a Delta-4 manipulator, where the prescribed workspace has been given in the form of a set of points. Snyman *et al.* [19] propose an algorithm for designing the planar 3-RPR manipulator parameters, for a prescribed (2-D) physically reachable output workspace. Similarly in [20] the synthesis of 3-dof planar manipulators with prismatic joints is performed using GA, where the architecture of a manipulator and its position and orientation with respect to the prescribed workspace were determined.

In this paper, the three translational DOF DELTA robot is designed to have a specified workspace. The genetic algorithm (GA) is used to solve the optimization problem, because of its robustness and simplicity.

This paper is organized as follows: Section 2 is devoted to the kinematic analysis of the DELTA robot and to determine its workspace. In Section 3, we carry out the formulation of the optimization problem using the genetic algorithm technique. Section 4 deals with the implementation of the proposed method followed by the obtained results. Finally, Section 5 contains some conclusions.

2. Kinematic analysis and workspace of the DELTA robot

2.1 Direct and inverse geometric analyses

The Delta robot consists of a moving platform connected to a fixed base through three parallel kinematic chains. Each chain contains a rotational joint activated by actuators in the

base platform. The motion is transmitted to the mobile platform through parallelograms formed by links and spherical joints (See Figure 2).

We assume that all the 3 legs of the DELTA robot are identical in length. The geometric parameters of the DELTA robot are then given as: $L_1, L_2, r_A, r_B, \theta_j$ for $j = 1, 2, 3$ defined in Figure 2, as well as $\varphi_{1j}, \varphi_{2j}, \varphi_{3j}$ for $j = 1, 2, 3$ the joint angles defining the configuration of each leg. Let P be a point located on the moving platform, the geometric model can be written as :

$$\begin{aligned} X_P &= X_A + \cos \theta_j (L_2 \cos \varphi_{1j} + L_1 \cos \varphi_{3j} \cos (\varphi_{1j} + \varphi_{2j}) - r_A) \\ &\quad - L_1 \sin \theta_j \sin \varphi_{3j} \end{aligned} \quad (1)$$

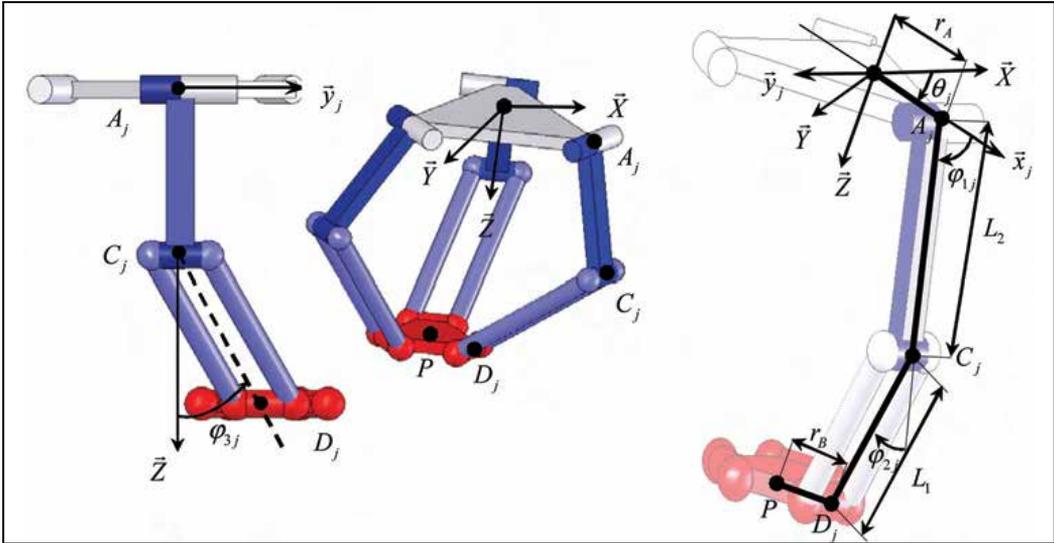


Fig. 2: The DELTA robot parameters.

$$\begin{aligned} Y_P &= Y_A + \sin \theta_j (L_2 \cos \varphi_{1j} + L_1 \cos \varphi_{3j} \cos (\varphi_{1j} + \varphi_{2j}) - r_A) \\ &\quad + L_1 \cos \theta_j \sin \varphi_{3j} \end{aligned} \quad (2)$$

$$Z_P = L_2 \sin \varphi_{1j} + L_1 \cos \varphi_{3j} \sin (\varphi_{1j} + \varphi_{2j}) \quad (3)$$

With $j = 1, \dots, 3$

Where $[X_P Y_P Z_P]$ are the coordinates of the point P .

In order to eliminate the passive joint variables we square and add these equations, which yields :

$$\begin{aligned} [(r + L_2 \cos \varphi_{1j}) \cos \theta_j - X_P]^2 &+ [(r + L_2 \cos \varphi_{1j}) \sin \theta_j - Y_P]^2 \\ &+ [-L_2 \cos \theta_j - Z_P]^2 - L_1^2 = 0 \end{aligned} \quad (4)$$

Where $j = 1, \dots, 3$ and $r = r_A - r_B$.

2.1.1 The direct geometric model

The direct problem is defined by (4), where the unknowns are the location of point $P = [X_P, Y_P, Z_P]$ for a given joint angles $\varphi_{1j}, \varphi_{2j}, \varphi_{3j}$ ($j = 1, \dots, 3$).

This equation can be put in the following form:

$$(X_P - X_j)^2 + (Y_P - Y_j)^2 + (Z_P - Z_j)^2 = L_1^2 \quad (5)$$

where,

$$\begin{cases} X_j = r + L_2 \cos \varphi_{1j} \\ Y_j = r + L_2 \sin \varphi_{1j} \\ Z_j = -L_2 \cos \theta_j \end{cases} \quad (6)$$

Equation (5) represents a sphere centred in point $S_j [X_j, Y_j, Z_j]$ and with radius L_1 .

The solution of this system of equations can be represented by a point defined as the intersection of these three spheres. In general, there are two possible solutions, which means that, for a given leg lengths, the top platform can have two possible configurations with respect to the base. For more details see ref [21].

2.1.2 Inverse geometric model

The inverse problem is defined by (4), where the unknowns are the joint angles $\varphi_{1j}, \varphi_{2j}, \varphi_{3j}$ ($j = 1, 2, 3$) for a given location of the point $P = [X_P, Y_P, Z_P]$.

$$(2rL_2 - 2L_2X_P \cos \theta_j - 2L_2Y_P \sin \theta_j) \cos \varphi_{1j} - 2rX_P \cos \theta_j + 2L_2Z_P \sin \varphi_{1j} - 2rY_P \sin \theta_j + X_P^2 + r^2 + L_2^2 + Z_P^2 + Y_P^2 - L_1^2 = 0 \quad (7)$$

which can be written as:

$$l_j \cos \varphi_{1j} + m_j \sin \varphi_{1j} - n_j = 0 \quad (8)$$

Where,

$$\begin{cases} l_j = 2rL_2 - 2L_2X_P \cos \theta_j - 2L_2Y_P \sin \theta_j \\ m_j = 2L_2Z_P \\ n_j = -2rX_P \cos \theta_j - 2rY_P \sin \theta_j + X_P^2 + r^2 + L_2^2 + Z_P^2 + Y_P^2 - L_1^2 \end{cases} \quad (9)$$

Equation (8) can have a solution if and only if:

$$\left| \frac{n_j}{\sqrt{l_j^2 + m_j^2}} \right| \leq 1 \Leftrightarrow n_j^2 - (l_j^2 + m_j^2) \leq 0 \quad (10)$$

For more details on the inverse geometric model of the DELTA robot see [21,22,23].

2.2 Workspace of the DELTA robot

The workspace of the DELTA robot is defined as a region of the three-dimensional cartesian space that can be attained by a point on the platform where the only constraints taken into account are the ones coming from the different chains given by Equations (10). Equation (10) can be written as:

$$h_j(X_P, Y_P, Z_P) = \left(\begin{array}{c} (X_P \cos \theta_j + Y_P \sin \theta_j - r)^2 \\ + (X_P \sin \theta_j - Y_P \cos \theta_j)^2 + Z_P^2 + L_2^2 - L_1^2 \end{array} \right)^2 - 4L_2^2 \left((X_P \cos \theta_j + Y_P \sin \theta_j - r)^2 + Z_P^2 \right) \leq 0 \quad (11)$$

Equation (11) in cartesian coordinates for a torus azimuthally symmetric about the y -axis can be written as follows :

$$\left((x - r)^2 + y^2 + z^2 + a^2 - b^2 \right) = 4d^2 (x^2 + z^2) \quad (12)$$

Where, $a = L_2$ and $b = L_1$

The set of points P satisfying $h_j(X_P, Y_P, Z_P) = 0$ are the ones located on the boundary of this workspace. This volume is actually the result of the intersection of three tori. Each torus is centered in point $O_j(r \cos \theta_j, r \sin \theta_j, 0)$ and with a minor radius given by L_2 and a major radius given by L_1 . Figure 3 shows the upper halves of these tori. In the following, we will be interested only in the upper half of the workspace.

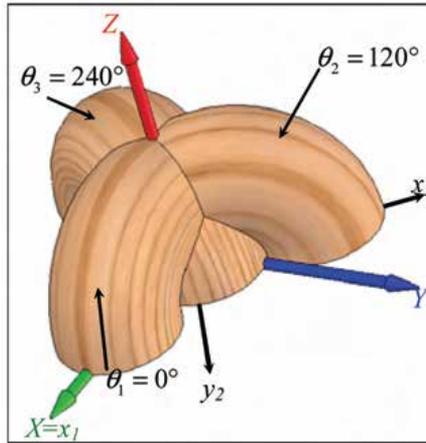


Fig. 3: The three upper halves of the tori given by $h_j(P) = 0$

Therefore, one can state that for a given point $P(X_P, Y_P, Z_P)$:

if P is inside the workspace then $h_j(P) < 0$ for $j = 1, 2, 3$.

if P is on the boundary of the workspace then $h_j(P) \leq 0$ for $j = 1, 2, 3$ and $h_j(P) = 0$ for $j = 1$ or $j = 2$ or $j = 3$.

if P is outside the workspace then $h_j(P) > 0$ for $j = 1$ or $j = 2$ or $j = 3$.

3. Dimensional synthesis of the DELTA robot for a given workspace

3.1 Formulation of the problem

The aim of this section is to develop and to solve the multidimensional, non linear optimization problem of selecting the geometric design variables for the DELTA robot having a specified workspace. This specified workspace has to include a desired volume in space, W . This approach is based on the optimization of an objective function using the genetic algorithm (GA) method.

The dimensional synthesis of the DELTA robot for a given workspace can be defined as follows:

Given : a specified volume in space W .

Find : the smallest dimensions of the DELTA robot having a workspace that includes the specified volume.

For example if the specified volume is a cube, then the workspace of the DELTA robot has to include the given cube.

The optimization problem can be stated as:

$$\min F(I)$$

Subject to

$$h_j(I, P) \leq 0 \text{ for all the points } P \text{ inside the specified volume } W. \quad (13)$$

$$x_i \in I$$

$$x_i \in [x_{i\min} \ x_{i\max}]$$

h_j : are the constraints applied on the system.

I : is a vector containing the independent design variables.

x_i , is an element of the vector I , called individual in the genetic algorithm technique.

$x_{i\min}$ and $x_{i\max}$ are the range of variation of each design variable.

If the volume can be defined by a set of vertices P_k ($k = 1, N_{pt}$), then the desired volume W is inside the workspace of the DELTA robot if:

$$h_j(I, P_k) \leq 0, j = 1, \dots, 3 \text{ and } k = 1, \dots, N_{pt}$$

In this work, we will take the case where W is a cube given by $N_{pt} = 8$ points (see Figure 4). For every workspace to be generated by a DELTA robot, the independent design variables are:

$$h_j(I, P_k) \leq 0, j = 1, \dots, 3 \text{ and } k = 1, \dots, N_{pt} \quad (14)$$

Where H is a parameter defining how far is the specified volume from the base of the DELTA robot (see Figure 4). This function h_j when applied to a point can be used as a measure of some kind of distance of this point with respect to the surface defined by $h_j = 0$. In geometry, this function is called the power of the point with respect to the surface. In the plane, $h_j = 0$ defines a curve. Annex I presents some theoretical background about the power of a point with respect to a circle. Moreover, the function h_j changes its sign depending on which side of the surface the point is located. Therefore minimizing the function $|h_j(P)|$, is

equivalent to finding the closest point to the given surface. In our case, we are looking for a volume bounded by three surfaces, therefore one has to minimize the function $f = |h_1(I, P)| + |h_2(I, P)| + |h_3(I, P)|$. Figure 5 represents a mapping $f(x, y)$, of the power of points at a given height $z_0 = 1$ as a function of x and y for a given design vector $I = [1.9, 1.2, 0.9, 1]$.

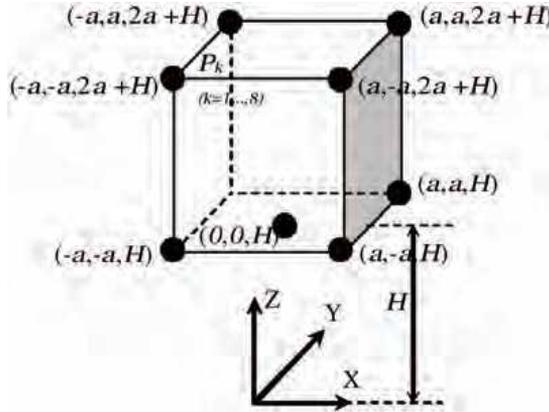


Fig. 4: The scheme of the prescribed workspace.

The function f is given by:

$$f = |h_1(I, P)| + |h_2(I, P)| + |h_3(I, P)|$$

One can notice that the minimum value of f is obtained when the point is located on the boundary of the workspace (see Figure 5).

Our objective is to find the smallest set of parameters, given by $I = [L_1, L_2, r, H]$ that can yield a DELTA robot having a workspace that includes the given volume in space W .

The methodology followed to solve this problem is based on minimizing the power of the vertices, defining the given volume, and to ensure that all these vertices have a negative power, i.e., they are inside the workspace of the DELTA robot. This minimization problem will be solved using the GA method.

It is worth noting that this procedure is valid for any convex volume defined by a set of vertices.

3.2 GA optimization

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution [24]. GAs operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation. The GA differs substantially from more traditional search and optimization methods. The four most significant differences are:

- GAs search a population of points in parallel, not a single point.

- GAs do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- GAs use probabilistic transition rules, not deterministic ones.
- A number of potential solutions are obtained for a given problem and the choice of final solution can be made, if necessary, by the user.

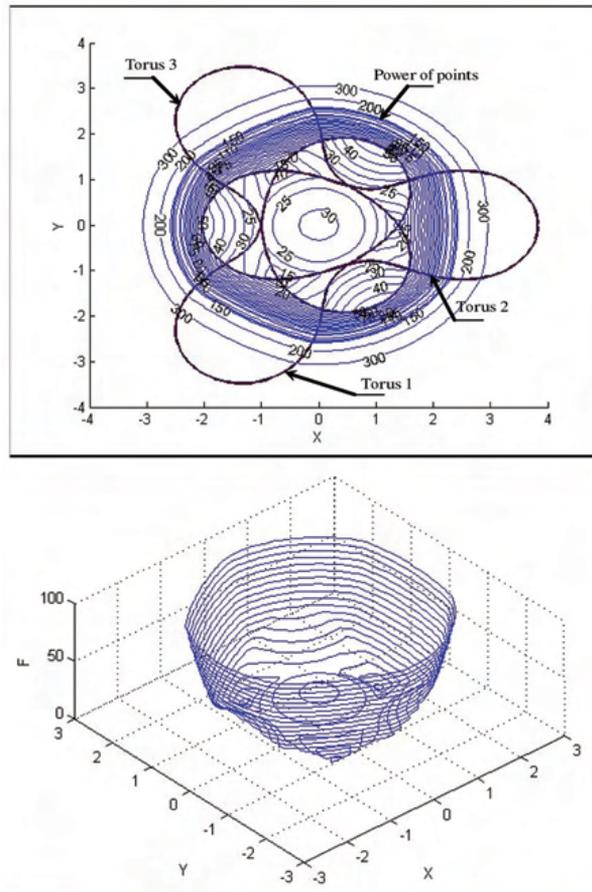


Fig. 5: Graphical representation of the power of a point $F(X, Y)$.

In most applications involving GAs, binary coding is used. However, Wright [32] showed that real-coded GAs have a better performance than binary-coded GAs [25,26,27,28,29]. A real-coded GA is used in this work. The description of the operations necessary for this type of code are presented by Figure 6, more details can be found in [30]. The parameters used in this work are shown in Table 1.

A penalty function method is used to handle the constraints and to ensure that the fitness of any feasible solution is better than infeasible ones.

The fitness function is constructed as:

$$F = -(F_1 + F_2) \quad (15)$$

Where F_1 is a penalty function defined as follows:

$$F_1 = \sum_{k=1}^{N_{pt}} \sum_{j=1}^3 \varphi_j(I, P_k) \quad (16)$$

where

$$\varphi_j(I, P_k) = \begin{cases} 0 & \text{if } h_j(I, P_k) \leq 0 \\ cf & \text{if } h_j(I, P_k) > 0 \end{cases} \quad (17)$$

Where, cf is a large positive constant.

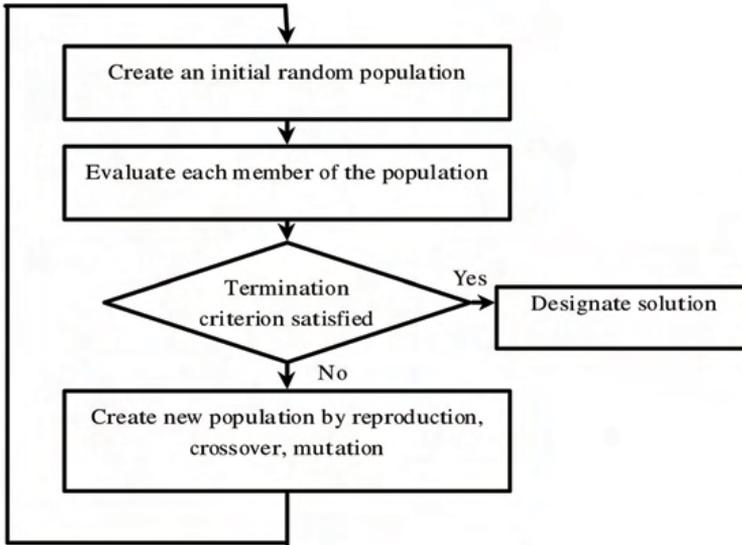


Fig. 6: Genetic algorithm flowchart.

The population size	N_{ind} :	100
The maximal generation number	$M_{ax}G_{en}$:	100
The variable number	N_{var} :	4
The generation gap	G_{gap} :	0.85
The mutation rate	F_1 :	0.64
The shrinking mutation factor	F_2 :	0.615
The recombination rate	Opt_1 :	0.623
The shrinking recombination factor	Opt_2 :	0.553

Tab. 1: Parameters used for the genetic algorithm.

$F_1 = 0$ means that all the vertices defining the volume W are contained within the workspace of the DELTA robot. In this case, the fitness F_2 is given by

$$F_2 = \sum_{k=1}^{N_{pt}} \sum_{j=1}^3 |h_j(I, P_k)|$$

In the case when $F_1 \neq 0$, i.e., at least one of the vertices is outside the workspace, F_2 is set to zero ($F_2 = 0$).

4. Results

All the results, presented in this section, are obtained on a Pentium M processor of 1500 Mhz and the programs are developed under MATLAB . The calculation time, necessary for obtaining the optimum solution, is estimated at about 4s.

4.1 Example 1

In this example, the dimensions of the DELTA robot are to be determined to get the smallest workspace capable of containing a volume W , given by a cube with a side $2a = 2$ (Figure 4). The bounding interval for each one of the design variables is presented in Table 2:

I	L_1	L_2	r	H
x_{\min}	0.5	0.5	0	1
x_{\max}	5	5	5	5

Tab. 2: The bounding interval for design variables

The optimal solution obtained by the GA for this example is presented in Table 3:

	L_1	L_2	r	H
I	2.66	1.73	1.74	1.01

Tab. 3: The optimal dimension of DELTA robot (example 1)

Figure 7 presents a mapping, f , of the power of points at a given height equal to 1.01 as a function of x and y for the optimal solution. A 3D representation of the platform and the corresponding workspace along with the desired volume W , is shown on Figure 8. Figure 9 presents horizontal slices of the workspace at the lower and upper faces of the cube. One can notice that the upper vertices of the cube are exactly located on the boundary of the workspace; which means that the robot has to be in an extreme position (on the boundary of the workspace) to be able to reach these points. To avoid this problem, we propose to design a robot having a slightly bigger workspace defining this way a safety region. The following example illustrates this problem.

4.2 Example 2

In this second example, a distance is kept between the workspace of the DELTA robot and the desired volume. To have this safety region, we used the fact that a safety distance can be kept, during the optimization, between each vertex and the surface defining the boundary of the workspace. This safety distance can be translated in terms of the power of the point, which means that, during the optimization, a lower bound is set on the powers of all points. This lower bound ensures that in the final solution no point can be on the surface defining the boundary of the workspace, i.e., the power is zero in that case, but rather on a surface parallel to the boundary of the workspace. The distance between these two surfaces is defined as the safety distance.

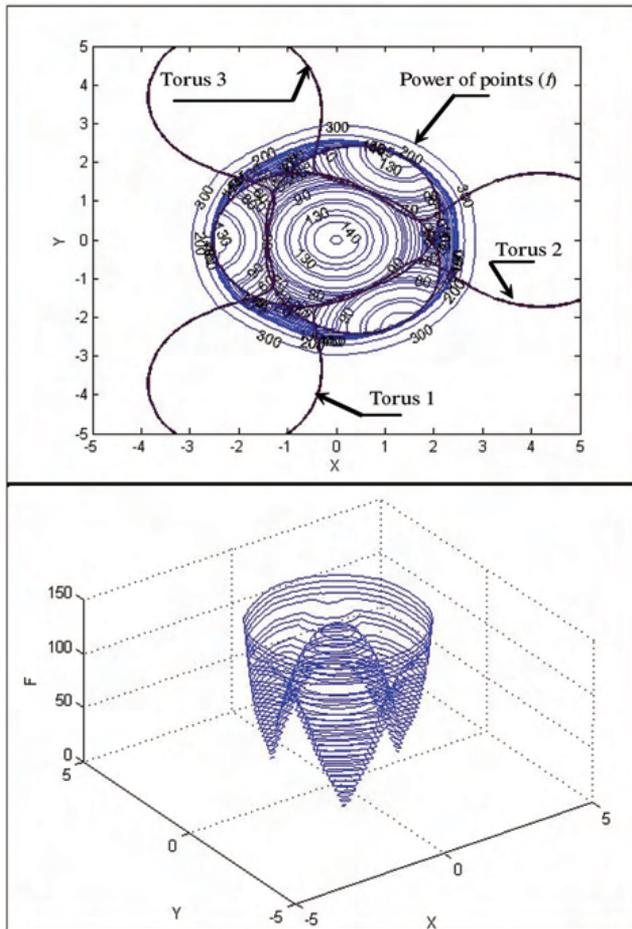


Fig. 7: Graphical representation of the power of a point $F(X, Y)$ (example 1).

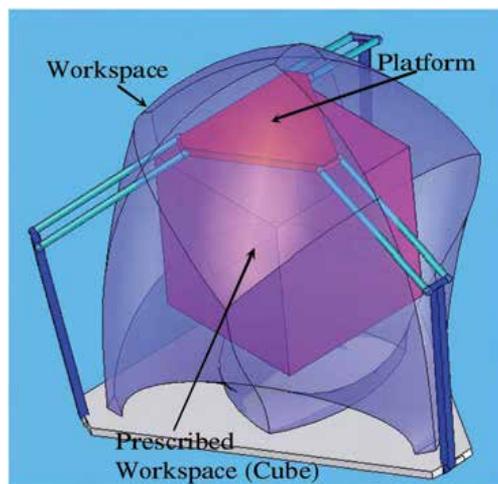


Fig. 8: The Optimal DELTA robot for example 1.

In our case, the workspace is the intersection of three tori, each with L_1 as a major radius and L_2 as the minor radius. Therefore, the three corresponding tori, each with a major radius L_1 and a minor radius $L_2 - e$, define a more restrictive volume of the workspace. The intersection of the smaller tori is now the bounding volume within which the desired volume W has to be located. In this case, we took $e = 0.1L_2$.

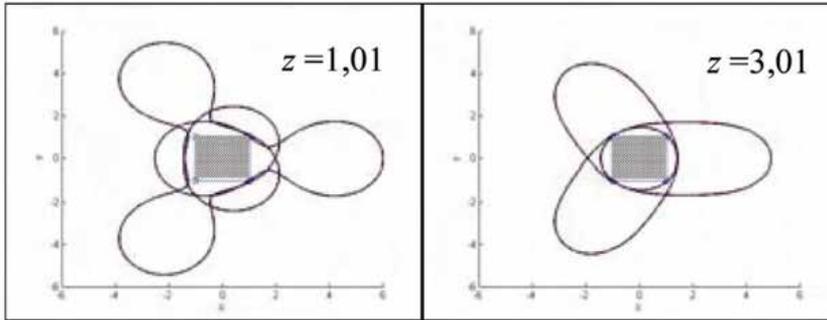


Fig. 9: Two slices of the workspace at the top and bottom of the cube.

The new optimal solution found for the DELTA robot is given by Table 4. One can notice that L_1 and r decreased, whereas L_2 increased, compared to the previous example. The height of the cube with respect to the base, H , stayed almost the same.

Figure 10 shows slices at the upper and lower faces of the cube of the workspace and the corresponding safety region. Figure 11 shows two cuts of the workspace with the cube inside it. One can notice that the vertices of the cube are kept at a minimum distance given by the safety distance e .

A 3D representation of the platform and the corresponding workspace along with the desired volume W , is shown on Figure 12. One can notice that all the points of the cube can be reached by the platform without reaching an extreme configuration as it was the case, in the previous example.

	L_1	L_2	r	H
I	2.5	1.91	1.48	1.02

Tab. 4: The optimal dimension of DELTA robot with safety zone (example 2)

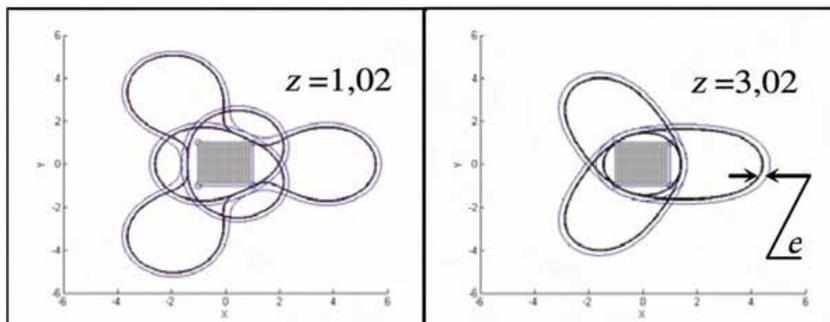


Fig. 10: Two slices of the workspace at the top and bottom of the cube with a safety zone.

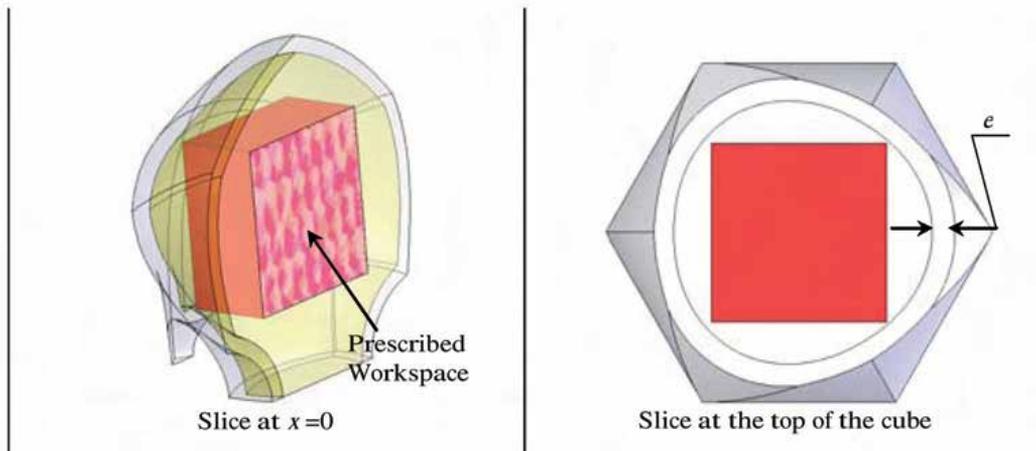


Fig. 11: Different slices of the Workspace.

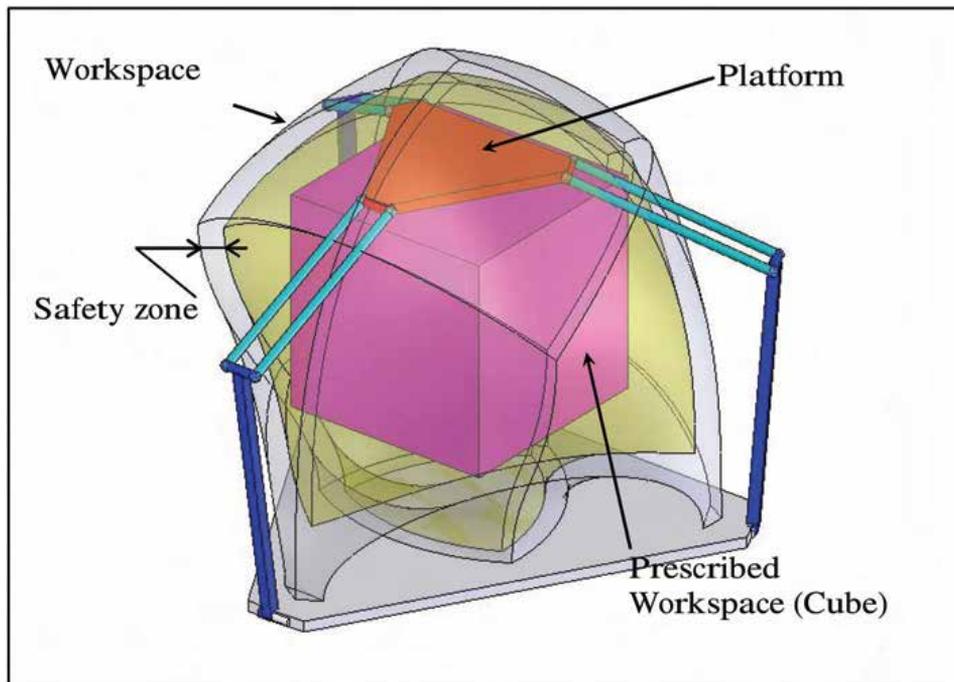


Fig. 12: The Optimal DELTA robot.

4.3 Example 3

In this example we propose an hexagonal prism as a prescribed workspace, given by $N_{pt} = 13$ points (see figure 13). The dimensions of the DELTA robot are to be determined to get the smallest workspace capable of containing a volume \mathbf{W} , given by an hexagonal prism with a side $\mathbf{b} = 1$. The bounding interval for each one of the design variables is presented in Table 5:

I	L_1	L_2	r	H
x_{\min}	0.5	0.5	0	1
x_{\max}	7	7	7	7

Tab. 5: The bounding interval for design variables

Figure 14 and 15 present a mapping, f , of the power of a point at a given height equal to 1.67 as a function of x and y for the optimal solution obtained by the GA presented in Table 6. A 3D representation of the platform and the corresponding workspace along with the desired volume W , are shown on Figure 16.

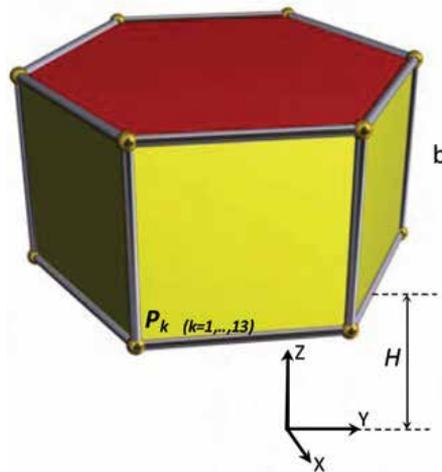


Fig. 13: The scheme of an hexagonal prism prescribed workspace.

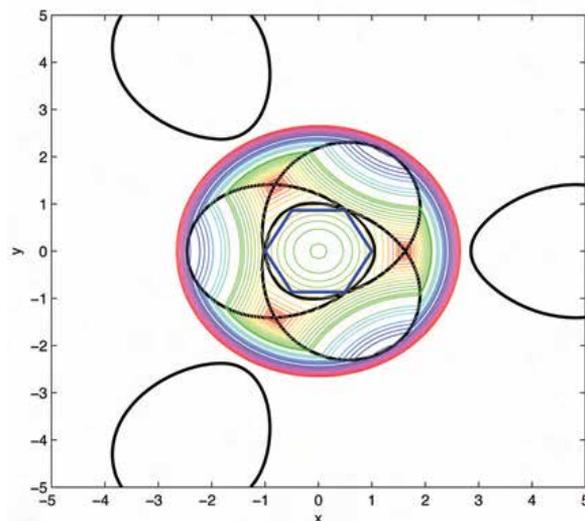


Fig. 14: Graphical representation of the power of a point $F(X, Y)$ for an hexagonal prism prescribed workspace.

	L_1	L_2	r	H
I	3.32	1.4	1.95	1.67

Tab. 6: The optimal dimension of the DELTA robot with a hexagonal prism as a prescribed workspace

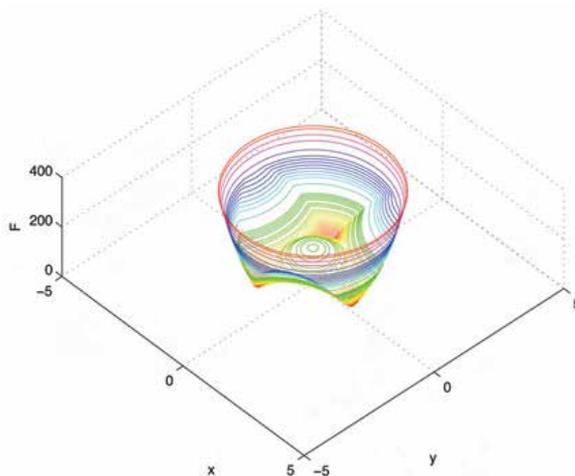


Fig. 15: Graphical representation of the power of a point $F(X, Y)$ for a hexagonal prism as a prescribed workspace.

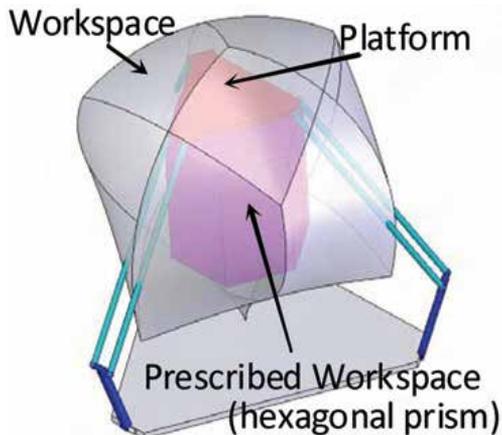


Fig. 16: The Optimal DELTA robot for an hexagonal prism as a prescribed workspace.

5. Conclusion

An optimal dimensional synthesis method suited for the DELTA robot was presented in this paper. An objective function, used the concept of the power of a point, which reflects the position of a point with respect to the boundary of the workspace. This objective function allowed us to find the robot having the smallest workspace containing a prescribed region.

The genetic algorithm method was used. The prescribed region was chosen as a cube then as a hexagonal prism. The obtained solution yields a workspace where some of the vertices of the cube or the hexagonal prism are located on the boundary of the workspace. To reach these points the DELTA robot has to get into extreme configurations. To avoid this problem, we introduced a safety distance allowing us to have all the prespecified region inside the workspace. The concept of the power of a point along with the GA method turned out to be an effective and easy tool to solve the problem of designing a DELTA robot for a specified workspace. This method can also be applied, in a similar manner, to any convex prismatoid prespecified region of the workspace.

6. References

- D. Stewart, 1965, "A platform with 6 degrees of freedom", Proc. of the Institution of mechanical engineers, Vol. 180 (Part 1, 15), pp. 371-386, 1965.
- E. F. Fichter, 1986, "A Stewart platform based manipulator: general theory and practical construction", International Journal of Robotic Research, Vol. 5, pp. 157-182.
- Griffis, M., and Duffy, J., "A Forward Displacement Analysis of a Class of Stewart Platforms," Trans. ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 6, No. 6, June 1989, pp. 703-720.
- Affi Z., Romdhane L. and Maalej A., 2004. Dimensional synthesis of a 3-translational-DOF in-parallel manipulator for a desired workspace. European Journal of Mechanics - A/Solids, Vol 23, Issue 2, pp 311-324.
- Clavel, R. 1986. Une nouvelle structure de manipulation parallèle pour la robotique légère. R.A.I.R.O. APII, Vol 23, N° 6.
- Vischer P. and Clavel R. 1998, "Kinematic Calibration of the Parallel Delta Robot", Robotica, 16, pp. 207-218.
- M. Stock and K. Miller 2003, "Optimal Design of Spatial Parallel Manipulators: Application to linear Delta Robot", ASME Journal of Mechanical Design, Vol. 125, pp 292-301.
- Hervé J. M. 1995, "Design of Parallel Manipulators via Displacement Group", Proceedings of the 9th World Congress on the Theory of Machines and Mechanisms. pp. 2079-2082.
- Hervé, J. M., Sparacino F. 1991. Structural synthesis of parallel robots generating spatial translation. 5th Int. Conf. On Adv. Robotics, IEEE n°91TH0367-4, Vol. 1, pp 808-813.
- Romdhane, L. 1999, Design and analysis of a hybrid serial-parallel manipulator. Mechanism and Machine Theory, Vol. 34, Issue 7, pp 1037-1055.
- Romdhane, L., Affi Z., Fayet M., 2002. Design and singularity analysis of a 3 translational-DOF in-parallel manipulator. ASME Journal of Mechanical Design, Vol. 124, pp 419-426.
- A. Tremblain and L. Baron 1999, "Geometrical synthesis of parallel manipulators of star-like topology with a geometric algorithm", IEEE International Conference on Robotics and Automation, Detroit, MI.
- Tsai L-W 1996, "Kinematics of three-dof platform with three extensible limbs" In J. Lenarcic V. Parenti-Castelli, editor, Recent Advances in Robot Kinematics, pp 401 410, Kluwer.
- C. Gosselin, 1990, "Determination of the workspace of 6-dof parallel manipulators", ASME Journal of Mechanical Design, Vol. 112, pp. 331-336.

- L. Romdhane, 1994, "Orientation workspace of fully parallel mechanisms", *Eur. J. of Mechanics* Vol. 13, pp. 541-553.
- R. Boudreau and C. M. Gosselin 1999, "The synthesis of planar parallel manipulators with a genetic algorithm", *ASME Journal of Mechanical Design*, Vol 121, pp 533-537.
- R. Boudreau and C. M. Gosselin 2001, "La synthèse d'une plate forme de Gough-Stewart pour un espace de travail atteignable prescrit", *Mech. Mach. Theory* 36 (2001) 327-342.
- Kosinska, A, Galicki, M. and Kedzior, K. 2003, "Design and optimization of parameters of Delta-4 Parallel Manipulator for a Given Workspace", *Journal of Robotic Systems* 20 (9), pp 539-548.
- J. A. Snyman and A. M. Hay 2005, "Optimal synthesis for a continuous prescribed dexterity interval of 3-DOF parallel planar manipulator for different prescribed output workspaces", *Proceeding of CK2005, 12th International Workshop on Computational Kinematics Cassino May 4-6*.
- M. Gallant and R. Boudreau 2002, "The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free workspace", *Journal of Robotic Systems* 19 (1), pp 13-24.
- F. Pierrot, C. Reynaud and A. Fourrier 1990, "DELTA : a simple and efficient parallel robot", *Robotica* Vol. 8, pp 105-109.
- Goudali, A. 1995. Contribution à l'étude d'un nouveau robot Parallèle 2- Delta à six degrés de liberté avec découplage. Thèse de doctorat Génie Mécanique L.M.S. Poitiers. France.
- J.P. Lallemand, A. Goudali and S. Zeghloul, "The 6 - D.o.f. 2 - Delta parallel robot" , *Robotica Journal*, Vol. 15, pp 407-416, 1997.
- Goldberg, D.E., 1994, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing, Reading, MA.
- Chipperfield A., Fleming P., Pohlheim H. and Fonseca C. 1994, "Genetic Algorithm TOOLBOX user's Guide" Department of automatic control and systems engineering university of Sheffield version (v 1.2)
- J.A. Lozano , P. Larranaga, M. Grana, F.X. Albizuri 1999, Genetic algorithms: bridging the convergence gap, *Theoretical Computer Science* Vol. 229, pp 11-22.
- R. Chelouah, P. Siarry 2000, A continuous genetic algorithm designed for the global optimization of multimodal functions, *Journal of Heuristics* Vol. 6, pp. 191-213.
- Schmitt L. M. 2001, *Fundamental Study Theory of genetic algorithms*, *Theoretical Computer science* n°259 pp 1-61.
- Laine, R., Zeghloul, S., Ramirez, G., 2002, A Method based on a Genetic Algorithm for the Optimal Design of Serial Manipulators, *Int. Symp. Rob. and Aut.*, pp. 15-20, Toluca, Mexique.
- M.A. Laribi, A. Mlika, L. Romdhane and S. Zeghloul, 2004, "A Combined Genetic Algorithm-Fuzzy Logic Method (GA-FL) in Mechanisms Synthesis", *Mech. Mach. Theory* 39, pp. 717-735.
- Coxeter, H. S. M. 1969, "Introduction to Geometry", 2nd ed. New York: Wiley.
- Alden H. Wright, 1991, "Genetic algorithms for real parameter optimization, *Foundations of Genetic Algorithms*", (edited by Gregory J. E. Rawlins), Morgan Kaufman, pp. 205-218.
- Steiner, J., 1826, "Einige geometrische Betrachtungen." *J. reine angew. Math.* 1, pp. 161-184.

A Appendix

The power of a fixed point A (see Figure 17) with respect to a circle of radius r and center O is defined by the product

$$f(A) \equiv AP \cdot AQ$$

Where, P and Q are the intersections of a line through A with the circle. The term "power" was first used in this way by Jacob Steiner [33,31]. $f(A)$ is independent of the choice of the line APQ .

Now consider a point A (see Figure 17) not necessarily on the circumference of the circle. If $d = OA$ is the distance between A and the circle's center O with equation $f(x, y) = x^2 + y^2 - r^2 = 0$, then the power of the point A relative to the circle is given by :

$$f(A) = f(x_A, y_A) = x_A^2 + y_A^2 - r^2 = d^2 - r^2$$

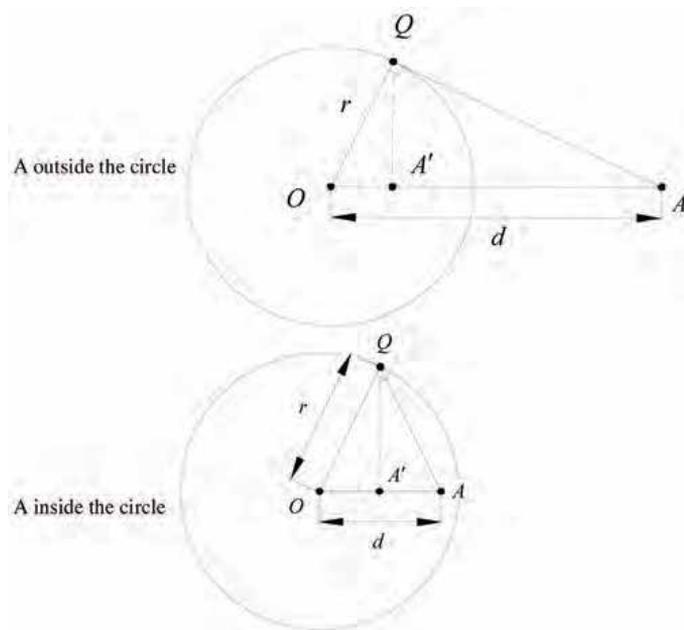


Fig. 17: The power of the point.

If A is outside the circle, its power is positive and it is equal to the square of the length of the segment AQ from A to the tangent Q to the circle through A ,

$$f(A) = AQ^2$$

If A is inside the circle, then the power is negative.

Size-adapted Parallel and Hybrid Parallel Robots for Sensor Guided Micro Assembly

Kerstin Schöttler, Annika Raatz and Jürgen Hesselbach
*Technische Universität Braunschweig, Institute of Machine Tools and Production
Technology (IVF), Langer Kamp 19 B, D-38106 Braunschweig
Germany*

1. Introduction

Miniaturized products and components are part of today's daily life. The comfort and security of automobiles is increased by use of micro sensors and actuators. Electronic devices, such as mobile phones and MP3-players, have reached very small sizes and miniaturized medical instruments facilitate endoscopic surgery.

Due to the advantages of micro technological solutions, such as small dimensions and low weight, Micro Systems Technology (MST) is worldwide considered a key technology of the 21st century. The new NEXUS market analysis forecasts a yearly growth of the world markets of 16% for products based on MST (Wicht & Bouchaud, 2005).

Miniaturization and simultaneous function integration are leading to increased requirements regarding production technology as a result of scaling effects, technical and assembly related problems (van Brussel et al., 2000). For MST products, micro assembly uncertainties in the range of a few micrometers or even less than one micrometer are required.

1.1 Approaches to meet the requirements for micro assembly

At present industrial applications for micro assembly predominantly incorporate systems which were originally developed for 2D chip assembly in semi-conductor back-end production. They can be classified into three groups according to their attainable assembly uncertainty. Most of the positioning units of the first class are pick-and-place machines based on Cartesian axes with uncertainties between 30 μm and 60 μm at 3σ . The second group, die-bonding machines, reaches pick-and-place uncertainties of 10 μm to 12 μm at 3σ by means of high-precision linear drives, high-resolution camera systems as well as systems for controlling and compensating for influences caused by changing temperatures. Ultra-precision die-bonders form the third class. They can be regarded as special machines for specific applications which were developed for the assembly of micro-optical components, optical fibres and especially for flip-chip assembly. They reach assembly or pick-and-place uncertainties of about 1 μm at 3σ . These low uncertainties can only be achieved with the help of special camera systems and positioning strategies. At present, these assembly uncertainties are always tied to a highly customized design of the assembly system adjusted

to the requirements of the products. This way the assembly uncertainties described are reached at the expense of a very low flexibility (Raatz & Hesselbach, 2007).

For the design of micro assembly systems it is necessary to gain a high product flexibility of the assembly units. Solutions that provide enough flexibility to reconfigure the system design need to be found. Here, modularity is the key when striving for high flexibility of the number of quantities, product variants and manufacturing base. The precision robot represents the central component within the assembly system. Some fundamental techniques to lower the uncertainty of the precision robot and the assembly system are choosing an adequate kinematic structure, developing size adapted handling devices, integrating ultra-precision machine elements and/or using sensor guidance (Fig. 1).

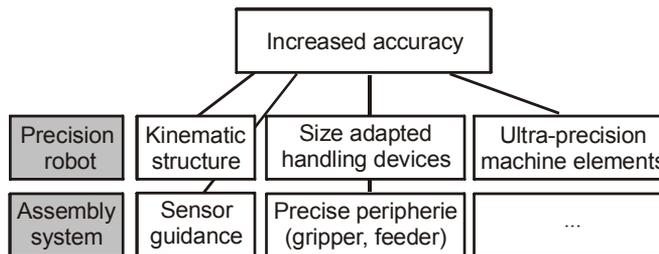


Fig. 1. Approaches to meet the requirements of accuracy (Raatz & Hesselbach, 2007)

1.2 Kinematic structures

Robots can be classified in terms of their kinematic structure into serial, parallel and hybrid (serial/parallel) robots. Most industrial robots are based on a serial structure between the frame and the working platform. All joints of open kinematic chains have a single degree of freedom (DOF) and are active, i.e. they are actuated. The serial structure offers in principle a large workspace in relation to the size of the robot as well as a high orientation range. The relatively large moved masses are a disadvantage of serial structures regarding the dynamics and accuracies of the robot, since each drive must be moved along with the entire kinematic chain. In micro assembly, large moved masses lead to massive construction of the frames and the robot links related to the size of the assembled parts.

Parallel robots are based on closed kinematic chains, i.e. they have several guiding chains between the base frame and the working platform, which provide a high structural stiffness. It is possible to install all drives in a fixed frame or at least to locate them nearby the frame, which results in low inertia. Drive positioning errors or tolerances in the legs are not necessarily added. Usually they partially compensate each other and only affect the positioning uncertainty of the end effector to a small extent.

Parallel robots are well suited for highly precise handling operations, due to their high structural stiffness with low moved masses at the same time. Compared to serial robots, the miniaturization of a parallel robot is much easier because all joints are passive. In addition the passive joints offer the potential for integrating flexure hinges as ultra-precision machine elements. The small workspace compared to the robot dimensions does not become severe in micro assembly tasks due to the size of the objects.

Combining a parallel structure with a serial structure the limited and position dependent mobility of the end effector can be overcome. For example by integrating a serial rotational axis into the working platform of a parallel robot, the end effector can be very well oriented.

1.3 Robots for micro assembly

A number of commercial robot manufacturers and many research institutions are developing robots which have sufficient positioning uncertainties for micro assembly tasks. Serial, parallel and hybrid robot structures are used. Most serial robots for micro assembly use a Cartesian structure. Often they incorporate modular precision linear axes. In nearly all cases, direct measuring systems are used in order to rule out inaccuracies due to mechanical play. The repeatability of those linear axes lies typically between $0.1\ \mu\text{m}$ and $1\ \mu\text{m}$. Some manufacturers and researchers claim that robots build with these axes reach an overall repeatability of $1\ \mu\text{m}$. A typical exponent of this class of robots is the Sysmelec Autoplace 411 (Fig. 2) (Hesselbach et al., 2005). Another solution for micro assembly robots are conventional Scara robots in combination with redundant high-precision axes in order to reach a high resolution. This approach is always combined with additional sensors to achieve a good repeatability (Höhn, 2001).



Fig. 2. Serial robot Sysmelec Autoplace 411 with Cartesian structure

The development of size-adapted robots is another solution of robots for micro assembly. Saving costs is only possible by reducing the footprint of an assembly system due to the demand of a clean room environment for the production of MST products. In recent years, the reduction of size and costs of micro production systems has been widely discussed in various papers. Most of these concepts relate to one of the two general groups explained in the following.

The first group consists of piezo driven, small walking micro robots and handling machines. These autonomous robots are suitable for positioning small objects such as the MINIMAN of (Fatikow, 2000), a handling device for samples in a scanning electron microscope. On the one hand, these micro robots are very promising for new trends such as nano assembly. By using autonomous robots, difficulties occur regarding the coordination and interaction of these robots, movement on rough surfaces and energy supply.

The second group describes cost-efficient, size-adapted handling devices, which fill the gap between piezo driven, small walking micro robots and conventional robots. A possible solution for this strategy is to determine the highest degree of miniaturization of conventional robot technology, using innovative, miniaturized machine parts. With these

size-adapted handling devices, in the range of several centimeters to a few decimeters, easily scalable and highly flexible production technology can be designed. Examples of size-adapted handling devices are the parallel robot structures Delta³ and Sigma 6 from (Clavel et al., 2005) and the pocket delta from (Coudourey et al., 2006).

This chapter presents a description of four size-adapted robot mechanisms based on parallel and hybrid structures. For three robot mechanisms, the structures are first designed with conventional joints and replaced by flexure hinges as ultra-precision machine element later on.

Furthermore, sensor guided assembly processes of hybrid micro systems are explained on the basis of one robot structure. Thus, the integration of sensor information into the robot control as well as the relative sensor guidance applied in the system will be presented. The positioning uncertainty and the assembly uncertainty of the process are described by means of an example of an assembly process.

2. Size-adapted parallel and hybrid robot structures

Various size-adapted parallel, parallel hybrid and serial hybrid robots for precision assembly were developed at the IWF. The main objective to develop size-adapted robot structures was to adapt the size of the robot cell to the size of the products. At the same time a good repeatability for highly precise micro assembly processes should be reached through the development of parallel and hybrid robot structures.

First, a functional model of a planar robot micabo^e (see section 2.2) with a parallel structure and 3 DOF for movement in the x-y plane and 1 DOF as a serial lifting table for movement in z-direction was implemented. Second, a spatial parallel hybrid robot structure micabo^h (see section 2.3) with 6 DOF was designed. Then, a spatial parallel robot structure, Triglide (see section 2.4), based on a parallel structure with 3 DOF and one serial rotational axis was realized.

These three robots were enhanced by integrating flexure hinges (see section 2.1) as ultra-precision machine elements and named micabo^{es}, micabo^{hs} and Triglide^s. With this machine elements, the conventional joints of the robot structures are replaced.

Based on the experiences with the above mentioned robot structures, the robot micabo^f, which provides 4 DOF for part handling and, as an advanced structure, the micabo^{f2} (see section 2.5), which provides 4 DOF for part handling and 1 additional DOF for focusing a vision sensor, were developed as planar serial hybrid robot structures.

2.1 Pseudo-elastic flexure hinges

One way to increase the accuracy of assembly systems is to enhance the positioning accuracy of the robot itself. Typical problems of parallel structures are the high number of joints and joints with more than one DOF. Backlash, friction and slip-stick effects in conventional joints often decrease the overall precision of the robot. As a result of the natural lack of the above mentioned disadvantages in flexure hinges, replacing conventional joints by flexure hinges seems to be a promising way to increase the accuracy of robots. Since flexure hinges gain their mobility exclusively from a deformation of matter, the attainable angle of rotation is limited. In order to achieve high life cycles of the hinges, the deformation should remain in the elastic part since plastic deformation normally induces defects in the material leading to an earlier crack failure (Hesselbach et al., 2004b).

The developed flexure hinges consist of a pseudo-elastic shape memory alloy (SMA). This material offers larger reversible strains than other materials, e.g. spring steel or thermoplastics, which are commonly used for flexure hinges. Due to the large reversible strains of SMA, deflections of the hinges of $\pm 30^\circ$ are possible. This approach offers the potential to design robots with high accuracy and resolution and with a sufficiently large workspace for micro assembly tasks (Hesselbach et al., 2004a).

SMA exists in austenite and martensite phases, depending on the temperature or the applied stress. The temperature and stress values for stable phases mainly depend on the basic material, their different alloy contents and the thermo-mechanical treatment of the material. The thermally induced phase transformation of SMA (one way effect) is typically used in applications in which the SMA device is used as an actuator. We use the stress induced phase transformation which offers large reversible strains (super-elasticity). Figure 3 shows a stress-strain diagram of pseudo-elastic SMA loaded with uniaxial tensile stress. In its initial condition, the material is in its austenitic phase at room temperature. First, it deforms linear elastic under load. With increasing loads a stress-induced transformation of austenite into martensite is initiated at the pseudo-yield stress R_{pe} . The phase transformation is accompanied by large pseudo-elastic strains ϵ_{pe} with nearly constant stresses. The pseudo-elastic strain is reverted at a lower stress σ_r with a stress hysteresis. Since pseudo-elastic strains are reversible, the specimen completely recovers to its undeformed shape. These strains are often called pseudo-elastic because the reversible deformation is caused by a reversible phase transformation and not only due to a translation of atoms out of their former equilibrium position.

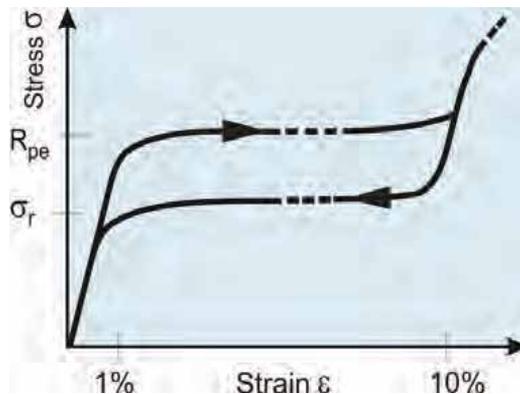


Fig. 3. Stress-strain diagram of shape memory alloy

A pseudo-elastic CuAlNiFe single crystal SMA is used for the design of flexure hinges because of its superior machinability and extremely large reversible strains up to 17%. The uniaxial stress-strain diagram of a CuAlNiFe single crystal has two pseudo-elastic stress plateaus, differing slightly from the example shown in Figure 3, but equivalent in principle. The first plateau, which is the area of interest, has a yield stress of about $R_{pe}=200 \text{ N/mm}^2$ and reversible strains of $\epsilon=10\%$.

A variety of different geometries of flexure hinges are proposed in the literature. They are designed in monolithic or hybrid processes allowing for up to 3 degrees of freedom (DOF) (Smith, 2000), (Paros & Weisbord, 1965). When designing flexure hinges for robots optimisation criteria are high accuracy and a large workspace of the robot. For the three

robots micabo^{es}, micabo^{hs} and Triglide^s (described in section 2.2, 2.3 and 2.4) pseudo-elastic flexure notch hinges with $R=15$ mm and $h=0.15$ mm are used. These geometrical dimensions are an optimum between small kinematic deviations compared to the kinematics of an ideal rotational joint and small occurring strains. The pseudo-elastic material can be modelled with a material model by Prandtl-Reuss if the deflection curve is calculated analytically (Howell & Midha, 1995), (Hesselbach & Ratz, 2000) or with a multilinear elastic material model using the FEM tool ANSYS. With the chosen geometry and geometrical dimensions maximal strains are $\varepsilon=2.1\%$ at deflections of 20° and $\varepsilon=4.2\%$ at 30° .

2.2 Planar parallel robot structures micabo^e and micabo^{es} with 3 DOF and one serial z-axis

The planar parallel robot structure micabo^e (Fig. 4 left) provides 3 DOF. Three linear drives move the platform and the gripper with three guiding chains in x-y-direction and enable a rotation φ around the z-axis. The movement in the direction of the z-axis is performed by an additional elevation platform. The robot is driven by three piezoelectric stick-slip drives with a smallest step size of 5 nm and is equipped with linear encoders with a resolution of $0.1 \mu\text{m}$. First, the passive rotary joints were built with conventional ball bearings. A repeatability of $4 \mu\text{m}$ with 3σ is reached, according to (EN ISO 9283, 1999). The characteristics of the robot micabo^e are shown in Table 1.

Figure 4 (right) shows the planar parallel robot micabo^{es} with 3 DOF in which the 6 conventional rotational joints are replaced by 6 flexure hinges. The motors, step sizes and resolution of the linear encoders are equal to the micabo^e.

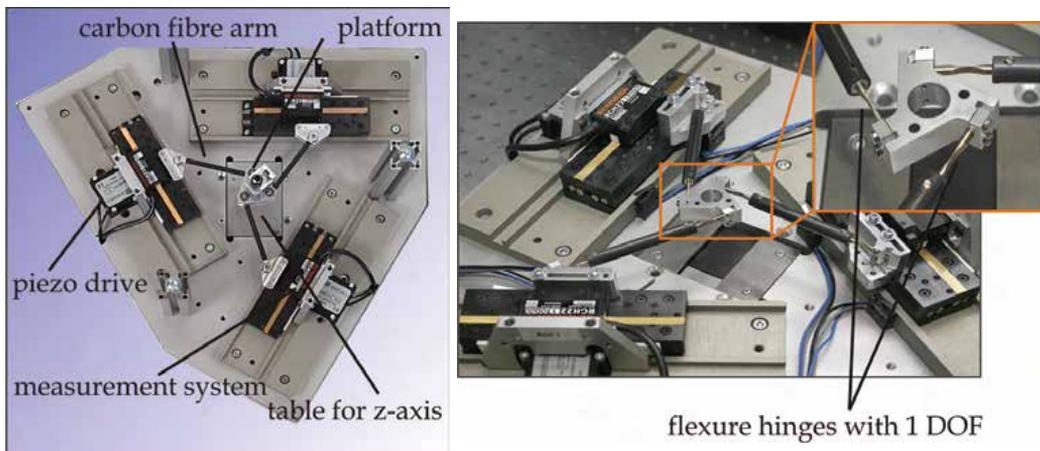


Fig. 4. Planar parallel robot structures micabo^e (left) and micabo^{es} (right)

With an assumed maximal angular deflection of $\pm 30^\circ$ at the hinges and maximal motion ranges of the drives of 70 mm the workspace of the robot is $40 \times 40 \text{ mm}^2$. If flexure hinges with conventional spring steel are used, the angular deflection is restricted to $\pm 5^\circ$. In this case the resulting workspace of the structure is about twenty times smaller compared to the workspace using pseudo-elastic flexure hinges. A repeatability of $1 \mu\text{m}$ with 3σ is reached with the flexure hinges. The characteristics of the robot micabo^{es} are shown in Table 1 as well.

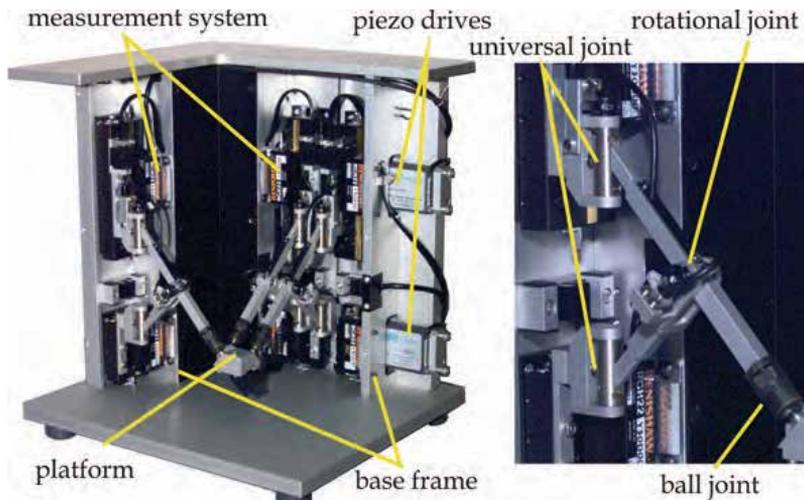
Performance Data	micabo ^e	micabo ^{es}
Max. velocity of the linear drives	0.23 m/s	
Max. velocity of the end effector	0.15 m/s	
Payload	0.1 kg	
Smallest step size	5 nm	
Resolution of linear encoders	0.1 μ m	
Footprint	440x440 mm ²	
Workspace translational	40x40 mm ²	
Repeatability	4 μ m	1 μ m

Table 1. Characteristics of the robots micabo^e and micabo^{es} (Raatz, 2006)

2.3 Spatial parallel hybrid robot structures micabo^h and micabo^{hs} with 6 DOF

Six degree of freedom in x -, y -, z -direction and around the Euler angles ψ , θ , ϕ are realized with the spatial parallel hybrid robot structure micabo^h (Fig. 5). Six translational high resolution piezo drives are used. The movement of the end effector can partly be decoupled from the drives by integrating planar sub chains inside a spatial structure and, at the same time, by defining certain dependencies between the geometric parameters. Two drives slide on one axis and are coupled in one planar sub chain. The sub arm links have the same length. The three sliding axes are parallel to the vertical axis of the coordinate system. The position of the lower drives defines the inclination (θ , ϕ) of the working platform as a result of the structural configuration. The distance between upper and lower drives defines the position of the working platform in the horizontal plane (x , y , ψ), and an equal movement of all drives leads to a movement in z -direction only (Hesselbach et al., 1997). Therefore, the structure is ideal for assembly tasks with a main working plane parallel to the base frame (Raatz, 2006).

Experimental measurements with the robot structure micabo^h with conventional joints lead to a worse repeatability than expected, with insufficient reproducibility. Reasons for the worse results of the repeatability measurements were largely attributed to the ball joints which were poorly adjustable (Hesselbach et al., 2004b).

Fig. 5. Spatial parallel hybrid robot structure micabo^h

As further development the spatial parallel hybrid robot structure micabo^{hs} (Fig. 6) is designed with spatial flexure hinges. Therefore, joints with one, two and three DOF have to be replaced. One planar sub chain consists of two links which are connected with a rotational joint (1 DOF). Each sub chain is connected with the drive over a universal joint (2 DOF) and to the working platform over a ball joint (3 DOF). The universal joints and the ball joints are realized by a spatial combination of flexure hinges with 1 DOF. A workspace of 40x40x18 mm³ without additional inclination of the platform is realized with the micabo^{hs} (Raatz, 2006). In Table 2 the characteristics of the robot structures micabo^h and micabo^{hs} with conventional joints and flexure hinges are depicted.

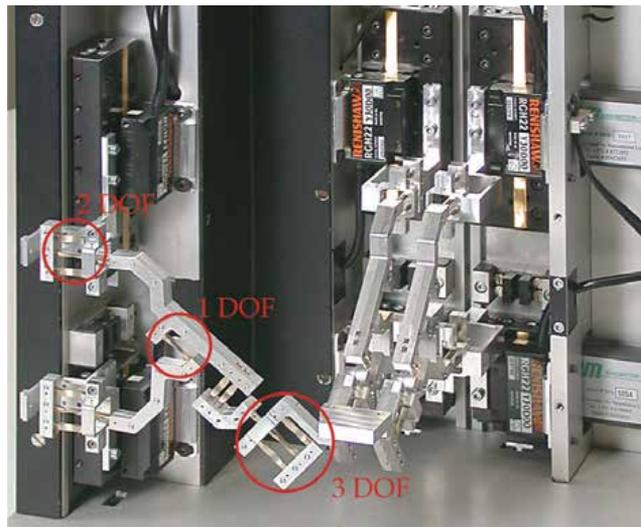


Fig. 6. Spatial parallel hybrid robot structure with flexure hinges micabo^{hs}

In accuracy measurements, a repeatability of 3 μm was measured for the micabo^{hs}. But these values have to be taken very carefully since the compliant robot is prone to vibrations (in z-direction with higher amplitudes than the repeatability values). This is due to the slip-stick effects of the piezo drives. Additionally, the structure has an inhomogeneous stiffness in the x-, y- and z- directions and the working platform is not symmetrically supported via the sub chains. As a consequence the damping mechanism of the spatial structure is not as good as the one of the planar parallel robot micabo^{es}.

Performance Data	micabo ^h	micabo ^{hs}
Max. velocity of the linear drives		0.23 m/s
Max. velocity of the end effector		0.1 m/s
Smallest step size		5 nm
Resolution of linear encoders		0.1 μm
Footprint		270x240 mm ²
Workspace translational	40x40x18 mm ³	40x40x18 mm ³
Repeatability	-	3 μm

Table 2. Characteristics of the robots micabo^h and micabo^{hs} (Raatz, 2006)

2.4 Spatial parallel robot structures Triglide and Triglide^s with 3 DOF and one serial rotational axis

The spatial parallel robot structure Triglide (Fig. 7) with 3 DOF (x-, y- and z-direction) was developed by the IWF and the Robert Bosch company as the main component of an assembly cell for micro assembly purposes. Three linear drives are arranged star-shaped in the base plane at intervals of 120°. This leads to a nearly triangle-shaped workspace. The working platform is connected to each drive by two links forming a parallelogram. This yields to translational movements of the platform and keeps the platform plane parallel to the base plane. An additional rotary axes is integrated serial into the working platform. The orientation of the working platform is therefore only limited by the gripper size and supply wires. This structure is very rigid and drive errors are reduced because the ratio of the “platform movement” to the “drive movement” is always <1 (Hesselbach et al., 2005). In this configuration, the resolution of the electrical linear motors with linear encoders is $0.125\ \mu\text{m}$. A repeatability of $0.9\ \mu\text{m}$ with 3σ is reached with conventional joints.

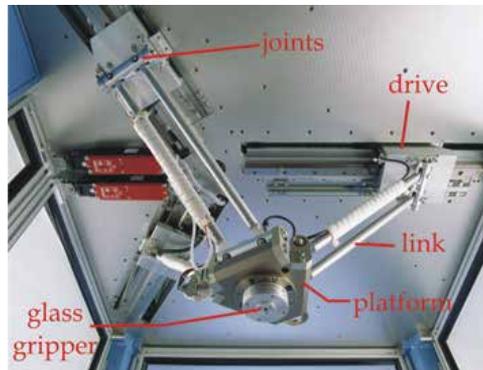


Fig. 7. Spatial parallel robot structure Triglide

Figure 8 shows the compliant spatial robot Triglide^s with 3 DOF and 6 integrated combined flexure hinges. The motors, footprint, workspace and the resolution of the linear encoders are equal to the Triglide.

For spatial mechanisms, flexure hinges with more than 1 DOF have to be designed. Those flexure hinges are realised by a spatial combination of flexure hinges with 1 DOF. A problem of compliant mechanisms, especially of spatial mechanisms, is their tendency to vibrate. Actually, the flexure hinges act as springs without any damping component, except for the inner damping of the deformed material. Figure 8 shows an example of increasing stiffness and optimising the distribution of occurring forces by a suitable design of a combined flexure hinge. Torsional moments can better be absorbed and transformed into tension and compression forces, since the hinges are arranged in a parallel and angular pattern (Hesselbach et al., 2004a).

Although the workspace is nearly triangular, a cube with a dimension of $112 \times 112 \times 112\ \text{mm}^3$ would fit into the workspace with the present configuration. If flexure hinges made of spring steel were used, the resulting workspace would be hundred times smaller than the workspace of the present design with pseudo-elastic flexure hinges. A repeatability of $0.3\ \mu\text{m}$ with 3σ is reached with the flexure hinges. In Table 3, the characteristics of Triglide and Triglide^s are listed.

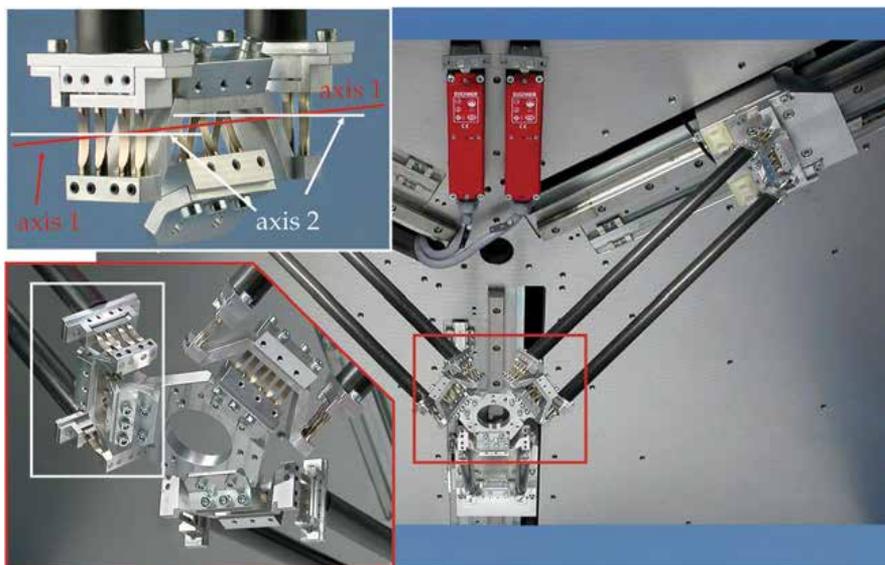


Fig. 8. Spatial parallel robot structure with flexure hinges Triglides

Performance Data	Triglide	Triglides
Max. velocity of the linear drives	0.2 m/s	
Max. velocity of the end effector	0.2 m/s	
Payload	1 kg	
Resolution of linear encoders	0.125 μm	
Footprint	1280x980 mm^2	
Workspace translational	112x112x112 mm^3	
Repeatability	0.9 μm	0.3 μm

Table 3. Characteristics of the robots Triglide and Triglides (Raatz, 2006)

2.5 Planar serial hybrid robot structures micabo^f and micabo^{f2} with 4 DOF

Another hybrid robot is the planar serial hybrid robot micabo^f with 4 DOF (Fig. 9 left). For movement in the xy -plane, two parallel linear axes with a resolution of 0.1 μm are used. Inside the robot head, two serial mounted drives for motion in z -direction and around the z -axis are located. Furthermore, the robot head is designed hollow for the integration of a camera. The micabo^f carries a 3D vision sensor (Tutsch & Berndt, 2003) which is integrated in the hollow robot head. This sensor is used for a sensor guided micro assembly with high accuracy. The workspace of the robot measures 120x200x15 mm^3 .

This hybrid robot structure combines the advantages of parallel and serial robotic structures. The parallel linear axes in the xy -plane offer high stiffness. A high accuracy is reached in this plane because of the combination of high resolution encoders and high precision motors. With the help of the serial drives in the robot head, a higher range of rotation than in a fully parallel structure is possible. In accuracy measurements, the micabo^f reached a repeatability of 2.6 μm (see the characteristics of the robot in Table 4).

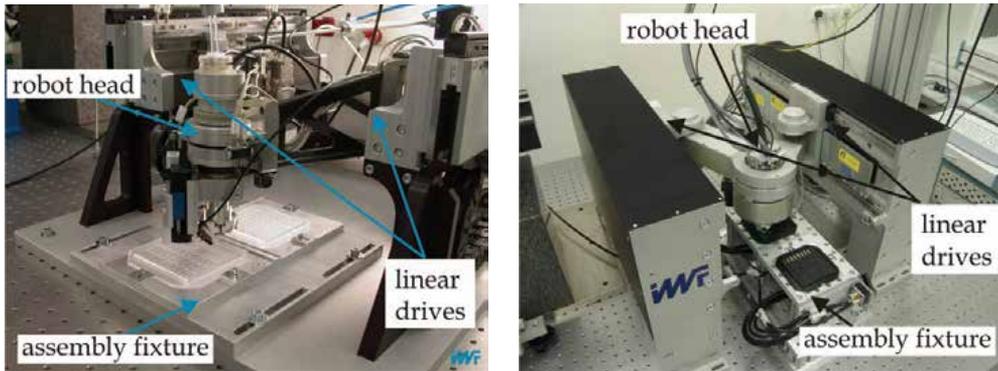


Fig. 9. Serial hybrid robot structures micabo^f (left) and micabo^{f2} (right)

Although a good repeatability is reached with the micabo^f, a self-induced vibration of the parallel linear drives occur, which is caused by the interaction between air bearings and linear drives. Neither the usage of additional dampers for the air bearings, nor optimisation of the control could eliminate this vibration. The parallel drives move around the desired position with a deviation of $\pm 1 \mu\text{m}$ because of this vibration. Furthermore, the workspace does not offer enough flexibility for part feeding, clamping different work pieces or extending the flexibility with two additional rotational drives in the workspace for 3D assembly operations. This leads to a demand for a redesign that reaches the required repeatability in the range of $1 \mu\text{m}$ and, at the same time, offers more flexibility by a larger workspace with better accessibility.

The robot micabo^{f2} (Fig. 9 right) has 4 DOF for part handling and one additional DOF for focus adjustment of the former mentioned 3D vision sensor. Two parallel linear drives impart motion in the xy-plane. Each of them is connected to a slide that is coupled to the arms of the structure through rotational bearings. A hollow axis between the arms takes up the robot head, which is designed like a cartridge and forms the tool center point (TCP). Inside the robot head, two drives are installed. One of them moves a platform with a gripper and the other one moves the 3D vision sensor (Simnofske, 2005). The workspace is enlarged to $160 \times 400 \times 15 \text{ mm}^3$ with better accessibility than before. In accuracy measurements, the micabo^{f2} reached a repeatability of $0.6 \mu\text{m}$ (Table 4).

Performance Data	micabo ^f	micabo ^{f2}
Max. velocity of the linear drives (x, y)	0.1 m/s	
Max. acceleration of the linear drives (x, y)	2 g	
Payload	0.2 kg	
Resolution of linear encoders	0.1 μm	
Footprint	480x600 mm ²	500x600 mm ²
Workspace translational / rotary angle	120x200x15 mm ³ / $\pm 45^\circ$	160x400x15 mm ³ / $\pm 180^\circ$
Repeatability	2.6 μm	0.6 μm

Table 4. Characteristics of the robots micabo^f and micabo^{f2}

2.6 Results of the development of size-adapted parallel and hybrid robot structures

In the previous sections, the development of four size-adapted robot structures, each with two different designs of the kinematic chain, was presented. Diverse requirements for the workspace, accuracies and flexibility can be fulfilled as a result of the different structures.

A small footprint is realized with the planar parallel robot structures micabo^e/micabo^{es} as well as with the spatial parallel hybrid structures micabo^h/micabo^{hs}. A larger workspace is offered by the spatial parallel robot structures Triglides/Triglides^s and by the planar serial hybrid robot structures micabo^f/micabo^{f2}. Therefore, the footprint is larger than that those of the micabo^e and micabo^h.

The integration of flexure hinges as ultra-precision machine elements into the size-adapted robot structures micabo^{es}, micabo^{hs} and Triglides^s leads to a better repeatability than with conventional joints. All robot structures presented here offer a sufficient repeatability for micro assembly. The most applicable robot structure can be chosen, depending on the assembly task.

The precision robot with its high accuracy is an important part of an assembly system for micro assembly tasks. Besides the precision robot, most assembly tasks require the use of additional sensors with high resolutions and measurement accuracies to reach a low assembly uncertainty. Furthermore, the technology of the gripper, the joining process and the adjustment of the assembly place influence the reachable assembly uncertainty. In section 3, an example of sensor guided micro assembly by use of the planar serial hybrid robot structure micabo^{f2} is described. The reachable assembly uncertainty is shown on the basis of an assembly process.

3. Sensor guided micro assembly

The assembly of hybrid micro systems typically demands assembly uncertainties in the range of a few micrometers. To achieve this high accuracy, the precision robot is supported by at least one sensor. Sensors for micro assembly can be optical sensors with resolutions in the range of a micrometer and below or force sensors with resolutions much below 1 N.

In the described example, the planar serial hybrid robot structure micabo^{f2} is supported by a 3D vision sensor and a 6D force sensor. The 3D vision sensor possesses a repeatability of 0.22 μm in x-, 0.29 μm in y- and 0.83 μm in z-direction. The field of view covers 11 mm in length and 5.5 mm in width. A beam splitter is arranged in front of a miniature camera, which directs the images of two perspectives to the CCD chip of the camera (Berndt, 2007).

With this vision sensor, 3D micro assembly tasks can be implemented. A positioning uncertainty lower than 0.5 μm can be reached with the combination of robot micabo^{f2} and 3D vision sensor. The reachable positioning uncertainty depends on the design of the assembly group and varies between 0.5 μm and 1 μm .

The 6D force sensor features a force measuring range (x-, y- and z-direction) of ± 12.5 N with a resolution of 0.0125 N. A measuring range of ± 125 Nmm with a resolution of 0.0625 Nmm for torques is given by the manufacturer. A defined joining force can be guaranteed by using the 6D force sensor for the implementation of a force controller inside the robot control. The robot micabo^{f2} is controlled by a real-time control system that is described in section 3.1. A description of the two methods for sensor guided micro assembly is given in section 3.2 and the chosen method for the presented assembly process is presented in section 3.3. The results of the sensor guided assembly process are shown in section 3.4.

3.1 Robot control

A real-time system is used to control the robot. The hardware of the control system features a PowerPC750 digital signal processor (DSP) running at 480 MHz, a digital I/O board, analog I/O boards, an encoder board and a serial I/O board. For programming, the control codes in “Matlab/ Simulink” and “C” are used. An open architecture control is realized that can deal with almost every robotic system with up to six axes. It consists of “structure specific” and “not structure specific” blocks (Fig. 10).

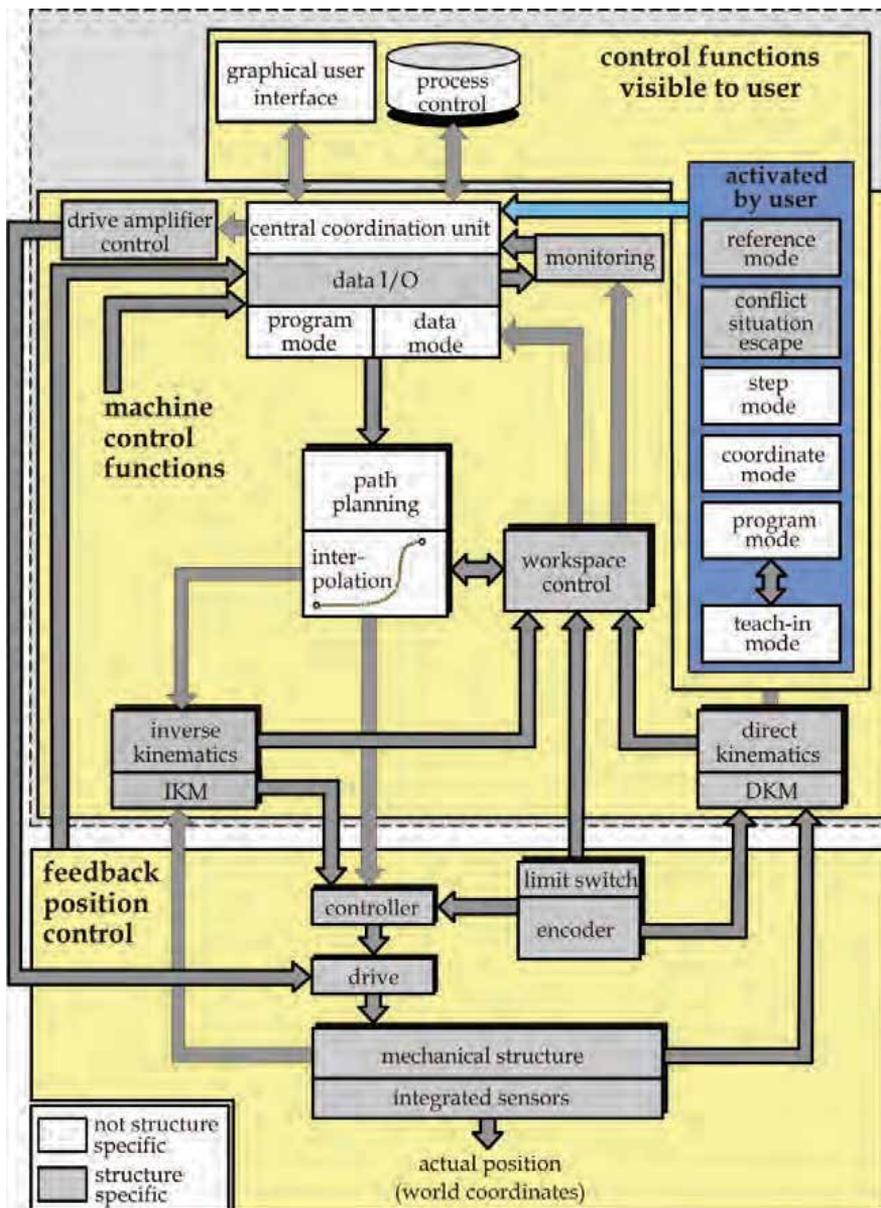


Fig. 10. Robot control concept

For use with a special robot, the “structure specific” blocks of the control have to be adapted to this robot structure. “Structure specific” blocks include inverse and direct kinematics, workspace control, monitoring, drive amplifier control, feedback position control as well as the allocation of data inputs and outputs. The “not structure specific” blocks, e.g. path planning and interpolation, do not have to be adapted.

3.2 Sensor guidance in micro assembly processes

Sensor guidance means that a feedback of position and/or force information is used to direct the positioning of the handling device during an assembly process. The information is given by optical or force sensors. Two different ways of data acquisition and data processing lead to the distinction of “absolute sensor guidance” and “relative sensor guidance”.

In a (micro) assembly process with absolute sensor guidance, the measurements of the handled part and the measurements of the assembly position on a substrate are carried out separately. The measurements are related by transformation of the sensor information into the world coordinate system. A position difference is calculated and carried out by the handling device. Only one position correction loop is possible with this method, which is used e.g. for pick-and-place assembly of SMD components.

With the method of relative sensor guidance, a simultaneous measurement of the handled part and the assembly position on a substrate is performed. The sensor information is transformed in the world coordinate system, too, and a position difference is calculated. The position correction can be performed in as many loops as desired. Naturally, as few correction loops as possible are carried out to ensure a low cycle time.

Relative sensor guidance is used for micro assembly tasks in this example. Sensor information from the vision sensor must be transmitted to the robot control. Therefore, two different control loops are used in the robot control (Fig. 11).

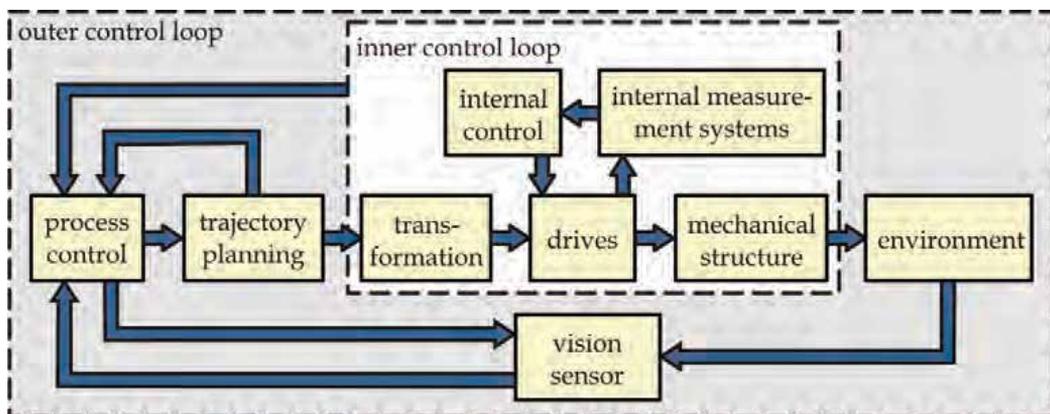


Fig. 11. Control loop with the use of sensor information

The process control gives commands to the robot control and demands information from the vision sensor system. The internal control loop works in a clock frequency of 5 kHz. The outer control loop contains the vision sensor, which gives relative position information to the process control. A resulting vector of the last desired position from the robot control and the relative position vector from the vision system is calculated inside the process control and transmitted to the robot control.

At present, the sensor guidance works in a so called “look-and-move” procedure. This means that the robot’s movement stops before a new measurement of the vision sensor is done and a new position correction is executed.

3.3 Example of assembly process

As an example, the assembly of a micro linear stepping motor, according to the reluctance principle, is described. The motor parts are mainly manufactured with micro technologies. One assembly task is the joining of guides on the surface of the motor’s stator element. In Figure 12 (left) the assembly group of two guides on a stator is shown. Figure 12 (right) shows the view of the 3D vision sensor of the assembly scene.

Circular positioning marks on the stator and guides are used by the 3D vision sensor for the relative positioning process. The reachable assembly uncertainty depends on the arrangement of the positioning marks and the length of the handled part. It is essential that the distances between the positioning marks are as large and the part length as small as possible.

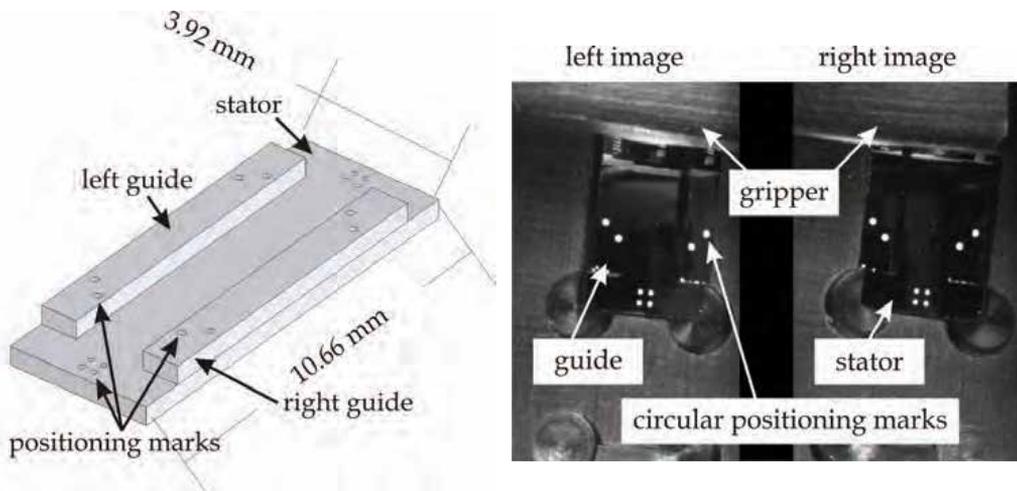


Fig. 12. Micro linear stepping motor – principle (left) and sensor view (right)

The sequence of the assembly process is shown in Figure 13. First, the robot moves over a stator element and checks the positioning marks. If the marks can be recognized, the robot moves over one left guide and checks the positioning marks, too. If the guide is recognized, it is picked up with a vacuum gripper by use of sensor information for a repeatable gripping process. Afterwards, the robot moves with the left guide over the stator and starts the relative positioning process. In this process, a measurement and calculation of a relative positioning vector is followed by the comparison with a limit value. If the relative position vector is larger than the limit value, a position correction is executed with the robot. Otherwise, the left guide is placed on the stator by use of the previously mentioned 6D force sensor to assure a defined contact force and reproducible process parameters. Cyanoacrylate is used for the bonding process.

Afterwards the relative positioning process is repeated for the right guide. During the assembly process, according to the method of relative sensor guidance, a limit value of $0.8 \mu\text{m}$ can be reached with the combination of the 3D vision sensor and the robot micabo^{l2}.

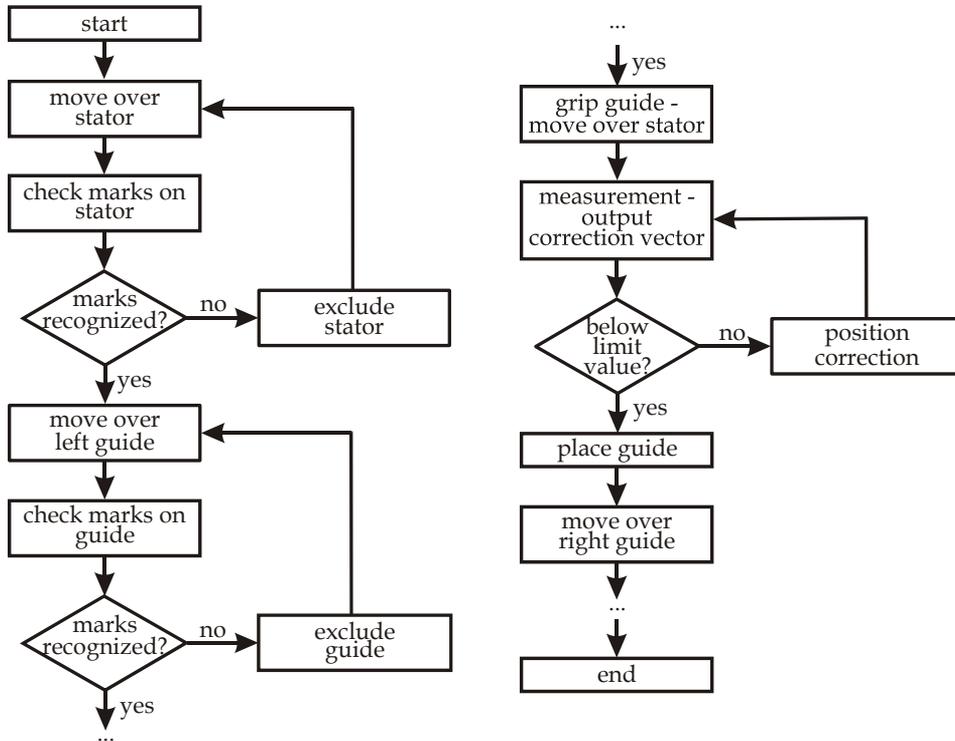


Fig. 13. Sequence of the assembly process

3.4 Results of the sensor guided assembly process

To quantify the precision assembly process, two terms were defined - positioning uncertainty and assembly uncertainty. According to (DIN ISO 230-2, 2000) the positioning uncertainty is the combination of the mean positioning deviation and the double standard deviation.

For precision assembly processes the term positioning uncertainty refers to the reached relative position between the two parts of the assembly group before the bonding process is carried out (in this case the guide is above the stator and is not in contact with it). The term assembly uncertainty describes the relative position between the assembled parts, measured after the assembly process has been completed. This is the combination of the mean assembly deviation and the double standard deviation, too.

Positioning marks are used as an inspection criterion. They are used for quality control of the parts before the process and during the process for the relative sensor guidance and evaluating the positioning uncertainty. After the process the positioning marks are used for evaluating the assembly uncertainty. During the process only one end of the assembled parts can be measured because the gripper covers half of the guide and the stator (see Fig. 12

right). Therefore, the 3D vision sensor observes only the visible sides of the assembled parts. This means that the measured positioning error and the resulting positioning uncertainty are only determined by the visible part side. After the process, both ends of the assembly group can be inspected and the overall assembly deviation can be measured. The assembly uncertainty is calculated from the deviations. This value is comprised of the overall errors during the assembly of the micro system.

An assembly uncertainty of $38\ \mu\text{m}$ and a positioning uncertainty of $0.82\ \mu\text{m}$ are reached for the assembly process. The difference between assembly uncertainty and positioning uncertainty is a result of the relatively long part length of $10.66\ \text{mm}$. A small angular deviation causes a positioning error (in xy-direction). This error is larger at the side of the part which is invisible during the positioning process than the error on the visible side. With a greater part length, this positioning error will be higher than with smaller parts. Furthermore, deviations occurring during the bonding process cause an increased assembly error. Figure 14 shows the positioning uncertainty and figure 15 the assembly uncertainty of the assembled groups. The circles in the diagrams show the radius of the uncertainties.

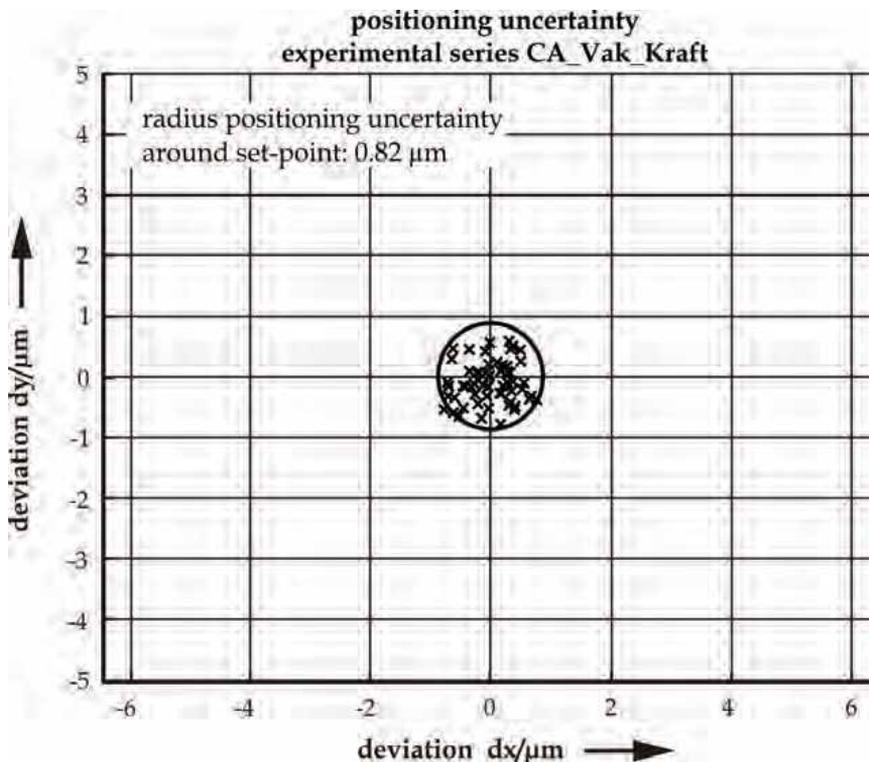


Fig. 14. Reached positioning uncertainty

In another assembly task, assembly uncertainties of $25\ \mu\text{m}$ were reached with another design of the assembly group. Therefore, the distance between the positioning marks has been enlarged. A positioning uncertainty and a limit value of $0.5\ \mu\text{m}$ was reached with this arrangement of the positioning marks. This demonstrates the potential for further improvement of the assembly uncertainty.

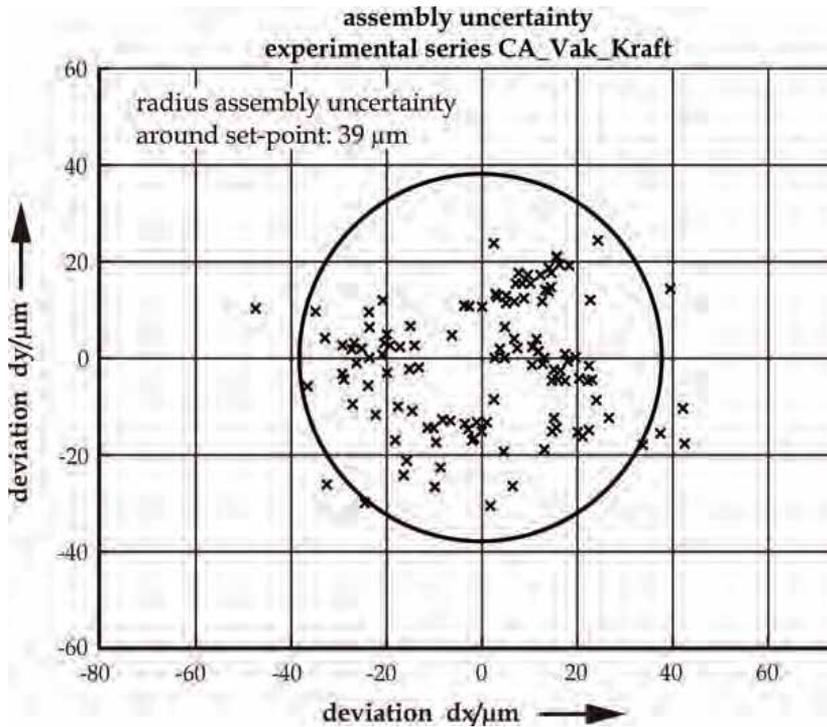


Fig. 15. Reached assembly uncertainty

4. Conclusion

Micro assembly tasks demand low assembly uncertainties in the range of a few micrometers. This request results from the small part sizes in the production of MST components and the resulting small valid tolerances. Since precision robots represent the central component of an assembly system, an appropriate kinematic structure is crucial. These kinematic structures can be serial, parallel or hybrid (serial/parallel). Although serial structures can be used for micro assembly, they have large moved masses and need a massive construction of the frame and robot links to obtain an appropriate repeatability.

Therefore, some size-adapted parallel and hybrid parallel robot structures were presented in the previous sections. Very good repeatabilities were reached with the presented robots due to the chosen structures, the miniaturized design and the use of flexure hinges as ultra-precision machine components.

Besides the precision robot, most assembly tasks require the use of additional sensors with high resolutions and measurement accuracies to reach a low assembly uncertainty. Therefore, optical and/or force sensors are used for sensor guided micro assembly processes.

The terms "absolute sensor guidance" and "relative sensor guidance" were introduced. Both methods offer an enhancement of the accuracy within micro assembly processes. The "relative sensor guidance" promises a lower positioning and assembly uncertainty because of the user defined number of position correction loops. Therefore, relative sensor guidance was used in the presented example for micro assembly.

With the use of relative sensor guidance, positioning uncertainties below $0.5\ \mu\text{m}$ can be reached. The assembly uncertainty has to be further improved to fulfil the demand for assembly uncertainties in the range of a few micrometers. Therefore, the design of the product and positioning marks as well as the gripping and joining technology has to be examined in future developments.

5. References

- Berndt, M. (2007). *Photogrammetrischer 3D-Bildsensor für die automatisierte Mikromontage*, Schriftenreihe des Institutes für Produktionsmesstechnik, No. 3, Shaker Verlag, ISBN 978-3-8322-6768-1, Aachen
- van Brussel, H. ; Peirs, J. ; Delchambre, A. ; Reinhart, G. ; Roth, N. ; Weck, M. & Zussman, E. (2000). Assembly of Microsystems, *Annals of CIRP*, Vol. 49, No. 2, pp. 451-472
- Clavel, R.; Helmer, P.; Niaritsiry, T.; Rossopoulos, S.; Verettas, I. (2005). High Precision Parallel Robots for Micro-Factory Applications, *Robotic Systems for Handling and Assembly - Proc. of 2nd International Colloquium of the Collaborative Research Center 562*, Fortschritte in der Robotik Band 9, Shaker Verlag, ISBN 3-832-3866-2, Aachen, pp. 285-296
- Coudourey, A.; Perroud, S.; Mussard, Y. (2006). Miniature Reconfigurable Assembly Line for Small Products, *Proc. Third International Precision Assembly Seminar (IPAS'2006)*, Springer Verlag, ISBN 0-387-31276-5, Berlin, pp. 193-200
- DIN ISO 230-2 (2000). Prüfregeln für Werkzeugmaschinen, Teil 2: Bestimmung der Positionierunsicherheit und der Wiederholpräzision der Positionierung von numerisch gesteuerten Achsen, *Beuth Verlag*, Berlin
- EN ISO 9283 (1999). Industrieroboter: Leistungskenngrößen und zugehörige Prüfmethode. *Beuth Verlag*, Berlin
- Fatikow, S. (2000). Miniman. In: *Mikroroboter und Mikromontage*, p. 277, Teubner Verlag, ISBN 3-519-06264-X, Stuttgart – Leipzig
- Hesselbach, J.; Plitea, N. ; Thoben, R. (1997). Advanced technologies for micro assembly, *Proc. of SPIE*, Vol. 3202, pp. 178-190
- Hesselbach, J.; Raatz, A. (2000). Pseudo-Elastic Flexure-Hinges in Robots for Micro Assembly, *Proc. of SPIE*, Vol. 4194, pp. 157-167
- Hesselbach, J. ; Raatz, A. & Kunzmann, H. (2004a). Performance of Pseudo-Elastic Flexure Hinges in Parallel Robots for Micro-Assembly Tasks, *Annals of CIRP*, Vol. 53, No. 1, pp. 329-332
- Hesselbach, J.; Wrege, J.; Raatz, A.; Becker, O. (2004b) Aspects on Design of High Precision Parallel Robots, *Journal of Assembly Automation*, Vol. 24, No. 1, pp. 49-57
- Hesselbach, J. ; Wrege, J. ; Raatz, A. ; Heuer, K. & Soetebier, S. (2005). Microassembly - Approaches to Meet the Requirements of Accuracy, In: *Advanced Micro & Nanosystems Volume 4 - Micro-Engineering in Metals and Ceramics Part II*, Löhe, D. (Ed.) & Haußelt, J. (Ed.), pp. 475-498, Wiley-VCH Verlag, ISBN 3-527-31493-8, Weinheim
- Höhn, M. (2001). *Sensorgeführte Montage hybrider Mikrosysteme*, Forschungsberichte iwB, Herbert Utz Verlag, ISBN 3-8316-0012-0, München
- Howell, L.L. ; Midha, A. (1995). Parametric Deflection Approximations for End-Loaded, Large-Deflection Beams in Compliant Mechanisms, *Journal of Mechanical Design*, Vol. 117, No. 3, pp. 156-165

- Paros, J.M. ; Weisbord, L. (1965). How to Design Flexure Hinges, *Machine Design*, Vol. 25, pp. 151-156
- Raatz, A. (2006). *Stoffschlüssige Gelenke aus pseudo-elastischen Formgedächtnislegierungen in Parallelrobotern*, Vulkan Verlag, ISBN 3-8027-8691-2, Essen
- Raatz, A. & Hesselbach, J. (2007). High-Precision Robots and Micro Assembly, *Proceedings of COMA '07 International Conference on Competitive Manufacturing*, pp. 321-326, Stellenbosch, South Africa, 2007
- Simnofske, M. ; Schöttler, K. ; Hesselbach, J. (2005). Micabo² – robot for micro assembly, *Production Engineering*, Vol. 12, No. 2, pp. 215-218
- Smith, S.T. (2000). *Flexures - Elements of Elastic Mechanisms*. Gordon & Breach Science Publishers, ISBN 90-5699-261-9, Amsterdam
- Tutsch, R.; Berndt, M. (2003). Optischer 3D-Sensor zur räumlichen Positionsbestimmung bei der Mikromontage, *Applied Machine Vision*, VDI-Report No. 1800, Stuttgart, pp. 111-118
- Wicht, H. & Bouchaud, J. (2005). NEXUS Market Analysis for MEMS and Microsystems III 2005-2009, *mst news*, Vol. 5, 2005, pp. 33-34

Dynamics of Hexapods with Fixed-Length Legs

Rosario Sinatra^a and Fengfeng Xi^b

^a*Università di Catania, 95125, Catania,*

^b*Ryerson University Toronto, Ontario,*

^a*Italy*

^b*Canada*

1. Introduction

Hexapod is a new type of machine tool based on the parallel closed-chain kinematic structure. Compared to the conventional machine tool, parallel mechanism structure offers superior stiffness, lower mass and higher acceleration, resulting from the parallel structural arrangement of the motion systems. Moreover, hexapod has the potential to be highly modular and re-configurable, with other advantages including higher dexterity, simpler and fewer fixtures, and multi-mode manufacturing capabilities.

Initially, hexapod was developed based on the Stewart platform, i.e. the prismatic type of parallel mechanism with the variable leg length. Commercial hexapods, such as VARIAX from Giddings & Lewis, Tornado from Hexel Corp., and Geodetic from Geodetic Technology Ltd., are all based on this structure. One of the disadvantages for the variable leg length structure is that the leg stiffness varies as the leg moves in and out. To overcome this problem, recently the constant leg length hexapod has been envisioned, for instance, HexaM from Toyada (Susuki et al., 1997). Hexaglibe from the Swiss Federal Institute of Technology (Honegger et al., 1997), and Linapod from University of Stuttgart (Pritschow & Wurst, 1997). Between these two types, the fixed-length leg is stiffer (Tlustý et al., 1999) and, here, becoming popular.

Dynamic modeling and analysis of the parallel mechanisms is an important part of hexapod design and control. Much work has been done in this area, resulting in a very rich literature (Fichter, 1986; Sugimoto, 1987; Do & Yang, 1988; Geng et al., 1992; Tsai, 2000; Hashimoto & Kimura, 1989; Fijany & Bejezy, 1991). However, the research work conducted so far on the inverse dynamics has been focused on the parallel mechanisms with extensible legs.

In this chapter, first, in the inverse dynamics of the new type six d.o.f. hexapods with fixed-length legs, shown in Fig. 1, is developed with consideration of the masses of the moving platform and the legs. (Xi & Sinatra, 2002) This system consists of a moving platform *MP* and six legs sliding along the guideways that are mounted on the support structure. Each leg is connected at one end to the guideway by a universal joint and at another end to the moving platform by a spherical joint. The natural orthogonal complement method (Angeles & Lee, 1988; Angeles & Lee, 1989) is applied, which provides an effective way of solving multi-body dynamics systems. This method has been applied to studying serial and parallel manipulators (Angeles & Ma, 1988; Zanganesh et al., 1997) automated vehicles (Saha & Angeles, 1991) and flexible mechanisms (Xi & Sinatra, 1997). In this development, the

Newton-Euler formulation is used to model the dynamics of each individual body, including the moving platform and the legs. All individual dynamics equations are then assembled to form the global dynamics equations. Based on the complete kinematics model developed, an explicit expression is derived for the natural orthogonal complement which effectively eliminates the constraint forces in the global dynamics equations. This leads to the inverse dynamics equations of hexapods that can be used to compute required actuator forces for given motions.



Fig. 1. New hexapod design

Finally, for completeness of the dynamic study of the parallel manipulator with the fixed-length legs, the static balancing is studied (Xi et al., 2005).

A great deal of work has been carried out and reported in the literature for the static balancing problem. For example, in the case of serial manipulator, Nathan (Nathan, 1985) and Hervé (Hervé, 1986) applied the counterweight for gravity compensations. Streit et al. (Streit & Gilmore, 1991), (Walsh et al., 19) proposed an approach to static balanced rotary bodies and two degrees of freedom of the revolute links using springs. Streit and Shin presented a general approach for the static balancing of planar linkages using springs (Streit & Shin, 1980). Ulrich and Kumar presented a method of passive mechanical gravity compensation using appropriate pulley profiles (Ulrich & Kumar, 1991). Kazerooni and Kim presented a method for statically-balanced direct drive arm (Kazerooni & Kim, 1990).

For the parallel manipulator much work was done by Gosselin et al. Research reported in (Gosselin & Wang, 1998) was focused on the design of gravity-compensated of a six-degree-of-freedom parallel manipulator with revolute joints. Each leg with two links is connected by an actuated revolute joint to the base platform and by a spherical joints the moving platform. Two methods are used, one approach using the counterweight and the other using springs. In the former method, if the centre of mass of a mechanism can be made stationary, the static balancing is obtained in any direction of the Cartesian space. In the second approach, if the total energy is kept constant, the mechanism is statically balanced only in the direction of gravity vector. The static balancing conditions are derived for the three-degree-of-freedom spatial parallel manipulator (Wang & Gosselin, 1998) and in similar

universal joint, and S_i indicates the position of the i th spherical joint. Six legs are numbered from 1 to 6.

Furthermore, a local coordinate frame $O_i-x_iy_iz_i$ is defined for each leg, with its origin located at the center of the i th universal joint. Two unit vectors are used. Unit vector \mathbf{u}_i^l is along the leg length representing the direction of the i th leg, and unit vector \mathbf{u}_i^s is along the guideway representing the direction of the i th guideway. The orientation of the i th coordinate frame with respect to the base can therefore be defined by a 3×3 rotation matrix, for $i = 1, \dots, 6$, as

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{u}_i^a & \mathbf{u}_i^l \times \mathbf{u}_i^a & \mathbf{u}_i^l \end{bmatrix} \quad (1)$$

where \mathbf{u}_i^a is expressed as

$$\mathbf{u}_i^a = \frac{\mathbf{u}_i^s \times \mathbf{u}_i^l}{\|\mathbf{u}_i^s \times \mathbf{u}_i^l\|} \quad (2)$$

Note that vector \mathbf{u}_i^l is configuration-dependent and determined for the given location of the moving platform; vector \mathbf{u}_i^s is constant and defined by the geometry of the hexapod.

For the purpose of carrying out the inverse dynamics analysis of the hexapod, the following symbols are defined. As shown in Figure 2, C_i is the center of mass of the i th leg, C_p is the center of mass of the moving platform, \mathbf{c} , $\dot{\mathbf{c}}$ and $\ddot{\mathbf{c}}$ are the position, velocity and acceleration vectors, respectively, of C_i with respect to the fixed coordinate frame, $\bar{\mathbf{p}}$ is the vector pointing from O_i to C_p with respect to the local coordinate frame $O_i-x_iy_iz_i$.

2.2 Kinematics

Consider one branch of the leg-guideway system, as shown in Figure 2, the following loop equation for $i = 1, \dots, 6$, holds,

$$\mathbf{h} + \mathbf{R}\bar{\mathbf{p}}_i - \mathbf{b}_i - \mathbf{s}_i - \mathbf{l}_i = \mathbf{0} \quad (3)$$

where \mathbf{h} and \mathbf{R} are the vector and rotation matrix that define the position and orientation of the moving platform relative to the base, respectively, $\bar{\mathbf{p}}_i$ is the vector representing the position of the i th spherical joint on the moving platform in the local coordinates.

Since the leg always moves along the guideway, \mathbf{s}_i can be expressed as

$$\mathbf{s}_i = s_i \mathbf{u}_i^s \quad (4)$$

where s_i is a scalar representing the displacement of the i th actuator along the guideway. Likewise, leg vector \mathbf{l}_i can be expressed as

$$\mathbf{l}_i = l_i \mathbf{u}_i^l \quad (5)$$

where l_i is a scalar representing the fixed length of the i th leg. As mentioned in Section 2.1, the leg axis is parallel to the z_i axis of the local coordinate frame $O_i-x_iy_iz_i$. In the light of eq.(1), \mathbf{u}_i^l can be expressed as

$$\mathbf{u}_i^l = \mathbf{Q}_i \mathbf{z}_i \quad (6)$$

Substituting eqs.(4 & 5) into eq.(3) and rearranging it yields the following kinematics equations for the fixed-length leg hexapod, for $i = 1, \dots, 6$,

$$s_i \mathbf{u}_i^s = \mathbf{h} + \mathbf{R} \bar{\mathbf{p}}_i - \mathbf{b}_i - l_i \mathbf{u}_i^l \quad (7)$$

To obtain the velocity of the moving platform, taking the time derivative of eq. (7) yields

$$\dot{s}_i \mathbf{u}_i^s = \mathbf{v} + (\boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i) - (\boldsymbol{\omega}_i \times \mathbf{l}_i) \quad (8)$$

where \mathbf{v} and $\boldsymbol{\omega}$ are the vectors representing the velocity and angular velocity of the moving platform, respectively, and $\boldsymbol{\omega}_i$ is the vector representing the angular velocity of the i th leg.

Furthermore, by taking dot product on both sides of eq.(8) by \mathbf{l}_i , it leads to

$$\dot{s}_i \mathbf{u}_i^s \cdot \mathbf{l}_i = [\mathbf{v} + (\boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i)] \cdot \mathbf{l}_i \quad (9)$$

It is well known that the kinematic analysis of parallel manipulator leads to two Jacobian matrices, namely, the forward and the inverse Jacobian (Gosselin & Angeles, 1990). To find the Jacobians for the hexapod under study, rearranging eq.(9) yields the following form

$$\dot{s}_i (\mathbf{u}_i^s \cdot \mathbf{l}_i) = [\mathbf{l}_i^T, (\mathbf{R} \bar{\mathbf{p}}_i \times \mathbf{l}_i)^T] \mathbf{t}_p \quad (10)$$

where $\mathbf{t} = [\mathbf{v}, \boldsymbol{\omega}]$ is the 6×1 twist vector of the moving platform. Consider all six legs it leads to the following expression

$$\mathbf{B} \dot{\mathbf{s}} = \mathbf{A} \mathbf{t}_p \quad (11)$$

where $\dot{\mathbf{s}} = [\dot{s}_1, \dots, \dot{s}_6]^T$ is the 6×1 vector of the actuator speeds, and \mathbf{A} and \mathbf{B} are the 6×6 matrices representing the inverse and forward *Jacobian* of the hexapod and they are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{l}_1^T & (\mathbf{R} \bar{\mathbf{p}}_1 \times \mathbf{l}_1)^T \\ \vdots & \vdots \\ \mathbf{l}_6^T & (\mathbf{R} \bar{\mathbf{p}}_6 \times \mathbf{l}_6)^T \end{bmatrix} \quad (12)$$

$$\mathbf{B} = \text{diag}(\mathbf{u}_1^s \cdot \mathbf{l}_1, \dots, \mathbf{u}_6^s \cdot \mathbf{l}_6) \quad (13)$$

Eq.(11) defines the differential relationship between the actuator speeds $\dot{\mathbf{s}}$ and the twist of the moving platform \mathbf{t}_p . Rewriting eq.(11) gives

$$\dot{\mathbf{s}} = \mathbf{J}_p \mathbf{t}_p \quad (14)$$

Provided that \mathbf{B} is invertible, the Jacobian matrix of the moving platform \mathbf{J}_p can be given as

$$\mathbf{J}_p = \mathbf{B}^{-1} \mathbf{A} = [\mathbf{J}_{p1}^T, \dots, \mathbf{J}_{p6}^T]^T \quad (15)$$

where

$$\mathbf{J}_{pi} = \begin{bmatrix} \mathbf{1}_i^T \\ \mathbf{1}_i^T \mathbf{u}_i^s \end{bmatrix}, \quad \frac{(\mathbf{R}\bar{\mathbf{p}}_i \times \mathbf{1}_i)^T}{\mathbf{1}_i^T \mathbf{u}_i^s} \quad (16)$$

for $i = 1, \dots, 6$. From eq.(14), \mathbf{t}_p can be expressed in terms of $\dot{\mathbf{s}}$ as,

$$\mathbf{t}_p = \mathbf{T}_p \dot{\mathbf{s}} \quad (17)$$

where $\mathbf{T}_p = \mathbf{J}_p^{-1}$.

To obtain the acceleration of the moving platform, taking the time derivative of eq. (14) yields

$$\ddot{\mathbf{s}} = \dot{\mathbf{J}}_p \mathbf{t}_p + \mathbf{J}_p \dot{\mathbf{t}}_p \quad (18)$$

where $\ddot{\mathbf{s}} = [\ddot{s}_1, \dots, \ddot{s}_6]^T$ is the 6×1 vector of the actuator accelerations, $\dot{\mathbf{t}}_p = [\mathbf{a}^T, \dot{\boldsymbol{\omega}}^T]^T$ is the time derivative of the twist of the moving platform, $\dot{\mathbf{J}}_p$ is the time derivative of the Jacobian matrix of the moving platform obtained by differentiating \mathbf{J}_p with respect to time, that is

$$\dot{\mathbf{J}}_p = \mathbf{B}^{-1}(\dot{\mathbf{A}} - \dot{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}) \quad (19)$$

where $\dot{\mathbf{A}}$ and $\dot{\mathbf{B}}$ given as

$$\dot{\mathbf{A}} = \begin{bmatrix} (\boldsymbol{\omega}_1 \times \mathbf{1}_1)^T & ((\boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_1) \times \mathbf{1}_1 + \mathbf{R}\bar{\mathbf{p}}_1 \times (\boldsymbol{\omega}_1 \times \mathbf{1}_1))^T \\ \vdots & \vdots \\ (\boldsymbol{\omega}_6 \times \mathbf{1}_6)^T & ((\boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_6) \times \mathbf{1}_6 + \mathbf{R}\bar{\mathbf{p}}_6 \times (\boldsymbol{\omega}_6 \times \mathbf{1}_6))^T \end{bmatrix} \quad (20)$$

$$\dot{\mathbf{B}} = \text{diag}(\mathbf{u}_1^s \cdot (\boldsymbol{\omega}_1 \times \mathbf{1}_1), \dots, \mathbf{u}_6^s \cdot (\boldsymbol{\omega}_6 \times \mathbf{1}_6)) \quad (21)$$

If the mass of the leg is uniformly distributed, then the center of mass is in its middle. The velocity of the center of mass can be given as

$$\dot{\mathbf{c}}_i = \dot{\mathbf{s}}_i + \boldsymbol{\omega}_i \times \frac{\mathbf{1}_i}{2} \quad (22)$$

Upon differentiating eq.(22), the acceleration of the center of mass can be given as

$$\ddot{\mathbf{c}}_i = \ddot{\mathbf{s}}_i + \dot{\boldsymbol{\omega}}_i \times \frac{\mathbf{1}_i}{2} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \frac{\mathbf{1}_i}{2}) \quad (23)$$

To obtain the leg angular velocity and acceleration, denote by \mathbf{E}_i the 3×3 cross-product matrix associated with vector \mathbf{u}_i^l , then eq.(9) may be re-written as

$$\mathbf{E}_i \boldsymbol{\omega}_i = \frac{1}{l_i} [\mathbf{v} + \boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_i - \dot{\mathbf{s}}_i \mathbf{u}_i^s] \quad (24)$$

Consider all six legs, it forms a set of linear equations containing the unknowns of the leg angular velocity. There are three components of $\boldsymbol{\omega}_i$ for each leg. Because matrix \mathbf{E}_i is a skew symmetric and singular, it is impossible to directly solve eq.(24). However, since the leg does not spin about its longitudinal axis, this indicates (Tsai, 2000)

$$\boldsymbol{\omega}_i \cdot \mathbf{l}_i = 0 \quad (25)$$

In the light of eq.(25), eq. (24) may be rewritten as

$$\mathbf{A}_i \boldsymbol{\omega}_i = \mathbf{e}_i \quad (26)$$

where \mathbf{A}_i is a 4×3 matrix and \mathbf{e}_i is a 4-dimensional vector, and they are defined as

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{E}_i \\ \mathbf{l}_i^T \end{bmatrix} \quad (27)$$

$$\mathbf{e}_i = \frac{1}{l_i} \begin{bmatrix} \mathbf{v} + \boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i - \dot{s}_i \mathbf{u}_i^s \\ 0 \end{bmatrix} \quad (28)$$

Solving eq. (26) leads to the expression for the leg angular velocity

$$\boldsymbol{\omega}_i = \frac{\mathbf{l}_i}{l_i^2} \times \left[\mathbf{v} + \boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i - \dot{s}_i \mathbf{u}_i^s \right] \quad (29)$$

Now eq.(29) is substituted back into eq.(22), and the velocity becomes

$$\dot{\mathbf{c}}_i = \frac{1}{2} \left[\mathbf{v} + \boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i - \dot{s}_i \mathbf{u}_i^s \right] \quad (30)$$

By examining eqs.(29 & 30), it may be noted that the two terms in the brackets are identical. The first term may be expressed as

$$\mathbf{v} + \boldsymbol{\omega} \times \mathbf{R} \bar{\mathbf{p}}_i = [\mathbf{1}, \mathbf{E}_{pi}] \mathbf{t}_p \quad (31)$$

where \mathbf{E}_{pi} is the cross-product matrix of $\mathbf{R} \bar{\mathbf{p}}_i$. In the light of eq.(17), eq.(31) may be related to \dot{s} as

$$[\mathbf{1}, \mathbf{E}_{pi}] \mathbf{t}_p = \mathbf{T}_{1i} \dot{s} \quad (32)$$

where $\mathbf{1}$ is the 3×3 identity matrix and \mathbf{T}_{1i} is the 3×6 matrix pertaining to the first term defined as

$$\mathbf{T}_{1i} = [\mathbf{1}, \mathbf{E}_{pi}] \mathbf{T}_p \quad (33)$$

The second term in eqs.(29) and (30) can also be expressed in terms of \dot{s}

$$\dot{s}_i \mathbf{u}_i^s = \mathbf{T}_{2i} \dot{s} \quad (34)$$

where \mathbf{T}_{2i} is the 3×6 matrix pertaining to the second term defined as

$$\mathbf{T}_{2i} = [\mathbf{0}_3, \dots, \mathbf{u}_i^s, \dots, \mathbf{0}_3] \quad (35)$$

In eq.(35), $\mathbf{0}_3$ is the 3-dimensional null vector. The twist of the i th leg can be expressed in terms of $\dot{\mathbf{s}}$ as

$$\mathbf{t}_i = \mathbf{T}_i \dot{\mathbf{s}} \quad (36)$$

where \mathbf{t}_i is the twist of the i th leg, i.e. $\mathbf{t}_i = [\dot{\mathbf{c}}_i^T, \boldsymbol{\omega}_i^T]^T$, and the 6×6 matrix \mathbf{T}_i is given as

$$\mathbf{T}_i = [\mathbf{T}_{1i}^T, \mathbf{T}_{2i}^T]^T \quad (37)$$

Furthermore, the leg angular acceleration can be obtained by differentiating eq.(26) with respect to time, that is

$$\mathbf{A}_i \dot{\boldsymbol{\omega}}_i = \dot{\mathbf{e}}_i - \dot{\mathbf{A}}_i \boldsymbol{\omega}_i \quad (38)$$

where

$$\dot{\mathbf{A}}_i \boldsymbol{\omega}_i = \begin{bmatrix} (\boldsymbol{\omega}_i \times \mathbf{u}_i) \times \boldsymbol{\omega}_i \\ 0 \end{bmatrix} \quad (39)$$

$$\dot{\mathbf{e}}_i = \frac{1}{l_i} \begin{bmatrix} \mathbf{a} + \dot{\boldsymbol{\omega}} \times \mathbf{R}\bar{\mathbf{p}}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_i) - \dot{s}_i \mathbf{u}_i^s \\ 0 \end{bmatrix} \quad (40)$$

From eq.(38), vector $\dot{\boldsymbol{\omega}}_i$ representing the angular acceleration of the i th leg is given as

$$\dot{\boldsymbol{\omega}}_i = \frac{1}{l_i^2} \left[(\boldsymbol{\omega}_i \times \mathbf{l}_i) \times (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_i - \dot{s}_i \mathbf{u}_i^s) + \mathbf{l}_i \times (\mathbf{a} + \dot{\boldsymbol{\omega}} \times \mathbf{R}\bar{\mathbf{p}}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{R}\bar{\mathbf{p}}_i) - \dot{s}_i \mathbf{u}_i^s) \right] \quad (41)$$

3. Dynamic modeling

3.1 The natural orthogonal complement method

Prior to performing dynamic modeling of the hexapod, a brief review of the natural orthogonal complement method (Angeles & Lee, 1988) is provided. Consider a system composed of p rigid bodies under holonomic constraints, the Newton-Euler equations for each individual body can be written, for $i = 1, \dots, p$, as

$$\mathbf{M}_i \dot{\mathbf{t}}_i = -\mathbf{W}_i \mathbf{M}_i \mathbf{t}_i + \mathbf{w}_i \quad (42)$$

where \mathbf{t}_i is the twist of the i th body, $\mathbf{w}_i = [\mathbf{n}_i^T, \mathbf{f}_i^T]^T$ represent the wrench acting on the i th body, \mathbf{n}_i and \mathbf{f}_i are the resultant moment and the resultant force acting at the center of mass. In general \mathbf{w}_i can be decomposed into *working wrench* \mathbf{w}_i^w and *non-working wrench* \mathbf{w}_i^N . The former can further be decomposed as

$$\mathbf{w}_i^w = \mathbf{w}_i^a + \mathbf{w}_i^g + \mathbf{w}_i^d \quad (43)$$

where \mathbf{w}_i^a , \mathbf{w}_i^g and \mathbf{w}_i^d are the actuator, gravity and dissipate wrenches, respectively.

In eq. (42), the 6×6 angular velocity matrix \mathbf{W}_i and the 6×6 inertia matrix \mathbf{M}_i are defined as

$$\mathbf{W}_i = \begin{bmatrix} \boldsymbol{\Omega}_i & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \quad \mathbf{M}_i = \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i \mathbf{1} \end{bmatrix} \quad (44)$$

with

$$\boldsymbol{\Omega}_i = \frac{\partial(\boldsymbol{\omega}_i \times \mathbf{e})}{\partial \mathbf{e}} \quad (45)$$

where \mathbf{I}_i is the 3×3 matrix of the moment of inertia of the i th body, m_i is the body mass, \mathbf{O} denotes the 3×3 null matrix, and \mathbf{e} is an arbitrary vector.

If consider all p bodies, the assembled system dynamics equations are given as

$$\mathbf{M}\dot{\mathbf{t}} = -\mathbf{W}\mathbf{M}\mathbf{t} + \mathbf{w}^W + \mathbf{w}^N \quad (46)$$

where the $6p \times 6p$ generalized mass matrix \mathbf{M} and generalized angular velocity matrix \mathbf{W} are defined as

$$\mathbf{M} \equiv \text{diag}(\mathbf{M}_1, \dots, \mathbf{M}_p), \quad (47)$$

$$\mathbf{W} \equiv \text{diag}(\mathbf{W}_1, \dots, \mathbf{W}_p) \quad (48)$$

and the $6p$ -dimensional generalized twist \mathbf{t} , generalized working wrench \mathbf{w}^W and generalized non-working wrench \mathbf{w}^N are defined as

$$\mathbf{t} \equiv \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_p \end{bmatrix}, \quad \mathbf{w}^W \equiv \begin{bmatrix} \mathbf{w}_1^W \\ \vdots \\ \mathbf{w}_p^W \end{bmatrix}, \quad \mathbf{w}^N \equiv \begin{bmatrix} \mathbf{w}_1^N \\ \vdots \\ \mathbf{w}_p^N \end{bmatrix} \quad (49)$$

It can be shown that the kinematic constraints hold the following relation with the generalized twist

$$\mathbf{K}\mathbf{t} = \mathbf{0}_{6p} \quad (50)$$

where $\mathbf{0}_{6p}$ is the $6p$ -dimensional null vector, \mathbf{K} is the $6p \times 6p$ velocity constraint matrix with a rank of m which is equal to the number of independent holonomic constraints. The number of degrees of freedom of the system, i.e. independent variables, is determined as $n = 6p - m$. Denote the independent variables by \mathbf{s} , they can be related to the twist as

$$\mathbf{t} = \mathbf{T}\dot{\mathbf{s}} \quad (51)$$

$$\dot{\mathbf{t}} = \mathbf{T}\dot{\mathbf{s}} + \dot{\mathbf{T}}\mathbf{s} \quad (52)$$

where \mathbf{T} is a $6p \times n$ twist-mapping matrix.

By substituting eq.(51) into eq.(50), the following relation can be obtained

$$\mathbf{K}\mathbf{T} = \mathbf{0}_{6p} \quad (53)$$

where \mathbf{T} is the natural orthogonal complement of \mathbf{K} . As shown in (Angeles & Lee, 1988, 1989) the non-working vector \mathbf{w}^N lies in the null space of the transpose of \mathbf{T} . Thus, if both sides of eq. (46) are multiplied by \mathbf{T}^T , in the aid of eqs. (51 & 52), the system dynamics equations can be obtained as

$$\mathbf{I}\ddot{\mathbf{s}} + \mathbf{C}\dot{\mathbf{s}} = \mathbf{T}^T(\mathbf{w}^a + \mathbf{w}^g + \mathbf{w}^d) \quad (54)$$

where the $n \times n$ generalized inertia matrix \mathbf{I} and coupling matrix \mathbf{C} are defined as

$$\mathbf{I} \equiv \mathbf{T}^T \mathbf{M} \mathbf{T}, \quad \mathbf{C} \equiv \mathbf{T}^T (\mathbf{M} \dot{\mathbf{T}} + \mathbf{W} \mathbf{M} \mathbf{T}) \quad (55)$$

Furthermore, by defining the following generalized forces

$$\tau^a = \mathbf{T}^T \mathbf{w}^a, \quad \tau^g = \mathbf{T}^T \mathbf{w}^g, \quad \tau^d = \mathbf{T}^T \mathbf{w}^d, \quad \tau^l = \mathbf{I}\ddot{\mathbf{s}} + \mathbf{C}\dot{\mathbf{s}} \quad (56)$$

the inverse dynamics of the system can be given as

$$\tau^a = \tau^l - \tau^g - \tau^d \quad (57)$$

where τ^a is the vector representing the applied actuator forces.

3.2 Inverse dynamics

The key in applying the natural orthogonal complement method is to derive the expression for the twist-mapping matrix \mathbf{T} , which relates the speeds of the independent variables to the generalized twist. For the hexapod under study, the independent variable s is the vector representing the actuator displacement, with the total number of six, as defined before. The generalized twist is expressed as

$$\mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_6 \\ \mathbf{t}_{pc} \end{bmatrix} \quad (58)$$

Note that \mathbf{t}_1 to \mathbf{t}_6 are the twists for the six legs. Since the twist in eq.(36) is defined at the center of mass of the leg, \mathbf{T}_i represents the twist-mapping for the legs. For the moving platform, \mathbf{t}_{pc} is defined as the center of mass which may be expressed as

$$\mathbf{c}_p = \mathbf{h} + \mathbf{R}\bar{\mathbf{p}} \quad (59)$$

Differentiating eq.(59) gives

$$\dot{\mathbf{c}}_p = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{R}\bar{\boldsymbol{\rho}} \tag{60}$$

In the light of eq.(60), the following relation can be obtained

$$\mathbf{t}_{pc} = \mathbf{H}_p \mathbf{t}_p \tag{61}$$

where \mathbf{H}_p is the 6×6 matrix defined as

$$\mathbf{H}_p = \begin{bmatrix} \mathbf{1} & \mathbf{E}_\rho \\ \mathbf{O} & \mathbf{1} \end{bmatrix} \tag{62}$$

In eq.(62), \mathbf{E}_ρ is the cross-product matrix of $\mathbf{R}\bar{\boldsymbol{\rho}}$. Note that when $\bar{\boldsymbol{\rho}}$ is zero, i.e. the center of mass coincides with the coordinate origin, \mathbf{H}_p becomes an identity matrix, and $\mathbf{t}_{pc} = \mathbf{t}_p$.

The twist-mapping matrix \mathbf{T} for the hexapod under study can be given in the light of eqs.(17), (36) and (62) as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_6 \\ \mathbf{H}_p \mathbf{T}_p \end{bmatrix} \tag{63}$$

where \mathbf{T} is a 42×6 matrix. With \mathbf{T} , the generalized forces can be defined according to eq.(56), and the applied actuator forces can be determined according to eq.(57).

4. Simulation

4.1 Geometric and inertial parameters

The geometry of the base and the moving platform is shown in Figure 3.

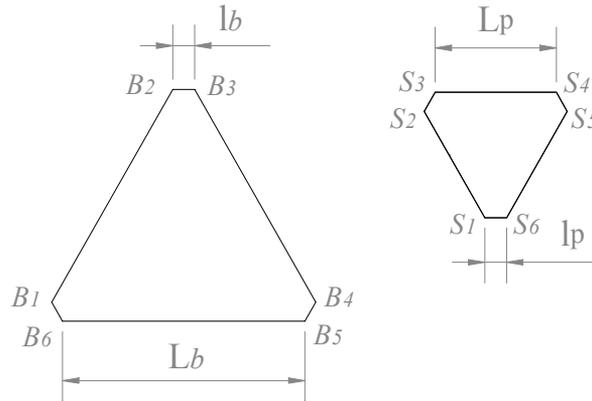


Fig. 3. Geometry of the base and the moving platform

Accordingly, the coordinates of vector \mathbf{b}_i with respect to the fixed frame are given as

$$\mathbf{b}_1 \equiv [L_b/2, -y_b, 0]^T \quad (64a)$$

$$\mathbf{b}_2 \equiv [(L_b + l_b)/2, -l_b c_s, 0]^T \quad (64b)$$

$$\mathbf{b}_3 \equiv [l_b/2, (L_b + l_b)c_s - y_b, 0]^T \quad (64c)$$

$$\mathbf{b}_4 \equiv [-l_b/2, (L_b + l_b)c_s - y_b, 0]^T \quad (64d)$$

$$\mathbf{b}_5 \equiv [(L_b + l_b)/2, l_b c_s - y_b, 0]^T \quad (64e)$$

$$\mathbf{b}_6 \equiv [-L_b/2, -y_b, 0]^T \quad (64f)$$

$$(64g)$$

$$(64h)$$

$$(64i)$$

where L_b and l_b are the long and short side of the base hexagon, $c_s = \cos(30^\circ)$, and $y_b = (L_b/2 + l_b) \tan(30^\circ)$. Likewise, the coordinates of vector $\bar{\mathbf{p}}_i$ with respect to the local frame are given as

$$\bar{\mathbf{p}}_1 \equiv [L_p/2, -y_p, 0]^T \quad (65a)$$

$$\bar{\mathbf{p}}_2 \equiv [(L_p + l_p)/2, -l_p c_s, 0]^T \quad (65b)$$

$$\bar{\mathbf{p}}_3 \equiv [l_p/2, (L_p + l_p)c_s - y_p, 0]^T \quad (65c)$$

$$\bar{\mathbf{p}}_4 \equiv [-l_p/2, (L_p + l_p)c_s - y_p, 0]^T \quad (65d)$$

$$\bar{\mathbf{p}}_5 \equiv [(L_p + l_p)/2, l_p c_s - y_p, 0]^T \quad (65e)$$

$$\bar{\mathbf{p}}_6 \equiv [-L_p/2, -y_p, 0]^T \quad (65f)$$

$$(65g)$$

$$(65h)$$

$$(65i)$$

where L_p and l_p are the long and short side of the moving platform hexagon, and $y_p = (L_p/2 + l_p) \tan(30^\circ)$. The geometric parameters and inertial parameters are given in Tables 1 and 2, respectively. In Table 1, S is the guideway length, γ is the guideway angle between the guideway and the vertical direction, and l is the length of the leg. These three parameters are the same for all the guideways and legs. Parameters L_b , l_b , L_p and l_p are defined in Figure 3. In Table 2, m is the mass, and I_{xx} , I_{yy} and I_{zz} are the moments of inertia.

S Guideway length	γ Guideway angle	l Leg length	L_b Long side BP	l_b Short side BP	L_p Long side MP	l_p Short side MP
0.60 m	45°	0.50 m	1.00 m	0.09 m	0.50 m	0.09m

Table 1. Geometric parameters

	m (kg)	$I_{xx} = I_{yy}$ (kg·m ²)	I_{zz} (kg·m ²)
Platform	3.983	0.068	0.136
Leg	0.398	0.0474	-

Table 2. Inertial parameters

4.2 Numerical example

A simulation program has been developed using Matlab based on the method described in the previous sections. In terms of computation, as can be seen from eq. (54), the inverse dynamics of the hexapods mainly involves the twist-mapping matrix \mathbf{T} and its derivative, which could be computed numerically for each time interval. This way, it is computationally more efficient. To further speed up computation, parallel computation techniques could be used. As shown in Figure 4, the motion part including actuator speeds and accelerations could be computed in parallel to the inertia part including mass matrix \mathbf{I} and coupling matrix \mathbf{C} . Since the program is done in Matlab, parallel computation is not realized. However, this strategy can certainly be applied to model-based control using the dynamic equations.

In terms of singularity, as can be shown from eq. (63), the twist-mapping matrix \mathbf{T} becomes degenerate when the moving platform Jacobian \mathbf{J}_p ($\mathbf{T}_p = \mathbf{J}_p^{-1}$) is singular.

The movement of the moving platform is defined in terms of 3-4-5 polynomials that guarantee zero velocities and zero accelerations at the beginning and at the end. The selection of a smooth motion profile is very important for the hexapod as it is operated under high speeds. The conventional machine tools are run at a maximum velocity of 30m/min with a maximum acceleration of 0.3 g. Hexapods can run at a maximum velocity of 100 m/min with a maximum acceleration over 1 g.

The first simulation is for high speed, with a maximum velocity of 102 m/min. The second simulation is for low high speed with a maximum velocity of 30 m/min. In both cases, the hexapod moves a distance of 0.1m along the z axis. The initial position of the moving platform is at $x_o = 0$, $y_o = 0$ and $z_o = 0.7$ m. Figure 4 shows the velocity profiles of the moving platform. Figures 5, 6 and 7 show the displacements, velocities and accelerations of the six actuators, respectively. Figure 8 shows the computed actuator forces. The simulations show that high speed motions result in large actuator forces.

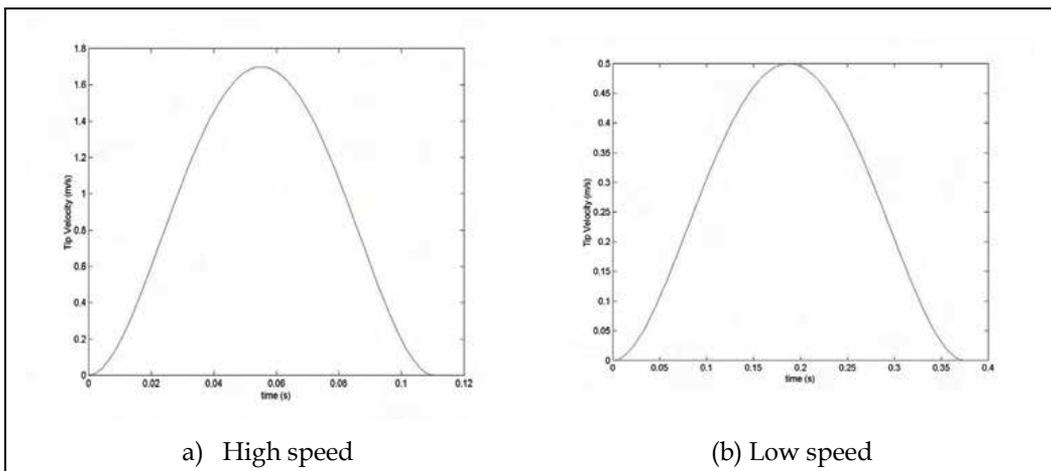


Fig. 4. Motion Profile of moving platform

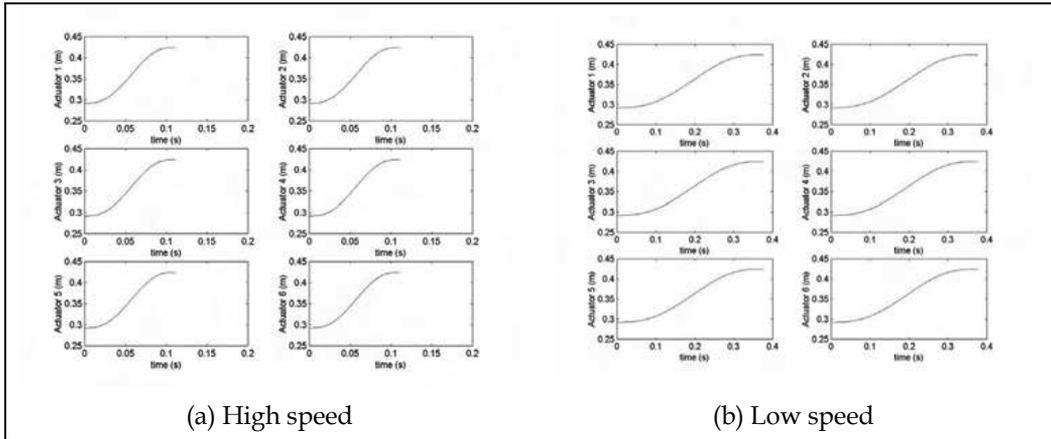


Fig. 5. Actuator displacements

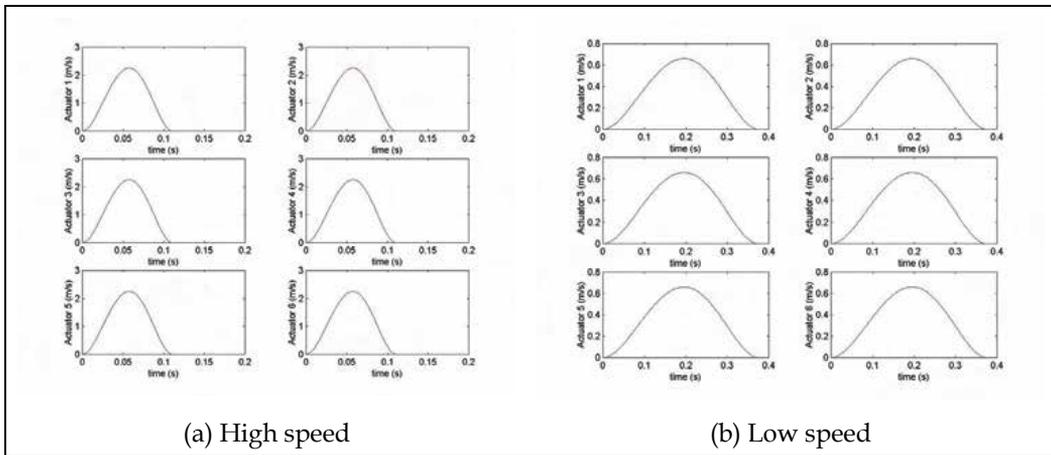


Fig. 6. Actuator velocities

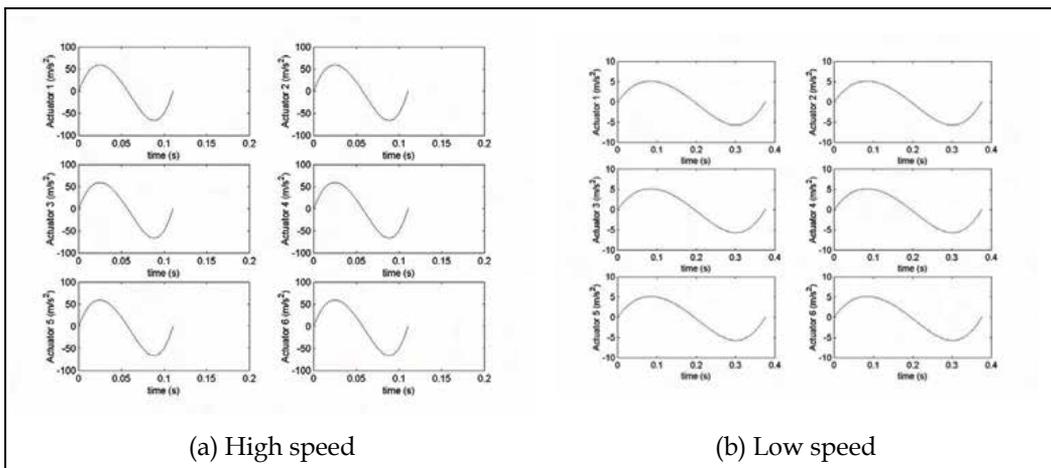


Fig. 7. Actuator accelerations

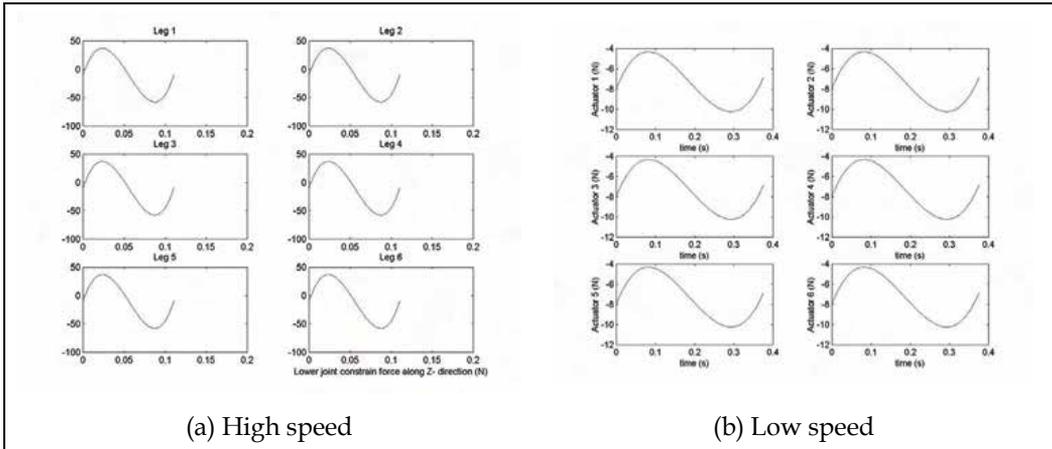


Fig. 8. The computed actuator forces

5. Static balancing of the hexapod

The static and dynamic balancing is a classic problem in the theory of machines and mechanisms. In particular, when a mechanism is not statically balanced, the weight of linkage produces force or torque at actuators under static conditions and actuators have to contribute to support the weight of the moving links for any configurations. The problem becomes more serious for the parallel manipulator applied as flight simulator where the weight of the moving platform is very large with respect to the masses of the links. Static balancing also called *gravity compensation* is important. If the forces/torques exerted by joint actuators are reduced, the full potential of machine will be improved.

In this paragraph, following the same approach presented by Gosselin, the static balancing of the hexapod with the fixed-length legs is studied.

5.1 Static balancing using counterweight

The static balancing of the parallel manipulator under study is investigated using counterweights. The base coordinate frame $Oxyz$ frame, is fixed to the base with Z -axis pointing vertically upward and the moving coordinate frame $O'x'y'z'$ is attached to the moving platform. The Cartesian coordinates used to describe the pose of the platform are as shown in Fig. 9 given by the position of O' with respect to the fixed frame and the orientation of the platform represented by the rotation matrix \mathbf{Q}

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} \quad (66)$$

Using the counterweights, static balancing is obtained if the global center mass of the mechanism is kept stationary at any values of the independent variables. To choose an suitable constant, namely

$$M\mathbf{r} = \mathbf{0} \quad (67)$$

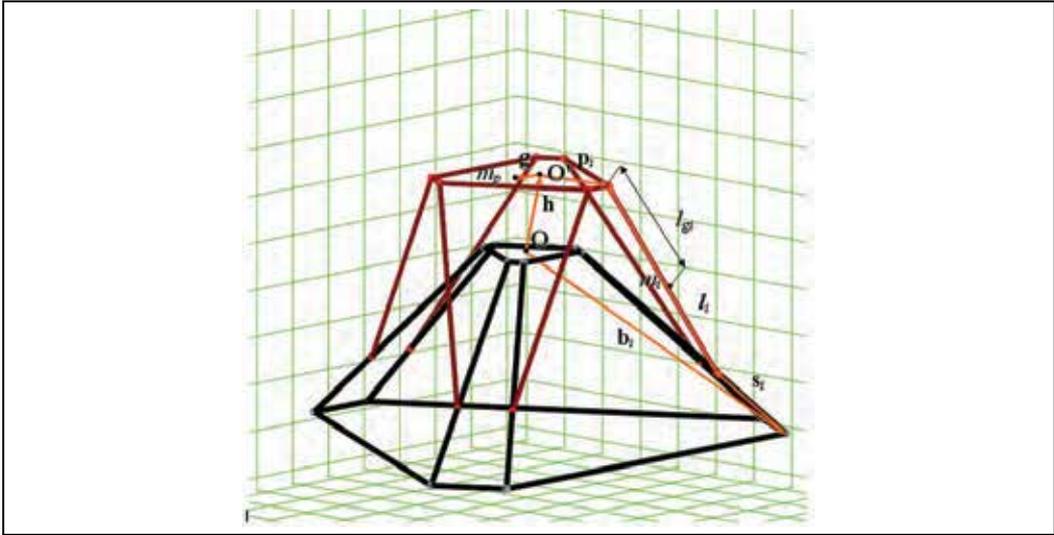


Fig.9 Kinematic mode

where \mathbf{r} is the position vector of the global mass center, and M is:

$$M = m_p + \sum_{i=1}^6 m_i \tag{68}$$

where m_p is the mass of the platform, m_i is the mass of the leg. The global centre of the mass of the manipulator is written as

$$M\mathbf{r} = m_p\mathbf{r}_p + \sum_{i=1}^6 m_i\mathbf{r}_i \tag{69}$$

where \mathbf{r}_p is the platform center of the mass, \mathbf{r}_i is the leg center of the mass. From Fig. 9, vectors \mathbf{r}_p , \mathbf{r}_i can be derived, and substituted into eq.(69), yielding

$$M\mathbf{r} = m_p(\mathbf{h} + \mathbf{Q} \cdot \mathbf{g}) + \sum_{i=1}^6 m_i \left[(\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i) - (\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i - \mathbf{b}_i - \mathbf{s}_i) \frac{l_{gi}}{l_i} \right] \tag{70}$$

where \mathbf{g} is the vector center of mass of the moving platform with respect to the frame $O'x'y'z'$, \mathbf{h} is the position of O' with respect to the fixed frame, \mathbf{p}_i is the position of the spherical joint with respect to the moving coordinate frame, \mathbf{b}_i is the position of the lower end of the guideway with respect to the fixed frame, l_i is the length of the leg, \mathbf{s}_i can be written, for $i=1, \dots, 6$, as

$$\mathbf{s}_i = \rho_i \cdot \hat{\mathbf{s}}_i \tag{71}$$

where $\hat{\mathbf{s}}_i$ is the unit vector of guideway, ρ_i is the independent variable of the prismatic joint. In concise form, eq. (70) is expressed as

$$M\mathbf{r} = A_1\mathbf{h} + \mathbf{QB} + \sum_{i=1}^6 A_{5i}\mathbf{s}_i + \mathbf{A}_0 \quad (72)$$

where

$$A_1 = m_p + \sum_{i=1}^6 m_i \left(1 - \frac{l_{gi}}{l_i}\right) \quad (73)$$

$$\mathbf{B} = m_p\mathbf{g} + \sum_{i=1}^6 m_i\mathbf{p}_i \left(1 - \frac{l_{gi}}{l_i}\right) \quad (74)$$

$$A_{5i} = m_i \frac{l_{gi}}{l_i}, \text{ for } i=1,\dots,6 \quad (75)$$

$$\mathbf{A}_0 = \sum_{i=1}^6 A_{5i}\mathbf{b}_i \quad (76)$$

The conditions for static balancing can be given for $i=1,\dots,6$, as follows:

$$A_1 = 0, \mathbf{B} = 0, A_{5i} = 0, \mathbf{A}_0 = 0 \quad (77)$$

From conditions $A_{5i} = 0, i=1,\dots,6$, one can obtain

$$l_{gi} = 0 \quad (78)$$

By condition $A_1 = 0$, one can obtain

$$m_p = - \sum_{i=1}^6 m_i \quad (79)$$

Eq.(79) shows that the balancing by counterweight is impossible. If it was substituted in the condition $\mathbf{B} = 0$,

$$m_p\mathbf{g} + \sum_{i=1}^6 m_i\mathbf{p}_i = \mathbf{0} \quad (80)$$

then one can obtain

$$\mathbf{g} = \frac{\sum_{i=1}^6 m_i\mathbf{p}_i}{\sum_{i=1}^6 m_i} \quad (81)$$

From eq. (81), it shows that the manipulator could be balanced by a device that provide a force that is

1. equal to the weight of the links and the platform;
2. in opposite direction of the weight.

5.2 Static balancing wit a pantograph counterweight

Since it is shown that the static balancing of the examined mechanism is impossible with the help of counterweights, we propose a method to add a pantograph connecting the moving platform O' to the fixed platform O , as shown in Fig. 10. The pantograph is a device that allows to keep two end points on the same line and keep their distance at the centre with a constant ratio. In this application it is possible to use a pantographs with two or more mesh as shown in Fig. 10 and Fig. 11, respectively. In both case the manipulator is balanced. The pantograph is fixed to the moving platform on the point O' by a spherical joint and fixed to the point O by an universal joint. The leg counterweight is shown in Fig.12.

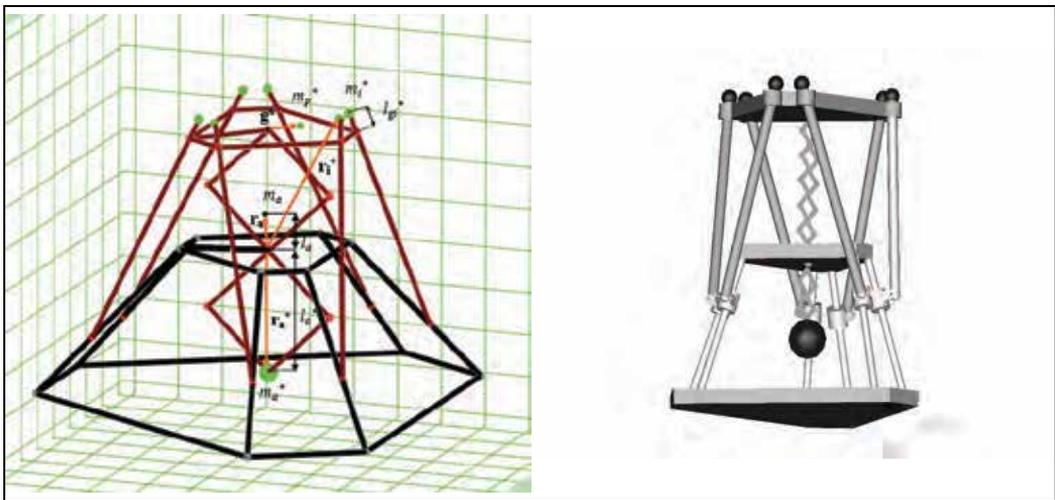


Fig. 10. Model with counterweights mass

Fig. 11. Balanced Hexapod using pantograph

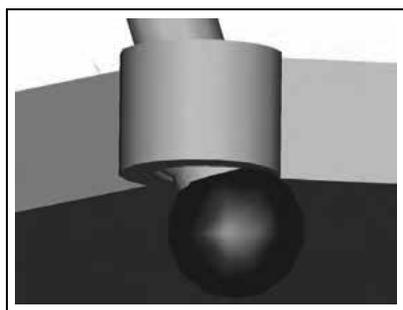


Fig. 12. Leg counterweight

In this case, the mass M becomes,

$$M = m_p + m_p^* + m_a + m_a^* + \sum_{i=1}^6 m_i + \sum_{i=1}^6 m_i^* \quad (82)$$

where m_p and m_p^* are the mass of the platform and the mass of the platform counterweight, m_i and m_i^* are the mass of the legs and the mass of the legs counterweights, m_a and m_a^* are the mass of the pantograph and the mass of the of the pantograph counterweight. In this case, the global center of the mass of the manipulator is written as

$$M\mathbf{r} = m_p\mathbf{r}_p + m_p^*\mathbf{r}_p^* + m_a\mathbf{r}_a + m_a^*\mathbf{r}_a^* + \sum_{i=1}^6 m_i\mathbf{r}_i + \sum_{i=1}^6 m_i^*\mathbf{r}_i^* \quad (83)$$

where \mathbf{r}_p and \mathbf{r}_p^* are the platform center of the mass and the platform counterweight position, \mathbf{r}_i and \mathbf{r}_i^* are the legs center of the mass and the legs counterweight position, \mathbf{r}_a and \mathbf{r}_a^* are the pantograph center of the mass and the pantograph counterweight position. From Figs. 9-10, vectors \mathbf{r}_p , \mathbf{r}_p^* , \mathbf{r}_a , \mathbf{r}_a^* , \mathbf{r}_i and \mathbf{r}_i^* can be derived and substituted into eq.(83), yielding

$$\begin{aligned} M\mathbf{r} = & m_p(\mathbf{h} + \mathbf{Q} \cdot \mathbf{g}) + m_p^*(\mathbf{h} + \mathbf{Q} \cdot \mathbf{g}^*) + m_a \left(\mathbf{h} \frac{l_a}{|\mathbf{h}|} \right) + m_a^* \left(\mathbf{h} \frac{l_a^*}{|\mathbf{h}|} \right) + \\ & + \sum_{i=1}^6 m_i \left[(\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i) - (\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i - \mathbf{b}_i - \mathbf{s}_i) \frac{l_{gi}}{l_i} \right] + \\ & + \sum_{i=1}^6 m_i^* \left[(\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i) - (\mathbf{h} + \mathbf{Q} \cdot \mathbf{p}_i - \mathbf{b}_i - \mathbf{s}_i) \frac{l_{gi}^*}{l_i} \right] \end{aligned} \quad (84)$$

where, l_a is the center of mass of the pantograph with respect to the fixed frame, l_a^* is the pantograph counterweight position with respect to the fixed frame, l_{gi} is the length of the leg counterweight link, l_i is the length of the leg, \mathbf{s}_i can be written, for $i=1, \dots, 6$, as

$$\mathbf{s}_i = \rho_i \cdot \hat{\mathbf{s}}_i \quad (85)$$

In concise form, eq.(84) can be expressed as

$$M\mathbf{r} = A_1\mathbf{h} + \mathbf{QB} + \sum_{i=1}^6 A_{5i}\mathbf{s}_i + \mathbf{A}_0 \quad (86)$$

where

$$A_1 = m_p + m_p^* + m_a \frac{l_a}{|\mathbf{h}|} + m_a^* \frac{l_a^*}{|\mathbf{h}|} + \sum_{i=1}^6 m_i \left(1 - \frac{l_{gi}}{l_i} \right) + \sum_{i=1}^6 m_i^* \left(1 - \frac{l_{gi}^*}{l_i} \right) \quad (87)$$

$$\mathbf{B} = m_p\mathbf{g} + m_p^*\mathbf{g}^* + \sum_{i=1}^6 m_i\mathbf{p}_i \left(1 - \frac{l_{gi}}{l_i} \right) + \sum_{i=1}^6 m_i^*\mathbf{p}_i \left(1 - \frac{l_{gi}^*}{l_i} \right) \quad (88)$$

$$A_{5i} = m_i \frac{l_{gi}}{l_i} + m_i^* \frac{l_{gi}^*}{l_i}, \quad i=1, \dots, 6 \quad (89)$$

$$\mathbf{A}_0 = \sum_{i=1}^6 A_{5i} \mathbf{b}_i \quad (90)$$

The conditions for static balancing can be given, for $i=1,\dots,6$, as follows

$$A_i = 0, \mathbf{B} = 0, A_{5i} = 0, \mathbf{A}_0 = 0 \quad (91)$$

From conditions $A_{5i} = 0$, for $i=1,\dots,6$, one can obtain

$$m_i \frac{l_{gi}}{l_i} + m_i^* \frac{l_{gi}^*}{l_i} = 0 \quad (92)$$

From eq. (92), for $i=1,\dots,6$, the following is obtained

$$l_{gi}^* = -\frac{m_i l_{gi}}{m_i^*} \quad (93)$$

By condition $A_i = 0$, i.e.,

$$m_p + m_p^* + \sum_{i=1}^6 (m_i + m_i^*) + m_a \frac{l_a}{|\mathbf{h}|} + m_a^* \frac{l_a^*}{|\mathbf{h}|} = 0 \quad (94)$$

one can obtain

$$l_a^* = -\frac{|\mathbf{h}|}{m_a^*} \left(m_p + m_p^* + \sum_{i=1}^6 (m_i + m_i^*) + m_a \frac{l_a}{|\mathbf{h}|} \right) \quad (95)$$

Finally, condition $\mathbf{B} = 0$ leads to the following

$$m_p \mathbf{g} + m_p^* \mathbf{g}^* + \sum_{i=1}^6 (m_i + m_i^*) \mathbf{p}_i = \mathbf{0} \quad (96)$$

Eq.(96) shows that the static balancing can be achieved by fixing the global center of the mass of the moving platform, that of the legs and their counterweights at the same position, O' . In order to obtain it, the platform counterweight should be placed in the position:

$$\mathbf{g}^* = \frac{m_p \mathbf{g} + \sum_{i=1}^6 (m_i + m_i^*) \mathbf{p}_i}{m_p^*} \quad (97)$$

Simulation is carried out to demonstrate the proposed method. The results are shown in Figs. 13-14, from which it can be seen that the centre of mass of the robot is non-stationary for non balanced case, while it is fixed for the balanced case.

After static balancing the global mass of the device increases by

$$\Delta M = m_p^* + m_a + m_a^* + \sum_{i=1}^6 m_i^* \quad (98)$$

The negative effect for the dynamic performance by the increasing global mass can be reduced by optimum design of the pantograph. A graph can be arranged to provide such help. Fig. 15 shows the ratios,

$$\frac{M + \Delta M}{M}, \frac{I_i + I_i^*}{I_i}, \frac{I_a + I_a^*}{I_a}, \tag{99}$$

which vary respect to the ratio r_a^*/h and l_{gi}^*/l_{gi} and where I_i is the moment of inertia of the leg, I_i^* is the moment of inertia of the leg counterweight with respect of P_i , I_a is the moment of inertia of the moving platform and I_a^* is the moment of inertia of the pantograph counterweight with respect of O. It should be noted that with a suitable design it is possible to reduce ΔM at the same time, it may increase I_i and I_a . The effect of gravity compensation on the dynamic performances was studied in detail in (Xi, 1999).

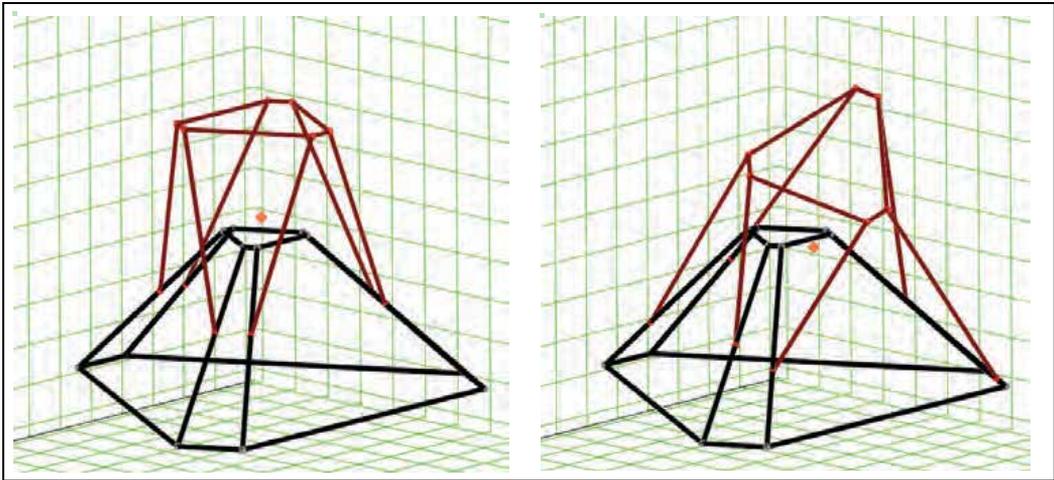


Fig. 13. Mobile center of mass Hexapod

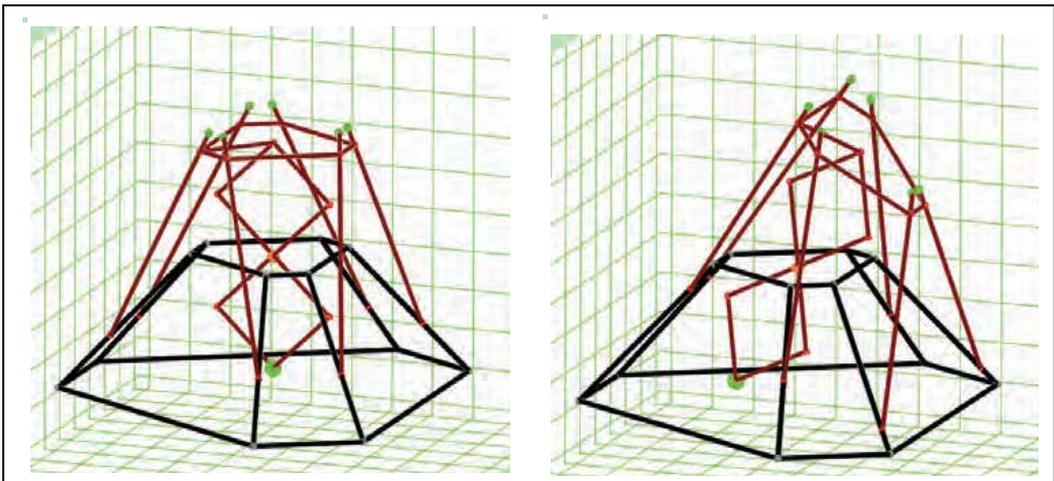


Fig. 14. Fixed center of mass of Balanced Hexapod

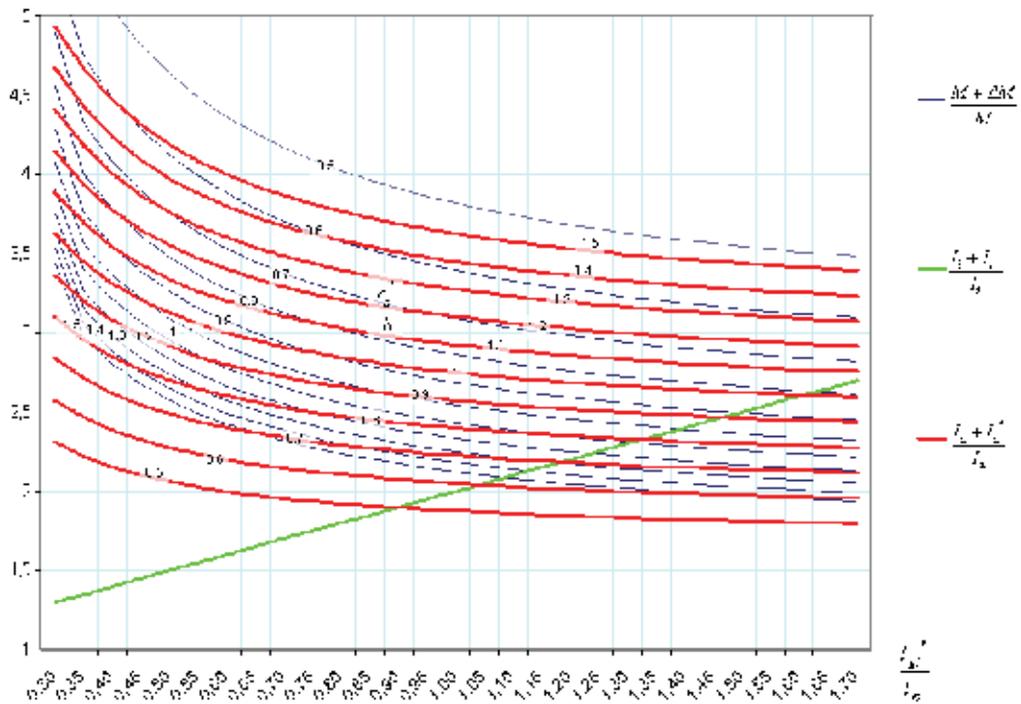


Figure 15. Graph for optimum design

	Input	
Mobile platform		
mass [kg]	short side [mm]	long side [mm]
8	200	800
Fixed platform 1		
mass [kg]	short side [mm]	long side [mm]
/	100	400
Fixed platform 2		
mass [kg]	short side [mm]	long side [mm]
/	250	1000
leg		
mass [kg]	l_i [mm]	l_{gi} [mm]
0.5	750	375
Pantograph		
mass [kg]	side length [mm]	r_a [mm]
3	100	0
	Output	
m_a^* [kg]	m_i^* [kg]	
17	1	

Table 3. Geometric and inertial parameters

7. Conclusion

In this chapter, the inverse dynamics of hexapods with fixed-length legs is analyzed using the natural orthogonal complement method, with considering the mass of the moving platform and those of the legs. A complete kinematics model is developed, which leads to an explicit expression for the twist-mapping matrix. Based on that, the inverse dynamics equations are derived that can be used to compute the required applied actuator forces for the given movement of the moving platform. The developed method has been implemented and demonstrated by simulation.

Successively, the static balancing of hexapods is addressed. The expression of the global center of mass is derived, based on which a set of static balancing equations has been obtained. It is shown that this type of parallel mechanism cannot be statically balanced by counterweights because prismatic joints do not have a fixed point to pivot as revolute joints. A new design is proposed to connect the centre of the moving platform to that of the fixed platform by a pantograph. The conditions for static balancing are derived. This mechanism is able to release the actuated joints from the weight of the moving legs for any configurations of the robot.

In the future research the leg inertia will be include for modeling the dynamics of the hexapod for high-speed applications.

8. References

- Angeles, J. & Lee, S. (1988). The Formulation of Dynamical Equations of Holonomic Mechanical Systems Using a Natural Orthogonal Complement, *ASME J. Applied Mechanics*, Vol. 55, pp. 243-244, ISSN: 0021-8936
- Angeles, J. & Lee, S. (1989). The Modeling of Holonomic Mechanical Systems Using a Natural Orthogonal Complement, *Trans. Canadian Society of Mechanical Engineering*, Vol. 13, No. 4, pp. 81-89, ISSN: 0315-8977
- Angeles, J. & Ma, O. (1988). Dynamic Simulation of n-axis Serial Robotic Manipulators Using a Natural Orthogonal Complement, *The International Journal of Robotics Research*, Vol. 7, No. 5, pp. 32-47, ISSN: 0278-3649
- Do, W. Q. D. & Yang, D. C. H. (1988). Inverse Dynamics and Simulation of a Platform Type of Robot, *The International Journal of Robotics Research*, Vol. 5, No. 3, pp. 209-227, ISSN: 0278-3649
- Fichter, E. F. (1986). A Stewart Platform-based Manipulator: General Theory and Practical Construction, *The International Journal of Robotics Research*, Vol. 5, No. 2, pp. 157-182, ISSN: 0278-3649
- Fijany, A., & Bejezy, A. K., (1991). Parallel Computation of Manipulator Inverse Dynamics, *Journal of Robotic Systems*, Vol. 8, No. 5, pp. 599-635, ISSN: 0741-2223
- Geng, Z.; Haynes, L. S.; Lee, T. D. & Carroll, R. L. (1992). On the Dynamic and Kinematic Analysis of a Class of Stewart Platforms, *Robotics and Autonomous Systems*, Vol. 9, No. 4, pp. 237-254, ISSN: 0921-8890
- Gosselin, C. M. & Wang, J. (1998). On the design of gravity-compensated six-degree-of-freedom parallel mechanisms, *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, ISBN: 0-7803-4300-X
- Hashimoto, K. and Kimura, H., (1989). A New Parallel Algorithm for Inverse Dynamics, *The International Journal of Robotics Research*, Vol. 8, No. 1, pp. 63-76, ISSN: 0278-3649
- Hervé, J. M. (1986). Device for counter-balancing the forces due to gravity in a robot arm, *United States Patent*, 4,620,829

- Honegger, M.; Codourey, A. & Burdet, E. (1997). Adaptive Control of the Hexaglide a 6 DOF Parallel Manipulator, *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997, ISBN: 0-7803-3612-7
- Kazerooni, H. & Kim, S. (1990). A new architecture for direct drive robot, *Proceedings of the ASME Mechanism Conference*, Vol. DE-25, pp. 21-28, Sept. 16-19, 1990, Chicago, IL
- Nathan, R. H., (1985). A constant force generation mechanism, *ASME Journal of Mechanism, Transmissions, and Automation in Design*, Vol. 107, No. 4, pp 508-512, ISSN: 0738-0666
- Pritschow, G., & Wurst, K.-H., (1997). Systematic Design of Hexapods and Other Parallel Link Systems, *Annals of the CIRP*, Vol. 46/1, pp. 291-295, Elsevier, ISSN: 0007-8506
- Saha, K.S. & Angeles, J. (1991). Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement, *ASME Journal of Applied Mechanics*, Vol. 58, pp.238-243, ISSN: 0021-8936
- Streit, D.A. & Gilmore, B.J. (1989). Perfect spring equilibrators for rotatable bodies, *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, No. 4, pp. 451-458, ISSN: 0738-0666
- Streit, D.A. & Shin, E. (1990). Equilibrators for planar linkage. *Proceedings of the ASME Mechanism Conference*, Vol. DE-25, pp. 21-28, Cincinnati, OH, , May 1990, Chicago, ISBN: 0-8186-9061-5
- Sugimoto, K. (1987). Kinematic and Dynamic Analysis of Parallel Manipulators by Means of Motor Algebra, *ASME Journal Mechanisms, Transmissions, and Automation in Design*, Vol. 109, pp. 3-7, ISSN: 0738-0666
- Susuki, M.; Watanabe K.; Shibukawa, T.; Tooyama, T. & Hattori, K. (1997). Development of Milling Machine with Parallel Mechanism, *Toyota Technical Review*, Vol. 47, No. 1, pp. 125-130
- Tlustý, J.; Ziegert, J. & Ridgeway, S. (1999). Fundamental Comparison of the Use of Serial and Parallel Kinematics for Machine Tools, *Annals of the CIRP*, Vol. 48, pp. 351-356, Elsevier, ISSN: 0007-8506
- Tsai, L. W. (2000). Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work, *ASME Journal of Mechanical Design*, Vol. 122, pp. 3-9, ISSN: 1050-0472
- Ulrich, N. & Kumar, V. (1991). Passive mechanical gravity compensation for robot manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1536-1541, Sacramento, CA, USA, April 1991, ISBN: 0-8186-2163-X
- Walsh, G.J.; Strei, D.A & Gilmore, B.J. (1991). Spatial spring equilibrators theory, *Mechanism and Machine Theory*, Vol. 26, No 2, pp. 155-170, ISSN: 0094-114X
- Wang, J. & Gosselin, C. M. (2000). Static balancing of spatial four-degree-of freedom parallel mechanisms, *Mechanism and Machine Theory*, Vol. 35, pp. 563-592, ISSN: 0094-114X
- Wang, J. & Gosselin, C.M. (1999). Static balancing of spatial three degree freedom parallel mechanism, *Mechanism and Machine Theory*, Vol. 34, pp. 437-452, ISSN 0094-114X
- Xi F.; Russo, A. & Sinatra, R., (2005). Static Balancing of parallel robots, *Mechanism and Machine Theory*, Vol. 40, No 2, pp. 131-258, ISSN: 0094-114X
- Xi, F. & Sinatra, R. (1997). Effect of Dynamic Balancing on Four-bar Linkage Vibrations, *Mechanism and Machine Theory*, Vol. 32, No. 6, pp.715-728, ISSN: 0094-114X
- Xi, F. & Sinatra, R. (2002). Inverse Dynamics of Hexapods using the Natural Orthogonal Complement Methods, *Journal of Manufacturing Systems*, Vol. 21, No 2, pp.73-82, ISSN: 0258-6125
- Xi, F. (1999). Dynamic Balancing of Hexapods for High-speed Applications, *Robotica*, Vol. 17, pp. 335-342, ISSN: 0263-5747
- Zanganesh, K. E.; Sinatra, R. & Angeles J. (1997). Dynamics of a Six-Degree-of-Freedom Parallel Manipulator with Revolute Legs, *Robotica*, Vol. 15, pp. 385-394, ISSN: 0263-5747.

Cartesian Parallel Manipulator Modeling, Control and Simulation

Ayssam Elkady¹, Galal Elkobrosy², Sarwat Hanna² and Tarek Sobh¹

*University of Bridgeport¹,
Alexandria University²,
USA¹,
Egypt²*

1. Introduction

Parallel manipulators are robotic devices that differ from the more traditional serial robotic manipulators by their kinematic structure. Parallel manipulators are composed of multiple closed kinematic loops. Typically, these kinematic loops are formed by two or more kinematic chains that connect a moving platform to a base, where one joint in the chain is actuated and the other joints are passive. This kinematic structure allows parallel manipulators to be driven by actuators positioned on or near the base of the manipulator. In contrast, serial manipulators do not have closed kinematic loops and are usually actuated at each joint along the serial linkage. Accordingly, the actuators that are located at each joint along the serial linkage can account for a significant portion of the loading experienced by the manipulator, whereas the links of a parallel manipulator generally need not carry the load of the actuators. This allows the parallel manipulator links to be made lighter than the links of an analogous serial manipulator. The most noticeable interesting features of parallel mechanisms being:

- High payload capacity.
- High throughput movements (high accelerations).
- High mechanical rigidity.
- Low moving mass.
- Simple mechanical construction.
- Actuators can be located on the base.

However, the most noticeable disadvantages being:

- They have smaller workspaces than serial manipulators of similar size.
- Singularities within working volume.
- High coupling between the moving kinematic chains.

1.1 Prior work

Among different types of parallel manipulators, the Gough-Stewart platform has attracted most attention because it has six degrees of freedom (DOF). It was originally designed by (Stewart, 1965). Generally, this manipulator has six limbs. Each one is connected to both the base and the moving platform by spherical joints located at each end of the limb.

Actuation of the platform is typically accomplished by changing the lengths of the limbs. Although these six-limbed manipulators offer good rigidity, simple inverse kinematics, and high payload capacity, their forward kinematics are difficult to solve, position and orientation of the moving platform are coupled and precise spherical joints are difficult to manufacture at low cost.

To overcome the above shortcomings, parallel manipulators with fewer than six degrees of freedom have been investigated. For examples, (Ceccarelli, 1997) proposed a 3-DOF parallel manipulator (called CaPaMan) in which each limb is made up of a planar parallelogram, a prismatic joint, and a ball joint. But these manipulators have coupled motion between the position and orientation of the end-effector. The 3-RRR (Revolute Revolute Revolute) spherical manipulator was studied in detail by (Gosselin & Angeles, 1989).

Several spatial parallel manipulators with a rotational moving platform, called rotational parallel manipulators (RPMs), were proposed (Di Gregorio, 2001), (Karouia & Herve, 2000) and (Vischer & Clavel, 2000). (Clavel, 1988) at the Swiss Federal Institute of Technology designed a 3-DOF parallel manipulator that does not suffer from the first two of the listed disadvantages of the Stewart manipulator. Closed-form solutions for both the inverse and forward kinematics were developed for the DELTA robot (Gosselin & Angeles, 1989). The DELTA robot has only translational degrees of freedom. Additionally, the position and orientation of the moving platform are uncoupled in the DELTA design. However, the DELTA robot construction does employ spherical joints. (Tsai, 1996) presented the design of a spatial 3-UPU (Universal Prismatic Universal) manipulator and pointed out the conditions that lead to pure translational motion and its kinematics was studied further by (Di-Gregorio & Parenti-Castelli, 1998). (Tsai, 1996) and (Tsai et al., 1996) designed a 3-DOF TPM (Translational Parallel Manipulator) that employs only revolute joints and planar parallelograms. (Tsai & Joshi, 2002) analyzed the kinematics of four TPMs for use in hybrid kinematic machines. (Carricato & Parenti-Castelli, 2001) developed a family of 3-DOF TPMs. (Fang & Tsai, 2002) presented a systematic methodology for structure synthesis 3-DOF TPMs using the theory of reciprocal screws (Kim & Tsai, 2002).

Han Sung Kim and Lung-Wen Tsai (Kim & Tsai, 2002) presented a parallel manipulator called CPM (figure 1) that employs only revolute and prismatic joints to achieve translational motion of the moving platform. They described its kinematic architecture and discussed two actuation methods. For the rotary actuation method, the inverse kinematics provides two solutions per limb, and the forward kinematics leads to an eighth-degree polynomial. Also, the rotary actuation method results in many singular points within the workspace. On the other hand, for the linear actuation method, there exists a one-to-one correspondence between the input and output displacements of the manipulator. Also, they discussed the effect of misalignment of the linear actuators on the motion of the moving platform. They suggested a method to maximize the stiffness to minimize the deflection at the joints caused by the bending moment because each limb structure is exposed to a bending moment induced by the external force exerted on the end-effector.

2. Manipulator description and kinematics

2.1 Manipulator structure

The Cartesian Parallel Manipulator, shown in figure 1, consists of a moving platform that is connected to a fixed base by three limbs. Each limb is made up of one prismatic and three revolute joints and all joint axes are parallel to one another.

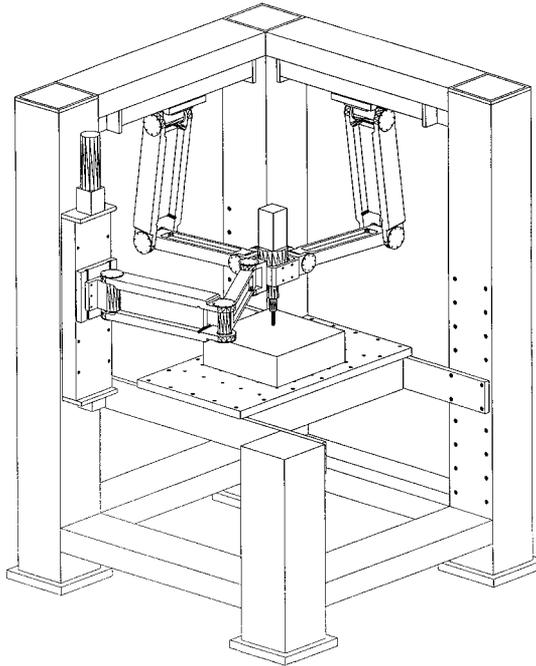


Figure 1: Assembly drawing of the prototype parallel manipulator.

2.2 Kinematic structure

The kinematic structure of the CPM is shown in figure 2 where a moving platform is connected to a fixed base by three PRRR (Prismatic Revolute Revolute Revolute) limbs. The origin of the fixed coordinate frame is located at point O and a reference frame XYZ is attached to the fixed base at this point. The moving platform is symbolically represented by a square whose length side is $2L$ defined by B_1 , B_2 , and B_3 and the fixed base is defined by three guide rods passing through A_1 , A_2 , and A_3 , respectively. The three revolute joint axes in each limb are located at points A_i , M_i , and B_i , respectively, and are parallel to the ground-connected prismatic joint axis. Furthermore, the three prismatic joint axes, passing through point A_i , for $i = 1, 2$, and 3 , are parallel to the X , Y , and Z axes, respectively. Specifically, the first prismatic joint axis lies on the X -axis; the second prismatic joint axis lies on the Y axis; and the third prismatic joint axis is parallel to the Z axis. Point P represents the center of the moving platform. The link lengths are L_1 , and L_2 . The starting point of a prismatic joint is defined by d_{0i} and the sliding distance is defined by $d_i - d_{0i}$ for $i = 1, 2$, and 3 .

2.3 Kinematics constraints

For this analysis, the position of the end-effector is considered known, and is given by the position vector $P = [x, y, z]$ which defines the location of P at the center of the moving platform in the XYZ coordinate frame. The inverse kinematics analysis produces a set of two joint angles for each limb (θ_{i1} and θ_{i2} for the i^{th} limb) that define the possible postures for each limb for the given position of the moving platform.

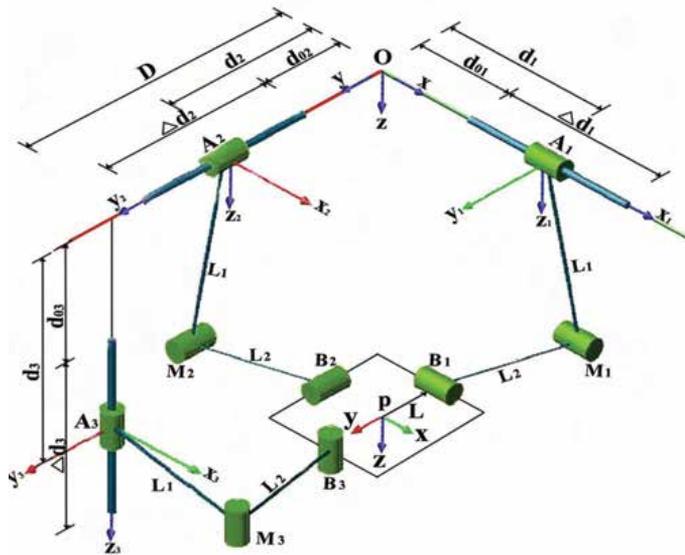


Figure 2: Spatial 3-PRRR parallel manipulator.

2.3.1 The first limb

A schematic diagram of the first limb of the CPM is sketched in figure 3, and then the relationships for the first limb are written for the position P[x, y, z] in the coordinate frame XYZ.

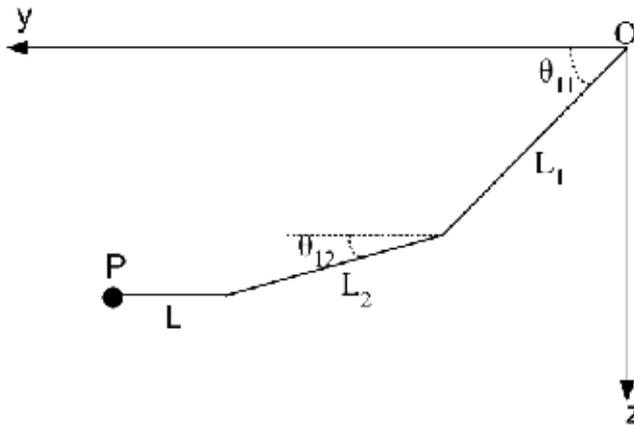


Figure 3: Description of the joint angles and link lengths for the first limb.

$$y = L_1 \cos \theta_{11} + L_2 \cos \theta_{12} + L \tag{1}$$

$$z = L_1 \sin \theta_{11} + L_2 \sin \theta_{12} \tag{2}$$

$$L_2^2 = (y - L_1 \cos \theta_{11} - L)^2 + (z - L_1 \sin \theta_{11})^2 \tag{3}$$

2.3.2 The second limb

A schematic diagram of the second limb of the CPM is sketched in figure 4, and then the relationships for the second limb are written for the position $P[x, y, z]$ in the coordinate frame XYZ.

$$z = L_1 \cos \theta_{21} + L_2 \cos \theta_{22} \tag{4}$$

$$x = L_1 \sin \theta_{21} + L_2 \sin \theta_{22} + L \tag{5}$$

$$L_2^2 = (z - L_1 \cos \theta_{21})^2 + (x - L_1 \sin \theta_{21} - L)^2 \tag{6}$$

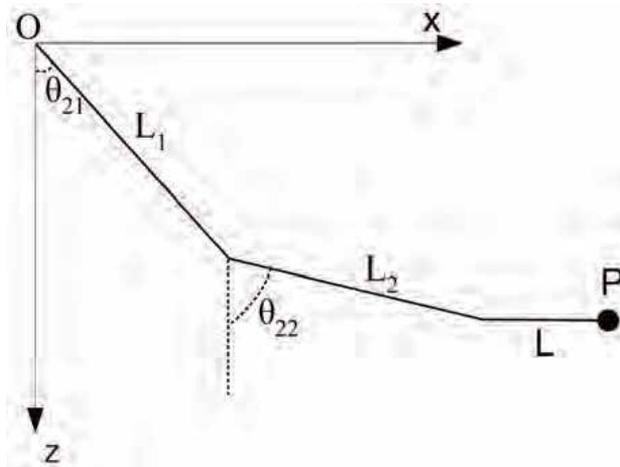


Figure 4: Description of the joint angles and link for the second limb.

2.3.3 The third limb

A schematic diagram of the third limb of the CPM is sketched in figure 5, and then the relationships for the third limb are written for the position $P[x, y, z]$ in the coordinate frame XYZ.

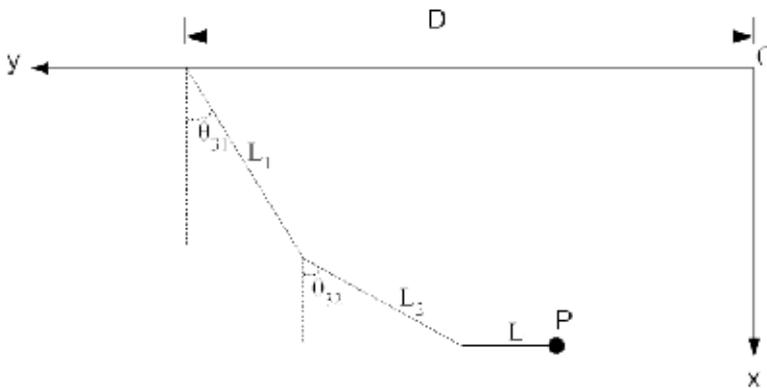


Figure 5: Description of the joint angles and link lengths for the third limb.

$$x = L_1 \cos \theta_{31} + L_2 \cos \theta_{32} \quad (7)$$

$$y = D - L - L_1 \sin \theta_{31} - L_2 \sin \theta_{32} \quad (8)$$

$$L_2^2 = (x - L_1 \cos \theta_{31})^2 + (y - D + L_1 \sin \theta_{31} + L)^2 \quad (9)$$

2.4 Linear actuation method

As described by Han Sung Kim and Lung-Wen Tsai (Kim & Tsai, 2002), for the linear actuation method, a linear actuator drives the prismatic joint in each limb whereas all the other joints are passive. This method has the advantage of having all actuators installed on the fixed base. The forward and inverse kinematic analyses are trivial since there exists a one-to-one correspondence between the end-effector position and the input joint displacements. Referring to figure 2, each limb constrains point P to lie on a plane which passes through point A_i and is perpendicular to the axis of the linear actuator. Consequently, the location of P is determined by the intersection of three planes. A simple kinematic relation can be written as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad (10)$$

3. Analysis of manipulator dynamics

An understanding of the manipulator dynamics is important from several different perspectives. First, it is necessary to properly size the actuators and other manipulator components. Without a model of the manipulator dynamics, it becomes difficult to predict the actuator force requirements and in turn equally difficult to properly select the actuators. Second, a dynamics model is useful for developing a control scheme. With an understanding of the manipulator dynamics, it is possible to design a controller with better performance characteristics than would typically be found using heuristic methods after the manipulator has been constructed. Moreover, some control schemes such as the computed torque controller rely directly on the dynamics model to predict the desired actuator force to be used in a feedforward manner. Third, a dynamical model can be used for computer simulation of a robotic system. By examining the behavior of the model under various operating conditions, it is possible to predict how a robotic system will behave when it is built. Various manufacturing automation tasks can be examined without the need of a real system. Several approaches have been used to characterize the dynamics of parallel manipulators. The most common approaches are based upon application of the Newton-Euler formulations, and Lagrange's equations of motion (Tsai, 1999). The traditional Newton-Euler formulation requires the equations of motion to be written once for each body of a manipulator, which inevitably leads to a large number of equations and results in poor computational efficiency. The Lagrangian formulation eliminates all of the unwanted reaction forces and moments at the outset. It is more efficient than the Newton-Euler formulation. However, because of the numerous constraints imposed by the closed loops of a manipulator, deriving explicit equations of motion in terms of a set of independent generalized coordinates becomes a prohibitive task. To simplify the problem, additional coordinates along with a set of Lagrangian multipliers are often introduced (Tsai, 1999).

3.1 Lagrange based dynamic analysis

It can be assumed that the first rod of each limb is a uniform and its mass is m_1 . The mass of second rod of each limb is evenly divided between and concentrated at joints M_i and B_i . This assumption can be made without significantly compromising the accuracy of the model since the concentrated mass model of the connecting rods does capture some of the dynamics of the rods. Also, the damping at the actuator is disregarded since the Lagrangian model does not readily accommodate viscous damping as is assumed for the actuators.

The Lagrangian equations are written in terms of a set of redundant coordinates. Therefore, the formulation requires a set of constraint equations derived from the kinematics of a mechanism. These constraint equations and their derivatives must be adjoined to the equations of motion to produce a number of equations that is equal to the number of unknowns. In general, the Lagrange multiplier approach involves solving the following system of equations (Tsai, 1999):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^k (\lambda_i \frac{\partial f_i}{\partial q_j}) \quad (11)$$

For $j=1$ to n , where

j : is the generalized coordinate index,

n : is the number of generalized coordinates,

i : is the constraint index,

q_j : is the j^{th} generalized coordinate,

k : is the number of constraint functions,

L : is the Lagrange function, where $L = T - V$,

T : is the total kinetic energy of the manipulator,

V : is the total potential energy of the manipulator,

f_i : is a constraint equation,

Q_j : is a generalized external force, and

λ_i : is the Lagrange multiplier.

Theoretically, the dynamic analysis can be accomplished by using just three generalized coordinates since this is a 3 DOF manipulator. However, this would lead to a cumbersome expression for the Lagrange function, due to the complex kinematics of the manipulator. So we choose three redundant coordinates which are θ_{11} , θ_{21} and θ_{31} beside the generalized coordinates x , y , and z . Thus we have θ_{11} , θ_{21} , θ_{31} , x , y , and z as the generalized coordinates. Equation 11 represents a system of six equations in six variables, where the six variables are λ_i for $i = 1, 2$, and 3 , and the three actuator forces, Q_j for $j = 4, 5$, and 6 . The external generalized forces, Q_j for $j=1, 2$, and 3 , are zero since the revolute joints are passive. This formulation requires three constraint equations, f_i for $i = 1, 2$, and 3 , that are written in terms of the generalized coordinates.

3.2 Derivation of the manipulator's dynamics

3.2.1 The kinetic and potential energy of the first limb

Referring to figure 6, the velocities of A_1 (the prismatic joint of the first limb), A_2 and A_3 are \dot{x} , \dot{y} and \dot{z} . The angular velocity of the rod $A_1 M_1$ is $\dot{\theta}_{11}$. We can consider the moment of

inertia of rods $A_1 M_1$, $A_2 M_2$, and $A_3 M_3$ is $I = \frac{m_1}{12} L_1^2$. m_3 is the mass of each prismatic joint (A_1 , A_2 , and A_3). So, the total kinetic energy of the first limb, T_1 , is

$$T_1 = [m_1 + m_2 + m_3] \frac{\dot{x}^2}{2} + \frac{m_2}{4} (\dot{y}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{11}^2 \tag{12}$$

The total potential energy of the first limb, V_1 , is

$$V_1 = -\frac{m_1 + m_2}{2} g L_1 \sin \theta_{11} - \frac{m_2}{2} g z \tag{13}$$

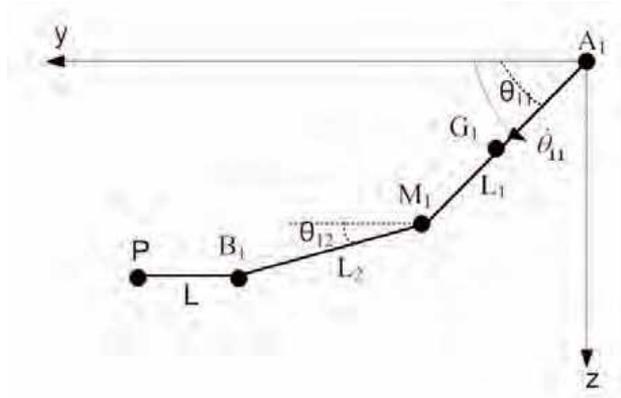


Figure 6: Schematic diagram of the first limb for the dynamic analysis.

3.2.2 The kinetic and potential energy of the second limb

Referring to figure 7, if the angular velocity of the rod $A_2 M_2$ is $\dot{\theta}_{21}$, the total kinetic energy of the second limb, T_2 is

$$T_2 = [m_1 + m_2 + m_3] \frac{\dot{y}^2}{2} + \frac{m_2}{4} (\dot{x}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{21}^2 \tag{14}$$

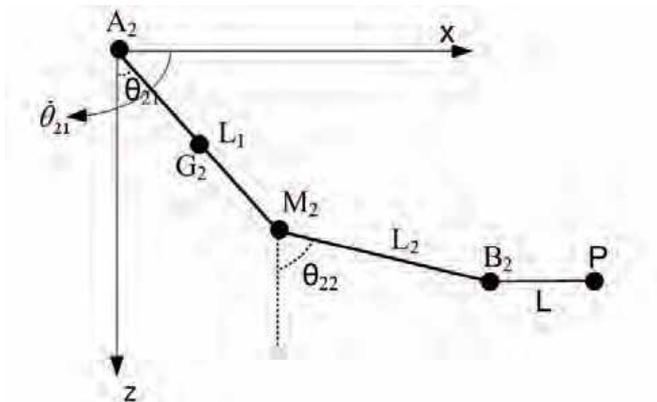


Figure 7: Schematic diagram of the second limb for the dynamic analysis.

The total potential energy of the second limb, V_2 , is given by

$$V_2 = -\frac{m_1 + m_2}{2} g L_1 \cos \theta_{21} - \frac{m_2}{2} g z \quad (15)$$

3.2.3 The kinetic and potential energy of the third limb

Referring to figure 8, the total kinetic energy of the second limb, T_3 is

$$T_3 = [m_1 + m_2 + m_3] \frac{\dot{z}^2}{2} + \frac{m_2}{4} (x^2 + y^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 \dot{\theta}_{31}^2 \quad (16)$$

The total potential energy of the third limb, V_3 , is

$$V_3 = -(m_1 + m_2 + m_3) g z \quad (17)$$

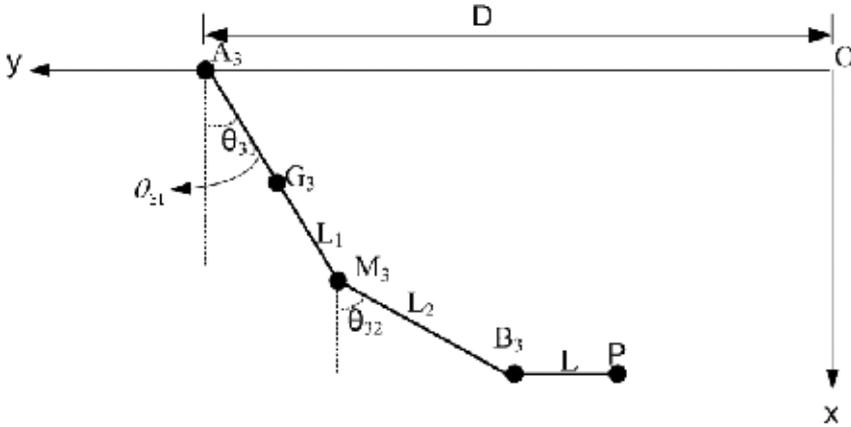


Figure 8: Schematic diagram of the third limb for the dynamic analysis.

3.2.4 Derivation of the Lagrange equation

From equations 12, 14, and 16, the total kinetic energy of the manipulator T is given by:

$$T = \frac{1}{2} [m_1 + 2m_2 + m_3 + m_4] (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \left(\frac{m_1}{6} + \frac{m_2}{4}\right) L_1^2 (\dot{\theta}_{11}^2 + \dot{\theta}_{21}^2 + \dot{\theta}_{31}^2) \quad (18)$$

where m_4 is the mass of the tool. From equations 13, 15, and 17, the total potential energy V of the manipulator is calculated relative to the plane of the stationary platform of the manipulator, and is found to be:

$$V = -\frac{m_1 + m_2}{2} g L_1 (\sin \theta_{11} + \cos \theta_{21}) - (m_1 + 2m_2 + m_3 + m_4) g z \quad (19)$$

The Lagrange function is defined as the difference between the total kinetic energy, T , and the total potential energy V : $L = T - V$

$$L = A(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + B(\dot{\theta}_{11}^2 + \dot{\theta}_{21}^2 + \dot{\theta}_{31}^2) + C(\sin \theta_{11} + \cos \theta_{21}) + Ez \quad (20)$$

Where:

$$A = \frac{1}{2}[m_1 + 2m_2 + m_3 + m_4], \quad B = \left(\frac{m_1}{6} + \frac{m_2}{4}\right)L_1^2, \quad C = \frac{m_1 + m_2}{2}gL_1 \quad \text{and} \quad E = (m_1 + 2m_2 + m_3 + m_4)g$$

3.2.5 The constraint equations

Differentiation of equation 3 with respect to time yields

$$0 = \frac{(y - L_1 \cos \theta_{11} - L)}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \dot{y} + \frac{(z - L_1 \sin \theta_{11})}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \dot{z} + \dot{\theta}_{11} \quad (27)$$

Differentiation of equation 6 with respect to time yields

$$0 = \frac{(z - L_1 \cos \theta_{21})}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \dot{z} + \frac{(x - L_1 \sin \theta_{21} - L)}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \dot{x} + \dot{\theta}_{21} \quad (28)$$

Differentiation of equation 9 with respect to time yields

$$0 = \frac{(x - L_1 \cos \theta_{31})}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} \dot{x} + \frac{(y - D + L_1 \sin \theta_{31} + L)}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} \dot{y} + \dot{\theta}_{31} \quad (29)$$

The equations 27, 28, and 29 are rearranged so as to produce:

$$\begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{21} \\ \dot{\theta}_{31} \end{bmatrix} = \Gamma \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (30)$$

Where

$$\Gamma = - \begin{bmatrix} 0 & \frac{(y - L_1 \cos \theta_{11} - L)}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} & \frac{(z - L_1 \sin \theta_{11})}{((y - L) \sin \theta_{11} - z \cos \theta_{11})L_1} \\ \frac{(x - L_1 \sin \theta_{21} - L)}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} & 0 & \frac{(z - L_1 \cos \theta_{21})}{L_1(z \sin \theta_{21} + (L - x) \cos \theta_{21})} \\ \frac{(x - L_1 \cos \theta_{31})}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} & \frac{(y - D + L_1 \sin \theta_{31} + L)}{(x \sin \theta_{31} + \cos \theta_{31}(y - D + L))L_1} & 0 \end{bmatrix}$$

3.2.6 Taking the derivatives of the Lagrange function with respect to θ_{11}

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{11}} \right) = 2B \ddot{\theta}_{11}, \quad \frac{\partial L}{\partial \theta_{11}} = C \cos \theta_{11}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{11}} \right) - \frac{\partial L}{\partial \theta_{11}} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{11}}) + Q_1$$

(Q_1, Q_2 and $Q_3 = 0$ since the revolute joints are passive)

$$2B \ddot{\theta}_{11} - C \cos \theta_{11} = \lambda_1 \quad (31)$$

3.2.7 Taking the derivatives of the Lagrange function with respect to θ_{21}

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{21}}\right) &= 2B\ddot{\theta}_{21}, \quad \frac{\partial L}{\partial \theta_{21}} = -C \sin \theta_{21} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{21}}\right) - \frac{\partial L}{\partial \theta_{21}} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{21}}) + Q_2 \\ 2B\ddot{\theta}_{21} + C \sin \theta_{21} &= \lambda_2\end{aligned}\quad (32)$$

3.2.8 Taking the derivatives of the Lagrange function with respect to θ_{31}

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{31}}\right) &= 2B\ddot{\theta}_{31}, \quad \frac{\partial L}{\partial \theta_{31}} = 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{31}}\right) - \frac{\partial L}{\partial \theta_{31}} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial \theta_{31}}) + Q_3 \\ 2B\ddot{\theta}_{31} &= \lambda_3\end{aligned}\quad (33)$$

Rearrangement of equations 31, 32, and 33 produces:

$$2B \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{21} \\ \ddot{\theta}_{31} \end{bmatrix} + C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}\quad (34)$$

Differentiation equation 30 with respect to time yields

$$\begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{21} \\ \ddot{\theta}_{31} \end{bmatrix} = \Gamma \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \frac{d\Gamma}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

Substituting into equation 34 yields

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = 2B\Gamma \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + 2B \frac{d\Gamma}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix}\quad (35)$$

3.2.9 Taking the derivatives of the Lagrange function with respect to X

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) &= 2A\ddot{x}, \quad \frac{\partial L}{\partial x} = 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} &= \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial x}) + Q_4\end{aligned}$$

$$2A\ddot{x} = F_x - \Gamma_{11}\lambda_1 - \Gamma_{21}\lambda_2 - \Gamma_{31}\lambda_3 \quad (36)$$

where F_x , F_y and F_z are the forces applied by the actuator for the first, second and third limbs. Γ_{ij} is the (i, j) element of the Γ matrix.

3.2.10 Taking the derivatives of the Lagrange function with respect to Y

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) = 2A\ddot{y}, \quad \frac{\partial L}{\partial y} = 0$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial L}{\partial y} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial y}) + Q_5$$

$$2A\ddot{y} = F_y - \Gamma_{12}\lambda_1 - \Gamma_{22}\lambda_2 - \Gamma_{32}\lambda_3 \quad (37)$$

3.2.11 Taking the derivatives of the Lagrange function with respect to Z

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}} \right) = 2A\ddot{z}, \quad \frac{\partial L}{\partial z} = E$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}} \right) - \frac{\partial L}{\partial z} = \sum_{i=1}^3 (\lambda_i \frac{\partial f_i}{\partial z}) + Q_6$$

$$2A\ddot{z} - E = F_z - \Gamma_{13}\lambda_1 - \Gamma_{23}\lambda_2 - \Gamma_{33}\lambda_3 \quad (38)$$

Rearrangement of equations 36, 37, and 38 produces:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = 2A \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$

Substituting into equation 35 yields

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T C \begin{bmatrix} -\cos\theta_{11} \\ \sin\theta_{21} \\ 0 \end{bmatrix} + (2AI + \Gamma^T 2B\Gamma) \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \Gamma^T 2B \frac{d\Gamma}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

The dynamic equation of the whole system can be written as

$$F = M(q)\ddot{q} + G(q, \dot{q})\dot{q} + K(q) \quad (39)$$

Where

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \quad \ddot{q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \quad \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad q = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad M(q) = 2AI + \Gamma^T 2B\Gamma, \quad G(q, \dot{q}) = \Gamma^T 2B \frac{d\Gamma}{dt}, \quad \text{and}$$

$$K(q) = - \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} + \Gamma^T C \begin{bmatrix} -\cos \theta_{11} \\ \sin \theta_{21} \\ 0 \end{bmatrix}$$

where q is the 3×1 vector of joint displacements, \dot{q} is the 3×1 vector of joint velocities, F is the 3×1 vector of applied force inputs, $M(q)$ is the manipulator inertia matrix, $G(q, \dot{q})$ is the manipulator centripetal and coriolis matrix which is the 3×3 matrix of centripetal and coriolis forces, and $K(q)$ is the vector of gravitational forces which is the 3×1 vector of gravitational forces due to gravity. The Lagrangian dynamics equation, equation 39, possess important properties that facilitates analysis and control system design. Among these are (Lewis et al., 1993): the $M(q)$ is a 3×3 symmetric and positive definite matrix and the matrix $W(q, \dot{q}) = \dot{M}(q) - 2G(q, \dot{q})$ is a skew symmetric matrix.

4. Controller design

4.1 Introduction

The control problem for robot manipulators is the problem of determining the time history of joint inputs required to cause the end-effector to execute a commanded motion. The joint inputs may be joint forces or torques depending on the model used for controller design.

Position control and trajectory tracking are the most common tasks for robot manipulators; given a desired trajectory, the joint force is chosen so that the manipulator follows that trajectory. The control strategy should be robust with respect to initial condition errors, sensor noise, and modeling errors. The primary goal of motion control in joint space is to make the robot joints q track a given time varying desired joint position q_d . Rigorously, we say that the motion control objective in joint space is achieved provided that $\lim_{t \rightarrow \infty} e(t) = 0$

where $e(t) = q_d(t) - q(t)$ denotes the joint position error. Although the dynamics of the manipulator's equation is complicated, it nevertheless is an idealization, and there are a number of dynamic effects that are not included in this equation. For example, friction at the joints is not accounted for in this equation and may be significant for some manipulators. Also, no physical body is completely rigid. A more detailed analysis of robot dynamics would include various sources of flexibility, such as elastic deformation of bearings and gears, deflection of the links under load, and vibrations.

4.2 PID control versus model based control

The PID controller is a single-input/single-output (SISO) controller that produces a control signal that is a sum of three terms. The first term is proportional (P) to the positioning error, the second term is proportional to the integral (I) of the error, and the third is proportional to the derivative (D) of the error. The PID (or PD) type is usually employed in industrial robot manipulators because it is easy to implement and requires little computation time during real time operation. This approach views each actuator of the manipulator independently, and essentially ignores the highly coupled and nonlinear nature of the manipulator. The error between the actual and desired joint position is used as feedback to control the actuator associated with each joint. However, independent joint controllers can not achieve a satisfactory performance due to their inherent low rejection of disturbances and parameter variations. Because of such limitation, model based control algorithms were proposed (Sciavicco et al., 1990) that have the potential to perform better than independent

joint controllers that do not account for manipulator dynamics. However, the difficulty with the model based controller is that it requires a good model of the manipulator inverse dynamics and good estimates of the model parameters, making this controller more complex and difficult to implement than the non-model based controller. The model based control scheme was intensively experimentally tested for example the experimental evaluation of computed torque control was presented in (Griffiths et al., 1989).

4.3 PD control with position and velocity reference

The first PD control law is based on the position and velocity error of each actuator in the joint space. To implement the joint control architecture, the values for the joint position error and the joint rate error of the closed chain system are used to compute the joint control force F (Spong & Vidyasagar, 1989).

$$F = K_p e + K_D \dot{e} \quad (40)$$

Where $e = q_d - q$, which is the vector of position error of the individual actuated joints, and $\dot{e} = \dot{q}_d - \dot{q}$, which is the vector of velocity error of the individual actuated joints. Where \dot{q}_d and q_d are the desired joint velocities and positions, and K_D and K_P are 3×3 diagonal matrices of velocity and position gains. Although this type of controller is suitable for real time control since it has very few computations compared to the complicated nonlinear dynamic equations, there are a few downsides to this controller. It needs high update rate to achieve reasonable accuracy. Using local PD feedback law at each joint independently does not consider the couplings of dynamics between robot links. As a result, this controller can cause the motor to overwork compared to other controllers presented next.

4.4 PD Control with gravity compensation

This is a slightly more sophisticated version of PD control with a gravitational feedforward term. Consider the case when a constant equilibrium posture is assigned for the system as the reference input vector q_d . It is desired to find the structure of the controller which ensures global asymptotic stability of the above posture. The control law F is given by (Spong & Vidyasagar, 1989):

$$F = K_p e + K_D \dot{e} + K(q_d) \quad (41)$$

It has been shown (Spong, 1996) that the system is asymptotically stable but it is only proven with constant reference trajectories. Although with varying desired trajectories, this type of controller cannot guarantee perfect tracking performance. Hence, more dynamic modeling information is needed to incorporate into the controller.

4.5 PD control with full dynamics feedforward terms

This type of controller augments the basic PD controller by compensating for the manipulator dynamics in the feedforward way. It assumes the full knowledge of the robot parameters. The key idea for this type of controller is that if the full dynamics is correct, the resulting force generated by the controller will also be perfect. The controller is given by (Gullayanon, 2005)

$$F = M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d) + K_p e + K_D \dot{e} \quad (42)$$

If the dynamic knowledge of the manipulator is accurate, and the position and velocity error terms are initially zero, the applied force $F = M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d)$

is sufficient to maintain zero tracking error during motion. This controller is very similar to the computed torque controller, which is presented next. The difference between these two controllers is the location of the position and velocity correction terms. This controller is less sensitive to any mass changes in the system. For example, if the robot picks up a heavy load in the middle of its operation, this controller is likely to respond to this change slower compared to computed torque controller. The advantage of this controller is that once the desired trajectory for a given task has been specified, then the feedforward terms relying on the robot dynamics $M(q_d)\ddot{q}_d + G(q_d, \dot{q}_d)\dot{q}_d + K(q_d)$ can be computed offline to reduce the computational burden.

4.6 Computed torque control

This controller is developed for the manipulator to examine if it is possible to improve the performance of the trajectory tracking of the manipulator by utilizing a more complete understanding of the manipulator dynamics in the controller design. This controller employs a computed torque control approach, and it uses a model of the manipulator dynamics to estimate the actuator forces that will result in the desired trajectory. Since this type of controller takes into account the nonlinear and coupled nature of the manipulator, the potential performance of this type of controller should be quite good. The disadvantage of this approach is that it requires a reasonably accurate and computationally efficient model of the inverse dynamics of the manipulator to function as a real time controller. The controller computes the dynamics online, using the sampled joint position and velocity data. The key idea is to find an input vector, F , as a function of the system states, which is capable to realize an input/output relationship of linear type. It is desired to perform not a local linearization but a global linearization of system dynamics obtained by means of a nonlinear state feedback. Using the computed torque approach with a proportional-derivative (PD) outer control loop, the applied actuator forces are calculated at each time step using the following force law as described by Lewis, 1993:

$$F = M(q)[\ddot{q}_d + K_D\dot{e} + K_P e] + G(q, \dot{q})\dot{q} + K(q) \quad (43)$$

where F is the force applied to input links, K_D is the diagonal matrix of the derivative gains, K_P is the diagonal matrix of the proportional gains, and e is the vector of the position errors of the input links, $e = q_d - q$. To show that the computed torque control scheme linearizes the controlled system, the force computed by equation 43 is substituted into equation 39, yielding: $M(q)\ddot{q} = M(q)\ddot{q}_d + M(q)[K_D\dot{e} + K_P e]$. Then multiplying each term by $M^{-1}(q)$, and substituting the relationship, $\ddot{e} = \ddot{q}_d - \ddot{q}$, provides the following linear relationship for the error:

$$\ddot{e} + k_D\dot{e} + k_P e = 0 \quad (44)$$

This relationship can be used to select the gains to give the desired nature of the closed loop error response since the solution of equation 44 provides a second order damped system with a natural frequency of ω_n , and a damping ratio of ζ where:

$$\omega_n = \sqrt{K_p}, \quad \zeta = \frac{K_D}{2\sqrt{K_p}} \quad (45)$$

Since the equation 44 is linear, it is easy to choose K_D and K_p so that the overall system is stable and $e \rightarrow 0$ exponentially as $t \rightarrow \infty$. Usually, if K_D and K_p are positive diagonal matrices, the control law 43 applied to the system 39 results in exponentially trajectory tracking.

It is customary in robot applications to take the damping ration $\zeta = 1$ so that the response is critically damped. This produces the fastest non-oscillatory response. The natural frequency ω_n determines the speed of the response. So, the values for the gain matrices K_D and K_p are determined by setting the gains to maintain the following relationship:

$$K_D = 2\sqrt{K_p} \quad (46)$$

If the error response is critically damped. Hence, the general solution of equation 44 is:

$$e(t) = (c_1 + c_2 t) e^{-\frac{K_D t}{2}} \quad (47)$$

where C_1 and C_2 are constants.

5. Trajectory planning and simulation

5.1 Introduction

The computer simulation is the first step to verify the performance of the controllers because it is an ideal way of comparing performance of various motion controllers. Although computer simulation has much fewer disturbances compared to real experiments, factors such as the integration estimation and sampling rate can cause the controllers to behave differently than the mathematical prediction.

5.2 Tracking accuracy

In this research, the main purpose for developing the motion controllers is to obtain a good trajectory tracking capability. The performance of each control method is evaluated by comparing the tracking accuracy of the end-effector. The tracking accuracy is evaluated by the *Root Square Mean Error* (RSME). The end-effector error is defined as

$$E_{xyz} = \sqrt{(e_x^2 + e_y^2 + e_z^2)} \quad (48)$$

where e_x , e_y and e_z are the position errors in x-, y-, and z-axis given in manipulator's workspace coordinates.

$$RSME = \sqrt{\frac{\sum E_{xyz}^2}{n}} \quad (49)$$

Where n is the number of the samples.

5.3 Trajectory planning

In controlling the manipulator using any types of joint space controllers, any sudden changes in desired joint angle, velocity, or acceleration can result in sudden changes of the

commanded force. This can result in damages of the motors and the manipulator. Here, the manipulator is given a task to move along careful preplanned trajectories without any external disturbances or no interaction with environment. The desired trajectory is simulated using all motion controllers presented in Section 4 and the tracking accuracy *RSME* is obtained to be compared. The simulation is used to find a set of minimum proportional gain K_P and derivative gain K_D that minimized *RSME*. It must be considered that the actuators can not generate forces larger than 120 Newtons. The values of the physical kinematic and dynamic parameters of the CPM are given in table 1 and table 2.

Parameters	$L(m)$	$L_1(m)$	$L_2(m)$	$D(m)$
Values	0.105	0.5	0.373	0.9144

Table 1: Kinematic parameters of the CPM

Parameters	$m_1(kg)$	$m_2(kg)$	$m_3(kg)$	$m_4(kg)$
Values	1.892994	0.695528	0.2	0.3

Table 2: Dynamic parameters of the CPM.

The sample trajectory of the end-effector is chosen to be a circular path with the radius of 0.175 meters and its center is $O(0.425, 0.425, 0.3)$. This path is designed to be completed in 4 seconds when the end-effector reaches the starting point $P1(0.6, 0.425, 0.3)$ again with constant angular velocity $\omega = 0.5\pi$ rad/sec. The end-effector path is shown in figure 9. The desired end-effector position along x-axis is $x = 0.425 + 0.175\cos(\omega t)$ meters, along y-axis is $y = 0.425 + 0.175\sin(\omega t)$ meters and along z-axis is $Z=0.3$ meters. The desired force obtained from the actuators to move the end-effector along the desired trajectory is shown in figure 10.

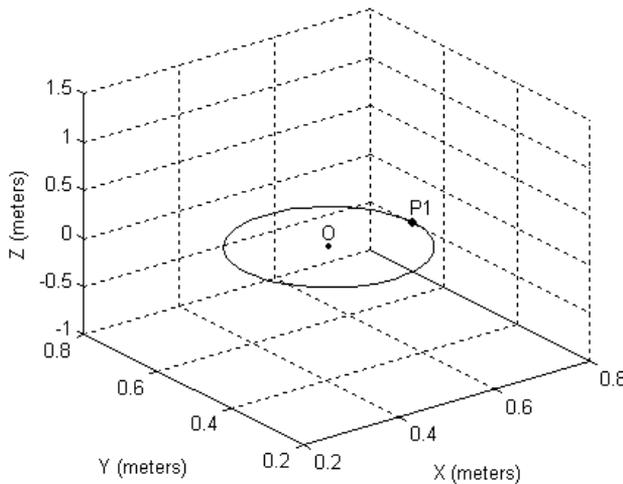


Figure 9: End-effector path for the circular trajectory.

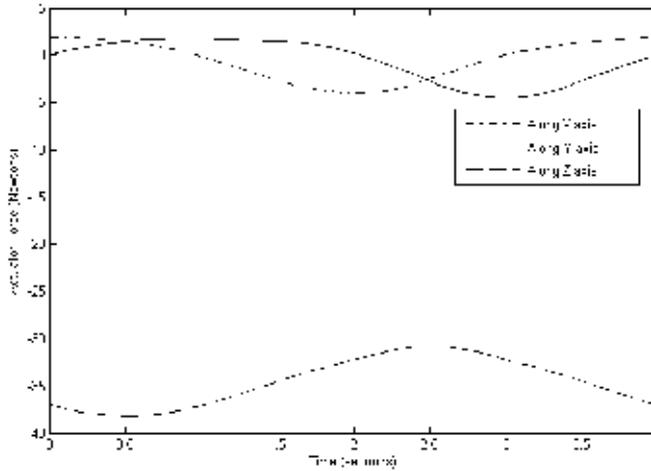


Figure 10: The desired force obtained from the actuators

5.4 Simulation results

To investigate each controller's performance, computer simulation, carried out in Matlab, is used in this thesis. The robot dynamic model, developed earlier, is constructed. The sample trajectory, presented in the previous section, is generated and stored offline. The environmental disturbances are ignored and full knowledge of the manipulator dynamics can be assumed. Hence, the optimal performance of each controller can be obtained and compared. The simulation results are presented in table 3.

5.4.1 PD control with position and velocity reference

It was required that the robot achieved the desired trajectory with a position error less than 3×10^{-3} m after 0.3 seconds.

5.4.2 PD control with gravity compensation

It was required that the robot achieved the desired trajectory with a position error less than 3×10^{-4} m after 0.3 seconds.

5.4.3 PD control with full dynamics Feedforward terms

It was required that the robot achieved the desired trajectory with a position error less than 10^{-5} m after 0.3 seconds.

5.4.4 Computed torque control

The initial conditions of the error and its derivative of our sample trajectory of the End-effector, $e(0) = 0$, and $\dot{e}(0) = \dot{e}_0$, are used to find c_1 and c_2 in equation 47. Then, the solution of this equation is

$$e = \dot{e}_0 t e^{-\frac{K_D t}{2}} \quad (50)$$

Equation 50 suggests that the derivative gain K_D should be a maximum value to achieve the desired critical damping but the actuator force cannot exceed more than 120 Newtons. Using this criterion, the simulation results are presented in table 3. The position and velocity errors of the end-effector obtained from the controllers and the actuator forces required by these controllers are shown in figures 11 to 22.

Controller	K_P	K_D	Position RSME	Velocity RSME
Pd Control with Position and Velocity Reference	12691	436	0.0027	0.0223
Pd Control with Gravity Compensation	8507	436	3.4804×10^{-4}	0.021
Pd Control with Full Dynamics Feedforward	7053	436	3.0256×10^{-4}	0.0182
Computed Torque Control	2550.25	101	2.3469×10^{-4}	0.0161

Table 3: The performance of various controllers

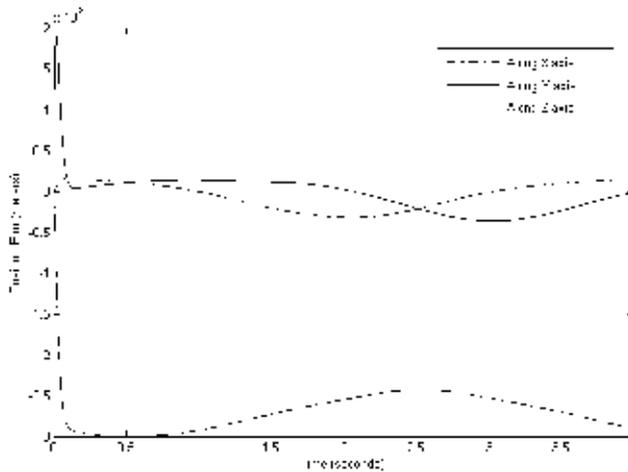


Figure 11 Position error of the end-effector obtained from the Simple PD Controller.

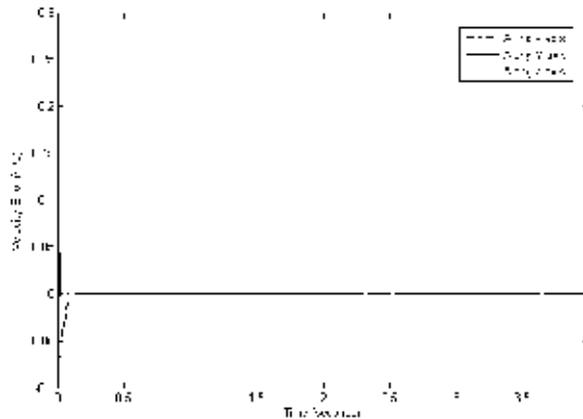


Figure 12: Velocity error of the end-effector obtained from the Simple PD Controller.

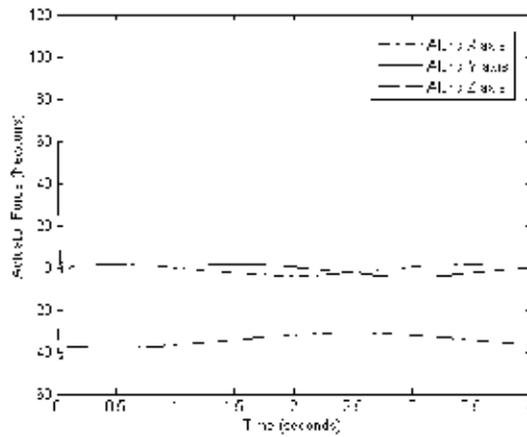


Figure 13: The actuator force required by the Simple PD Controller.

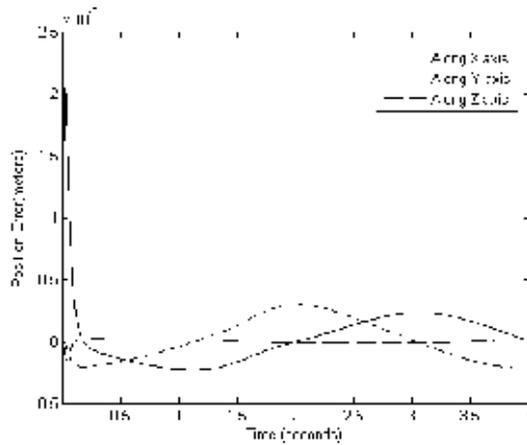


Figure 14: Position error of the end-effector obtained from the second PD Controller.

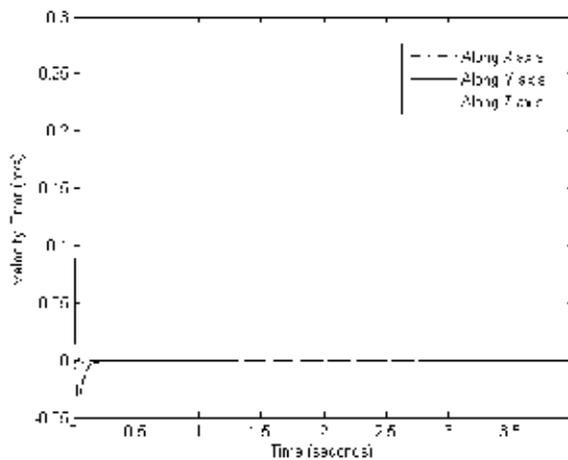


Figure 15: Velocity error of the end-effector obtained from the second PD Controller.

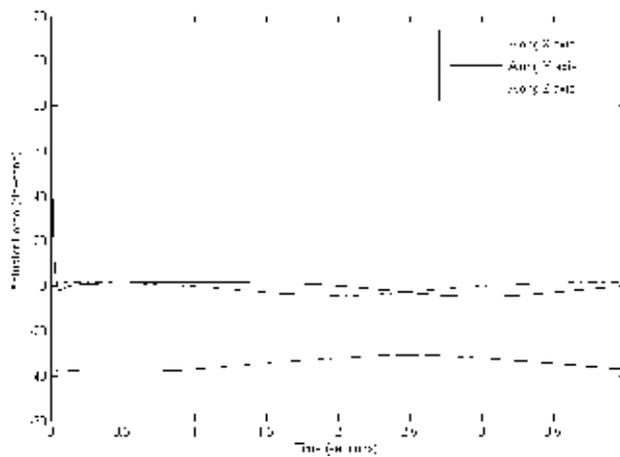


Figure 16: The actuator force required by the second PD Controller.

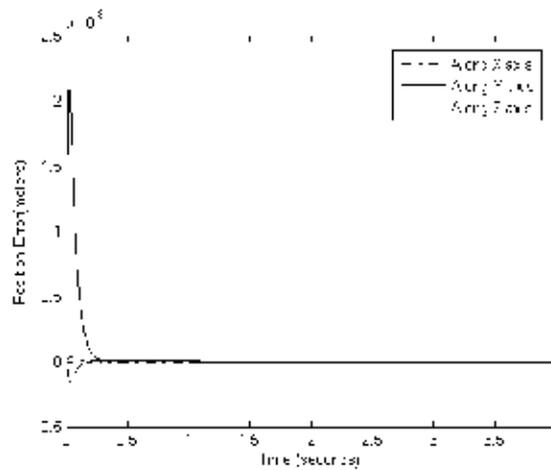


Figure 17: Position error of the end-effector obtained from the third PD Controller.

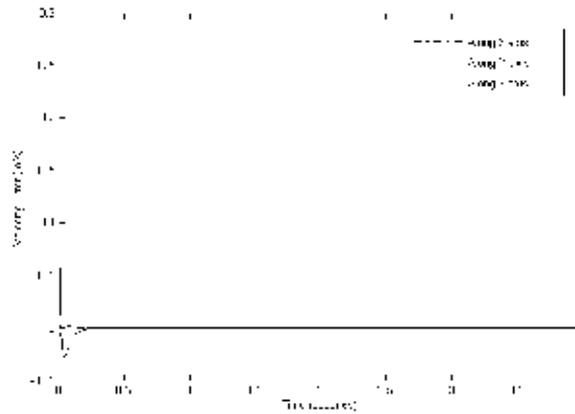


Figure 18: Velocity error of the end-effector obtained from the third PD Controller.

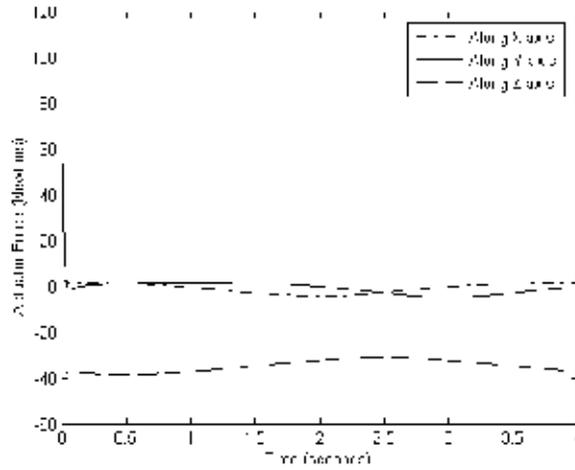


Figure 19: The actuator force required by the third PD Controller.

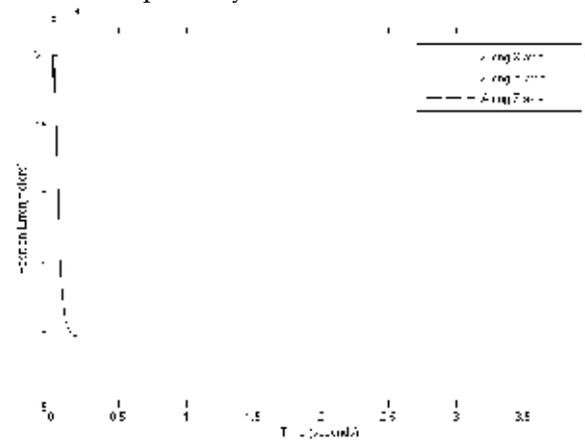


Figure 20: Position error of the end-effector obtained from the computed torque controller.

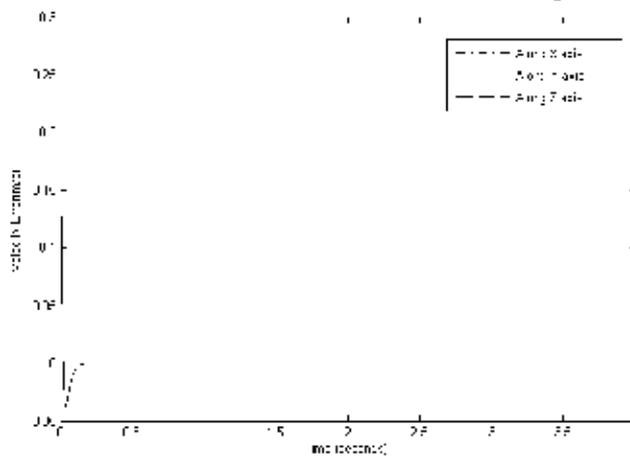


Figure 21: Velocity error of the end-effector obtained from the computed torque controller.

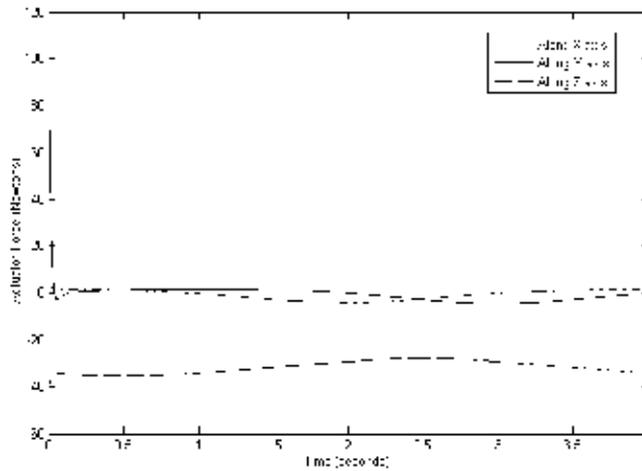


Figure 22: The actuator force required by the computed torque controller.

6. Conclusions

The research presented in this chapter establishes the CPM as a viable robotic device for three degrees of freedom manipulation. The manipulator offers the advantages associated with other parallel manipulators, such as light weight construction; while avoiding some of the traditional disadvantages of parallel manipulators such as the extensive use of spherical joints and coupling of the platform orientation and position. The CPM employs only revolute and prismatic joints to achieve translational motion of the moving platform. The main advantages of this parallel manipulator are that all of the actuators can be attached directly to the base, closed-form solutions are available for the forward and inverse kinematics, and the moving platform maintains the same orientation throughout the entire workspace. From simulations done in this research, performance of various motion controllers are studied and compared. Although the simple PD controller with only position and velocity reference is easy to implement and no knowledge of the system is needed to develop this type of controller, the tracking ability is very poor compared to the rest of the controllers used in this thesis. At the next step, when partial dynamic modeling information is incorporated into the controller, the PD controller with gravity compensation is implemented. The simulation results show a significant improvement in tracking ability from a simple PD controller. Next, the verification is needed to determine if complete mathematical modeling knowledge is needed to give the controller complete advantage in motion control. Hence, the PD controller with full dynamic feedforward terms and computed torque controller are implemented and put to the test. The model based controllers such as computed torque and PD control with full dynamic feedforward terms can generate force commands more intelligently and accurately than simple non-model based controllers. Hence, the need for studying dynamics of robot manipulator as well as having a good understanding of various basic motion controller theories are important in designing and controlling motion of the robot to achieve the highest quality and quantity of work. The simulation results show that the computed torque controller gives the best

performance. This is a result of the computed torques canceling the nonlinear components of the controlled system. From the observations seen in this work, one can see the motivation for engineers to develop more advanced controllers that not only know the dynamic model of the manipulator, but can also detect if the dynamic is changed and can tune itself accordingly (i.e. adaptive control).

7. Future work

1. The effect of some unknown parameters such as the friction and the nonlinear factors introduced by the motors and the gear boxes which may be obtained by experimental measurements and through the identification methods can be studied.
2. The performance of model based control relies on an accurate model of a system. However, identifying the accurate dynamic model of a system is very difficult. Therefore, effective controllers for the versatile application of parallel robots should be developed. Adaptive control has the potential to improve the tracking accuracy because it updates the unknown parameters online. Adaptive control algorithm is too complicated to be utilized in high speed applications. In such applications, robust independent joint control is a prospective method to improve the performance of simple PD control.
3. *Adaptive Neuro Fuzzy Inference System (ANFIS)* controller can be used for each active joint to generate the required control system, then its performance is compared with the conventional controllers. Although many of model based methods have been found and they provide satisfactory solutions, these solutions have been subordinated to the development of the mathematical theories that deal with over idealized problems bearing little relation to practice.

8. Acknowledgment

The authors would like to thank Prof. Han Sung Kim for his valuable suggestions and his kind assistance during this work.

9. References

- Carricato, M., and Parenti-Castelli, V., (2001), "A Family of 3-DOF Translational Parallel Manipulators", Proceedings of the 2001 ASME Design Engineering Technical Conferences, Pittsburgh, PA, DAC-21035.
- Ceccarelli, M., (1997), "A New 3 D.O.F. Spatial Parallel Mechanism", Mechanism and Machine Theory, Vol. 32, No. 8, pp. 895-902.
- Clavel, R., (1988), "Delta, A Fast Robot with Parallel Geometry", Proceedings of the 18th International Symposium on Industrial Robots, pp. 91-100.
- Di Gregorio, R., (2001), "A New Parallel Wrist Using only Revolute Pairs: The 3 RUU Wrist", Robotica, Vol. 19, No. 3, pp. 305-9.
- Di Gregorio, R. and Parenti-Castelli, V., (1998), "A Translational 3-DOF Parallel Manipulator", in Advances in Robot Kinematics, Edited by J. Lenarcic and M. L. Husty, Kluwer Academic Publishers, London, pp. 49-58.

- Fang, Y. and Tsai, L. W., 2002, "Enumeration of 3-DOF Translational Parallel Manipulators Using the Theory of Reciprocal Screws", accepted for publication in ASME Journal of Mechanical Design.
- Gosselin, C. and Angeles, J., 1989, "The Optimum Kinematic Design of a Spherical Three-Degree-of-Freedom Parallel Manipulator", ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 111, No. 2, pp. 202-7.
- Griffiths, J.D., An. C.H., Atkeson, C.G. and Hollerbach, J.M., 1989, "Experimental evaluation of feedback and computed torque control", International Journal of Robotics and Automation, 5(3):368-373, June.
- Gullayanon R., 2005, "Motion Control of 3 Degree-of-Freedom Direct-Drive Robot.", A master thesis presented to the School of Electrical and Computer Engineering, Georgia Institute of Technology.
- Karouia, M., and Herve, J. M., 2000, "A Three-DOF Tripod for Generating Spherical Rotation", in Advances in Robot Kinematics, Edited by J. Lenarcic and V. Parenti-Castelli, Kluwer Academic Publishers, pp. 395-402.
- Kim H.S., and Tsai L.W., 2002, "Design optimization of a Cartesian parallel manipulator", Department of Mechanical Engineering, Bourns College of Engineering, University of California.
- Lewis, F., Abdallah, C. and Dawson, D., 1993, "Control of Robot Manipulators", MacMillan Publishing Company.
- Pierrot, F., Reynaud, C. and Fournier, A., 1990, "Delta: A Simple and Efficient Parallel Robot", Robotica, Vol. 6, pp. 105-109.
- Sciavicco, L., Chiacchio, P. and Siciliano, B., 1990, "The potential of model-based control algorithms for improving industrial robot tracking performance", IEEE International Workshop on Intelligent Motion Control, pp. 831-836, August.
- Spong, M. W., 1996, "Motion Control of Robot Manipulators", University of Illinois at Urbana-Champaign.
- Spong, M.W. and Vidyasagar, M., 1989, "Robot dynamics and control", John Wiley & Sons.
- Stewart, D., 1965, "A Platform with Six Degrees of Freedom", Proceedings Institute of Mechanical Engineering, Vol. 180, pp. 371-386.
- Tsai, L. W., and Joshi, S., 2002, "Kinematic Analysis of 3-DOF Position Mechanism for Use in Hybrid Kinematic Machines", ASME Journal of Mechanical Design, Vol. 124, No. 2, pp. 245-253.
- Tsai, L. W., 1999, "Robot Analysis: the mechanics of serial and parallel manipulators", John Wiley & Sons.
- Tsai, L. W., 1996, "Kinematics of a Three-DOF Platform Manipulator with Three Extensible Limbs", in Advances in Robot Kinematics, Edited by J. Lenarcic and V. Parenti-Castelli, Kluwer Academic Publishers, pp. 401-410.
- Tsai, L. W., Walsh, G. C. and Stamper, R., 1996, "Kinematics of a Novel Three DOF Translational Platform", IEEE International Conference on Robotics and Automation, Minneapolis, MN, pp. 3446-3451.

Vischer, P. and Clavel, R., 2000, "Argos: a Novel 3-DOF Parallel Wrist Mechanism", The International Journal of Robotics Research, Vol. 19, No. 1, pp. 5-11.

Optimal Design of Parallel Kinematics Machines with 2 Degrees of Freedom

Sergiu-Dan Stan, Vistrian Mătieș and Radu Bălan
*Technical University of Cluj-Napoca
Romania*

1. Introduction

The mechanical structure of today's machine tools is based on serial kinematics in the overwhelming majority of cases. Parallel kinematics with closed kinematics chains offer many potential benefits for machine tools but they also cause many drawbacks in the design process and higher efforts for numerical control and calibration.

The Parallel Kinematics Machine (PKM) is a new type of machine tool which was firstly showed at the 1994 International Manufacturing Technology in Chicago by two American machine tool companies, Giddings & Lewis and Ingersoll.

Parallel Kinematics Machines seem capable of answering the increase needs of industry in terms of automation. The nature of their architecture tends to reduce absolute positioning and orienting errors (Stan et al., 2006). Their closed kinematics structure allows them obtaining high structural stiffness and performing high-speed motions. The inertia of its mobile parts is reduced, since the actuators of a parallel robot are often fixed to its base and the end-effector can perform movements with higher accelerations. One drawback with respect to open-chain manipulators, though, is a typically reduced workspace and a poor ratio of working envelope to robot size.

In theory, parallel kinematics offer for example higher stiffness and at the same time higher acceleration performance than serial structures. In reality, these and other properties are highly dependent on the chosen structure, the chosen configuration for a structure and the position of the tool centre point (TCP) within the workspace. There is a strong and complex link between the type of robot's geometrical parameters and its performance. It's very difficult to choose the geometrical parameters intuitively in such a way as to optimize the performance. The configuration of parallel kinematics is more complex due to the high sensitivity to variations of design parameters. For this reason the design process is of key importance to the overall performance of a Parallel Kinematics Machines. For the optimization of Parallel Kinematics Machines an application-oriented approach is necessary. In this chapter an approach is presented that includes the definition of specific objective functions as well as an optimization algorithm. The presented algorithm provides the basis for an overall multiobjective optimization of several kinematics structures.

An important objective of this chapter is also to propose an optimization method for planar Parallel Kinematics Machines that combines performance evaluation criteria related to the following robot characteristics: workspace, design space and transmission quality index.

Furthermore, a genetic algorithm is proposed as the principle optimization tool. The success of this type of algorithm for parallel robots optimization has been demonstrated in various papers (Stan et al., 2006).

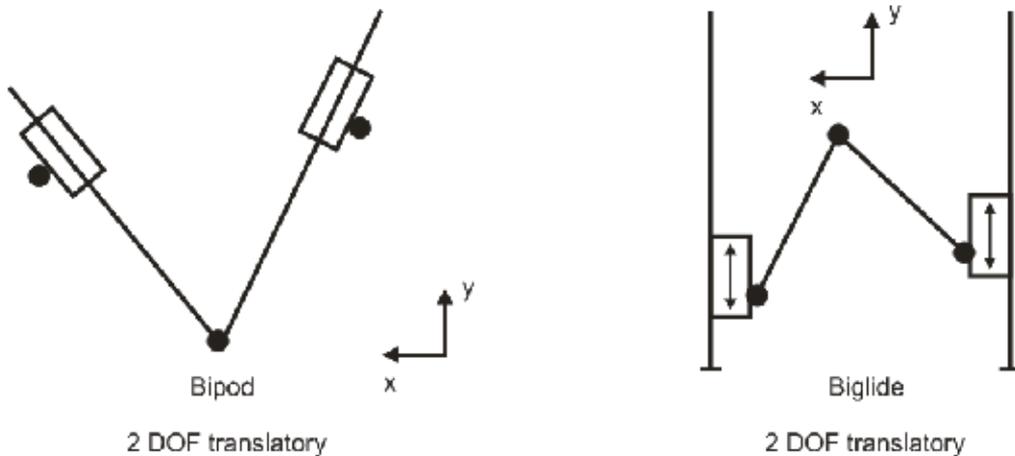


Fig. 1. Parallel kinematics for milling machines

For parallel kinematics machines with reduced number of degrees of freedom kinematics and singularity analyses can be solved to obtain algebraic expressions, which are well suited for an implementation in optimum design problems.

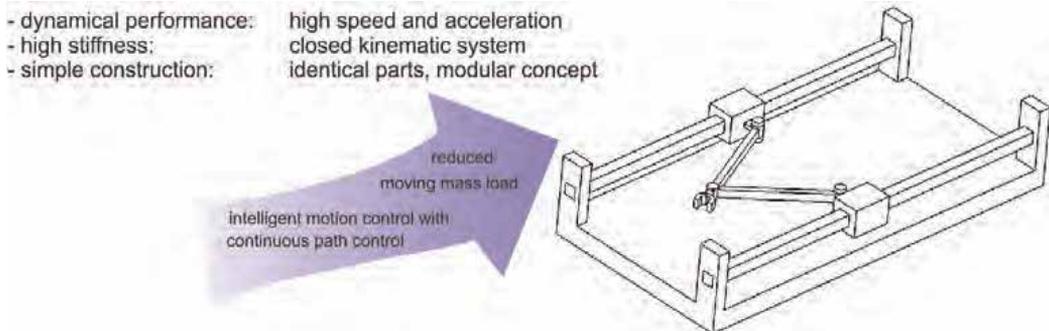


Fig. 2. Benefits of Parallel Kinematics Machines

High dynamical performance is achieved due to the low moved masses. Due to the closed kinematics the movements of parallel kinematics machines are vibration free for which the accuracy is improved. Finally, the modular concept allows a cost-effective production of the mechanical parts.

In this chapter, the optimization workspace index is defined as the measure to evaluate the performance of two degree of freedom Parallel Kinematics Machines. Another important contribution is the optimal dimensioning of the two degree-of-freedom Parallel Kinematics Machines of type Bipod and Biglide for the largest workspace using optimization based on Genetic Algorithms.

2. Objective functions used for optimization of machine tools with parallel kinematics

One of the main influential factors on the performance of a machine tool with parallel kinematics is its structural configuration. The performance of a machine tool with parallel kinematics can be evaluated by its kinematic, static and dynamic properties. Optimal design is one of the most important issues in the development of a parallel machine tool. Two issues are involved in the optimal design: performance evaluation and dimensional synthesis. The latter one is one of the most difficult issues in this field. In the optimum design process, several criteria could be involved for a design purpose, such as workspace, singularity, dexterity, accuracy, stiffness, and conditioning index.

After its choice, the next step on the machine tool with parallel kinematics design should be to establish its dimensions. Usually this dimensioning task involves the choice of a set of parameters that define the mechanical structure of the machine tool. The parameter values should be chosen in a way to optimize some performance criteria, dependent upon the foreseen application.

The optimization of machine tools with parallel kinematics can be based on the following objectives functions:

- workspace,
- the overall size of the machine tool,
- kinematic transmission of forces and velocities,
- stiffness,
- acceleration capabilities,
- dexterity,
- accuracy,
- the singular configurations,
- isotropy.

In the design process we want to determine the design parameters so that the parallel kinematics machine fulfills a set of constraints. These constraints may be extremely different but we can mention:

- workspace requirement,
- maximum accuracy over the workspace for a given accuracy of the sensors,
- maximal stiffness of the Parallel Kinematics Machines in some direction,
- minimum articular forces for a given load,
- maximum velocities or accelerations for given actuator velocities and accelerations.

Determination of the architecture and size of a mechanism is an important issue in the mechanism design. Several objectives are contradictory to each other. An optimization with only one objective runs into unusable solutions for all other objectives. Unfortunately, any change that improves one performance will usually deteriorate the other. This trade-off occurs with almost every design and this inevitable generates the problem of design optimization. Only a multiobjective approach will result in practical solutions for machine tool applications.

The classical methods of design optimization, such as iterative methods, suffer from difficulties in dealing with this problem. Firstly, optimization problems can take many iterations to converge and can be sensitive to numerical problems such as truncation and round-off error in the calculation. Secondly, most optimization problems depend on initial

guesses, and identification of the global minimum is not guaranteed. Therefore, the relation between the design parameters and objective function is difficult to know, thus making it hard to obtain the most optimal design parameters of the mechanism. Also, it's rather difficult to investigate the relations between performance criteria and link lengths of all mechanisms. So, it's important to develop a useful optimization approach that can express the relations between performance criteria and link lengths.

2.1 Workspace

The workspace of a robot is defined as the set of all end-effector configurations which can be reached by some choice of joint coordinates. As the reachable locations of an end-effector are dependent on its orientation, a complete representation of the workspace should be embedded in a 6-dimensional workspace for which there is no possible graphical illustration; only subsets of the workspace may therefore be represented.

There are different types of workspaces namely constant orientation workspace, maximal workspace or reachable workspace, inclusive orientation workspace, total orientation workspace, and dextrous workspace. The constant orientation workspace is the set of locations of the moving platform that may be reached when the orientation is fixed. The maximal workspace or reachable workspace is defined as the set of locations of the end-effector that may be reached with at least one orientation of the platform. The inclusive orientation workspace is the set of locations that may be reached with at least one orientation among a set defined by ranges on the orientation parameters. The set of locations of the end-effector that may be reached with all the orientations among a set defined by ranges on the orientations on the orientation parameters constitute the total orientation workspace. The dextrous workspace is defined as the set of locations for which all orientations are possible. The dextrous workspace is a special case of the total orientation workspace, the ranges for the rotation angles (the three angles that define the orientation of the end-effector) being $[0, 2\pi]$.

In the literature, various methods to determine workspace of a parallel robot have been proposed using geometric or numerical approaches. Early investigations of robot workspace were reported by (Gosselin, 1990), (Merlet, 1995), (Kumar & Waldron, 1981), (Tsai and Soni, 1981), (Gupta & Roth, 1982), (Sugimoto & Duffy, 1982), (Gupta, 1986), and (Davidson & Hunt, 1987). The consideration of joint limits in the study of the robot workspaces was presented by (Delmas & Bidard, 1995). Other works that have dealt with robot workspace are reported by (Agrawal, 1990), (Gosselin & Angeles, 1990), (Cecarelli, 1995). (Agrawal, 1991) determined the workspace of in-parallel manipulator system using a different concept namely, when a point is at its workspace boundary, it does not have a velocity component along the outward normal to the boundary. Configurations are determined in which the velocity of the end-effector satisfies this property. (Pernkopf & Husty, 2005) presented an algorithm to compute the reachable workspace of a spatial Stewart Gough-Platform with planar base and platform (SGPP) taking into account active and passive joint limits. Stan (Stan, 2003) presented a genetic algorithm approach for multi-criteria optimization of PKM (Parallel Kinematics Machines). Most of the numerical methods to determine workspace of parallel manipulators rest on the discretization of the pose parameters in order to determine the workspace boundary (Cleary & Arai, 1991), (Ferraresi et al., 1995). In the discretization approach, the workspace is covered by a regularly arranged grid in either Cartesian or polar form of nodes. Each node is then examined to see whether it belongs to the workspace. The accuracy of the boundary depends upon the sampling step that is used to create the grid.

The computation time grows exponentially with the sampling step. Hence it puts a limit on the accuracy. Moreover, problems may occur when the workspace possesses singular configurations. Other authors proposed to determine the workspace by using optimization methods (Stan, 2003). Numerical methods for determining the workspace of the parallel robots have been developed in the recent years. Exact computation of the workspace and its boundary is of significant importance because of its impact on robot design, robot placement in an environment, and robot dexterity.

Masory, who used the discretisation method (Masory & Wang, 1995), presented interesting results for the Stewart-Gough type parallel manipulator:

- The mechanical limits on the passive joints play an important role on the volume of the workspace. For ball and socket joints with given rotation ability, the volume of the workspace is maximal if the main axes of the joints have the same directions as the links when the robot is in its nominal position.
- The workspace volume is roughly proportional to the cube of the stroke of the actuators.
- The workspace volume is not very sensitive to the layout of the joints on the platforms, even though it is maximal when the two platforms have the same dimension (in this case, the robot is in a singular configuration in its nominal position).

Even though powerful three-dimensional Computer Aided Design and Dynamic Analysis software packages such as Pro/ENGINEER, IDEAS, ADAMS and Working Model 3-D are now being used, they cannot provide important visual and realistic workspace information for the proposed design of a parallel robot. In addition, there is a great need for developing methodologies and techniques that will allow fast determination of workspace of a parallel robot. A general numerical evaluation of the workspace can be deduced by formulating a suitable binary representation of a cross-section in the taskspace. A cross-section can be obtained with a suitable scan of the computed reachable positions and orientations \mathbf{p} , once the forward kinematic problem has been solved to give \mathbf{p} as function of the kinematic input joint variables \mathbf{q} . A binary matrix P_{ij} can be defined in the cross-section plane for a cross-section of the workspace as follows: if the (i, j) grid pixel includes a reachable point, then $P_{ij} = 1$; otherwise $P_{ij} = 0$, as shown in Fig. 3. Equations (1)-(4) for determining the workspace of a robot by discretization method can be found in Ref. (Ottaviano et al., 2002).

Then is computed i and j :

$$i = \left\lfloor \frac{x + \Delta x}{x} \right\rfloor, j = \left\lfloor \frac{y + \Delta y}{y} \right\rfloor \quad (1)$$

where i and j are computed as integer numbers. Therefore, the binary mapping for a workspace cross-section can be given as:

$$P_{ij} = \begin{cases} 0 & \text{if } P_{ij} \notin W(H) \\ 1 & \text{if } P_{ij} \in W(H) \end{cases} \quad (2)$$

where $W(H)$ indicates workspace region; \in stands for "belonging to" and \notin is for "not belonging to".

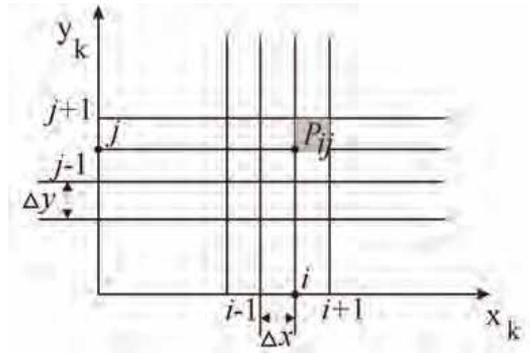


Fig. 3. The general scheme for binary representation and evaluation of robot workspace

In addition, the proposed binary representation is useful for a numerical evaluation of the position workspace by computing the sections areas A as:

$$A = \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta x \Delta y) \quad (3)$$

This numerical approximation of the workspace area has been used for the optimum design purposes.

2.2 Kinematics accuracy

The kinematics accuracy is a key factor for the design and application of the machine tools with parallel kinematics. But the research of the accuracy is still in initial stage because of the various structures and the nonlinear errors of the parallel kinematics machine tools.

To analyze the sensitiveness of the structural error is one of the directions for the research of structural accuracy. An approach was introducing a dimensionless factor of sensitiveness for every leg of the structure. Other approach includes the use of the value of Jacobian matrix as sensitivity index for the whole legs or the use of condition number of Jacobian matrix as a quantity index to describe the error sensitivity of the whole system.

2.3 Stiffness

Stiffness describes the ratio "deformation displacement to deformation force" (*static stiffness*). In case of dynamic loads this ratio (*dynamic stiffness*) depends on the exciting frequencies and comes to its most unfavorable (smallest) value at resonance (Hesselbach et al., 2003). In structural mechanics deformation displacement and deformation force are represented by vectors and the stiffness is expressed by the *stiffness matrix* K .

2.4 Singular configurations

Because singularity leads to a loss of the controllability and degradation of the natural stiffness of manipulators, the analysis of Parallel Kinematics Machines has drawn considerable attention. This property has attracted the attention of several researchers because it represents a crucial issue in the context of analysis and design. Most Parallel Kinematics Machines suffer from the presence of singular configurations in their workspace

that limit the machine performances. The singular configurations (also called singularities) of a Parallel Kinematics Machine may appear inside the workspace or at its boundaries. There are two main types of singularities (Gosselin & Angeles, 1990). A configuration where a finite tool velocity requires infinite joint rates is called a serial singularity or a type 1 singularity. A configuration where the tool cannot resist any effort and in turn, becomes uncontrollable is called a parallel singularity or type 2 singularity. Parallel singularities are particularly undesirable because they cause the following problems:

- a high increase of forces in joints and links, that may damage the structure,
- a decrease of the mechanism stiffness that can lead to uncontrolled motions of the tool though actuated joints are locked.

Thus, kinematics singularities have been considered for the formulated optimum design of the Parallel Kinematics Machines.

2.5 Dexterity

Dexterity has been considered important because it is a measure of a manipulator's ability to arbitrarily change its position and orientation or to apply forces and torques in arbitrary direction. Many researchers have performed design optimization focusing on the dexterity of parallel kinematics by minimization of the condition number of the Jacobian matrix. In regards to the PKM's dexterity, the condition number ρ , given by $\rho = \sigma_{\max} / \sigma_{\min}$ where σ_{\max} and σ_{\min} are the largest and smallest singular values of the Jacobian matrix J .

2.6 Manipulability

The determinant of the Jacobian matrix J , $\det(J)$, is proportional to the volume of the hyper ellipsoid. The condition number represents the sphericity of the hyper ellipsoid. The manipulability measure w , given by $w = \sqrt{\det(JJ^T)}$ was defined to describe the ability of machine tool with parallel structure to change its position and direction in its workspace.

3. Two DOF Parallel Kinematics Machines

3.1 Geometrical description of the Parallel Kinematics Machines

A planar Parallel Kinematics Machines is formed when two or more planar kinematic chains act together on a common rigid platform. The most common planar parallel architecture is composed of two RPR chains (Fig. 4), where the notation RPR denotes the planar chain made up of a revolute joint, a prismatic joint, and a second revolute joint in series. Another common architecture is PRRRP (Fig. 5). Two general planar Parallel Kinematics Machines with two degrees of freedom activated by prismatic joints are shown in Fig. 4 and Fig. 5.

There are a wide range of parallel robots that have been developed but they can be divided into two main groups:

- Type 1) Parallel Kinematics Machine with variable length struts,
- Type 2) Parallel Kinematics Machine with constant length struts.

Since mobility of these Parallel Kinematics Machines is two, two actuators are required to control these Parallel Kinematics Machines. For simplicity, the origin of the fixed base frame {B} is located at base joint A with its x-axis towards base joint B, and the origin of the moving frame {M} is located in TCP, as shown in Fig. 7. The distance between two base joints is b . The position of the moving frame {M} in the base frame {B} is $\mathbf{x} = (x_P, y_P)^T$ and the actuated joint variables are represented by $\mathbf{q} = (q_1, q_2)^T$.

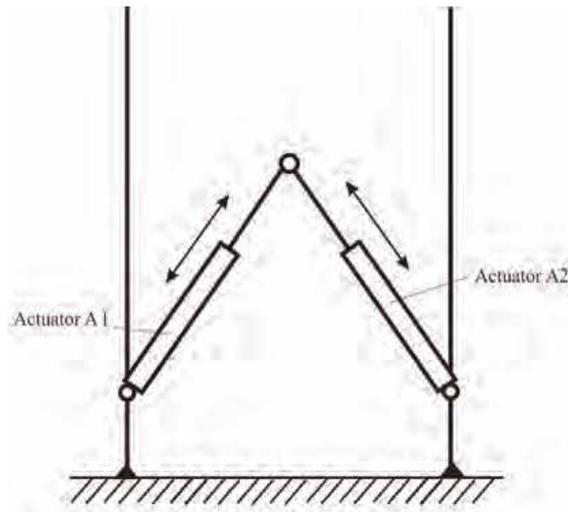


Fig. 4. Variable length struts Parallel Kinematics Machine

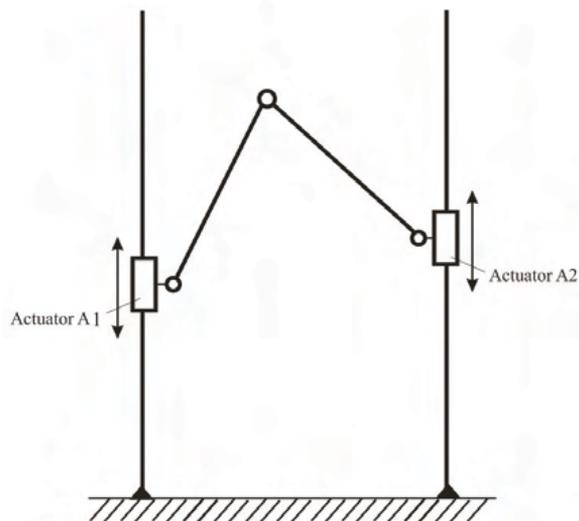


Fig. 5. Constant length struts Parallel Kinematics Machine

3.2 Kinematic analysis of the Parallel Kinematics Machines

PKM kinematics deal with the study of the PKM motion as constrained by the geometry of the links. Typically, the study of the PKMs kinematics is divided into two parts, inverse kinematics and forward (or direct) kinematics. The inverse kinematics problem involves a known pose (position and orientation) of the output platform of the PKM to a set of input joint variables that will achieve that pose. The forward kinematics problem involves the mapping from a known set of input joint variables to a pose of the moving platform that results from those given inputs. However, the inverse and forward kinematics problems of our PKMs can be described in closed form.

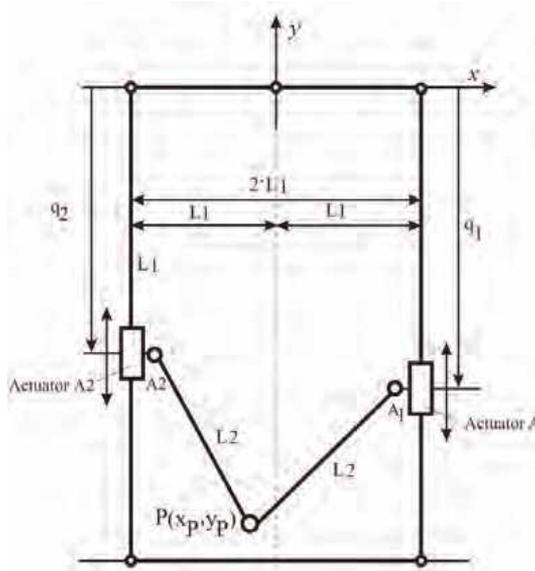


Fig. 6. The general kinematic scheme of a PRRRP Parallel Kinematics Machine

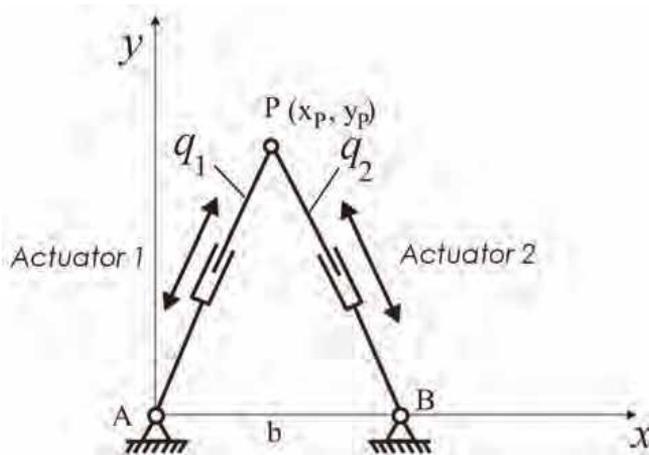


Fig. 7. The general kinematic scheme of a RPRPR Parallel Kinematics Machine

The kinematics relation between \mathbf{x} and \mathbf{q} of these 2 DOF Parallel Kinematics Machines can be expressed solving the following equation:

$$f(\mathbf{x}, \mathbf{q})=0 \tag{4}$$

Then the inverse kinematics problem of the PKM from Fig. 6 can be solved by writing the following equations:

$$q_1 = y_p \pm \sqrt{L_2^2 - (x_p - L_1)^2} \tag{5}$$

$$q_2 = y_p \pm \sqrt{L_2^2 - (x_p + L_1)^2}$$

Then the inverse kinematics problem of the PKM from Fig. 7 can be solved by writing the following equations:

$$q_1 = \sqrt{x_p^2 + y_p^2} \quad (6)$$

$$q_2 = \sqrt{(b - x_p)^2 + y_p^2}$$

The TCP position can be calculated by using inverted transformation, from (6), thus the direct kinematics of the PKM can be described as:

$$x_p = \frac{q_1^2 + b^2 - q_2^2}{2 \cdot b} \quad (7)$$

$$y_p = \sqrt{q_1^2 - x_p^2}$$

where the values of the x_p, y_p can be easily determined.

The forward and the inverse kinematics problems were solved under the MATLAB environment and it contains a user friendly graphical interface. The user can visualize the different solutions and the different geometric parameters of the PKM can be modified to investigate their effect on the kinematics of the PKM. This graphical user interface can be a valuable and effective tool for the workspace analysis and the kinematics of the PKM. The designer can enhance the performance of his design using the results given by the presented graphical user interface.

The Matlab-based program is written to compute the forward and inverse kinematics of the PKM with 2 degrees of freedom. It consists of several MATLAB scripts and functions used for workspace analysis and kinematics of the PKM. A friendly user interface was developed using the MATLAB-GUI (graphical user interface). Several dialog boxes guide the user through the complete process.



Fig. 8. Graphical User Interface (GUI) for solving inverse kinematics of the 2 DOF planar Parallel Kinematics Machine of type Bipod in MATLAB environment.

The user can modify the geometry of the 2 DOF PKM. The program visualizes the corresponding kinematics results with the new inputs.

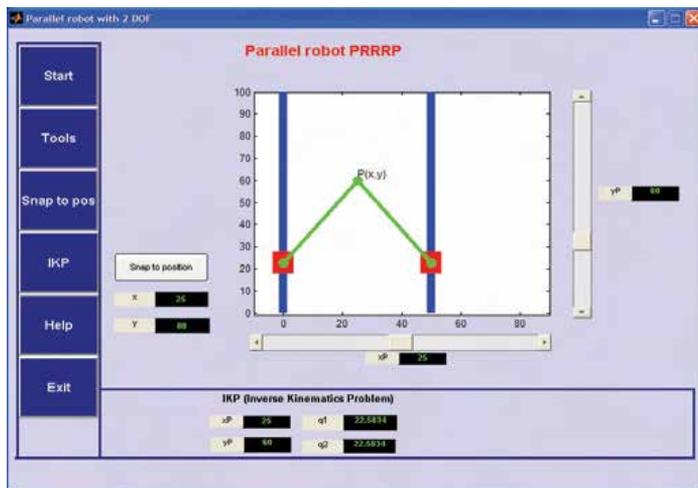


Fig. 9. Parallel Kinematics Machine configuration for $X_p=25$ mm $Y_p=60$ mm

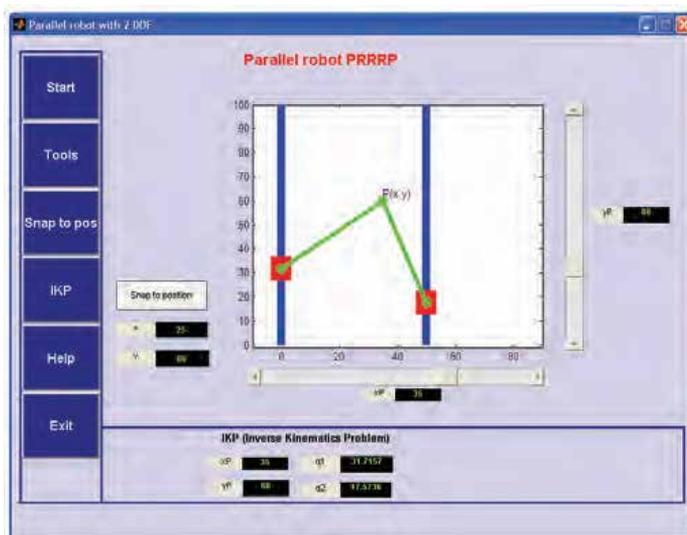


Fig. 10. Parallel Kinematics Machine configuration for $X_p=35$ mm $Y_p=60$ mm

4. Performance evaluation of Parallel Kinematics Machines

4.1 Workspace determination and optimization of the Parallel Kinematics Machines

The workspace is one of the most important kinematics properties of manipulators, even by practical viewpoint because of its impact on manipulator design and location in a workcell (Ceccarelli et al., 2005). Workspace is a significant design criterion for describing the kinematics performance of parallel robots. The planar parallel robots use area to evaluate the workspace ability. However, is hard to find a general approach for identification of the

workspace boundaries of the parallel robots. This is due to the fact that there is not a closed form solution for the direct kinematics of these parallel robots. That's why instead of developing a complex algorithm for identification of the boundaries of the workspace, it's developed a general visualization method of the workspace for its analysis and its design.

A general numerical evaluation of the workspace can be deduced by formulating a suitable binary representation of a cross-section in the taskspace. Other authors proposed to determine the workspace by using optimization (Stan, 2003). A fundamental characteristic that must be taken into account in the dimensional design of robot manipulators is the area of their workspace. It is crucial to calculate the workspace and its boundaries with perfect precision, because they influence the dimensional design, the manipulator's positioning in the work environment, and its dexterity to execute tasks. Because of this, applications involving these Parallel Kinematics Machines require a detailed analysis and visualization of the workspace of these PKMs. The algorithm for visualization of workspace needs to be adaptable in nature, to configure with different dimensions of the parallel robot's links. The workspace is discretized into square and equal area sectors. A multi-task search is performed to determine the exact workspace boundary. Any singular configuration inside the workspace is found along with its position and dimensions. The area of the workspace is also computed.

The workspace is the area in the plane case where the tool centre point (TCP) can be controlled and moved continuously and unobstructed. The workspace is limited by *singularities*. At singularity poses it is not possible to establish definite relations between input and output coordinates. Such poses must be avoided by the control.

The robotics literature contains various indices of performance (Du Plessis & Snyman, 2001) (Schoenherr & Bemessen, 1998), such as the workspace index W and the general equation is given in (8). Workspace for this kind of robot may be easily generated by intersection of the enveloping surfaces and the area can be also computed.

$$W = \int_W dW \quad (8)$$

The workspace of the 2 DOF planar PKM of type Bipod is often represented as a region of the plane, which can be obtained by the reachable points of the TCP.

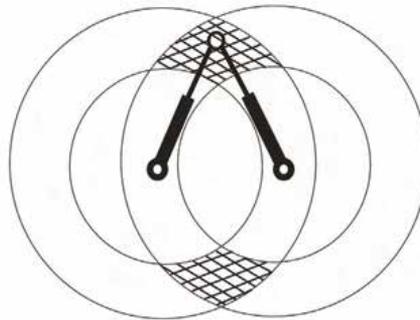


Fig. 11. The workspace is the intersection of two enveloping surface of two legs.

The following presents the main factors affecting workspace. For ease of comparison a cubic working envelope with a common contour length is used together with a machine size that

is calculated from the maximum required strut length. Other design specific factors such as the end-effector size, drive volumes have been neglected for simplification.

The working envelope to machine size using variable length struts is dependent on the following factors:

1. The length of the extended and retracted actuator (L_{min} , L_{max});
2. Limitations due to the joint angle range.

The limiting effect of the joint limits is clearly illustrated in Fig. 12-13.

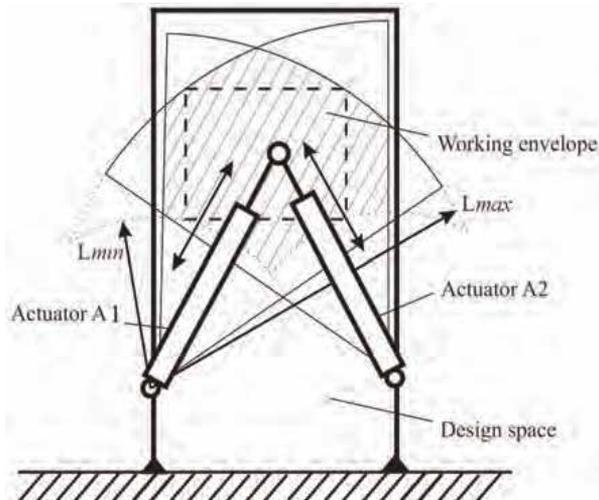


Fig. 12. Workspace of the Parallel Kinematics Machine with variable length struts

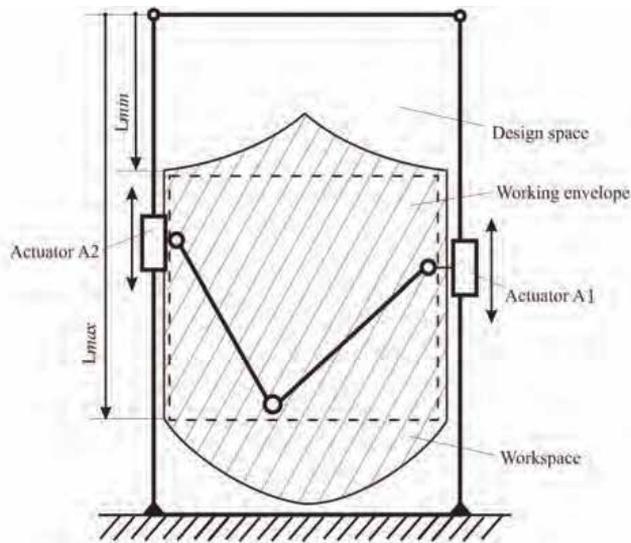


Fig. 13. Workspace of the Parallel Kinematics Machine with constant length struts

In this section, the workspace of the proposed Parallel Kinematics Machines will be discussed systematically. It's very important to analyze the area and the shape of workspace

for parameters given robot in the context of industrial application. The workspace is primarily limited by the boundary of solvability of inverse kinematics. Then the workspace is limited by the reachable extent of drives and joints, occurrence of singularities and by the link and platform collisions. The PKM mechanisms $\underline{P}RRRP$ and $RPRPR$ realize a wide workspace as well as high-speed. Analysis, visualization of workspace is an important aspect of performance analysis. A numerical algorithm to generate reachable workspace of parallel manipulators is introduced.

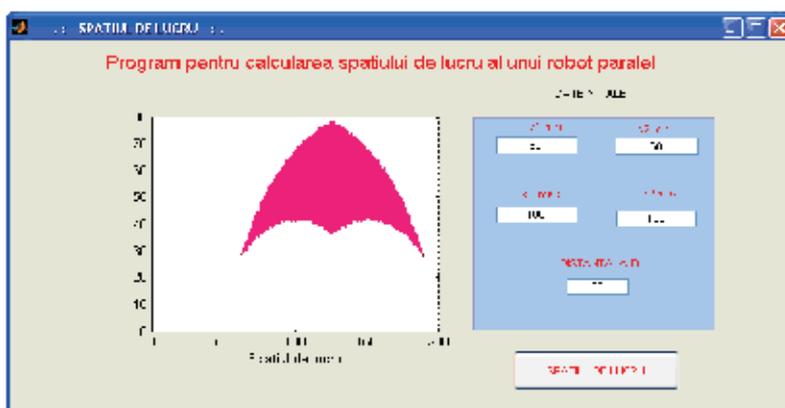


Fig. 14. The GUI for calculus of workspace for the planar 2 DOF Parallel Kinematics Machine with variable length struts

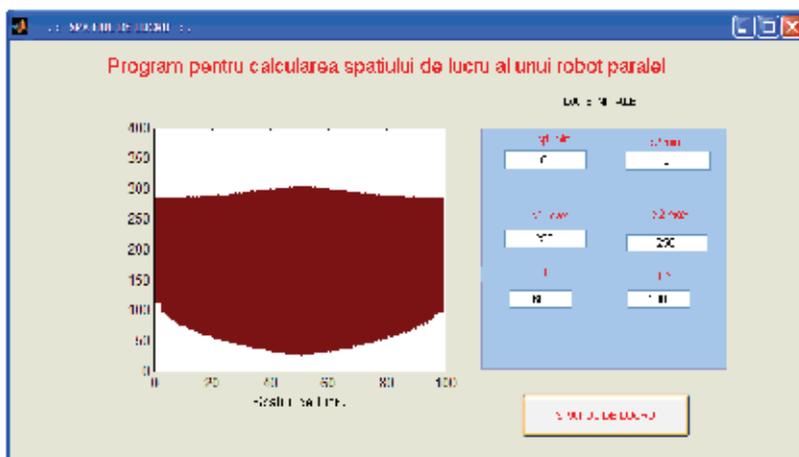


Fig. 15. The GUI for calculus of workspace for the planar 2 DOF Parallel Kinematics Machine with constant length struts

In the followings is presented the workspace analysis of 2 DOF Bipod PKM.

Case I:

Conditions:

$$q_{1min} + q_{2min} > b, q_{1max} > b, q_{2max} > b$$

a) for $y > 0$

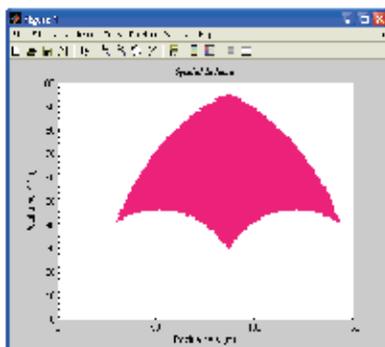


Fig. 16. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

b) for $-\infty < y < +\infty$, there exist two regions of the workspace

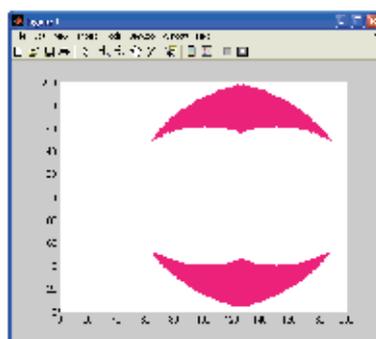


Fig. 17. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case II:

Conditions:

$$q_{1min} + q_{2min} > b, q_{1max} < b, q_{2max} < b$$

a) for $y > 0$

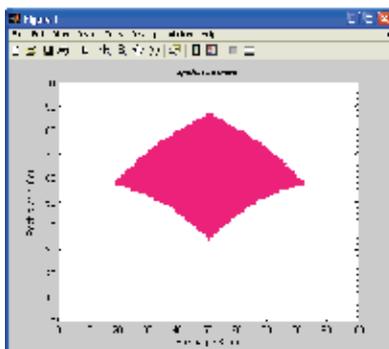


Fig. 18. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

b) for $-\infty < y < +\infty$, there exist two regions of the workspace

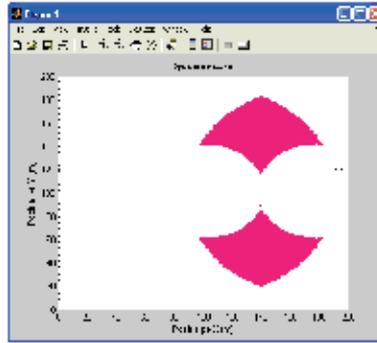


Fig. 19. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case III:

Conditions: $q_{1min} + q_{2min} < b$, $q_{1max} > b$, $q_{2max} > b$

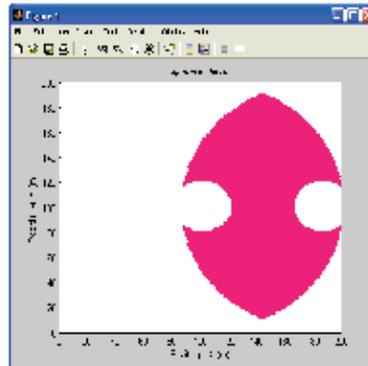


Fig. 20. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case IV:

Conditions: $q_{1min} + q_{2min} < b$, $q_{1max} < b$, $q_{2max} < b$

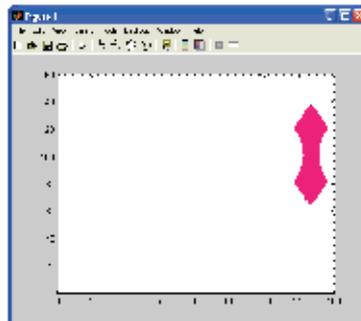


Fig. 21. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case V:

Conditions: $q_{1min} + q_{2min} < b$, $q_{1max} > b + q_{2min}$, $q_{2max} > b + q_{1min}$

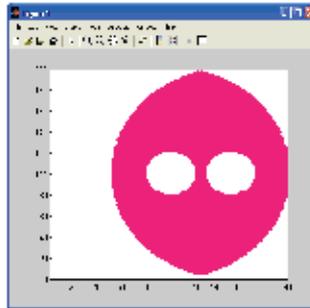


Fig. 22. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case VI:

Conditions: $q_{1min} + q_{2min} > b$, $q_{1max} > b + q_{2min}$, $q_{2max} > b + q_{1min}$

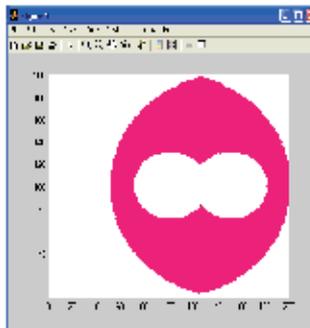


Fig. 23. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

Case VII:

Conditions: $q_{1min} < b$, $q_{1max} < b$, $q_{2min} < b$, $q_{2max} < b$, $q_{1min} + q_{2min} < b$,
 $q_{1max} + q_{2max} > b$

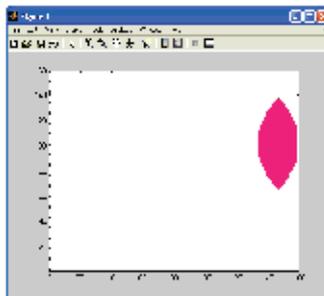
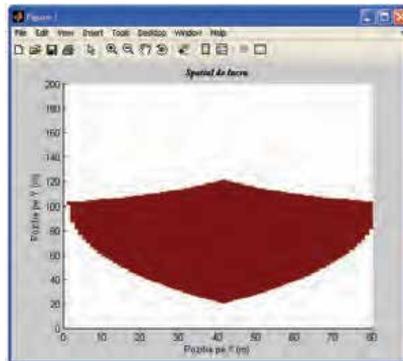
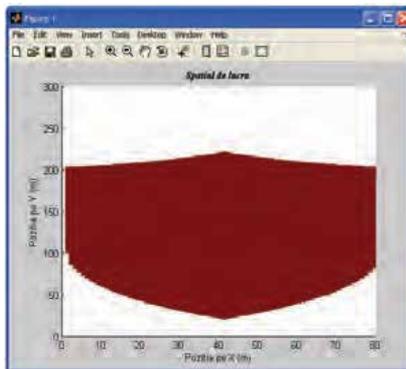


Fig. 24. The workspace of the planar 2 DOF Parallel Kinematics Machine is shown as the shading region.

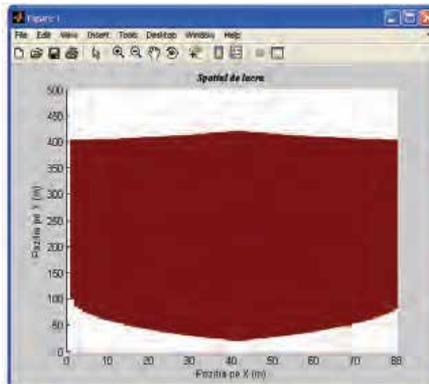
In the followings is presented the workspace analysis of 2 DOF Biglide Parallel Kinematics Machine.



a) Workspace for the planar 2 DOF Parallel Kinematics Machine, case $q_{1max} = q_{2max} = 100 \text{ mm}$



b) Workspace for the planar 2 DOF Parallel Kinematics Machine, case $q_{1max} = q_{2max} = 200 \text{ mm}$



c) Workspace for the planar 2 DOF Parallel Kinematics Machine, case $q_{1max} = q_{2max} = 400 \text{ mm}$

Fig. 25. Different regions of workspace for Biglide PKM for different lengths of stroke of actuators

4.2 Singularity analysis of the Biglide Parallel Kinematics Machine

Because singularity leads to a loss of the controllability and degradation of the natural stiffness of manipulators, the analysis of parallel manipulators has drawn considerable attention. Most parallel robots suffer from the presence of singular configurations in their workspace that limit the machine performances. Based on the forward and inverse Jacobian matrix, three cases of singularities of parallel manipulators can be obtained. Singular configurations should be avoided.

In the followings are presented the singular configurations of 2 DOF Biglide Parallel Kinematic Machine.

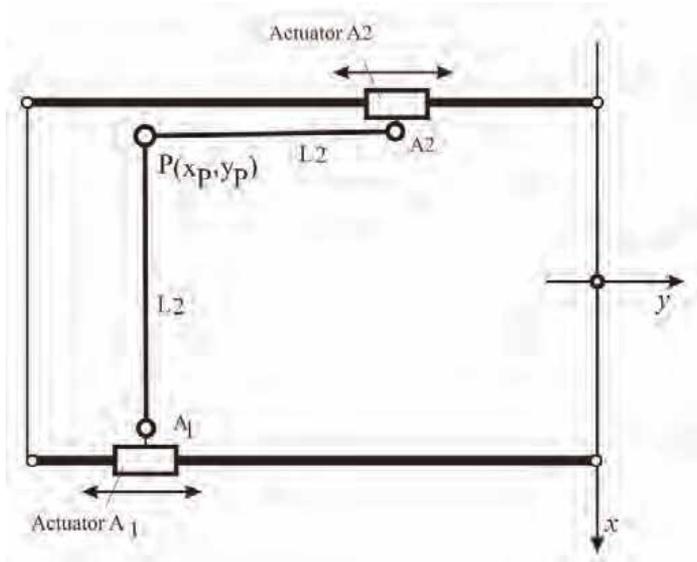


Fig. 26. Singular configuration for the planar 2 DOF Biglide Parallel Kinematic Machine

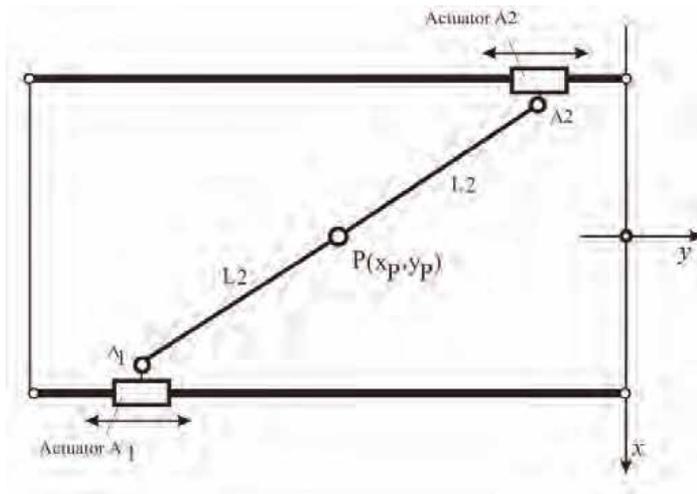


Fig. 27. Singular configuration for the planar 2 DOF Biglide Parallel Kinematic Machine

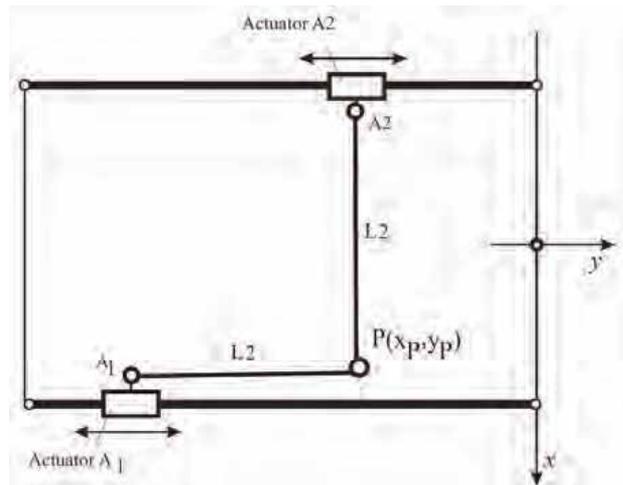


Fig. 28. Singular configuration for the planar 2 DOF Biglide Parallel Kinematic Machine

4.2 Performance evaluation

Beside workspace which is an important design criterion, transmission quality index is another important criterion. The transmission quality index couples velocity and force transmission properties of a parallel robot, i.e. power features (Hesselbach et al., 2004). Its definition runs:

$$T = \frac{\|I\|^2}{\|J\| \cdot \|J^{-1}\|} \quad (9)$$

where I is the unity matrix. T is between $0 < T < 1$; $T=0$ characterizes a singular pose, the optimal value is $T=1$ which at the same time stands for isotropy (Stan, 2003).

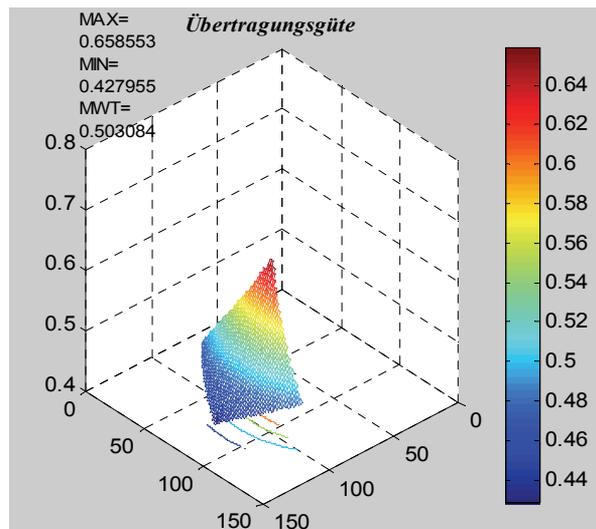


Fig. 29. Transmission quality index for RPRPR Bipod Parallel Kinematic Machine

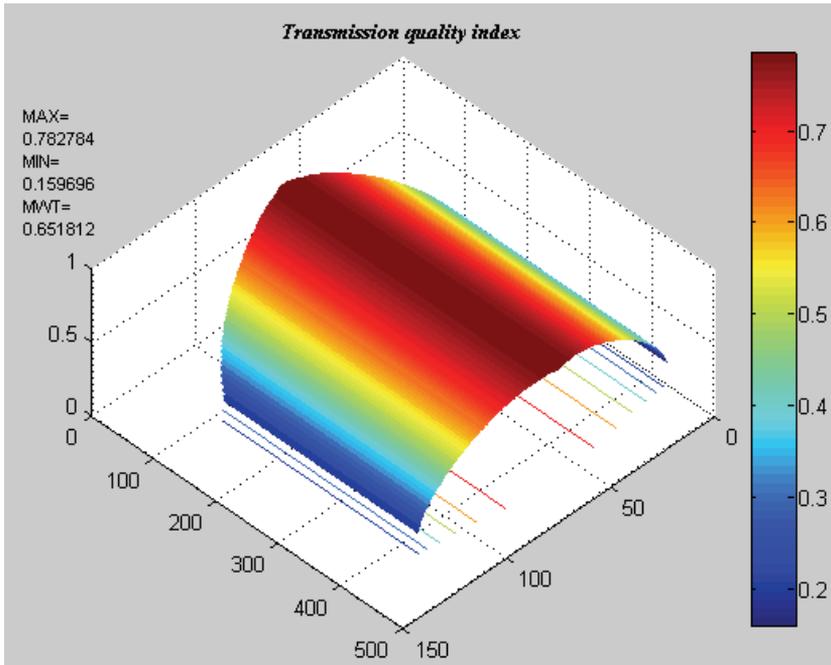


Fig. 30. Transmission quality index for PRRRP Biglide Parallel Kinematic Machine

As it can be seen from the Fig. 30, the performances of the PRRRP Biglide Parallel Kinematic Machine are constant along y -axis. On every y section of such workspace, the performance of the robot can be the same.

5. Optimal design of 2 DOF Parallel Kinematics Machines

5.1 Optimization results for RPRPR Parallel Kinematic Machine

The design of the PKM can be made based on any particular criterion. The chapter presents a genetic algorithm approach for workspace optimization of Bipod Parallel Kinematic Machine. For simplicity of the optimization calculus a symmetric design of the structure was chosen.

In order to choose the PKM's dimensions $b, q_{1min}, q_{1max}, q_{2min}, q_{2max}$, we need to define a performance index to be maximized. The chosen performance index is W (workspace) and T (transmission quality index).

An objective function is defined and used in optimization. It is noted as in Eq. (8), and corresponds to the optimal workspace and transmission quality index. We can formalize our design optimization problem as the following:

$$ObjFun=W+T \tag{10}$$

Optimization problem is formulated as follows: the objective is to evaluate optimal link lengths which maximize Eq. (10). The design variables or the optimization factor is the ratios of the minimum link lengths to the base link length b , and they are defined by:

$$q_{1min}/b \tag{11}$$

Constraints to the design variables are:

$$0,52 < q_{1min}/b < 1,35 \quad (12)$$

$$q_{1min} = q_{2min}, q_{1max} = q_{2max}, q_{1max} = 1,6q_{1min}, q_{2max} = 1,6q_{2min} \quad (13)$$

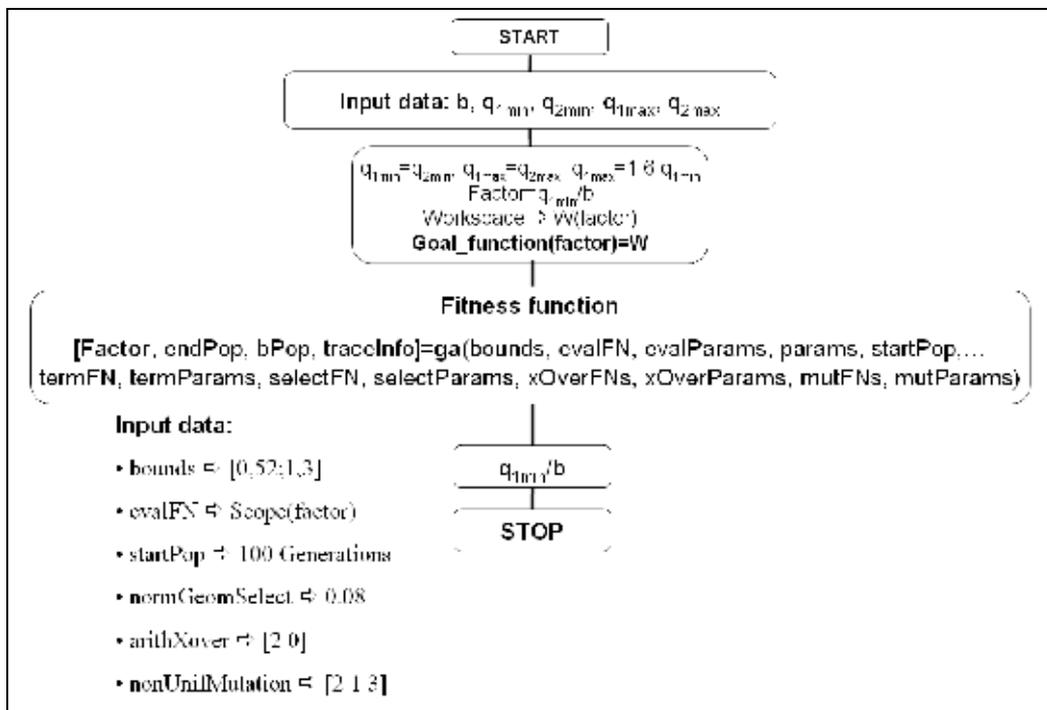


Fig. 31. Flowchart of the optimization Algorithm with GAOT (Genetic Algorithm Optimization Toolbox)

For this example the lower limit of the constraint was chosen to fulfill the condition $q_{1min} \geq b/2$ that means the minimum stroke of the actuators to have a value greater than the half of the distance between them in order to have a workspace only in the upper region. For simplicity of the optimization calculus the upper bound was chosen $q_{1min} \leq 1,35b$.

During optimization process using genetic algorithm it was used the following GA parameters, presented in Table 1.

Generations	100
Crossover rate	0.08
Mutation rate	0.005
Population	50

Table 1. GA Parameters

Researchers have used genetic algorithms, based on the evolutionary principle of natural chromosomes, in attempting to optimize the design parallel kinematics. Kirchner and Neugebauer (Kirchner & Neugebauer, 2000), emphasize that a parallel manipulator machine tool cannot be optimized by considering a single performance criterion. Also, using a

genetic algorithm, they consider a multiple design criteria, such as the “velocity relationship” between the moving platform and the actuator legs, the influence of actuator leg errors on the accuracy of the moving platform, actuator forces, stiffness, as well as a singularity-free workspace.

A genetic algorithm (GA) is used because its robustness and good convergence properties. The genetic algorithms optimization approach has the clear advantage over conventional optimization approaches in that it allows a number of solutions to be examined in a single design cycle.

The traditional methods searches optimal points from point to point, and are easy to fall into local optimal point. Using a population size of 50, the GA was run for 100 generations. A list of the best 50 individuals was continually maintained during the execution of the GA, allowing the final selection of solution to be made from the best structures found by the GA over all generations.

We performed a kinematic optimization in such a way to maximize the objective function. It is noticed that optimization result for Bipod when the maximum workspace of the 2 DOF planar PKM is obtained for $q_{1min} / b = 1,35$. The used dimensions for the 2 DOF parallel PKM were: $q_{1min}=80$ mm, $q_{1max}=130$ mm, $q_{2min}=80$ mm, $q_{2max}=130$ mm, $b=60$ mm. Maximum workspace of the Parallel Kinematics Machine with 2 degrees of freedom was found to be $W= 4693,33$ mm².

If an elitist GA is used, the best individual of the previous generation is kept and compared to the best individual of the new one. If the performance of the previous generation’s best individual is found to be superior, it is passed on to the next generation instead of the current best individual.

There have been obtained different values of the parameter optimization (q_1/b) for different objective functions. The following table presents the results of optimization for different goal functions. W_1 and W_2 are the weight factors.

	Method	GAOT Toolbox MATLAB
Goal functions	$Z=W_1 \cdot T+W_2 \cdot W$, $W_1=0,7$ and $W_2=0,3$	$q_1/b = 0.92$
	$Z=W_1 \cdot T+W_2 \cdot W$, $W_1=0,3$ and $W_2=0,7$	$q_1/b= 1.13$
	$Z= W_1 \cdot T$, $W_1=1$ and $W_2=0$	$q_1/b=0.71$
	$Z=W_2 \cdot W$, $W_1=0$ and $W_2=1$	$q_1/b=1.3$

Table 2. Results of Optimization for Different Goal Functions

The results show that GA can determine the architectural parameters of the robot that provide an optimized workspace. Since the workspace of a parallel robot is far from being intuitive, the method developed should be very useful as a design tool.

However, in practice, optimization of the robot geometrical parameters should not be performed only in terms of workspace maximization. Some parts of the workspace are more useful considering a specific application. Indeed, the advantage of a bigger workspace can

be completely lost if it leads to new collision in parts of it which are absolutely needed in the application. However, it's not the case of the presented structure.

5.2 Optimization results for PRRRP Parallel Kinematic Machine

An objective function is defined and used in optimization. Objective function contains workspace and transmission quality index. Optimization parameter was chosen as the link length L_2 . The constraints was established as $1 < L_2 < 1.2$. After performing the optimization the following results were obtained:

	Method	GAOT Toolbox MATLAB
Goal functions	$Z = W_1 \cdot T + W_2 \cdot W$, $W_1 = 0,7$ and $W_2 = 0,3$	$L_2 = 1.1$
	$Z = W_1 \cdot T + W_2 \cdot W$, $W_1 = 0,3$ and $W_2 = 0,7$	$L_2 = 1.1556$
	$Z = W_1 \cdot T$, $W_1 = 1$ and $W_2 = 0$	$L_2 = 1$
	$Z = W_2 \cdot W$, $W_1 = 0$ and $W_2 = 1$	$L_2 = 1.2$

Table 3. Results of Optimization for Different Goal Functions

Based on the presented optimization methodology we can conclude that the optimum design and performance evaluation of the Parallel Kinematics Machines is the key issue for an efficient use of Parallel Kinematics Machines. This is a very complex task and in this paper was proposed a framework for the optimum design considering basic characteristics of workspace, singularities and isotropy.

6. Conclusion

The fundamental guidelines for genetic algorithm to optimal design of micro parallel robots have been introduced. It is concluded that with three basic generators selection, crossover and mutation genetic algorithm could search the optimum solution or near-optimal solution to a complex optimization problem of micro parallel robots. In the paper, design optimization is implemented with Genetic Algorithms (GA) for optimization considering transmission quality index, design space and workspace. Genetic algorithms (GA) are so far generally the best and most robust kind of evolutionary algorithms. A GA has a number of advantages. It can quickly scan a vast solution set. Bad proposals do not affect the end solution negatively as they are simply discarded. The obtained results have shown that the use of GA in such kind of optimization problem enhances the quality of the optimization outcome, providing a better and more realistic support for the decision maker.

7. References

Agrawal, S. K., (1990). Workspace boundaries of in-parallel manipulator systems. *Int. J. Robotics Automat.* 1990, 6(3) 281-290.

- Cecarelli, M., (1995). A synthesis algorithm for three-revolute manipulators by using an algebraic formulation of workspace boundary. *ASME J. Mech. Des.*; 117(2(A)): 298-302.
- Ceccarelli, M., G. Carbone, E. Ottaviano, (2005). An Optimization Problem Approach For Designing Both Serial And Parallel Manipulators, In: *Proc. of MUSME 2005, the International Symposium on Multibody Systems and Mechatronics*, Uberlandia, Brazil, 6-9 March 2005.
- Ceccarelli, M., (2004). *Fundamentals of Mechanics of Robotic Manipulation*, Dordrecht, Kluwer/Springer.
- Cleary, K. and Arai, T. (1991). A prototype parallel manipulator: Kinematics, construction, software, workspace results, and singularity analysis. In: *Proceedings of International Conference on Robotics and Automation*, pages 566-571, Sacramento, California, USA, April 1991.
- Davidson, J. K. and Hunt, K. H., (1987). Rigid body location and robot workspace: some alternative manipulator forms. *ASME Journal of Mech. Transmissions Automat Des*, 109(2); 224-232.
- Du Plessis L.J. and J.A. Snyman, (2001). A numerical method for the determination of dextrous workspaces of Gough-Stewart platforms. *Int. Journal for Numerical Methods in Engineering*, 52:345-369.
- Ferraresi, C., Montacchini, G. and M. Sorli, (1995). Workspace and dexterity evaluation of 6 d.o.f. spatial mechanisms, In: *Proceedings of the ninth World Congress on the theory of Machines and Mechanism*, pages 57-61, Milan, August 1995.
- Gogu, G., (2004), Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations, *European Journal of Mechanics, A/Solids*, vol. 23, pp. 1021-1039.
- Gosselin, C. (1990). Determination of the workspace of 6-d.o.f. parallel manipulators. *ASME Journal of Mechanical Design*, 112:331-336.
- Gosselin, C., and Angeles J. (1990). Singularities analysis of closed loop kinematic chains. *IEEE Trans Robotics Automat*; 6(3) 281-290.
- Gupta, K. C. (1986). On the nature of robot workspaces, *International Journal of Robotics Research*. 5(2): 112-121.
- Gupta, K. G. and Roth B., (1982). Design considerations for manipulator workspace. *ASME J. Mech. Des.*, 104(4), 704-711.
- Hesselbach, J., H. Kerle, M. Krefft, N. Plitea, (2004). The Assessment of Parallel Mechanical Structures for Machines Taking Account of their Operational Purposes. In: *Proceedings of the 11th World Congress in Mechanism and Machine Science-IFTOMM 11*, Tianjin, China, 2004.
- Kirchner, J., and Neugebauer, R., (2000). How to Optimize Parallel Link Mechanisms - Proposal of a New Strategy. In: *Proceedings Year 2000 Parallel Kinematics Machines International Conference*, September 13-15, 2000, Ann Arbor, Mi. USA, [Orlandea, N. et al. (eds.)], pp. 307-315.
- Kumar, A. and Waldron, (1981). K.J. The workspace of mechanical manipulators. *ASME J. Mech. Des.*; 103:665-672.
- Masory, O. and Wang J. (1995). Workspace evaluation of Stewart platforms. *Advanced robotics*, 9(4):443-461.

- Merlet, J. P., (1995). Determination of the orientation workspace of parallel manipulators. *Journal of intelligent and robotic systems*, 13:143–160.
- Pernkopf, F. and Husty, M., (2005). Reachable Workspace and Manufacturing Errors of Stewart-Gough Manipulators, *Proc. of MUSME 2005, the Int. Sym. on Multibody Systems and Mechatronics Brazil*, p. 293-304.
- Schoenherr, J., (1998). Bemessen Bewerten und Optimieren von Parallelstrukturen, In: *Proc. 1st Chemnitzer Parallelstruktur Seminar*, Chemnitz, Germany, 85-96.
- Snyman, J. A., L.J. du Plessis, and J. Duffy. (2000). An optimization approach to the determination of the boundaries of manipulator workspaces. *Journal of Mechanical Design*, 122:447–455.
- Stan, S., (2003). Analyse und Optimierung der strukturellen Abmessungen von Werkzeugmaschinen mit Parallelstruktur, *Diplomarbeit, IWF-TU Braunschweig*, Germany.
- Stan, S., (2006). Workspace optimization of a two degree of freedom mini parallel robot, *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics – AQTR 2006 (THETA 15)*, May 25-28 2006, Cluj-Napoca, Romania, IEEE Catalog number: 06EX1370, ISBN: 1-4244-0360-X, pp. 278-283.
- Stan, S. and Lăpușan, C., (2006). Workspace analysis of a 2 dof mini parallel robot, *The 8th National Symposium with International Participation COMPUTER AIDED DESIGN - PRASIC'06*, Brașov, 9 - 10th November 2006, pag. 175-180, ISBN (10)973-653-824-0; (13)978-973-635-824-1.
- Stan, S., Vistriean M., Balan, R. (2007). Optimal Design of a 2 DOF Micro Parallel Robot Using Genetic Algorithms, *Proceedings of the 2007 IEEE-ICIT 2007, IEEE International Conference on Integration Technology*, March 20 - 24, 2007, Shenzhen, China, 1-4244-1092-4/07, p. 719-724, IEEE Catalog Number: 07EX1735, ISBN: 1-4244-1091-6, ISBN: 1-4244-1092-4.
- Stan, S., Balan, R., Vistriean M., (2007). Multi-objective Design Optimization of Mini Parallel Robots Using Genetic Algorithms, *IEEE-ISIE 2007 IEEE International Symposium on Industrial Electronics*, June 4-7, 2007, Caixanova - Vigo, Spain, IEEE Catalog Number: 07TH8928C, ISBN: 1-4244-0755-9, Library of Congress: 2006935487, pag. 1-4244-0755-9/07/ IEEE 2173-2178.
- Stan, S., Maties, V., Balan, R., (2007). Optimization of 2 DOF Micro Parallel Robots Using Genetic Algorithms, *IEEE-ICM 2007, IEEE - International Conference on Mechatronics 2007*, 8-10 May, 2007, Kumamoto, Japan, ISBN: 1-4244-1184-X IEEE Catalog Number of CD proceedings: 07EX1768C, ISBN of CD proceedings: 1-4244-1184-X, pp.1-6
- Sugimoto, K., Duffy J., Hunt K. H., (1982). Special configurations of spatial mechanisms and robot arms. *Mech Mach Theory* 1982, 117(2); 119-132.
- Tsai, Y. C. and Soni, A.H., (1981). Accessible region and synthesis of robot arm. *ASME J. Mech Des.*, 103: 803-811.

The Analysis and Application of Parallel Manipulator for Active Reflector of FAST

Xiao-qiang Tang and Peng Huang
Tsinghua University
People's Republic of China

1. Introduction

Since radio telescope is the main tool for human being to search the universe secret, the astronomer reached unanimity at the 24th URSI Conference in Kyoto, Japan, 1993, and proposed to construct the next generation of the large radio telescope (LT) (Nan & Peng, 2000). From then on, the astronomer of China began the project of Five-hundred meter Aperture Spherical radio Telescope (FAST) (Qiu, 1998; Li, 1998).

It is well known that Arecibo is the breakthrough of radio telescope. Its main mirror, 305m in diameter, is fixed on the karst base, and an elaborately designed feed system illuminates a part of the mirror which forms an effective aperture of the telescope with about 200m. The feed system is movable at a height of about 100 m for tracking the object to be observed. The enormous receiving area of the telescope will enable it to make many important astronomical discoveries inaccessible to lesser instruments, despite its small sky coverage (20° zenith scan angle), due to geometrical configuration, and narrow frequency bandwidth, originated from spherical aberration. An upgrade project has recently been carried out for the Arecibo telescope, in which a heavy and complex hence expensive Gregorian dual-reflector feed system is introduced for correcting the spherical aberration and a broad bandwidth is affected (Duan, 1999).

For the sake of satisfying the requirements of low cost and broad bandwidth, the project group of FAST decided to substitute the fixed spherical reflector with active reflector units. As shown in Fig. 1(a), the reflector consists of almost 2000 elementary reflector units. Fig. 1(b) shows some active reflector units and supporting mechanisms. The reflector unit is small part of spherical surface of regular hexagon and is driven by a supporting mechanism. The part of spherical reflector illuminated by the feed is continuously adjusted to fit a paraboloid of revolution in real-time, synchronous with the motion of the feed while tracking the object to be observed. As it is now free from spherical aberration, a simple, light, hence cheap feed system may be adopted to achieve broad bandwidth and full polarization.

In order to fit a paraboloid of revolution, it is necessary that every reflector units should be driven by a supporting mechanism with two rotational degrees of freedom and one translational degree of freedom (Luo et al., 2000). That means almost 6000 control nodes on the whole active reflector should be managed and controled at the same time. It is supposed to be very difficult, so a sharing strategy is derived to decrease the number of nodes, which

requires three adjacent nodes combined together to share one driver. Basically, there are two types of mechanism which can fulfill the required movement for each reflector unit and fit for the sharing strategy, 3-PSS mechanism with constraint leg (Wang et al., 2006), shown in Fig. 2(a), 3-PSS+C for abbreviation, and 3-PRS mechanism (Tang et al., 2007), shown in Fig. 2(b).

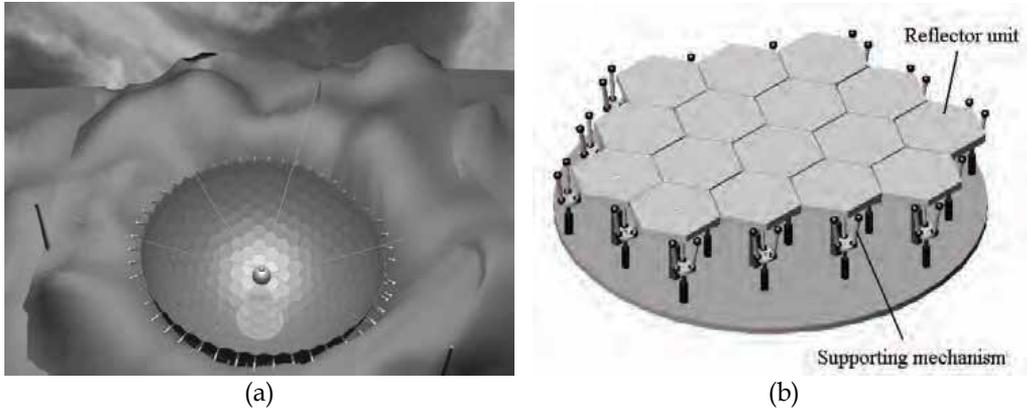


Fig. 1. The active reflector of spherical radio telescope

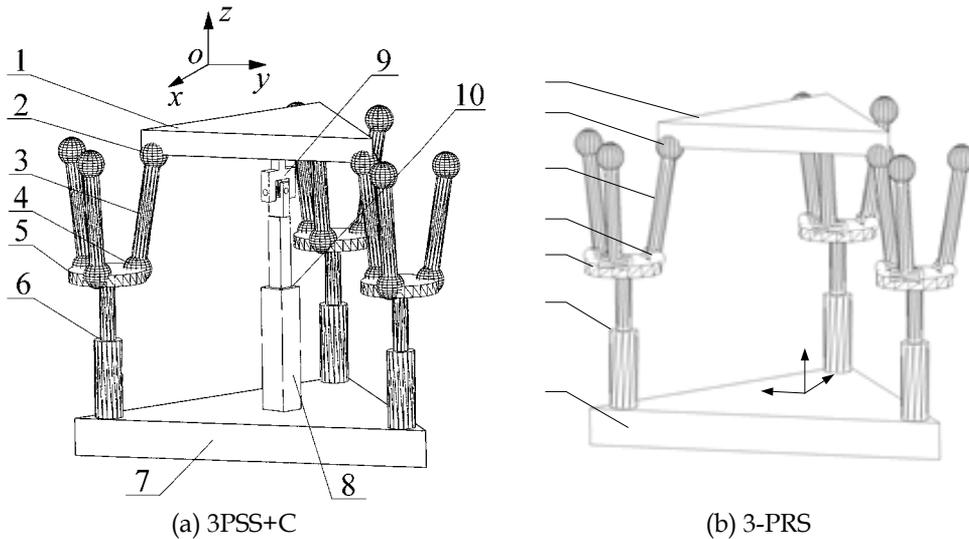


Fig. 2. The parallel supporting mechanism

These mechanisms will bring errors because of the control or dimensional factor. Moreover, the fitting surface of reflector will not match exactly with the nominal paraboloid, and the sharing strategy also brings accuracy problem. In order to guarantee the highest working frequency of large spherical radio telescope, 5GHz, the fitting accuracy of active reflector should be studied systematically. Based on the kinematics of 3-PSS+C mechanism, in this chapter, one-dimensional and two-dimensional fitting accuracy on the whole active reflector is analyzed considering control errors. However, about 2000 constraint legs increase almost one quarter of the cost. Thus 3-PRS mechanism is proposed and used as supporting

manipulator for reflector unit. Since 3-PRS mechanism has many problems such as parasitic motion, advanced research on kinematics with errors is necessary. Then three-dimensional fitting accuracy is analyzed based on error kinematics of 3-PRS mechanism.

2. The analysis of 3-PSS+C supporting mechanism

2.1 Supporting mechanism description

As shown in Fig. 2(a), the parallel supporting mechanism consists of a base plate, a movable platform, and four connecting legs, three of which have identical kinematic chains, PSS. Each of the three legs is composed of one fixed length link (3), and one union driven plate (5). The fixed length link (3) is connected to the movable platform (1) and the union driven plate (5) by two spherical joints (2) and (4), respectively. The union driven plate (5) is connected to the base plate (7) by a prismatic joint (6). The base plate and the movable platform are two regular triangles. The passive leg (8) connects the center points of the two regular triangles. One end of the passive leg has a 2-DOF universal joint (9), another end is fixed to the base plate (7) by a prismatic joint (10). The passive leg (8) can be extensible with the prismatic joint (10) along its axis line. Furthermore, when the supporting mechanism is assembled, the axis line of the prismatic joint (10) should pass the center of the spherical reflector. Since a supporting mechanism should be driven by three actuator legs, as shown in Fig. 2, the union driven plate (5) connects three fixed length links in order to reduce the actuator number. As a result, the number of actuators of the active reflector is equal to that of the reflector units.

From above description, one can see that the proposed mechanism is such a mechanism with n DOFs, which usually consists of n identical actuated legs with 6 DOFs and one passive leg with n DOFs connecting the movable platform and the base plate, i.e., the DOF of the mechanism is dependent on the passive leg's DOF. For the mechanism considered in this paper, the passive leg is with three DOFs, which means that n equals to 3. The three DOFs are one translation along z axis and two rotations about x and y axes.

2.2 Kinematics analysis

The mechanism kinematics deals with the study of the mechanism motion as constrained by the geometry of the links. Typically, the study of mechanism kinematics is divided into two parts, inverse kinematics and forward (or direct) kinematics (Wang & Tang, 2003). The inverse kinematics problem involves mapping a known pose (position and orientation) of the output platform of the mechanism to a set of input joint variables that will achieve that pose. The forward kinematics problem involves the mapping from a known set of input joint variables to a pose of the movable platform that results from those given inputs (Wang et al., 2001). Generally, as the number of closed kinematics loops in the parallel mechanism increases, the difficulty of solving the forward kinematics relationships increases while the difficulty of solving the inverse kinematics relationships decreases (Liu et al., 2001).

2.2.1 Inverse kinematics

A kinematics model of the mechanism is developed as shown in Fig. 3. The vertices of the movable platform are denoted as platform joints A_i ($i=1, 2, 3$), and the vertices of the base plate are denoted as b_i ($i=1, 2, 3$). A fixed global reference system $\mathfrak{R}: o-xyz$ is located at the center of the regular triangles $b_1b_2b_3$ with the z axis normal to the base plate and the

y axis parallel to the side b_1b_2 . The circumcircle radius of triangles $b_1b_2b_3$ is denoted as R . Another reference frame, called the top frame $\mathfrak{R}' : o' - x'y'z'$, is located at the center of regular triangles $A_1A_2A_3$. The z' axis is perpendicular to the movable platform and y' axis parallel to the side A_1A_2 . The circumcircle radius of triangles $A_1A_2A_3$ is denoted as r . Vector of fixed length links are denoted as L_i ($i = 1, 2, 3$), and the link length for each legs is denoted as l , where $A_iB_i = l$, ($i = 1, 2, 3$).

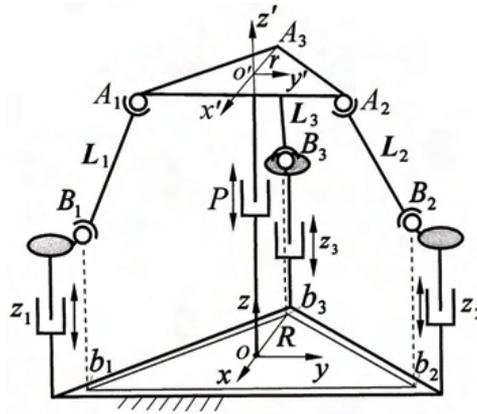


Fig. 3. The geometric parameters of the parallel mechanism

The objective of the inverse kinematics solution is to define a mapping from the pose of the output platform in the Cartesian space to the set of actuated inputs that achieve that pose. For this analysis, the pose of the movable platform is considered known, and the position is given by the position vector $[o']_{\mathfrak{R}}$ and the orientation is given by a matrix R_1 . Then there are

$$[o']_{\mathfrak{R}} = (x \ y \ z)^T \tag{1}$$

where $x = y = 0$,

$$R_1 = \begin{bmatrix} c\beta & s\beta s\alpha & s\beta c\alpha \\ 0 & c\alpha & -s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{bmatrix} \tag{2}$$

where c stands for cosine function, s stands for sine function, and α and β are the orientational DOFs of the movable platform with respect to x and y axes, respectively. The coordinate of point A_i in the frame \mathfrak{R}' can be described by the vector $[A_i]_{\mathfrak{R}'}$ ($i = 1, 2, 3$), and

$$[A_1]_{\mathfrak{R}'} = [r/2, -\sqrt{3}r/2, 0]^T, [A_2]_{\mathfrak{R}'} = [r/2, \sqrt{3}r/2, 0]^T, [A_3]_{\mathfrak{R}'} = [-r, 0, 0]^T \tag{3}$$

Vectors $[B_i]_{\mathfrak{R}}$ ($i = 1, 2, 3$) will be defined as the position vectors of base joints B_i in frame \mathfrak{R} , and

$$[\mathbf{B}_1]_{\mathcal{R}} = [R/2, -\sqrt{3}R/2, z_1]^T, [\mathbf{B}_2]_{\mathcal{R}} = [R/2, \sqrt{3}R/2, z_2]^T, [\mathbf{B}_3]_{\mathcal{R}} = [-R, 0, z_3]^T \quad (4)$$

Vectors $[\mathbf{A}_i]_{\mathcal{R}}$ ($i = 1, 2, 3$) in frame $o - xyz$ can be, therefore, written as

$$[\mathbf{A}_i]_{\mathcal{R}} = \mathbf{R}_1[\mathbf{A}_i]_{\mathcal{R}'} + [\mathbf{o}']_{\mathcal{R}}, \quad (i = 1, 2, 3) \quad (5)$$

Then the inverse kinematics of the parallel mechanism can be solved by writing following constraint equation

$$\|[\mathbf{B}_i]_{\mathcal{R}} - [\mathbf{A}_i]_{\mathcal{R}}\| = \|\mathbf{L}_i\| = l, \quad (i = 1, 2, 3) \quad (6)$$

Hence, for a given mechanism and prescribed position and orientation of the movable platform, the required actuator inputs can be directly computed from Eq. (6), that is

$$\begin{cases} z_1 = \sqrt{l^2 - A_{11}^2 - A_{12}^2} + A_{13} \\ z_2 = \sqrt{l^2 - A_{21}^2 - A_{22}^2} + A_{23} \\ z_3 = \sqrt{l^2 - A_{31}^2 - A_{32}^2} + A_{33} \end{cases} \quad (7)$$

where

$$\begin{aligned} A_{11} &= (R - r(c\beta - \sqrt{3}s\beta s\alpha))/2 \\ A_{12} &= -\sqrt{3}(R - r\alpha)/2 \\ A_{13} &= r(s\beta + \sqrt{3}c\beta s\alpha)/2 + z \\ A_{21} &= R - r(c\beta + \sqrt{3}s\beta s\alpha)/2 \\ A_{22} &= \sqrt{3}(R - r\alpha)/2 \\ A_{23} &= -r(s\beta - \sqrt{3}c\beta s\alpha)/2 + z \\ A_{31} &= -R + rc\beta \\ A_{32} &= 0 \\ A_{33} &= rs\beta + z \end{aligned}$$

2.2.2 Forward kinematics

The objective of the forward kinematics solution is to define a mapping from the known set of the actuated inputs to the unknown pose of the output platform. For the architecture with prismatic actuators, the inputs that are considered known are the lengths of the three actuator legs z_1 , z_2 and z_3 . The unknown pose of the output platform is described by the position vector $[\mathbf{o}']_{\mathcal{R}}$ and angles α and β . Because it is very difficult to describe the direct kinematics in closed form for this type of parallel mechanism, the forward kinematics solution should be obtained by numerical methodology as following:

1. Decide the non-singularity workspace of the mechanism;
2. Give the initial value of direct kinematics solution;

3. Calculate the position coordinates of spherical joints, construct the nonlinear equations set by the geometry constraint relationship of fixed length links;
4. Solve the nonlinear equations set by Quasi-Newton method (Press et al., 1995).

From the Eq. (6), the nonlinear equations are

$$f_i(z, \alpha, \beta) = l^2 - A_{i1}^2 - A_{i2}^2 - (z_i - A_{i3})^2 = 0, \quad (i = 1, 2, 3) \quad (8)$$

where the direct kinematics solutions are z , α and β .

2.2.3 Velocity equation

Eq. (6) can be differentiated with respect to time to obtain the velocity equation. This leads to an equation of the form.

$$\mathbf{J}_p \dot{\mathbf{p}} = \mathbf{J}_q \dot{\mathbf{q}} \quad (9)$$

where $\dot{\mathbf{q}}$ is the vector of Cartesian velocities defined as

$$\dot{\mathbf{q}} = [\dot{z}, \dot{\alpha}, \dot{\beta}]^T \quad (10)$$

and $\dot{\mathbf{p}}$ is the vector of input velocities defined as

$$\dot{\mathbf{p}} = [\dot{z}_1, \dot{z}_2, \dot{z}_2]^T \quad (11)$$

Matrices \mathbf{J}_p and \mathbf{J}_q are the 3×3 forward and inverse Jacobian matrices of the mechanism and can be expressed as

$$\mathbf{J}_p = \begin{bmatrix} (z_1 - A_{13})/l & 0 & 0 \\ 0 & (z_2 - A_{23})/l & 0 \\ 0 & 0 & (z_3 - A_{33})/l \end{bmatrix} \quad (12)$$

$$\mathbf{J}_q = \begin{bmatrix} (\mathbf{w}_1)_z & (\mathbf{v}_1 \times \mathbf{w}_1)_x & (\mathbf{v}_1 \times \mathbf{w}_1)_y \\ (\mathbf{w}_2)_z & (\mathbf{v}_2 \times \mathbf{w}_2)_x & (\mathbf{v}_2 \times \mathbf{w}_2)_y \\ (\mathbf{w}_3)_z & (\mathbf{v}_3 \times \mathbf{w}_3)_x & (\mathbf{v}_3 \times \mathbf{w}_3)_y \end{bmatrix} \quad (13)$$

where \mathbf{w}_i is the unit vector of \mathbf{L}_i , and $\mathbf{v}_i = \mathbf{R}_1[\mathbf{A}_i]_{\text{SR}}$. $(\mathbf{w}_i)_z$ is the element of vector \mathbf{w}_i with respect to z axis coordinate, $(\mathbf{v}_i \times \mathbf{w}_i)_x$ and $(\mathbf{v}_i \times \mathbf{w}_i)_y$ are the elements of vector $\mathbf{v}_i \times \mathbf{w}_i$ with respect to x and y axis coordinates.

2.3 Mechanism accuracy analysis

When the large spherical radio telescope works, the feed system will illuminate a working area, which is the paraboloid reflector with a three-hundred-meter aperture. The part of spherical reflector illuminated by the feed is continuously adjusted to fit a paraboloid of revolution in real-time, synchronous with the motion of the feed while tracking the object to

be observed. For the fitting, the spherical surface reflector is divided into some small elementary units. When the mechanisms drive the reflector units to fit the paraboloid, the fitting surface of reflector will not match exactly with the nominal paraboloid. Moreover, the mechanism has error because of the control or dimensional factor. In this section, the mechanism accuracy is analyzed firstly.

The mechanism accuracy involves the error caused by the actuator input error and the joint error of the mechanism. The actuator input error is denoted as $\delta p = [\delta z_1, \delta z_2, \delta z_3]^T$ and the joint error is denoted as $\delta e = [\delta A_i^T \quad \delta B_i^T]^T \in R^{18 \times 1} (i = 1, \dots, 3)$, where $\delta B_i^T \in R^{9 \times 1} (i = 1, \dots, 3)$ includes the joint error on the base platform and the input error $\delta p = [\delta z_1, \delta z_2, \delta z_3]^T$. The output error is denoted as $\delta q = [\delta z, \delta \alpha, \delta \beta]^T$.

From Eq. (5) and (6), the inverse kinematics equation can be written as

$$R_1 [A_i]_{\mathfrak{R}'} + [o']_{\mathfrak{R}'} - [B_i]_{\mathfrak{R}'} = L_i = w_i l \tag{14}$$

Differentiating Eq. (14) leads to

$$\delta l = J_q \delta q + J_e \delta e \tag{15}$$

where

$$J_e = \begin{bmatrix} w_1^T R_1 & -w_1^T & 0 & 0 & 0 & 0 \\ 0 & 0 & w_2^T R_1 & -w_2^T & 0 & 0 \\ 0 & 0 & 0 & 0 & w_3^T R_1 & -w_3^T \end{bmatrix} \in R^{3 \times 18} \tag{16}$$

and $\delta l = [\delta l_1, \delta l_2, \delta l_3]^T$, $\delta l_i (i = 1, 2, 3)$ is the manufacturing or measuring error of the i -th link. When J_q is nonsingular in the workspace, Eq. (15) can be rewritten as

$$\delta q = J_q^{-1} (\delta l - J_e \delta e) \tag{17}$$

3. Fitting accuracy analysis of active reflector

3.1 One dimensional fitting accuracy analysis

As shown in Fig. 4, the base active reflector of the radio telescope is a spherical surface with five-hundred-meter aperture, and the working reflector is a paraboloid with a three-hundred-meter aperture. When it works, the reflector units are driven by the parallel mechanism from the initial position to the fitting position to fit the paraboloid. Because the paraboloid is formed by the revolution of parabola, we can analyze the deviation about spherical surface and paraboloid in the reflector frame $\mathfrak{R}'' : o'' - y'' - z''$, which is built as shown in Fig. 4, where the spherical surface and the paraboloid in the frame \mathfrak{R}'' are circular arc and parabola, respectively.

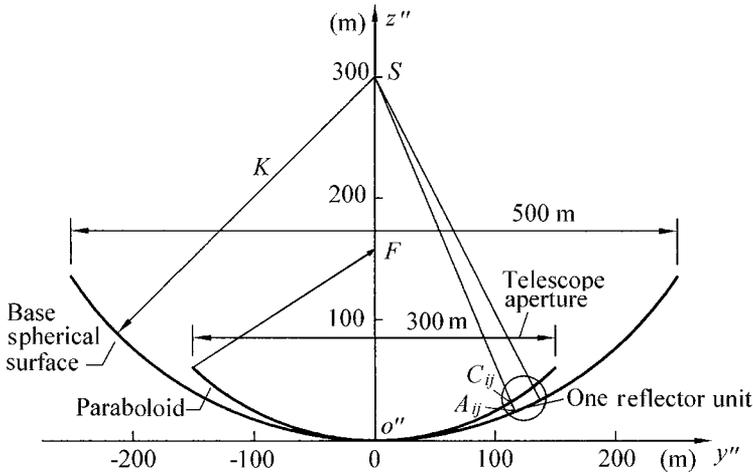


Fig. 4. Configuration of the active reflector

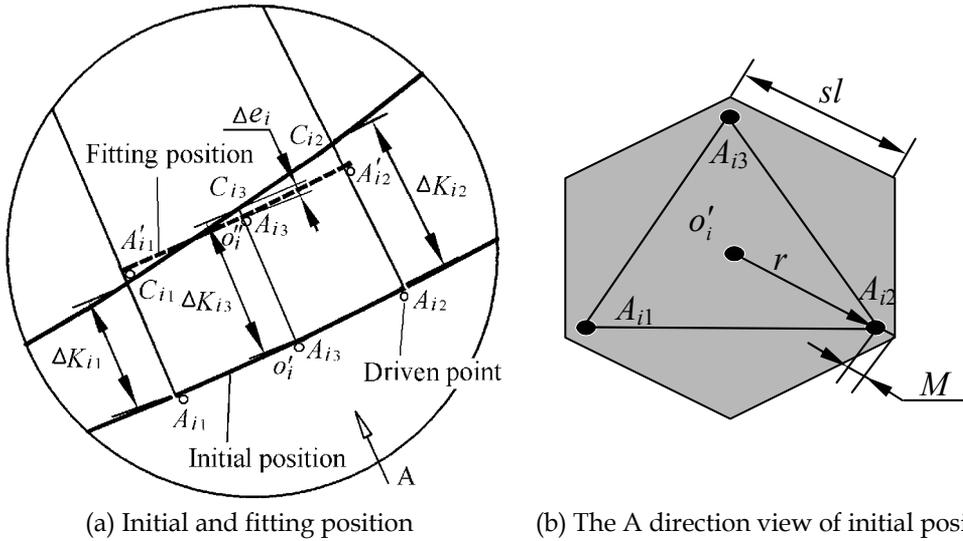


Fig. 5. The i -th reflector unit

Fig. 5 shows one reflector unit which is in the initial position and fitting position, respectively. The initial position is located at the base spherical reflector surface. The deviation from the circular arc to the parabola is denoted as ΔK_{ij} , and symbol i represents the i -th reflector unit which corresponds to the i -th mechanism. The symbol j ($j = 1, 2, 3$) represents the supporting point of the movable platform. The explanations of other symbols used in accuracy analysis are:

A_{ij} The supporting point while the reflector unit is in the initial position.

A'_{ij} The supporting point while the reflector unit is in the fitting position.

C_{ij} The intersecting points of line SA_{ij} and the parabola.

o'_i The reference center in the movable platform while the reflector unit is in the initial position.

o''_i The reference center in the movable platform while the reflector unit is in the fitting position.

S The center of spherical reflector.

K The radius of spherical reflector.

F The focal point of the paraboloid.

The absolute actuator input of the i -th mechanism is specified as ΔK_{ij} ($j=1,2,3$), while the i -th active reflector unit is driven to fit the paraboloid. Obviously, the driven reflector unit will not match exactly with the nominal paraboloid. In order to evaluate the fitting error, as shown in Fig. 5, Δe_i is defined as the center points deviation of the i -th reflector unit to the corresponding paraboloid and Δe_i is equal to $\|o''_i C_{i3}\|$, where the center points deviation Δe_i is called as one-dimensional fitting error.

3.1.1 Parabola equation and circle equation

According to the reference (Qiu 1998), the focal length of the parabola is specified as $0.476K$, then the parabola equation can be written as

$$z'' = \frac{1}{4 \times 0.467K} y''^2 \tag{18}$$

The base spherical surface in reflector coordinate system \mathfrak{R}'' is a circle. And the circle equation can be written as

$$z'' = K - \sqrt{K^2 - y''^2} \tag{19}$$

3.1.2 Actuator input range

The coordinate of the point A_{ij} in the frame \mathfrak{R}'' can be described by the vector $[A_{ij}]_{\mathfrak{R}''}$ ($j=1,2,3$), then

$$[A_{ij}]_{\mathfrak{R}''} = [y''_{ij}, z''_{ij}]^T, (j=1,2,3) \tag{20}$$

The equation of straight line SA_{ij} can be written as

$$z'' = \frac{(z''_{ij} - K)y''}{y''_{ij}} + K, (j=1,2,3) \tag{21}$$

According to Eqs. (19) and (21), the intersecting point C_{ij} between line SA_{ij} and the circle can be expressed by vector $[C_{ij}]_{\mathfrak{R}''}$, which is

$$[C_{ij}]_{\mathfrak{R}''} = [y''_{cij}, z''_{cij}]^T, (j=1,2,3) \tag{22}$$

Actuator input value of the i -th reflector unit can be written as

$$\Delta K_{ij} = \|K - SC_{ij}\| = K - \sqrt{(y''_{cij})^2 + (K - z''_{cij})^2}, \quad (j=1,2,3) \quad (23)$$

3.1.3 One-dimensional fitting error

When actuator input ΔK_{ij} , ($j=1,2,3$) is specified, the fitting error Δe_i can be reached. The first step is to calculate the position coordinate $[o''_i]_{\mathfrak{R}} = [y, z]_{\mathfrak{R}}^T$ in the frame \mathfrak{R} by the forward kinematics solution. The position vector of center point o''_i in the frame \mathfrak{R}'' is written as

$$[o''_i]_{\mathfrak{R}''} = [y''_{o''_i}, z''_{o''_i}]^T = \mathbf{R}_2 [o''_i]_{\mathfrak{R}} + [y''_{i3}, z''_{i3}]^T \quad (24)$$

where \mathbf{R}_2 is the rotation matrix about frame $\mathfrak{R} : o - yz$ to the frame $\mathfrak{R}'' : o'' - y''z''$, i.e.,

$$\mathbf{R}_2 = \begin{bmatrix} c\alpha' & -s\alpha' \\ s\alpha' & c\alpha' \end{bmatrix} \quad (25)$$

where $\alpha' = \sin^{-1}\left(\frac{y''}{K}\right)$. Then the fitting error is expressed as

$$\Delta e_i = So''_i - SC_{i3} = \sqrt{(y''_{o''_i})^2 + (z''_{o''_i} - K)^2} - \sqrt{(y''_{ci3})^2 + (z''_{ci3} - K)^2} \quad (26)$$

Since the three-hundred-meter aperture paraboloid is composed of a lot of reflector units, we should analysis all the error of reflector units. When the error is studied in the reflector frame $\mathfrak{R}'' : o'' - y''z''$ and the side length of reflector unit is specified, the one-dimensional root-mean-square (RMS) fitting error of the paraboloid reflector with three-hundred-meter aperture is defined as

$$Re = \sqrt{\frac{\sum_{i=1}^n \Delta e_i^2}{n}} \quad (27)$$

3.1.4 One-dimensional accuracy synthesis analysis

The accuracy synthesis analysis is defined as the composition RMS error that caused by the mechanism actuator input error and the fitting error. For the mechanism actuator input error has linear relationship with the value of ΔK_{ij} , Eq. (23) can be rewritten as

$$\Delta K_{ij} = \|K - SC_{ij}\| + \delta z_j = K - \sqrt{(y''_{cij})^2 + (z''_{cij} - K)^2} + \delta z_j, \quad (j=1,2,3) \quad (28)$$

where δz_j ($j=1,2,3$) is the actuator input error. Then Eqs. (24)- (27) can be used to calculate the one-dimensional composition RMS error $R'e$.

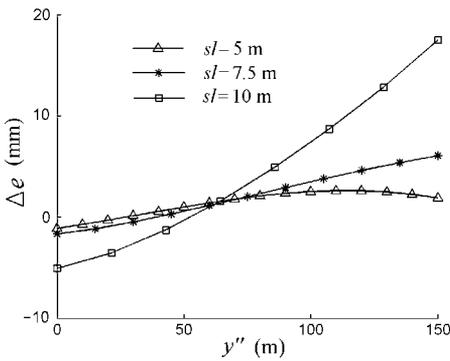
3.1.5 Simulation example

Since the position of supporting point A_{ij} should be limited in the range of reflector unit, as shown in Fig. 5(b), the base plate parameter r of the parallel mechanism is written as

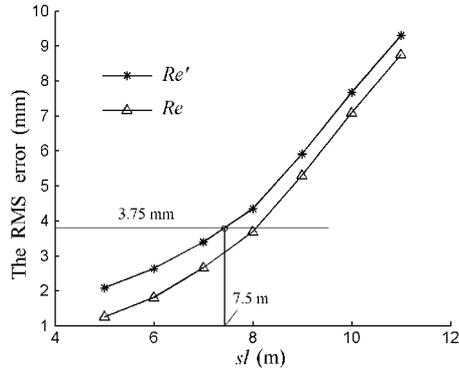
$$r = sl - M \tag{29}$$

where sl is the side length of reflector unit and M is the distance from the movable platform edge to reflector unit edge. In this work, $M = 0.5$ m and $R = r = 2l$.

Since the paraboloid reflector with the 300m aperture is symmetry, the error can be analyzed in the range of 150m. Fig. 6(a) shows the one-dimensional fitting error when the side length of reflector unit is specified.



(a) The fitting error of reflector unit



(b) The one-dimensional RMS fitting error

Fig. 6. The fitting accuracy of active reflector

According to Eqs. (26) and (28), the one-dimensional RMS fitting error and composition RMS error can be drawn as shown in Fig. 6(b) for $[\delta z_1, \delta z_2, \delta z_3]_{\max} = [1, 1, 1]$ mm. In this work, we assume that the maximal input error of the mechanism is 1 mm. When the side length of reflector unit is equal to 7.5m, the one-dimensional RMS fitting error is 3.75 mm.

3.2 Two-dimensional fitting accuracy analysis

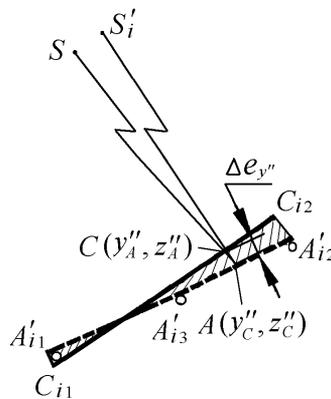


Fig. 7. Two-dimensional fitting error region of the i -th reflector unit

As shown in Fig. 5(a), the reflector unit fitting error is a closed region. And section 3.1 only considered the one-dimensional error. In this section, the area of the closed region will be used to analyze and evaluate the fitting error, which is called as two-dimensional fitting error. Obviously, the two-dimensional fitting error will provide more reliable index for us to analyzing the fitting accuracy of the large spherical radio telescope. Fig. 7 shows the two-dimensional fitting error of the i -th reflector unit, which is the sectional region.

3.2.1 Fitting position circle equation

When the reflector units are driven by mechanism, the circle arc equation of the i -th reflector unit will be changed in the frame \mathfrak{R}'' . As shown in Fig. 7, the centre of circle arc $A'_1A'_3A'_2$ is changed from S to S'_i . The coordinate of S'_i in the frame \mathfrak{R} is written as

$$[S'_i]_{\mathfrak{R}} = [y_{S'_i}, z_{S'_i}]^T = R_1 [S]_{\mathfrak{R}} + [y, z]^T \quad (30)$$

where

$$R_1 = \begin{bmatrix} c\alpha & s\alpha \\ s\alpha & c\alpha \end{bmatrix}, [S]_{\mathfrak{R}} = [0, K] \quad (31)$$

The coordinate of S'_i in the frame \mathfrak{R}'' is written as

$$[S'_i]_{\mathfrak{R}''} = [y''_{S'_i}, z''_{S'_i}]^T = R_2 [S'_i]_{\mathfrak{R}} + [y''_{i3}, z''_{i3}]^T \quad (32)$$

The circle arc equation is changed to

$$z'' = K - \sqrt{K^2 - (y'' - y_{S'_i})^2} + z''_{S'_i} \quad (33)$$

3.2.2 Two-dimensional fitting error

According to the circle arc equation in the frame \mathfrak{R}'' , the two-dimensional fitting error can be calculated. Firstly, as shown in Fig. 7, one point in parabola is denoted as $C(y''_c, z''_c)$, The equation of straight line SC can be written as

$$z'' = \frac{(z''_c - K)y''}{y''_c} + K \quad (34)$$

According to Eqs. (33) and (34), the intersecting point A between line SC and the circle arc $A'_1A'_3A'_2$ can be expressed by $A(y''_A, z''_A)$. The fitting error of given point is expressed as

$$\Delta e_{y''} = SA - SC = \sqrt{(y''_c - y''_A)^2 + (z''_c - z''_A)^2} \quad (35)$$

The area of the closed region can be written as

$$Se_i = \int_{y''_{c1}}^{y''_{c2}} |\Delta e_{y''}| dy'' \quad (36)$$

which is the two-dimensional fitting error of the i -th reflector unit. Then the average error of the two-dimensional fitting error is defined as

$$Qe_i = \frac{Se_i}{y''_{Ci2} - y''_{Ci1}} \tag{37}$$

In the end, the two-dimensional root-mean-square (RMS) fitting error of the paraboloid reflector with three-hundred-meter aperture is defined as

$$R_{Qe} = \sqrt{\frac{\sum_{i=1}^n Qe_i^2}{n}} \tag{38}$$

where n is the number of reflector units that consist of three-hundred meter aperture parabola.

3.2.3 Two-dimensional accuracy synthesis analysis

The two-dimensional accuracy synthesis analysis is defined as the composition RMS error that caused by the mechanism actuator input error and the two-dimensional fitting error, which is denoted as R'_{Qe} . The Eqs. (28), (37) and (38) can be used to calculate the two-dimensional composition RMS fitting error R'_{Qe} .

3.2.4 Simulation example

The two-dimensional RMS error and composition RMS fitting error are shown in Fig. 8, where all the dimensional design parameters are the same as the specified parameters in section 3.1.5. Comparing the Fig. 8 and Fig. 7, we can know although both the one-dimensional and two-dimensional RMS fitting error increase while the side length of reflector unit increases, the two-dimensional RMS error is larger than one-dimensional RMS error.

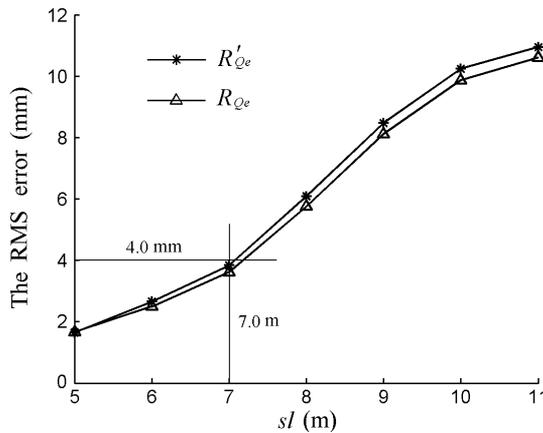


Fig. 8. The two-dimensional RMS fitting error of active reflector

According to the fitting error requirement given by reference (Qiu, 1998), when the highest working frequency of the radio telescope is 5GHz, the reflector RMS fitting error should be less than 3.75 mm. Now, we can decide the dimensional parameters and guarantee the implementation of the working frequency by the one-dimensional or two-dimensional RMS fitting error curves. For example, according to the Fig. 7, if the side length of reflector unit is specified as 7.5m, the specified dimension of reflector units can satisfy the requirement of 5GHz work frequency. However, as shown in Fig. 8, if the two-dimensional RMS fitting error is used to evaluate the fitting accuracy, the side length of reflector unit should be less than 7.0m for satisfying the requirement of 5GHz work frequency.

4. The error kinematics of 3-PRS mechanism

3-PRS mechanism has less chains which reduces its cost. The kinematics of 3-PRS mechanism has been fully analyzed (Carretero et al., 1997; Tsai & Shiau, 2003). Yet, when the mechanical manufacturing and assembling errors are brought into the model, kinematic analysis will become complicated. Therefore, analysis on parasitic motion and accuracy should be made to guarantee the application of 3-PRS mechanism as reflector unit supporting mechanism.

4.1 Kinematic modeling with errors

The magnitude of the reflector driving machine is always at meter, so input error, length error of the legs and location error of the spherical joint have little influence on motion error of the moving platform. On the other hand, the location and angle error of the rotational joint, which will be extended by the legs, will mix with parasitic motion so as to greatly affect the motion. Therefore, we introduce angle error of the rotational axis and location error of the joint point in the rotational joint as the main error resources in order to analyze kinematics of 3-PRS error model.

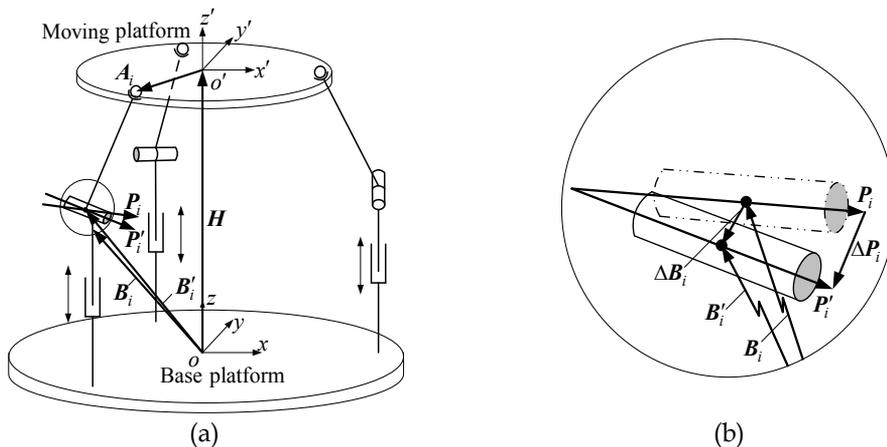


Fig. 9. Kinematic error model of 3-PRS mechanism

In the error model representation of 3-PRS mechanism, as shown in Fig. 9(a), P_i is ideal axis vector of the rotational joint, and P'_i is actual axis vector with angle error. Similarly, B_i

is ideal vector of the rotational joint point, whereas \mathbf{B}'_i is actual joint point vector with location error. Both \mathbf{P}'_i and \mathbf{B}'_i include three direction errors separately along x , y , and z axis, which means that there are six errors in each leg, as shown in Fig. 9(b), in which two-dot chain line represents ideal rotational joint and real line represents actual one.

The location error vector of the rotational joint is defined as

$$\Delta \mathbf{B}_i = [\Delta b_{ix} \quad \Delta b_{iy} \quad \Delta b_{iz}]^T$$

The angle error vector of the rotational joint is defined as

$$\Delta \mathbf{P}_i = [\Delta p_{ix} \quad \Delta p_{iy} \quad \Delta p_{iz}]^T$$

Then we can find

$$\mathbf{B}'_i = \mathbf{B}_i + \Delta \mathbf{B}_i \tag{39}$$

$$\mathbf{P}'_i = \mathbf{P}_i + \Delta \mathbf{P}_i \tag{40}$$

The three components of the vector $\Delta \mathbf{B}_i$ are independent while those of the vector $\Delta \mathbf{P}_i$ are not since the error on the direction can be given through two parameters only. So the relationship between the components of the vector $\Delta \mathbf{P}_i$ can be determined by $\|\mathbf{P}_i + \Delta \mathbf{P}_i\| = 1$, where $\|\mathbf{P}_i\| = \|\mathbf{P}'_i\| = 1$ are unit direction vectors. Thus, error resources are appropriately introduced and error modeling of 3-PRS mechanism is completed.

4.2 Inverse kinematics

The coordinate axes of the inertial frame fixed on the base platform are denoted by $\mathfrak{R} : o - xyz$ while those of the moving frame fixed on the moving platform are denoted by $\mathfrak{R}' : o' - x'y'z'$ (see Fig. 9). In order to simplify the kinematic model, the origin of the inertial frame is located on the center of the base platform and x axis of the inertial frame points to one of the spherical joint on the base platform. The y axis is also on the plane of the base platform while z axis points upward forming a right-handed orthogonal frame. The coordinate axes of the moving frame are also located on the moving platform in the same way. The rotation matrix from the coordinate axes of the moving platform to those of the base platform is denoted by \mathbf{R} which is expressed as

$$\mathbf{R} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}$$

where ψ , θ , and ϕ are variables which orderly specify the rotations around the x , y , and z axis, and s represents sin, while c represents cos.

$[\mathbf{H}]_{\mathfrak{R}} = [x \quad y \quad z]^T$ is the vector from the origin of the inertial frame to the origin of the moving frame expressed in the inertial frame.

$[\mathbf{B}_i]_{\mathcal{R}}$ is the vector \mathbf{B}_i expressed in the inertial frame.

$[\mathbf{B}'_i]_{\mathcal{R}}$ is the vector \mathbf{B}'_i expressed in the inertial frame.

$[\mathbf{P}_i]_{\mathcal{R}} = [p_{i1} \ p_{i2} \ 0]^T$ is the vector \mathbf{P}_i expressed in the inertial frame.

$[\mathbf{P}'_i]_{\mathcal{R}}$ is the vector \mathbf{P}'_i expressed in the inertial frame.

$[\mathbf{A}_i]_{\mathcal{R}'} = [a_{i1} \ a_{i2} \ 0]^T$ is the vector from the origin of the moving frame to the i -th upper attachment point expressed in the moving frame.

$[\mathbf{L}_i]_{\mathcal{R}}$ is the vector from the rotational point to the upper attachment point of the i -th leg. It should be noted that $\|[\mathbf{L}_i]_{\mathcal{R}}\| = l_i$ is constant for each leg.

$\mathbf{S} = (S_1, S_2, S_3)^T$ is the set of actuated joint variable of the 3-PRS mechanism which is the height of the rotational joint point. We can get

$$[\mathbf{B}_i]_{\mathcal{R}} = [b_{i1} \ b_{i2} \ S_i]^T$$

The inverse kinematic problem is supposed to determine the value of the actuated variables for a known position and orientation of the end-effector, that is: $\mathbf{S} = \mathbf{f}(x, y, z, \psi, \theta, \phi)$.

In those six variables, the known numbers are three desired motions which include z , ψ , and θ , while the unknown numbers are three parasitic motions which include x , y , and ϕ . The parasitic motions are determined by the target motions, that is: $(x, y, \phi) = \mathbf{g}(z, \psi, \theta)$.

The structure of mechanical joint leads to two geometrical constraints which are rotation constraint and length limitation of the leg.

(a) The rotation constraint

Each attachment point of the moving platform should be restricted in the rotation plane formed by the wheeling leg. The constraint equations are

$$[\mathbf{P}'_i]_{\mathcal{R}}^T (\mathbf{R}[\mathbf{A}_i]_{\mathcal{R}'} + [\mathbf{H}]_{\mathcal{R}}) + C_i = 0, i = 1, 2, 3 \quad (41)$$

where C_i is a constant of the rotation plane and determined by the following equation

$$[\mathbf{P}'_i]_{\mathcal{R}}^T [\mathbf{B}'_i]_{\mathcal{R}} + C_i = 0, i = 1, 2, 3 \quad (42)$$

Substituting Eq. (39) into Eq. (42)

$$([\mathbf{P}_i]_{\mathcal{R}} + [\Delta\mathbf{P}_i]_{\mathcal{R}})^T ([\mathbf{B}_i]_{\mathcal{R}} + [\Delta\mathbf{B}_i]_{\mathcal{R}}) + C_i = 0 \quad (43)$$

Expressing with the elements of those vectors, we get

$$m_i S_i + C_i + n_i = 0 \quad (44)$$

where

$$m_i = \Delta p_{iz}, \quad n_i = p_{i1} b_{i1} + p_{i2} b_{i2} + \Delta p_{ix} b_{i1} + \Delta p_{iy} b_{i2} + p_{i1} \Delta b_{ix} + p_{i2} \Delta b_{iy} + \Delta p_{ix} \Delta b_{ix} + \Delta p_{iy} \Delta b_{iy} + \Delta p_{iz} \Delta b_{iz}$$

Substituting Eq. (40) into Eq. (41)

$$([\mathbf{P}_i]_{\text{gr}} + [\Delta\mathbf{P}_i]_{\text{gr}})^T (\mathbf{R}[\mathbf{A}_i]_{\text{gr}'} + [\mathbf{H}]_{\text{gr}}) + C_i = 0 \quad (45)$$

Expressing with the elements of those vectors, we get

$$f_i c\phi + g_i s\phi + h_i x + k_i y + j_i + C_i = 0 \quad (46)$$

where

$$\begin{aligned} f_i &= a_{i1} p_{i1} c\theta + a_{i2} p_{i1} s\theta s\psi + a_{i2} p_{i2} c\psi + a_{i1} \Delta p_{ix} c\theta + a_{i2} \Delta p_{ix} s\theta s\psi + a_{i2} \Delta p_{iy} c\psi \\ g_i &= -a_{i2} p_{i1} c\psi + a_{i1} p_{i2} c\theta + a_{i2} p_{i2} s\theta s\psi - a_{i2} \Delta p_{ix} c\psi + a_{i1} \Delta p_{iy} c\theta + a_{i2} \Delta p_{iy} s\theta s\psi \\ h_i &= p_{i1} + \Delta p_{ix}, k_i = p_{i2} + \Delta p_{iy}, j_i = \Delta p_{iz} (-a_{i1} s\theta + a_{i2} c\theta s\psi + z) \end{aligned}$$

(b) The leg limitation

The distance between the attachment point of the moving platform and the rotational point of the rotational joint should be constant. The constraint equations are

$$\|\mathbf{R}[\mathbf{A}_i]_{\text{gr}'} + [\mathbf{H}]_{\text{gr}} - [\mathbf{B}'_i]_{\text{gr}}\| = \|\mathbf{L}_i\| = l_i, i = 1, 2, 3 \quad (47)$$

Substitute Eq. (39) into Eq. (47)

$$\|\mathbf{R}[\mathbf{A}_i]_{\text{gr}'} + [\mathbf{H}]_{\text{gr}} - ([\mathbf{B}_i]_{\text{gr}} + [\Delta\mathbf{B}_i]_{\text{gr}})\| = l_i \quad (48)$$

Expressing with the elements of those vectors, we get

$$S_i = \pm \sqrt{l_i^2 - u_i^2 - v_i^2} + w_i \quad (49)$$

where

$$\begin{aligned} u_i &= a_{i1} c\phi c\theta + a_{i2} c\phi s\theta s\psi - a_{i2} s\phi c\psi + x - b_{i1} - \Delta b_{ix} \\ v_i &= a_{i1} s\phi c\theta + a_{i2} s\phi s\theta s\psi + a_{i2} c\phi c\psi + y - b_{i2} - \Delta b_{iy} \\ w_i &= -a_{i1} s\theta + a_{i2} c\theta s\psi + z - \Delta b_{iz} \end{aligned}$$

In Eq. (49), there are two possible solutions for each leg, thereby yielding a total of 8 possible combinations of actuated height for a given position and orientation. In the present work, the negative square root is always selected to yield a solution where the legs are always beneath the moving platform.

Combining Eqs. (44) and (46), we get

$$f_i c\phi + g_i s\phi + h_i x + k_i y + j_i - m_i S_i - n_i = 0 \quad (50)$$

Combining Eqs. (49) and (50), we get equations set involved with variables S_i and three parasitic motions. We can find that if $m_i = \Delta p_{iz} = 0$, Eqs. (49) and (50) can be solved separately. But in ordinary condition, $m_i = \Delta p_{iz} \neq 0$, direct solution of the equations set will become impossible.

4.3 Arithmetic of inverse solution

In order to figure out the nonlinear equations set with the normal condition of $\Delta p_{iz} \neq 0$, a numerical iterative arithmetic is proposed as follows:

1. Decide the initial value of the actuated height S_0 and set the loop variable $i = 0$;
2. Calculate the parasitic motions of the moving platform according to Eq. (50) with S_i ;
3. Then calculate S_{i+1} according to Eq. (49);
4. If $\|S_{i+1} - S_i\| < \varepsilon$, where ε is the acceptable error limitation, end up iterative calculation with solution of S_{i+1} . Otherwise, $i = i + 1$, then turn back to step 2.

4.4 Forward kinematics

Forward kinematic solution is supposed to determine the position and orientation of the end-effector for known actuated variables. Since we already have the inverse kinematics, Similarly with the method used in section 2.2.2, we can also get forward solution.

5. Three-dimensional fitting accuracy analysis

5.1 Coordinate description for calculation

As shown in Fig. 10(a), the paraboloid of revolution covers a spherical surface with a 300-meter aperture, which has a coning angle of 60 degree. The paraboloid moves on the sphere to track the object in real-time. Since the axial line of the desired paraboloid always orients to the center of the base spherical surface when it works, the fitting accuracy is constant at any time when the paraboloid of revolution is at any location on the base spherical surface. Thus analysis of Fig. 10(a), where the peak of the paraboloid of revolution is located on the bottom of the base spherical surface, will be enough.

In Fig. 10(a), the global coordinate axes of the inertial frame fixed on the whole active reflector system are denoted by $\mathcal{R}^g : o^g - x^g y^g z^g$. The origin of the coordinate system \mathcal{R}^g is located on the bottom of the base sphere while the $x^g y^g z^g$ plane is the tangent plane of the point o^g .

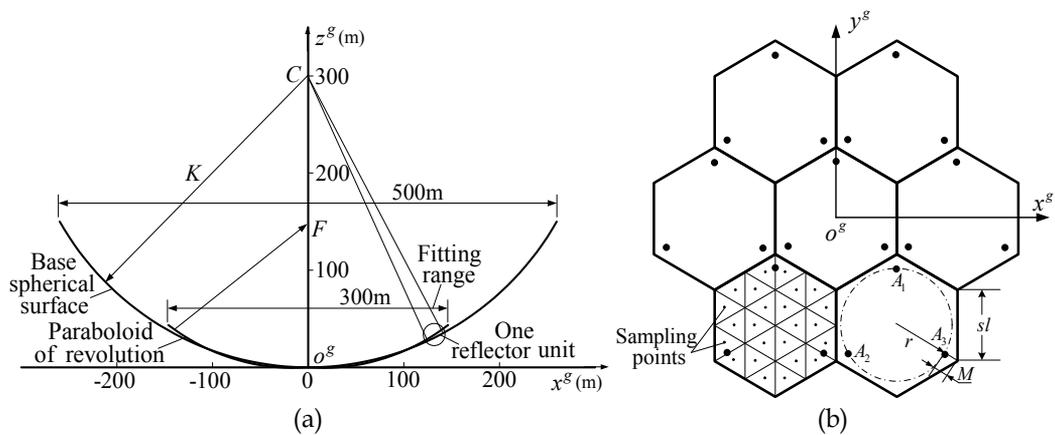


Fig. 10. Active spherical reflector and fitting paraboloid: (a) profile; (b) top view

Fig. 10(b) is a top view of base sphere along radial direction. The x^g axis is perpendicular with the side of the regular hexagon reflector unit. Fig. 10(b) shows the location of the three attachment points and twenty four sampling points for calculation. It is noted that the upper surface of the reflector unit is spherical which means that the sampling points are all on a spherical surface whose radius is K .

As shown in Fig. 11, the coordinate axes of the inertial frame fixed on the platform in initial position are denoted by $\mathcal{R} : o - xyz$, in which the location of the axes is similar with the location in the kinematic analyses introduced in section 4.1. The inertial frame is not moving with the reflector unit. Since the reflector unit in initial position is the tangent plane of the base sphere, z axis always points to the center of the sphere. It should be noticed that the directions of x axis and y axis in the coordinate frame $o - xyz$ is different from those in $o^g - x^g y^g z^g$. Similarly, the coordinate axes of the moving frame fixed on the platform in fitting position and orientation are denoted by $\mathcal{R}' : o' - x' y' z'$. The coordinate frame \mathcal{R} and \mathcal{R}' are similar with the kinematic inertial and moving frame analyzed in section 4 so that the inverse and forward solution can be used.

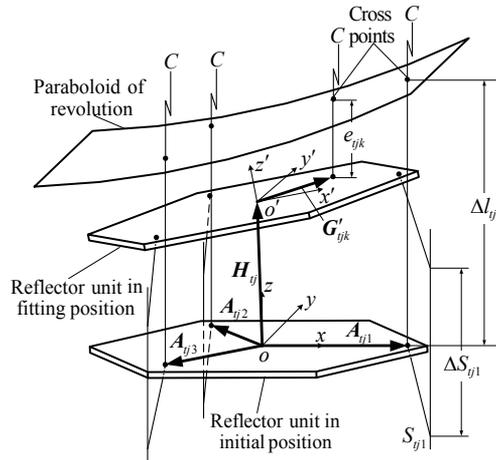


Fig. 11. One reflector unit in initial and fitting position

We can divide the surface by several circular arcs. These arcs are intersections of the base sphere and the planes which are parallel with $x^g y^g z^g$ plane, and are all through the centers of the reflector units. So the fitting accuracy of all reflector units can be calculated along these arcs.

For analyses, the symbols used in the Fig. 10 and 11 are defined as

A_i is the i -th attachment on the reflector unit.

C is the center of the base sphere.

K is the radius of the base sphere.

F is the focal point of the ideal paraboloid of revolution.

sl is the side length of the reflector unit.

M is the distance from each attachment to the border of the reflector unit on radial direction.

$[A_{tj}]_{\mathfrak{R}}$ is the vector from the origin of \mathfrak{R} to A_i on the (t, j) reflector unit in the initial position expressed in the inertial frame \mathfrak{R} (the subscript t, j mean the j -th reflector unit on the t -th circular arc, the same as below).

$[A_{tj}]_{\mathfrak{R}^g}$ is the vector from the origin of \mathfrak{R}^g to A_i in the initial position expressed in the global inertial frame \mathfrak{R}^g .

$[G_{tjk}]_{\mathfrak{R}'}$ is the vector from o' to the k -th sampling point in the fitting position expressed in the moving frame \mathfrak{R}' , so $[G_{tjk}]_{\mathfrak{R}'}$ is constant and known.

$[G_{tjk}]_{\mathfrak{R}}$ is the vector from o to the k -th sampling point in the fitting position expressed in the inertial frame \mathfrak{R} .

$[G_{tjk}]_{\mathfrak{R}^g}$ is the vector from o^g to the k -th sampling point in the fitting position expressed in the inertial frame \mathfrak{R}^g .

$[H_{tj}]_{\mathfrak{R}^g}$ is the vector from o^g to o expressed in the inertial frame \mathfrak{R}^g which is determine by the position of current analyzing reflector unit.

$[H_{tj}]_{\mathfrak{R}} = (x, y, z)^T$ is the vector from o to o' expressed in the inertial frame \mathfrak{R} .

R_{tj} is the rotation matrix from coordinate frame \mathfrak{R} to \mathfrak{R}^g .

R'_{tj} is the rotation matrix from coordinate frame \mathfrak{R}' to \mathfrak{R} .

$S_{tj} = (S_{tj1}, S_{tj2}, S_{tj3})^T$ is the actuated joint variable.

ΔS_{tji} is the increment of the actuated variable of the i -th leg from the initial position to the fitting position.

Δl_{tji} is the distance between the i -th attachment in the initial position and the paraboloid of revolution on radial direction.

$f(\psi, \theta, z)$ is inverse solution which is analyzed in section 4 with output of $(S_1, S_2, S_3)^T$.

$f^{-1}(S_1, S_2, S_3)$ is forward solution with output of $(x, y, z, \psi, \theta, \phi)^T$.

5.2 Paraboloid equation and circle equation

The focal length of the paraboloid is specified as $0.467K$, then the paraboloid equation can be written as

$$z^g = \frac{1}{4 \times 0.467K} (x^{g2} + y^{g2})$$

The distance from one point $(x_0^g, y_0^g, z_0^g)^T$ to the paraboloid on radial direction is

$$\Delta l = \left\| (x_0^g, y_0^g, z_0^g)^T - (x^g, y^g, z^g)^T \right\|$$

where

$$z^g = K + \frac{2(z_0^g - K)^2 \times 0.467K}{x_0^{g2} + y_0^{g2}} - \sqrt{\left(K + \frac{2(z_0^g - K)^2 \times 0.467K}{x_0^{g2} + y_0^{g2}}\right)^2 - K^2}$$

$$x^g = \frac{z^g - K}{z_0^g - K} x_0^g$$

$$y^g = \frac{z^g - K}{z_0^g - K} y_0^g$$

By combining the equations above, we define the function to calculate distance between one point $(x_0^g, y_0^g, z_0^g)^T$ and the paraboloid on radial direction as

$$dis((x_0^g, y_0^g, z_0^g)^T) = \Delta l$$

5.3 Driving strategy

Driving strategy determines the method to drive the reflector unit to fit for paraboloid. In order to simplify calculation, the actuated variable S_{ij} is the value of actuated variable in initial position plus Δl_{iji} .

5.4 Fitting accuracy calculation

Calculating the number of the arcs in the half base sphere of positive y^g axis surface, we get:

$$m = \text{ceil}\left(\frac{K\pi}{9sl}\right) \quad (51)$$

where $\text{ceil}(x)$ means the least integer which is no less than x .

The radius of the t -th arc is written as

$$K_t = K \cos(3t \cdot sl/2K) \quad (52)$$

The meeting point of the t -th arc to the border of the base sphere is written as

$$x_t^g = \sqrt{(K/2)^2 - (K \sin(3t \cdot sl/2K))^2} \quad (53)$$

Thus the length of the t -th positive half circular arc is written as

$$al_t = K_t \text{arc sin}(x_t^g / K_t) \quad (54)$$

So the number of reflector units on the positive half t -th arc can be obtained as

$$n_t = \text{ceil}(al_t / \sqrt{3}sl) \quad (55)$$

First, we focus on the coordinate frame \mathfrak{R} and \mathfrak{R}^g . The equation of the base sphere is

$$x^{g2} + y^{g2} + (z^g - K)^2 = K^2$$

or

$$z^g = K - \sqrt{K^2 - x^{g2} - y^{g2}}$$

So that along the arc of even number, the position of the j -th reflector unit on the t -th arc can be written as

$$x_{ij}^g = K_t \sin(\sqrt{3}j \cdot sl / K_t), \quad y_{ij}^g = K \sin(3t \cdot sl / 2K), \quad z_{ij}^g = K - \sqrt{K^2 - x_{ij}^{g2} - y_{ij}^{g2}}$$

While along the arc of odd number, the x_{ij}^g is

$$x_{ij}^g = K_t \sin(\sqrt{3}(j + 0.5)sl / K_t)$$

Thus position of the coordinate frame \mathfrak{R} which is fixed on the reflector unit in the initial position is determined by

$$[\mathbf{H}_{ij}]_{\mathfrak{R}^g} = (x_{ij}^g, y_{ij}^g, z_{ij}^g)^T \quad (56)$$

According to the orientation of the coordinate frame \mathfrak{R} , the rotation matrix to \mathfrak{R}^g can be written as

$$\mathbf{R}_{ij} = Rot(\mathbf{p}_{ij}, \arcsin(\frac{\sqrt{x_{ij}^{g2} + y_{ij}^{g2}}}{K}))Rot(z, 90) \quad (57)$$

where $Rot(a, b)$ is rotation matrix of rotating angle of b degree around vector a , and \mathbf{p}_{ij} is rotating vector in $o^g x^g y^g$ plane which is expressed as

$$\mathbf{p}_{ij} = (y_{ij}^g / \sqrt{x_{ij}^{g2} + y_{ij}^{g2}}, -x_{ij}^g / \sqrt{x_{ij}^{g2} + y_{ij}^{g2}}, 0)^T$$

The vector of sampling point can be written as

$$[\mathbf{G}_{ijk}]_{\mathfrak{R}^g} = \mathbf{R}_{ij} [\mathbf{G}_{ijk}]_{\mathfrak{R}} + [\mathbf{H}_{ij}]_{\mathfrak{R}^g} \quad (58)$$

while the vector of the upper attachment in the initial position can be written as

$$[\mathbf{A}_{tji}]_{\mathfrak{R}^g} = \mathbf{R}_{ij} [\mathbf{A}_{tji}]_{\mathfrak{R}} + [\mathbf{H}_{ij}]_{\mathfrak{R}^g} \quad (59)$$

Then, we focus on the coordinate frame \mathfrak{R}' and \mathfrak{R} . The value of actuated variable should be determined for the (t, j) unit. According to the driving strategy, the actuated variable of the reflector unit in fitting position can be obtained as

$$\mathbf{S}_{ij} = \mathbf{S}_0 + (\Delta l_{j1}, \Delta l_{j2}, \Delta l_{j3})^T \quad (60)$$

where $\mathbf{S}_0 = \mathbf{f}(0,0,0)$ is the initial value of the actuated variable and $\Delta l_{ji} = \text{dis}(\mathbf{A}_{ji}]_{\mathfrak{R}^g}$.

According to forward solution, the Cartesian variables which specify the position and orientation from the coordinate frame \mathfrak{R}' to the inertial frame \mathfrak{R} can be obtained as

$$(x, y, z, \psi, \theta, \phi)^T = \mathbf{f}^{-1}(\mathbf{S}_{ij}) \tag{61}$$

$$[\mathbf{H}_{ij}]_{\mathfrak{R}} = (x, y, z)^T \tag{62}$$

$$\mathbf{R}'_{ij} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi c\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} \tag{63}$$

The vector of the k -th sampling point expressed in inertial frame \mathfrak{R} can be obtained by

$$[\mathbf{G}_{ijk}]_{\mathfrak{R}} = \mathbf{R}'_{ij} [\mathbf{G}_{ijk}]_{\mathfrak{R}'} + [\mathbf{H}_{ij}]_{\mathfrak{R}} \tag{64}$$

Finally, we calculate the fitting accuracy based on the results obtained above. The fitting accuracy of one sampling point on the j -th reflector unit in the t -th arc can be written as

$$e_{ijk} = \text{dis}(\mathbf{G}_{ijk}^g) \tag{65}$$

Synthetically, substituting Eqs. (56)-(64) into (65), we can calculate e_{ijk} with specified t, j and k as

$$e_{ijk} = \text{dis}(\text{Rot}(\mathbf{p}_{ij}, \arcsin(\frac{\sqrt{x_{ij}^{g2} + y_{ij}^{g2}}}{K}))\text{Rot}(z, 90)(\mathbf{R}'_{ij} [\mathbf{G}_{ijk}]_{\mathfrak{R}'} + [\mathbf{H}_{ij}]_{\mathfrak{R}}) + (x_{ij}^g, y_{ij}^g, z_{ij}^g)^T)$$

The RMS fitting accuracy on the whole fitting surface can be written as

$$E_{RMS} = \sqrt{\left(\sum_{t=-m}^m \sum_{j=-n}^n \sum_{k=1}^{24} e_{ijk}^2 \right) / \left(24 \sum_{t=-m}^m (2n_t + 1) \right)}$$

5.5 Driving strategy optimization

In order to reduce the RMS fitting accuracy of the whole reflector, the current driving strategy should be optimized. Considering the real-time compensation, the optimization algorithm should be no more difficult than the current strategy.

According to the analyses above, we put forward one modified driving strategy as

$$\mathbf{S}_{ij} = \mathbf{S}_0 + (\Delta l_{ij1}, \Delta l_{ij2}, \Delta l_{ij3})^T + (3\text{dis}([\mathbf{H}_{ij}]_{\mathfrak{R}^g}) - \Delta l_{ij1} - \Delta l_{ij2} - \Delta l_{ij3})/6 \tag{66}$$

The strategy will compensate the actuated variable of each reflector unit with the algebraic average among three supporting points and the center of the reflector so as to get less RMS fitting accuracy on each unit spherical surface.

5.6 Simulation example

Fig. 12(a) shows the fitting accuracy on the whole reflector range with and without optimized driving strategy when the side length is changeable, which are respectively expressed as E_{RMS}^O and E_{RMS} . All the dimensional design parameters are the same as the specified parameters in section 3.1.5.

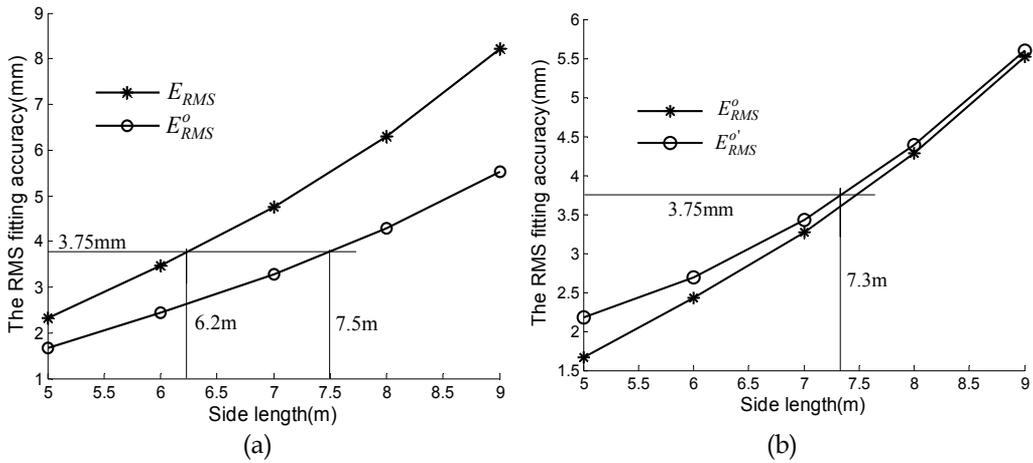


Fig. 12. The three-dimensional fitting accuracy of active reflector

As shown in Fig. 12(a), the fitting accuracy is reduced approximately by 40% when the driving strategy is optimized. Fig. 12(b) shows the fitting accuracy on the whole reflector range with and without errors when the optimized driving strategy is used, which are respectively expressed as E_{RMS}^O' and E_{RMS}^O . In this work, we assume that the axis angle tolerance of the rotational joint is $[0, 0.2^\circ]$ and the position tolerance is $[-10mm, 10mm]$, and E_{RMS}^O' is the worst situation with these tolerances. In order to guarantee the working frequency of the large radio telescope, the side length of the reflector unit should be less than 7.3mm.

6. Conclusion and future works

In order to guarantee the usage of active reflector and achieve the highest working frequency requirement, 5GHz, in FAST, fitting accuracy of the active reflector is supposed to be analyzed. In this chapter, a novel 3 DOFs parallel mechanism, 3-PSS with constraint leg, is proposed. This mechanism can fulfill the required movement to fit a paraboloid of revolution for the active reflector. The kinematics of 3-PSS+C mechanism is studied. Based on that, the one and two-dimensional fitting accuracy are calculated and the side length

limit of reflector units is evaluated as 7.0m. However, due to more expensive cost of the extra constraint chain, 3-PSS+C is not very appropriate as reflector supporting mechanism. So 3-PRS mechanism becomes more attractive and deserves to pay more attention. Then error kinematics with rotational joint tolerance is analyzed for actual application. Based on that, three-dimensional fitting accuracy is calculated with optimized driving strategy, and the side length limit turns out to be 7.3m.

The future work will still focus on the fitting accuracy not only on kinematics, but also on synthetical design, stiffness and control, as well as sharing strategy study. Further more, experiment research will be taken into account as certification for the theoretical analysis.

7. Acknowledgment

This research is sponsored by National Natural Science Foundation of China (No.50605035, 10778625), the State High-Technology Development Program of China (No.2006AA04Z133), and the National 973 Program of China (No. 2007CB714007).

8. References

- Carretero, J. A.; Nahon, M.; Buckham, B. & Gosselin, C. M. (1997). Kinematic analysis of a three-dof parallel mechanism for telescope applications, *Proceedings of The 1997 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pp. 1-8, California, Sep. 1997, ASME, Sacramento
- Duan, B.-Y. (1999). A new design project of the line feed structure for large spherical radio telescope and its nonlinear dynamic analysis. *Mechatronics*, Vol. 9, pp. 53-64
- Li, H. (1998). China hopes to move FAST on largest telescope. *Science*, Vol. 281, pp. 771-773
- Liu, X.-J; Wang, J.-S.; Gao, F. & Wang, L.-P. (2001). On the analysis of a new spatial three degrees of freedom parallel manipulator *IEEE Transactions on Robotics and Automation*, *IEEE Transactions on Robotics and Automation*, Vol. 17, pp. 959-968
- Luo, Y.-F.; Deng, C.-G. & Li, G.-Q. (2000). Structural Analysis of the Reflector Supporting System of the 500 m Active Spherical Telescope. *Journal of Tongji University*, Vol. 28, No. 4, pp. 497-500
- Nan, R.-D. & Peng, B. (2000). Chinese concept for the 1km² radio telescope. *Acta Astronautica*, Vol. 46, pp. 667-675
- Press, W. H.; Teukolsky, S. A.; Vetterling, W. T. & Flannery, B. F. (1995). *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University, U.K.
- Qiu, Y.-H. (1998). A novel design for a giant Arecibo-type spherical radio telescope with an active main reflector. *Mon. Not. R. Astron. Soc*, Vol. 301, pp. 222-228
- Tang, X.-Q.; Wang, J.-S.; Huang P. et al. (2007). Patent: 3-PRS Based Co-driving Array Parallel Mechanism for Large Radio Telescope, 200710177308.3
- Tsai, M.-S. & Shiau, T.-N. (2003). Direct kinematic analysis of a 3-PRS parallel mechanism. *Mechanism and Machine Theory*, Vol. 38, No. 1, pp. 71-83
- Wang, J.-S.; Tang, X.-Q.; Duan, G.-H. & Li, J.-F. (2001). Design methodology for a novel planar three degrees of freedom parallel machine tool. *Proceedings of the 2001 IEEE*

International Conference on Robotics & Automation, pp. 2448-2453, Korea, May. 2001, IEEE, Seoul

Wang, J.-S. & Tang, X.-Q. (2003). Analysis and dimensional design of a novel hybrid machine tool. *International Journal of Machine Tools and Manufacture*, Vol. 43, No. 7, pp. 647-655

Wang, Q.-M.; Tang, X.-Q.; Zhu, L.-C. et al. (2006). Patent: Co-driving Array Parallel Mechanism for Large Radio Telescope, 200610011941.0

A Reconfigurable Mobile Robots System Based on Parallel Mechanism

Wei Wang¹, Houxiang Zhang², Guanghua Zong¹ and Zhicheng Deng¹

¹ *Beijing University of Aeronautics and Astronautics,*

² *University of Hamburg,*

¹ *China,*

² *Germany*

1. Introduction

Reconfigurable robots consist of many modules which are able to change the way they are connected. As a result, these robots have the capability of adopting different configurations to match various tasks and suit complex environments. For mobile robots, the reconfiguration is a very powerful ability in some tasks which are difficult for a fixed-shape robot and during which robots have to confront unstructured environments (Granosik et al. 2005; Castano et al. 2000), e.g. navigation in rugged terrain. The basic requirement for this kind of robotic system is the extraordinary motion capabilities.

In recent years considerable progress has been made in the field of reconfigurable modular robotic systems, which usually comprise three or more rigid segments that are connected by special joints (Rus, D. and Vona, M. 2000). One group of the reconfigurable robots featuring in interconnected joint modules realizes the locomotion by virtue of the structure transform performed by the cooperative movements and docking/undocking actions of the modules (Suzuki et al. 2007; Kamimura et al. 2005; Shen et al. 2002; Suzuki et al. 2006; Vassilvitskii et al. 2002). Because the modules in these robots are not able to move independently and the possible structures of the robot are limited, these kinds of robots are not suitable for the field tasks.

The other kind of reconfigurable robots being composed of independently movable modules is more suitable for the field environment. The first prototype (Hirose et al. 1990) with powered wheels was designed by Hirose and Morishima in 1990, which consists several vertical cylindrical segments. The robot looks like a train, however with a weight over 300 kg it is too heavy. Klaassen developed a mobile robot with six active segments and a head for the inspection of sewage pipes (Klaassen et al. 1999). There are twelve wheels on each module to provide the driving force. Mark Yim proposed another reconfigurable robot PolyBot which is able to optimize the way its parts are connected to fit the specific task (Yim et al. 2000). PolyBot adopts its shape to become a rolling type for passing over flat terrain, an earthworm type to move in a narrow space and a spider type to stride over uncertain hilly terrain.

The application of powered tracks to field robots enriches their configurations and improves the adaptability to the environments. A serpentine robot from Takayama and Hirose

consists of three segments. Each segment is driven by a pair of tracks, but all tracks are powered simultaneously by a single motor located in the centre segment (Takayama et al., 2000). The special ability of adapting to irregular terrain is passive and provided by springs. The OmniTread serpentine robot (Granosik et al. 2005) for industrial inspection and surveillance was developed by Grzegorz Granosik in 2004. Optimal active joints are actuated by pneumatic cylinders in order to compromise the strength and compliance. However, the known robots usually have few configurations due to relatively simple docking and pose-adjusting mechanisms. For example, the Millibot Train robot consists of seven compact segments, which can connect by couplers with one DOF (Brown et al. 2002). A reconfigurable mobile robot designed by M. Park is not able to change its configuration actively at all (Park et al. 2004). The robot from Université Libre de Bruxelles has a one-DOF pose-adjusting mechanism and one coupler to change the configuration between the neighboring modules as well (Sahin et al. 2002).

From the mechanical point of view, the reconfiguration mechanism applied to mobile robot is composed of the posture-adjusting and connecting mechanism, and the most important technology is how to construct a posture-adjusting mechanism with large workspace and high driving ability in a limited robot body. However, in complex field terrain, the fact that the existing reconfigurable mobile robots can only assume limited configurations due to relatively simple posture-adjusting is a ubiquitous deficiency.

The project presented in this chapter aims at developing a reconfigurable mobile multi-robot platform made highly flexible and robust by its three-DOF posture-adjusting ability. The key object of the project is to develop a new posture-adjusting mechanism featuring in compact structure, large workspace and powerful driving ability. As a secondary object, the project has developed an effective connecting mechanism aligned to flat terrain and synthesized it with the posture-adjusting mechanism. The locomotion abilities of the system are expected to be as follows.

1. The single robots in the system have an independent omni-directional locomotion ability equivalent to that of a normal outdoor mobile robot.
2. Due to the posture-adjusting mechanism, which enables the robots to drive very well and to operate in a large workspace, the robots can adjust the posture of their partners.
3. The connecting mechanism tolerating large posture deviation in flat terrain can link two robots in a locked connection and transit large forces and torques between them.
4. Compared with a single robot, the connected robots are able to perform more demanding locomotion activities, such as stepping over high obstacles, crossing wide grooves, passing through narrow barriers and self-recovering from invalid postures and other actions which are impossible for a single robot.

To reach the above targets, a novel reconfigurable mobile robot system JL-1 based on a serial and parallel active spherical mechanism and a conic self-aligning connecting mechanism has been developed. This system is composed of three robot modules which are able to not only move independently, but also to connect to form a chain-structured group capable of reconfiguration. On flat terrain, each module of JL-1 can cooperate with each other by exchanging information to keep up its high efficiency; while on rugged terrain, the modules can actively adopt a reconfigurable chain structure to cope with the cragged landforms which will be a nightmare for a single robot (Zhang et al. 2006; Wang et al. 2006).

In the chapter, after giving an overview of JL-1, the discussion focuses on some special locomotion capabilities of it. Then the related kinematics analysis of the serial and parallel

mechanism is discussed thoroughly as well as the theory of the connecting mechanism. Based on the discussion, the mechanical realization of JL-1 is introduced in detail. The prototype shows the advantage of the parallel mechanism in realizing powerful driving force in a relative small size. In the end, a series of successful on-site tests, such as crossing high vertical obstacles, connecting action and getting self-recovery when the robot is upside-down, is presented to confirm the above principles and the locomotion abilities of JL-1.

2. Overview of the reconfigurable JL-1

2.1 Mechanical model of JL-1

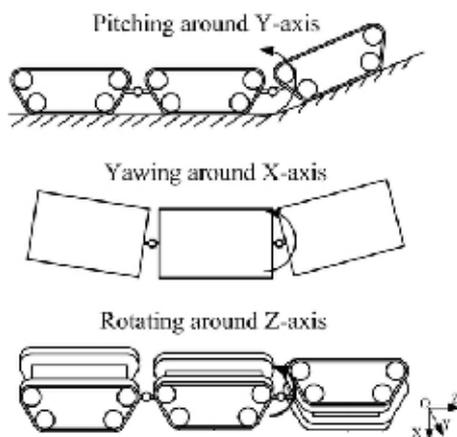


Fig. 1. Adapting to terrains by pitching, yawing and rotating

By virtue of three uniform modules being capable of docking with each other, JL-1 has various moving modes which enable JL-1 to move in almost all kinds of rough terrains. The principle of terrain adaptability is shown in Fig. 1. In connected state, JL-1 can change its posture by pitching around the Y axis, yawing around the X axis and rotating around the Z axis. JL-1 is endowed with the abilities of adopting optimized configurations to negotiate difficult terrains or splitting into several small units to perform tasks simultaneously, by the three DOF active spherical joints between the modules and the docking mechanism being capable of self-aligning within certain lateral and directional offsets.

In JL-1, the yawing and pitching movements are achieved by a parallel mechanism. The third rotation DOF around the joint's Z axis is achieved by a serial mechanism. There are two reasons for using the serial and parallel mechanisms for JL-1. Firstly, the JL-1 robot can be made lightweight and dexterous while allowing for a larger payload. Secondly, the advantages of the high rigidity of a parallel mechanism and the extended workspace of a serial mechanism can be combined, thus improving the flexibility of the robotic system.

2.2 Locomotion capabilities

It can be easily imagined that the locomotion abilities of JL-1 will be enhanced when it is in connected state, such as climbing up higher steps, spanning wider ditch and stepping up stairs. Furthermore, JL-1 is capable of some novel actions, which will be required in outdoor environment, e.g. self-recovery and passing through a narrow fence.

2.2.1 90° self-recovery

It is possible for the robot to implement a 90° recovering movement by adopting the proper configuration sequence as shown in Fig. 2.

- The robot is lying on its side
- The first module and the last module are yawing up around the X axes of the active joints.
- Then the first module and the last module rotate 90° around the Z axes.
- After that, they are pitching down around the Y axes of the active joints until they attach to the ground in order to raise the middle module up.
- The middle module rotates around the Z axis until it is parallel to the ground.
- In the end, the module is pitching down around the Y axes of the active joints until all three modules attach to the ground together. The robot is now in its home state again, and the process of 90° Self-recovery is over.

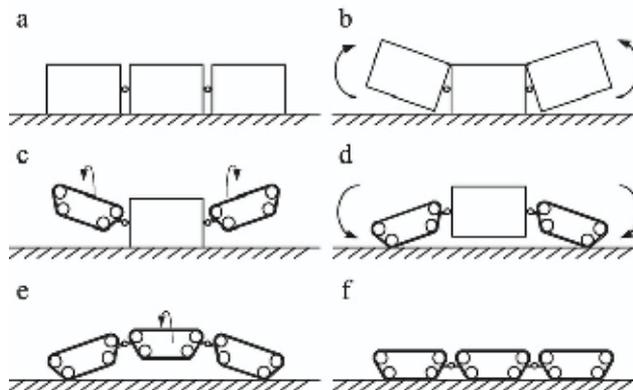


Fig. 2. 90° recovering movement

2.2.2 180° Self-recovery

It is also possible for the robot to tip over and realize the 180° recovery movement as shown in Fig. 3.

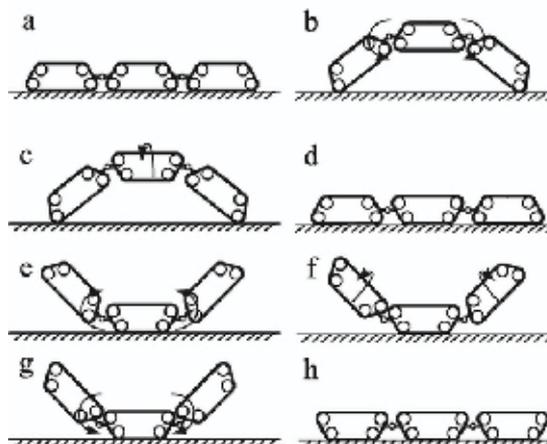


Fig. 3. 180° recovering movement

- The robot is in its home state.
- The first and the last modules are pitching down around the Y axes of the active joints until the middle module is in the air.
- The middle module rotates 180° according to the Z axis.
- The first module and the last module are pitching down around the Y axes of the active joints until the middle module attaches to the ground.
- The first module and the last module are pitching up around the Y axes of the active joints again.
- The first module and the last module are rotating 180° around the Z axes of the active joints.
- Then the first module and the last module are pitching down around the Y axes of the active joints again until all three modules attach to the ground.
- The process of 180° Self-recovery is over.

2.2.3 Crossing a narrow fence

As shown in Fig. 4, the train configuration robot is able to cross a fence narrower than the width of its modules.

- The robot is in its home state, and the sensor detects the fence in the moving direction.
- The robot stops before the fence, and then the first module pitches up around the Y axis and then rotates 90° according to the Z axis.
- The crossing movement does not stop until the first module passes through the fence.
- The first module rotates and pitches to get back into the home state, and then the three modules attach to the ground together again.

The following steps (e) to (k) of the second and third modules are similar to those of the first one. The process will be achieved until the robot crosses the fence entirely. In order to show the principle clearly, the lateral views of steps (e) and (f) are also given.

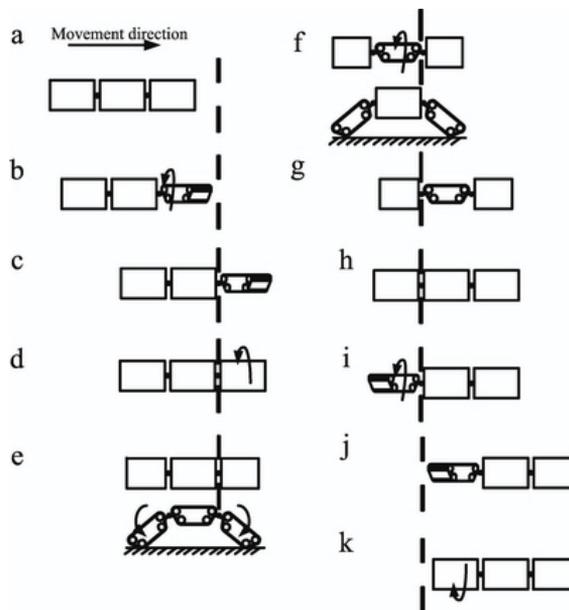


Fig. 4. The sequence of crossing a narrow fence

3. Kinematics analysis of the active spherical joint

As described above, the robot's reconfiguring abilities are achieved by the motion of the 3 DOF active spherical joints. Two of the DOF achieved by the parallel mechanism are yawing and pitching around the joint's X and Y axes respectively. The third rotation DOF around the joint's Z axis is achieved by the serial mechanism.

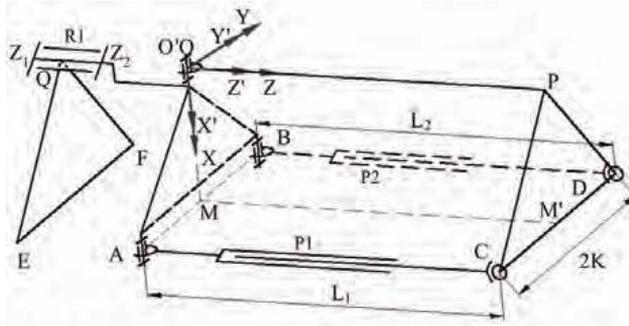


Fig. 5. The kinematics model of the active spherical joint

To demonstrate the reconfiguring possibility, the kinematics analysis of two connected modules should be studied first. Fig. 5 shows the kinematics model of the joint between two modules. Where $OXYZ$ is the world coordinate fixed at the plane QEF which represents the front unmovable module during the reconfiguration. The origin is located at the universal joint O , the Z -axis coincides with the axis of the serial mechanism and the X -axis points to the middle point of line AB . Another reference coordinate $O'X'Y'Z'$ is fixed at triangular prism $OABPCD$ which represents the back moveable module. The $O'X'Y'Z'$ is coincident with the $OXYZ$ when the spherical joint is in its home state. Equations (1) and (2) are satisfied due to the mechanical constraints. QF is perpendicular and equal to QE .

$$QEF // OAB // PCD \tag{1}$$

$$QEF = OAB = PCD \tag{2}$$

The required orientation for the reference frame $O'X'Y'Z'$ on the back module is achieved by a rotation of θ_z , a pitching angle θ_y and a yawing angle θ_x according to the relative axes. From the mechanical point of view, actually the pitching and yawing motions are realized by the outstretching and returning movement of the L_1, L_2 of the parallel mechanism, and the rotation of θ_z is actuated by the serial mechanism. The freedom of the reconfiguring movement is three and can be described with the generalized coordinate θ (3). The joint variants of the movement are named q , described as (4).

$$\theta = [\theta_x, \theta_y, \theta_z]^T \tag{3}$$

$$q = [L_1, L_2, \theta_z]^T \tag{4}$$

The purpose of the kinematics analysis is to deduce the relationship between q and θ . In Fig. 10, the points A, B, C, D are described as (5) in the $OXYZ$ coordinate.

$$A = \begin{bmatrix} K \\ -K \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} K \\ K \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} K \\ -K \\ L \end{bmatrix} \quad D = \begin{bmatrix} K \\ K \\ L \end{bmatrix} \tag{5}$$

The homogeneous transformation matrix $[T]$ from the world coordinate $OXYZ$ to the coordinate $O'X'Y'Z'$ is described as (6).

$$T = Rot(Y)Rot(X)Rot(Z) = \begin{bmatrix} c\theta_y & 0 & s\theta_y \\ 0 & 1 & 0 \\ -s\theta_y & 0 & c\theta_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta_x & -s\theta_x \\ 0 & s\theta_x & c\theta_x \end{bmatrix} \cdot \begin{bmatrix} c\theta_z & -s\theta_z & 0 \\ s\theta_z & c\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

After the reconfiguring movement, A , B , C , and D are changed to new positions described as A_1 , B_1 , C_1 , and D_1 . The Cartesian coordinates of the new points can be expressed as (7) and (8).

$$[A_1 \quad B_1] = Rot(Z)[A \quad B] \quad (7)$$

$$[C_1 \quad D_1] = T[C \quad D] \quad (8)$$

There are some constraints to the mechanical structure, as shown in (9) and (10). The lengths of the link L_1 and L_2 are equal to the distance between C_1A_1 and D_1B_1 respectively.

$$L_1 = C_1 - A_1 \quad (9)$$

$$L_2 = D_1 - B_1 \quad (10)$$

All these results are inserted into (9) and (10), then the kinematics expression results from them.

$$\begin{aligned} L_1^2 = & (K(c\theta_y c\theta_z + s\theta_x s\theta_y s\theta_z) - \\ & K(-s\theta_z c\theta_y + s\theta_x s\theta_y c\theta_z) + \\ & Lc\theta_x s\theta_y - Kc\theta_z - Ks\theta_z)^2 + \\ & (K(c\theta_x s\theta_z) - K(c\theta_x c\theta_z) - Ls\theta_x - Ks\theta_z + Kc\theta_z)^2 \\ & + (K(-s\theta_y c\theta_z + s\theta_x c\theta_y s\theta_z) - \\ & K(s\theta_y s\theta_z + s\theta_x c\theta_y c\theta_z) + Lc\theta_x c\theta_y)^2 \end{aligned} \quad (11)$$

$$\begin{aligned} L_2^2 = & (K(c\theta_y c\theta_z + s\theta_x s\theta_y s\theta_z) + \\ & K(-s\theta_z c\theta_y + s\theta_x s\theta_y c\theta_z) + \\ & Lc\theta_x s\theta_y - Kc\theta_z + Ks\theta_z)^2 + \\ & (K(c\theta_x s\theta_z) + K(c\theta_x c\theta_z) - Ls\theta_x - Ks\theta_z - Kc\theta_z)^2 \\ & + (K(-s\theta_y c\theta_z + s\theta_x c\theta_y s\theta_z) + \\ & K(s\theta_y s\theta_z + s\theta_x c\theta_y c\theta_z) + Lc\theta_x c\theta_y)^2 \end{aligned} \quad (12)$$

Named $T_{L1}=L_1^{1/2}$, $T_{L2}=L_2^{1/2}$, then the relation between q and θ can be concluded as (13).

$$q = [T_{L1}^{1/2}, T_{L2}^{1/2}, \theta_z]^T \quad (13)$$

The relationship of the world coordinate and the reference joint coordinate can be concluded. Furthermore the movements can be anticipated according to the joints' driving outputs.

4. System realization

4.1 Mechanical realization

The JL-I system consists of three connected, identical modules for crossing grooves, steps, obstacles and traveling in complex environment. The mechanical structure is flexible due to its uniform modules and special connection joints (Fig. 6a). Actually each module is an entire robot system that can perform distributed activities (Fig. 6b).

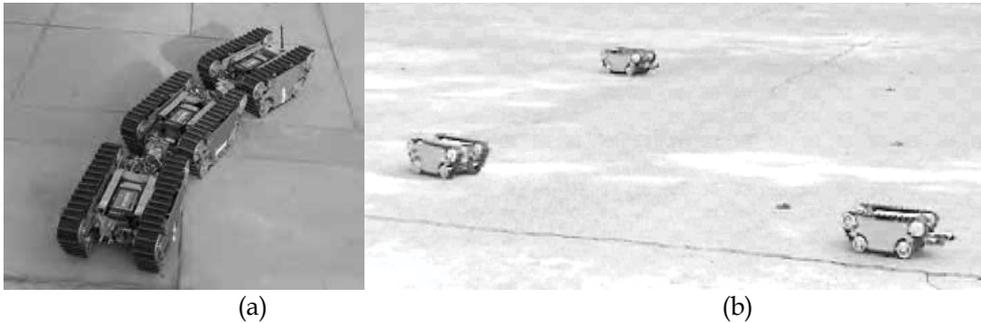


Fig. 6. The robotics system of JL-I

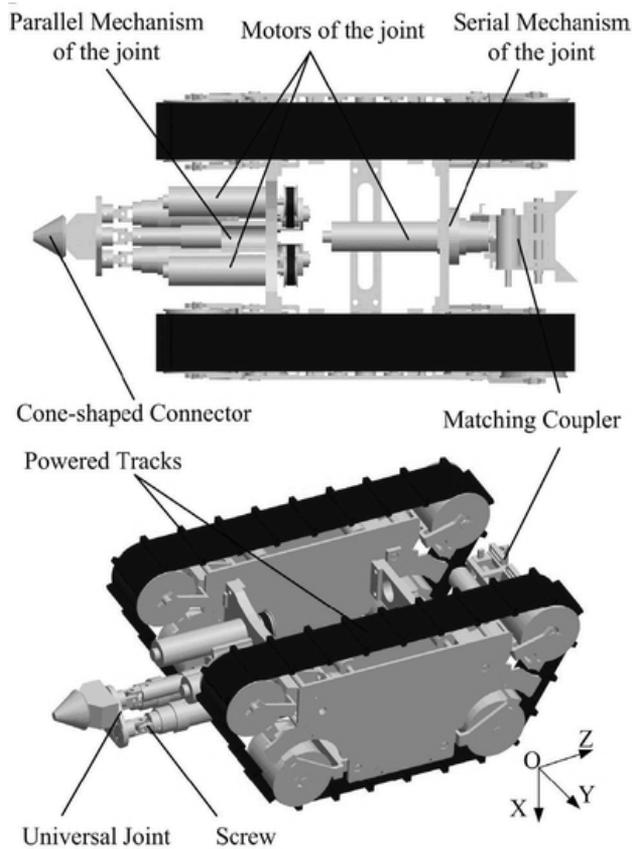


Fig. 7. An artistic impression of the module

The single module is about 35 centimeters long, 25 centimeters wide and 15 centimeters high. Fig. 7 shows the mechanical structure of the module which has two powered tracks, a serial mechanism, a parallel mechanism, and a docking mechanism. Two DC motors drive the tracks providing skid-steering ability in order to realize the flexible omni-directional movement. The docking mechanism consists of two parts: a cone-shaped connector at the front and a matching coupler at the back of the module. It enables any two adjacent modules to link, forming a train configuration.

4.1.1 Realizing the parallel mechanism

The realization of the parallel mechanism is also shown in Fig. 8. Each branch of it consists of a driving platform, a Hooker joint, a lead screw, a nut slider, a ball bearing, a synchronous belt system, a DC motor and a base platform. The Hooker joint connects the driving platform and the nut slider. The lead screw is supported by a ball bearing in the base platform. The cone-shaped connector fixed on the driving platform is called a buffer head, because its rubber is used to buffer the wallop during the docking process. Besides the two branches, there is a knighthead fixed on the base platform and connected to the driving platform by another Hooker joint. By revolving the two lead screws, the driving platform can be manipulated relative to the Hooker joint on the knighthead.

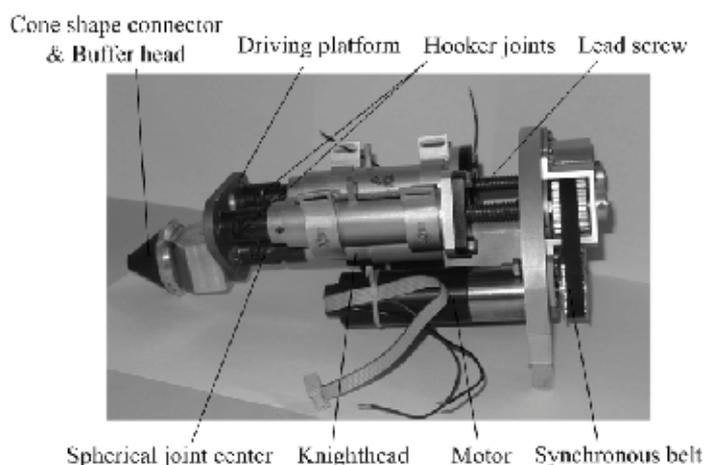


Fig. 8 The parallel mechanism

There are two advantages in applying the synchronous belt system.

- a) When the screw revolves, it rocks around the ball bearing. By using the synchronous belt system and an elastic connector, the rock motion of the screw is isolated from the motor.
- b) The motor and the lead screw can be installed on the same side of the base platform, and that decreases the dimension of the structure.

4.1.2 Realizing the serial and docking mechanism

The docking mechanism consists of two parts: a cone-shaped connector at the front (shown in Fig. 8) and a matching coupler at the back of the module, as shown in Fig. 9. The coupler is composed of two sliders propelled by a motor-driven screw. The sliders form a matching

funnel which guides the connector to mate with the cavity and enables the modules to self-align with certain lateral offsets and directional offsets. After that, two mating planes between the sliders and the cone-shaped connector constrain the movement, thus locking the two modules. This mechanism enables any two adjacent modules to link, forming a train configuration. Therefore the independent module has to be rather long in order to realize all necessary docking functions. In designing this mechanism and its controls, the equilibrium between flexibility and size has to be reached. A DC motor is connected to the coupler with its motor shaft aligned with the module's Z axis, which also passes through the center of the Hooker joint on the knighthead of the parallel mechanism. Therefore a full active spherical joint is formed when two modules are linked.

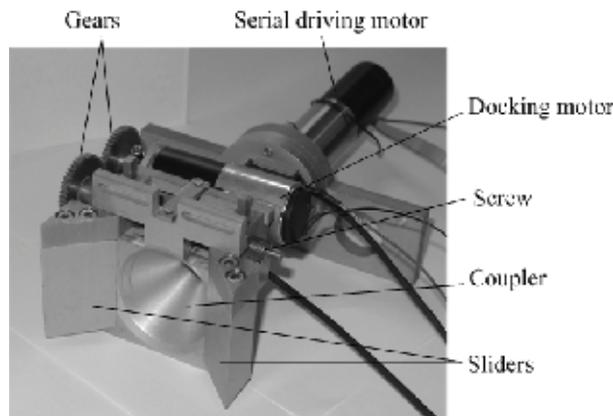


Fig. 9. The serial and docking mechanism

This docking mechanism can compensate a position deviation within $\pm 30\text{mm}$ and a posture deviation within $\pm 45^\circ$ between two modules. The self-locking characteristic of the screw-nut mechanism ensures a reliable connection between two modules to endure the vibration in motion.

4.2 Control system

The control system of the robot based on an industrial PC (IPC) and a master-slave structure meets the requirements of functionality, extensibility, and easy handling (Fig. 10). Multiple processes programming capability is guaranteed by the principle of the control structure. The hardware consists of an SBC-X255, an independent image processing unit and a low-level driving unit (SBC 2).

The SBC-X255 is the core part of the control system. It is a standard PC/104+ compliant, single-board computer with an embedded low power Intel Xscale PXA255 (400 MHz). This board operates without a fan at temperatures from -40°C up to 85°C and typically consumes fewer than 4.5 Watts while supporting numerous peripherals. The Ethernet port is used as a communication interface between the IPC and the image processing unit which is in charge of searching and monitoring. The IPC is a higher-level controller and does not take part in joint motion control. Its responsibilities include receiving orders from the remote controller, planning operational processes, receiving feedback information.

The SBC 2 is in charge of driving five DC motors and receives and processes all related sensor signals. It can directly count the pulse signals from the encoder, deal with the signals

from other magnetic sensors, and directly drive the DC motors forward and backward at different velocities. Meanwhile it sends all information to the IPC through another Ethernet port.

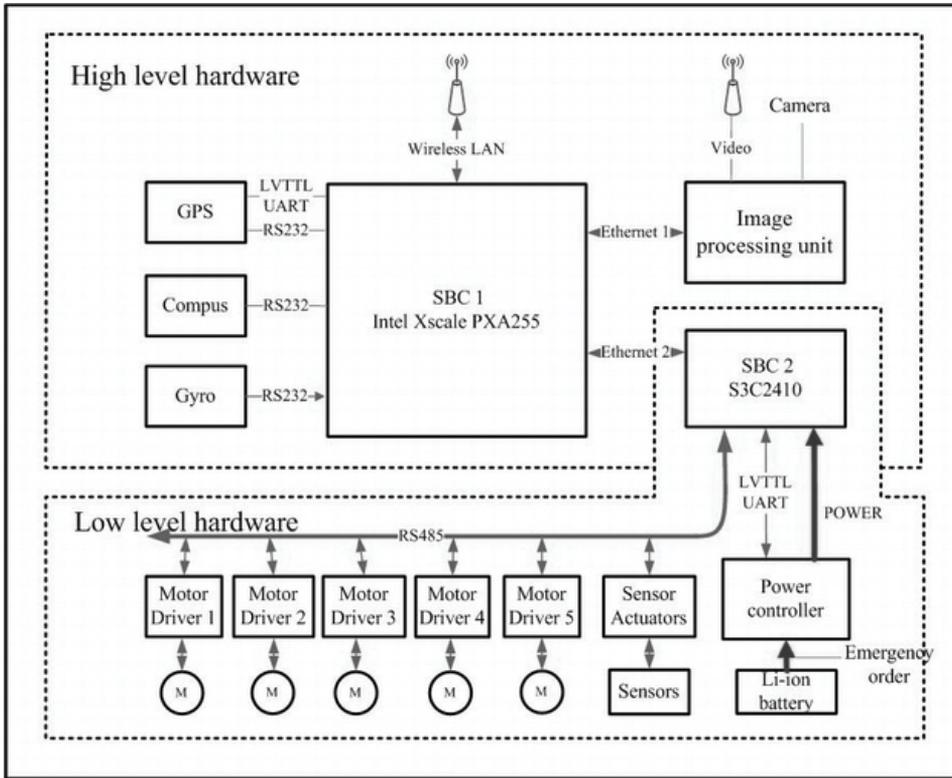


Fig. 10. The control system of JL-1's module

There are two kinds of external sensors on the robot: a CCD camera and touchable sensors, which are responsible for collecting information about the operational environment. The internal sensors such as GPS, digital compass, gyro sensors are used to reflect the self-status of the robot. The gesture sensor will send the global locomotion information of the robot θ_x , θ_y , and θ_z to the controller, which are essential to inverse kinematics. Meanwhile there are limit switches to give the controller the position of the joint. On the joint where the accurate position is needed, the optical encoder is used.

5. On-site tests

Relevant successful on-site tests with the mobile robot were carried out recently, confirming the principles described above and the robot's ability. Fig. 11 shows the docking process of the connection mechanism whose most distinctive features are its ability of self aligning and its great driving force. With the help of the powered tracks, the cone-shaped connector and the matching coupler can match well within $\pm 30\text{mm}$ lateral offsets and $\pm 45^\circ$ directional offsets.

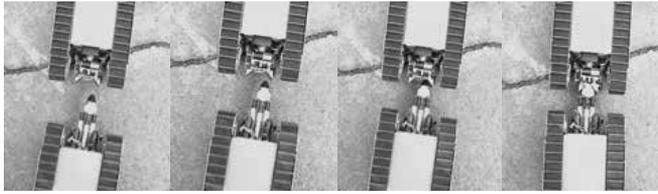


Fig. 11. The docking process

Compared with many configurable mobile robots, the JL-I improves its flexibility and adaptability by using novel active spherical joints between modules. The following figures show the typical motion functionalities one by one, whose principles are discussed above.



Fig. 12. Climbing stairs

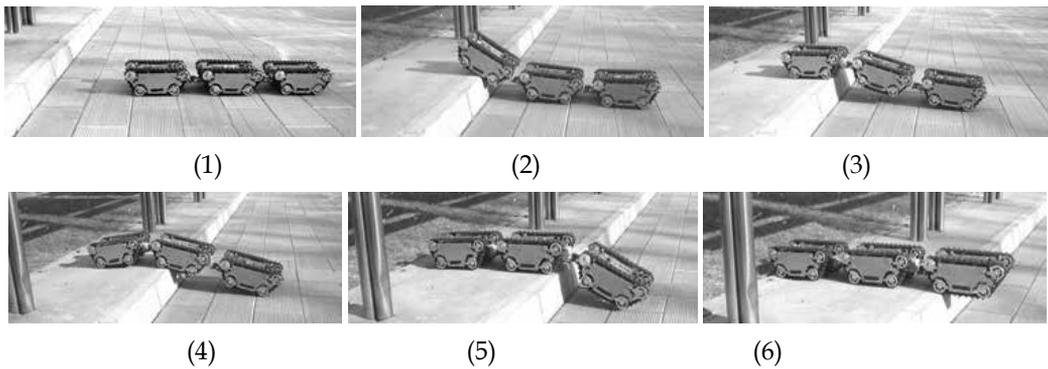


Fig. 13. Snapshots of crossing a step

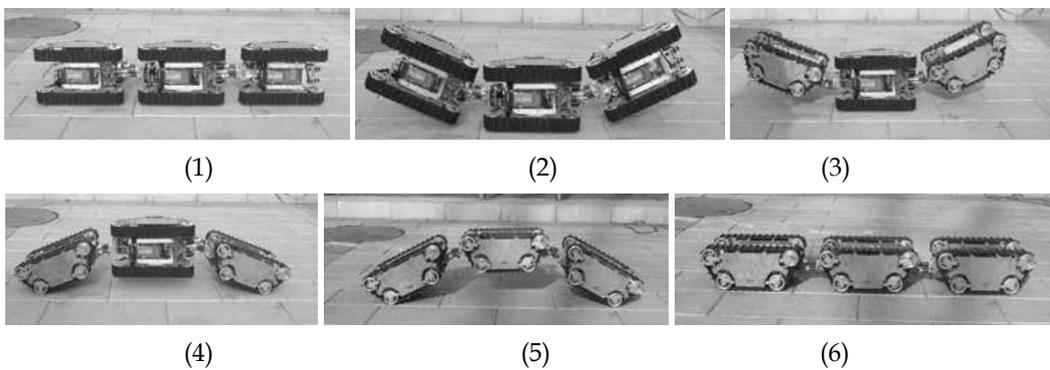


Fig. 14. Snapshots of the 90° self-recovery

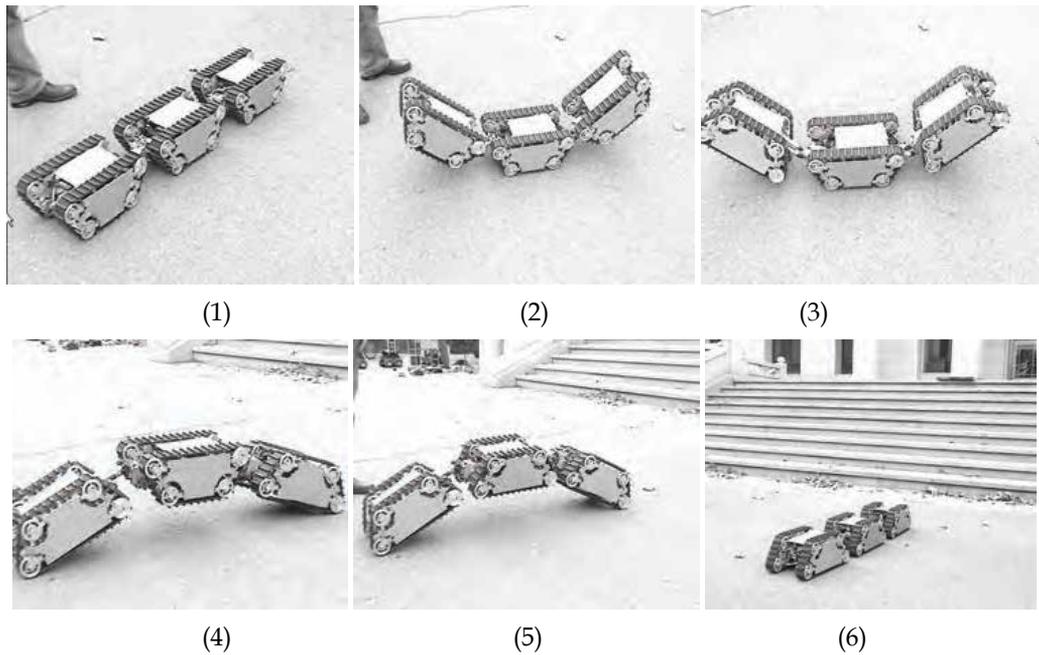


Fig. 15. Snapshots of the 180° self-recovery

The experimental results show that the 3 DOF active joints with serial and parallel mechanisms have the ability to achieve all the desired configurations. The performance specifications of JL-I are given in Table 1.

Parameters	Values
Posture adjustment angle around X-axis	$\pm 45^\circ$
Posture adjustment angle around Y-axis	$\pm 45^\circ$
Posture adjustment angle around Z-axis	0~360°
Maximum lateral docking offset	± 30 mm
Maximum directional docking offset	$\pm 45^\circ$
Maximum height of steps	280 mm
Maximum length of ditches	500 mm
Minimum width of the fence	200 mm
Maximum slope angle	40°
Self-recovering ability	0~180°
Maximum climbing velocity	180 mm/s
Maximum unchangeable working time	4 hours

Table 1. Performance specifications

6. Conclusions

The modular reconfiguration robot has the ability of changing its configuration which makes it more suitable for complex environments. In contrast to conventional theoretical research, the project introduced in this paper successfully completes the following innovative work.

- a) It proposes a robot named JL-I which is based on a modular reconfiguration concept. The advantages and the characteristics of the mechanism are analysed. The robot features a docking mechanism with which the modules can connect or disconnect flexibly. The active spherical joints formed by serial and parallel mechanisms endow the robot with the ability of changing shapes in three dimensions.
- b) A kinematics model of reconfiguration between two modules is given. The relationship of the world coordinate and the reference joint coordinate is concluded. Furthermore, the movements can be anticipated according to the joints' driving outputs. The analysed results are important for system design and the design of the controlling mechanism for the robot.
- c) Experimental tests have shown that the JL-I can implement a series of various locomotion capabilities such as 90° recovery, 180° recovery, and crossing steps. This implies the mechanical feasibility, the rationality of the analysis and the outstanding movement adaptability of the robot.

The future research will focus on the following aspects.

- a) Developing a new docking mechanism which tolerates larger offset in rugged terrain and can be used as a simple manipulator;
- b) Developing a more reliable track modules with shock absorption function;
- c) Developing a new mechanism which can actively undock a disable robot module.

7. Acknowledgement

The work in this chapter is proposed by National High-tech R&D Program (863 Program) of China (No. 2006AA04Z241).

8. Reference

- Granosik, G.; Hansen, M. G. & Borenstein, J. (2005). The omnitread serpentine robot for industrial inspection and surveillance. *Industrial Robot: An International Journal*, Vol.32, No.2, (Feb. 2005) page numbers (139-148), ISSN: 0143-991X
- Castano, A.; Shen, W.M. & Will, P. (2000). CONRO: towards miniature self-sufficient metamorphic robots. *Autonomous Robots*, Vol.13, No.4, (April 2000) page numbers (309-324), ISSN: 0929-5593
- Rus, D. & Vona, M. (2000). A basis for self reconfigurable robots using crystal modules, Proceedings of the 2000 IEEE Conference on Intelligent Robots and Systems, pp. 2194-2202, ISBN: 0-7803-6348-5 Takamatsu, Japan, October 2000, IEEE Service Center, Piscataway
- Suzuki, Y.; Inou, N.; Kimura, H & Koseki, M. (2007). Reconfigurable group robots adaptively transforming a mechanical structure. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2361-2367, ISBN: 1-

- 4244-0912-8, San Diego, CA, USA, Oct. 29 - Nov. 2, 2007, IEEE Service Center, Piscataway
- Kamimura, A.; Kurokawa, H.; Yoshida, E.; Murata, S.; Tomita, K. & Kokaji, S. (2005). Automatic locomotion design and experiments for a modular robotic system, *IEEE/ASME Transactions on Mechatronics*, Vol. 10, No. 3, (March 2005) page numbers (314-325), ISSN: 1083-4435
- Shen, W.-M.; Salemi B. & Will, P. (2002). Hormone-Inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, (Oct. 2002) page numbers (700-712), ISSN: 0882-4967
- Suzuki, Y.; Inou, N.; Kimura, H. & Koseki, M. (2006). Reconfigurable group robots adaptively transforming a mechanical structure - Crawl motion and adaptive transformation with new algorithms. *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS 2006)*, pp. 2200-2205, ISBN: 1-4244-0259-X, Beijing, China, October 2006, IEEE Service Center, Piscataway
- Vassilvitskii, S.; Yim, M. & Suh, J. (2002). A complete, local and parallel reconfiguration algorithm for cube style modular robots, *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, pp. 117- 122 ISBN: 0-7803-7272-7, Washington DC, USA, May 2002, IEEE Service Center, Piscataway
- Hirose, S. & Morishima, A. (1990). Design and control of a mobile robot with an articulated body. *The International Journal of Robotics Research*, Vol. 9 No. 2, (Feb. 1990) page numbers (99-113), ISSN: 0278-3649
- Klaassen, B. & Paap, K.L. (1999). GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control. *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 3014 - 3019, ISBN: 0792356012, Detroit, MI, USA, May 1999, IEEE Service Center, Piscataway
- Yim, M. & David, G. (2000). PolyBot: a module reconfigurable robot. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp.514-520, ISBN: 0-7803-5886-4, San Francisco, CA, USA, April 2000, IEEE Service Center, Piscataway
- Takayama, T. & Hirose, S. (2000). Development of Souryu-I connected crawler vehicle for inspection of narrow and winding space, *Proceedings of the 26th Annual Conference of the IEEE Industrial Electronics Society*, pp. 143-148 ISBN: 0-7803-6456-2, Nagoya, Aichi, Japan, Oct. 2000, IEEE Service Center, Piscataway
- Brown, H. B. & et al. (2002). Millibot trains for enhanced mobility. *IEEE/ASME Transactions on Mechatronics*, Vol.7, No.4, (March 2002) page numbers (452-461), ISSN: 1083-4435
- Park, M.; Chung W. & Yim M. (2004). Control of a mobile robot with passive multiple trailers. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 4369-4374, ISBN: 0-7803-8232-3, New Orleans, LA, United States, April-May 2004, IEEE Service Center, Piscataway
- Sahin, E.; Labella, T.H. & et al. (2002). SWARM-BOT: Pattern formation in a swarm of self-assembling mobile robots. *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, pp. 145-150, ISBN: 0-7803-7437-1, Yasmine Hammamet, Tunisia, October 2002, IEEE Service Center, Piscataway

- Zhang, H.X.; Wang, W.; Deng, Z.C. & Zong, G.H. (2006). A novel reconfigurable robot for urban search and rescue. *International Journal of Advanced Robotic Systems*, Vol.3 No.4 (2006), page numbers (259-366), ISSN: 1729-8806
- Wang, W.; Zhang H.X; Zong, G.H. & Zhang, J.W. (2006). Design and realization of a novel reconfigurable robot with serial and parallel mechanisms. *Proceedings of 2006 IEEE International Conference on Robotics and Biomimetics*, pp. 697-702, ISBN: 1-4244-0571-8, Kunming, China, Dec. 2006, IEEE Service Center, Piscataway

Hybrid Parallel Robot for the Assembling of ITER

Huapeng Wu, Heikki Handroos and Pekka Pessi
*Lappeenranta University of Technology
Finland*

1. Introduction

The international thermonuclear experimental reactor (ITER) is a joint international research and development project that aims to demonstrate the scientific and technical feasibility of fusion power. The reactor tokamak (vacuum vessel) is made of stainless steel, and contains nine sectors welded together; each sector has about the size of 10 meter high and 6 meter wide. The sectors of ITER vacuum vessel (VV) (Fig. 1) require more stringent tolerance ($\pm 5\text{mm}$) than normally expected for the size of the structure involved. The walls (inner wall and outer wall) of ITER sectors are of 60mm thick and are joined by high quality leak tight welds. In addition to the initial vacuum vessel assembly, the sectors may have to be replaced for repair. Meanwhile, the machining operations and lifting of a possible e-beam gun column system require extreme stiffness property and good accuracy for a machine tool. The payload to weight ratio also has to be significantly better than it is in the commercial industrial robots.

The conventional robots, providing a high nominal payload, are lack of stiffness and accuracy in such machining condition. Since commercially available machines capable of handling large payloads require floor mounting and their workspaces are insufficient for reaching the cross section at a single mounting position, a special robot is needed. Parallel robots have high stiffness, high dynamic performance and good payload to weight ratio in comparison with the conventional serial robots. Stewart [1] presented the novel idea of six-degree-of-freedom parallel robot in 1960's. A remarkable number of research articles and books about parallel manipulators have been published during the last two decades. There are also a number of successful industrial applications developed [2], [3], [4], [7]. The parallel manipulators have many potential advantages compared with the conventional serial link manipulators. Parallel manipulators are closed-loop mechanism presenting good performances in terms of accuracy, rigidity, high speed, and ability to handle large loads. They are becoming popular in applications such as machining, welding, assembly, flight and vehicle simulators, mining machines, and pointing devices [2], [3], [4], [7]. The most important drawback of parallel robots is the small workspace, which can be made larger by adding additional serial axes in the robot.

For the assembly of the ITER vacuum vessel sector, the precise positioning of welding end-effectors at some distance in a confined space from the available supports will be required,

while it is not possible using conventional machines or robots. The parallel robot presented in this paper is able to carry out welding and machining processes from inside the ITER vacuum vessel, consisting of a six degree-of-freedom parallel mechanism mounted on a carriage driven by electric motor/gearbox on a rack. The robot carries both welding gun (such as a TIG, hybrid laser or e-beam welding gun to weld the inner and outer walls of the ITER vacuum vessel sectors) and machining tools (to cut and mill the walls with necessary accuracy). It can also carry other heavy tools and parts to a required position inside the vacuum vessel.

The robot offers not only a device but also a methodology for assembling and repairing VV. For assembling, an on-line six-degree-of-freedom-seam-finding algorithm has been developed. The algorithm enables the robot to find welding seam automatically in a very complex environment. For machining, the multi flexible machining processes carried out automatically by the robot have also been investigated, including edge cutting, smoothing, and defect point milling. The kinematic design of the robot has been optimised for the ITER access and a hydraulically actuated prototype has been built. Finally the experimental results are presented and discussed. The earlier development phases of the robot are presented in [8] and [9].

2. Structure of VV and assembling process

The inner and outer walls of the VV of ITER are made of 60mm-thick stainless steel 316L and are welded together through an intermediate, so-called "splice plate", inserted between the sectors to be joined. The splice plates have two important functions: (i) to allow access to bolt together the thermal shield between the VV and coils; and (ii) to compensate the mismatch between adjacent sectors to give a good fit-up of the sector-sector butt weld. The robot's end-effector will have to pass through the inner wall splice plates opening to reach the outer wall. As shown in Fig.1, the assembly and repairing processes have to be carried out from inside the vacuum vessel.

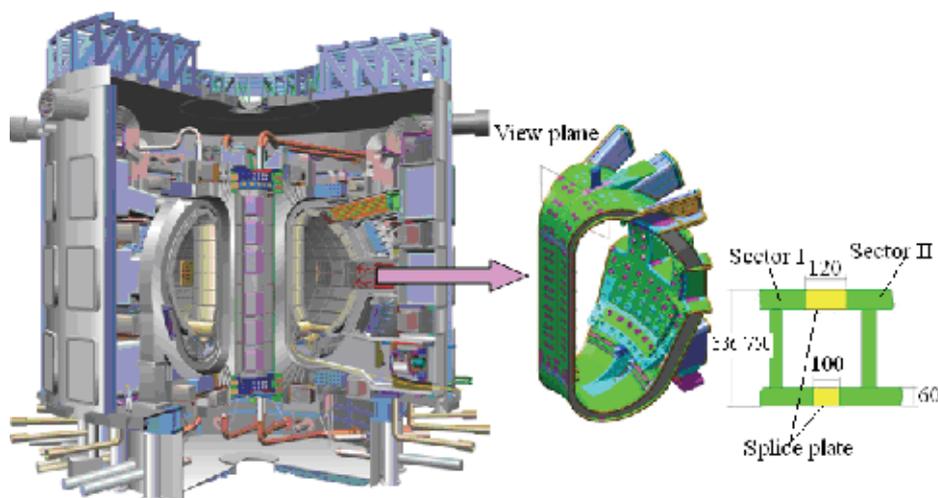


Fig.1 ITER and VV sectors to be welded

The assembly or repair will be performed according to four phases: cutting, edge machining and smoothing, welding, and non-destructive tests (NDT) control. The robot will carry out welding, machining, and inspecting inside the VV. The maximum robot force arises from cutting. It can be up to 3kN.

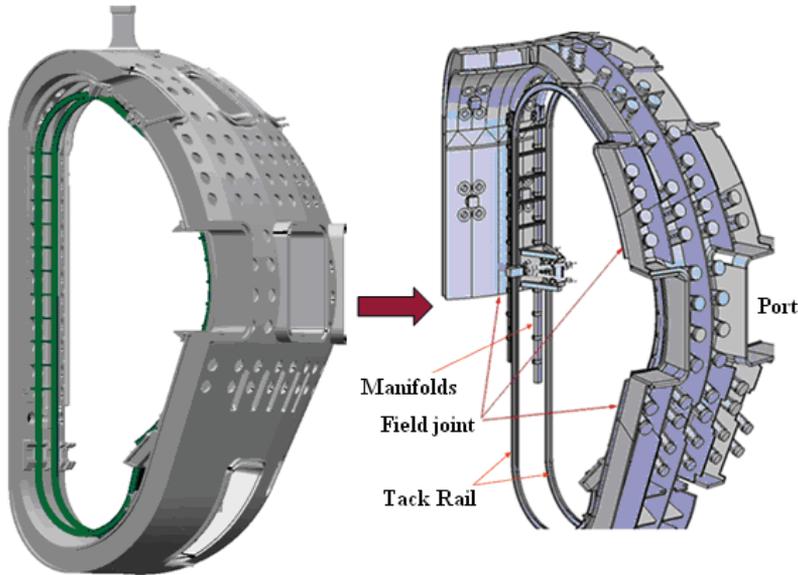


Fig.2 The track rail mounted inside VV and robot on the track

In order that the robot can operate in the cross section of the vessel, a track is assembled inside the sector. The track has a rack on one side of the rail and it is supported by manifolds and beams (shown in Fig. 2). The robot driven on the rail carries out welding or machining along the edge of the sector. After finishing the assembly of two sectors, the robot has to be moved to the next sector where there is also a track assembled. After finishing the assembly of all the sectors, the robot can be taken out via the port of VV.

3. Kinematics of parallel robot

3.1 Structure of parallel robot

The proposed parallel robot has ten degrees of freedom (Fig. 3). It consists of two relatively independent sub-structures: (i) carriage, which provides four additional degrees of freedom, i.e., rotation, linear motion, tilt rotation and tracking motion, and these four degrees are added to enlarge the workspace and to offer high mobility; and (ii) the Hexa-WH parallel mechanism driven by six hydraulic cylinders contributes six degrees of freedom for the end-effector. Thus the robot is a hybrid redundant manipulator with four extra degrees of freedom provided by serial kinematic links.

a). Carriage mechanism

The carriage mainly consists of 5 units. i) *Carriage frame*: The carriage frame is a complex structure welded by multi-steel-plates, and it is able to carry high payload and offers enough room to maintain mechanisms. Stiffness and weight are the most important considerations in the design, and they have been optimized to achieve necessary stiffness

with light weight. *ii) Tracking drive unit:* The tracking drive unit consists of electric motor, reduction unit CYCLO, V-shape bearing, and driving gear. The electric servo motor with

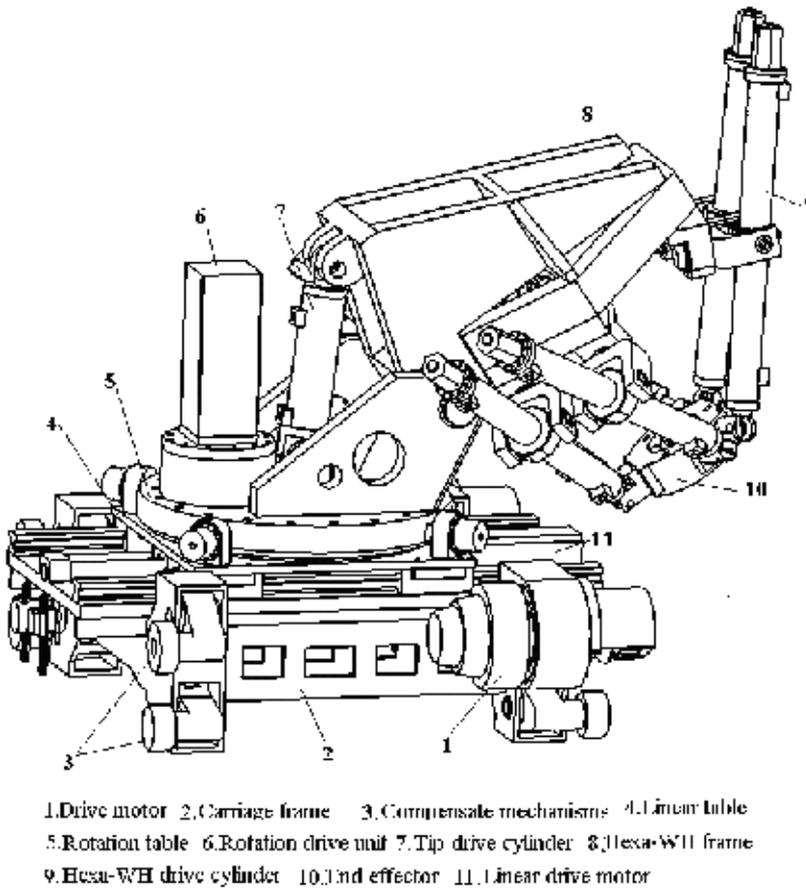


Fig. 3 Parallel robot

position feedback controller offers the high accurate motion. In order to output large torque to drive the heavy mass and payload, the reduction unit CYCLO is added to the motor to reduce speed and to transmit high torque to the driving gear. Two V-shaped wheels keep the carriage on the tracking rail at right position to avoid the cross motion. Two drive units are used in the carriage to offer enough torque in order to drive the robot and payload around inside the VV. *iii) Compensation mechanism:* The compensation system is an important unit that limits the backlash caused by the inaccurate assembling of the tracking rail and compensates the distance changing between the wheels in bending area. As the shape of the VV is very complicated, it is difficult to keep the tracking rails lying on the VV surface in the accurate position. The position tolerance can be up to $\pm 2\text{mm}$. The distance of the coupled wheels has to be adjusted to follow the changes of the rail, and all the wheels must touch the parallel rails with certain force during motion; hence an adaptive distance compensation system is needed and it should be able to undertake the summed weight of the robot and the payload, when the robot is upside down at the top position insider VV. Since the total

payload is very heavy, a hydraulic cylinder is applied to justify the compensation force according to the position where the robot is located. Fig. 4 shows the compensation system, where the upside is the tolerance adaptive mechanism that passively follows the changes of track rail and the downside is the hydraulic distance compensation system that ensures a constant force is applied to the rails. *iv) Linear drive unit:* The linear drive unit enlarges the workspace of the robot, and consists of five parts: ball screw drive unit, servo motor, rails, linear bearings, and a table. Two parallel rails are fixed on the carriage frame to offer the motion crossing the frame and to extend the distance of the robot in direction Y. In direction Y, the distance from the inner board to the outer board can be 900mm in one VV sector, i.e., the robot needs longer reach in this direction, and the linear drive unit helps the robot end-effector to reach the farther border of the VV. *v) Rotation drive unit:* This unit offers a rotation motion about the Z axis, so that the robot can machine the flexible houses on the inner wall at any position. The rotation drive unit consists of slewing bearing, drive gear, reduction unit CYCLO, and servo motor. The slewing bearing integrates bearing and gear together, leading to a compact structure with light weight. The rotation of the unit can reach $\pm 180^\circ$.

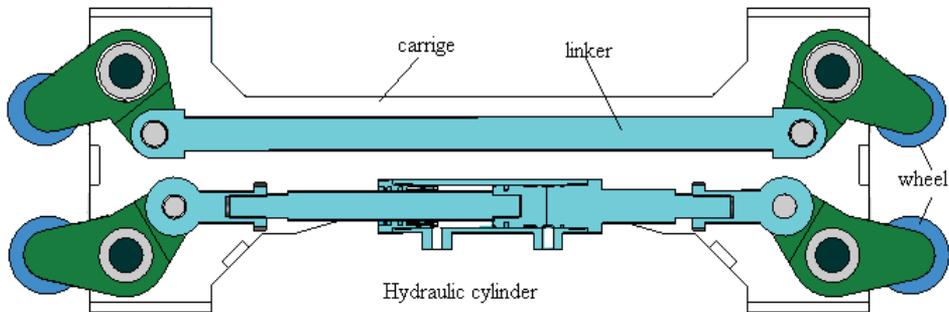


Fig. 4 Compensation mechanism

b). Hexa-WH

A Stewart based mechanism, driven by six servo control water hydraulic cylinders, offers six-degree freedom to the end-effector, where the machining head and welding gun are mounted. Because of the special shape of the VV, a full six-degree freedom motion for tool is needed to enable the robot to carry out welding and machining. The Hexa-WH can offer the required accuracy and the high force capacity due to its novel configuration and the hydraulic drive.

3.2 Kinematics model

The kinematics model is very important for the robot motion control. As the robot has redundant degree freedom, it is difficult to find the kinematics solution directly. The kinematics models can first be set up for the carriage and the Hexa-WH separately, and then be combined together by using an optimization algorithm in solving the redundant problem [4], [5].

a). Forward kinematics

As described above, the carriage offers the robot the four-degree freedom: two linear motions and two rotations; while the Hexa-WH offers the end-effector the full six-degree freedom. The transformation matrix of the robot can be defined as:

$$T_c = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5, \quad (2)$$

where

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & X_1 \\ 0 & 1 & 0 & Y_1 \\ 0 & 0 & 1 & Z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & X_2 \\ 0 & 1 & 0 & Y_2 \\ 0 & 0 & 1 & Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3 = \begin{bmatrix} c\phi & -s\phi & 0 & X_3 \\ s\phi & c\phi & 0 & Y_3 \\ 0 & 0 & 1 & Z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_4 = \begin{bmatrix} 1 & 0 & 0 & X_4 \\ 0 & c\phi & -s\phi & Y_4 \\ 0 & s\phi & c\phi & Z_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_5 = \begin{bmatrix} cac\beta & cas\beta s\gamma - sac\gamma & cas\beta c\gamma + sas\gamma & X_5 \\ sac\beta & sas\beta s\gamma + cac\gamma & sas\beta c\gamma - cas\gamma & Y_5 \\ -s\beta & c\beta s\gamma & c\beta c\gamma & Z_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Once the parameters of joints are given, the forward kinematics of the robot can be defined as

$$\bar{P} = T \bullet \bar{P}_0 = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \cdot \bar{P}_0. \quad (2)$$

To solve the forward kinematic model of the Hexa-WH, the numeric iterative method can be employed and it can be solved from its inverse kinematic model given later in the chapter.

b). *Inverse kinematic model of robot*

As the robot has four-degree freedom of redundancy, we give an inverse kinematic model first to the carriage, then to the Hexa-WH.

Inverse kinematic model of carriage: The inverse kinematic model of the carriage is defined to find the values of the four actuators with respect to the frame \mathcal{o} for a given position and an orientation of P_4 on the Hexa-frame. The principle of the carriage mechanism is shown in Fig. 5. In the application, rotation angle ϕ is fixed only at a few values, 0° , $\pm 90^\circ$, and 180° , and we can calculate the values of other actuators by fixing ϕ , i.e., for a given position $P_4(x, y, z)$, the centre of the Hexa-Frame, we have

$$X + (r_0 + r_1 \cos \varphi) \cos \phi = x, \tag{3}$$

$$Y + (r_0 + r_1 \cos \varphi) \sin \phi = y, \tag{4}$$

$$r_1 \sin \varphi = Z. \tag{5}$$

Then

$$\varphi = \arcsin(Z / r_1), \tag{6}$$

$$X = x - (r_0 + r_1 \cos \varphi) \cos \phi, \tag{7}$$

$$Y = y - (r_0 + r_1 \cos \varphi) \sin \phi. \tag{8}$$

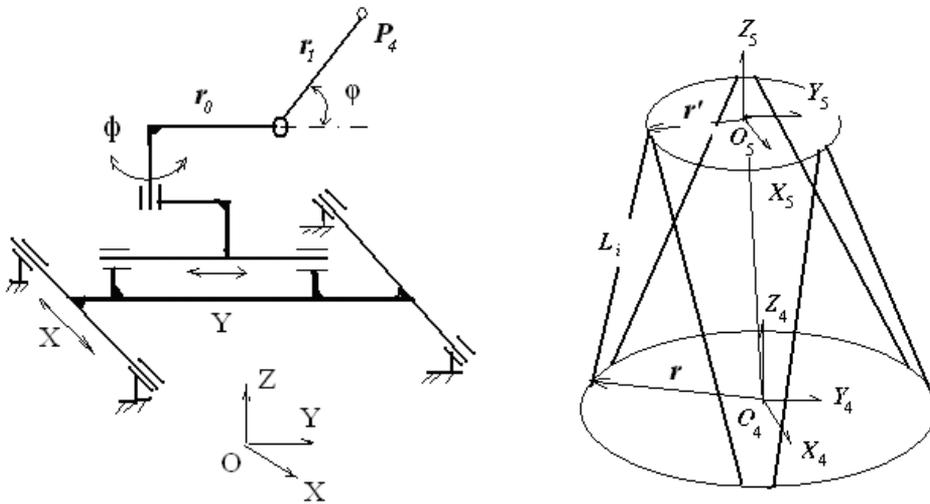


Fig. 5 a) Mechanism of carriage, b) Hexa-WH

Inverse kinematic model of Hexa-WH: The inverse kinematic model for the Hexa-WH is defined to find the values for each cylinder at a given position and an orientation of the end-effector with respect to the Hexa-frame. Here O_4 is coincided with P_4 on the carriage side. Fig.5 b) demonstrates the coordinates of the Hexa-WH.

The inverse kinematics model for the Hexa-WH is

$$\vec{L}_i = \vec{O_4O_5} + R \cdot \vec{r}'_i - \vec{r}_i \quad (i = 1, 2, \dots, 6), \tag{9}$$

where

$$R = \begin{bmatrix} cac\beta & cas\beta\gamma - sac\gamma & cas\beta c\gamma + sas\gamma \\ sac\beta & sas\beta\gamma + cac\gamma & sas\beta c\gamma - cas\gamma \\ -s\beta & c\beta\gamma & c\beta c\gamma \end{bmatrix},$$

\vec{r}_i denotes the vector of the joint of the i^{th} cylinder on the Hexa-frame with respect to frame o_4 and \vec{r}'_i is the vector of the joint of the i^{th} cylinder on the end-effector with respect to frame o_5 .

The length of each cylinder can be found, when $(x, y, z, \gamma, \beta, \alpha)$ is defined with respect to frame o_4 .

$$l_i = |\vec{L}_i| = \sqrt{(\vec{O}_4\vec{O}_5 + R \cdot \vec{r}'_i - \vec{r}_i) \cdot (\vec{O}_4\vec{O}_5 + R \cdot \vec{r}'_i - \vec{r}_i)} \tag{10}$$

There are two ways to combine the two inverse kinematic models to get the solution of the whole robot. One simple way is to calculate the coordinates $(x, y, z, \gamma, \beta, \alpha)$ of the end-effector with respect to $\{O_4\}$ and the values for each actuator from equations (3-10) for the given coordinates of the end-effector with respect to frame $\{o\}$, while fixing $\{O_4\}$ at a certain position with respect to frame $\{o\}$ according to experience. The another way is to use an optimization algorithm to find redundant solution, which is subjected to minimize the deflection of the robot during motion, i.e., $\min_f(\bar{q}, \bar{l})$, where f is the deflection model of the

robot, \bar{q} is the position vector of the end-effector, and \bar{l} is the value vector of ten actuators. For a given \bar{q} we can find \bar{l} by solving the optimization problem $\min_f(\bar{q}, \bar{l})$.

4. Control system

4.1 Hydraulic control system

Fig.6 shows the water hydraulic system. Pressure servo control is applied in locking cylinders. Position servo controller is used in cylinders 1-7. There are three loops in the servo position control: the position loop together with the speed loop that provide an accurate and fast trajectory tracking; and the load pressure feedback loop that is applied to damping the self-excited oscillations, which normally occur in the natural frequency. The speed loop can eliminate the speed error, while the pressure feedback damps the vibration of hydraulic actuators (Fig. 7).

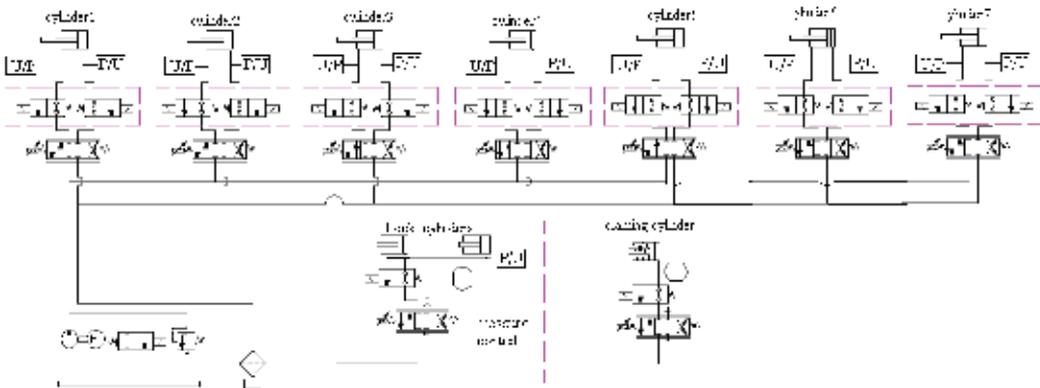


Fig.6 Water hydraulic circle

The hydraulic cylinders normally lack damping that makes the cylinder control difficult by using conventional PID-controllers. The damping can effectively be increased by means of load pressure feedback. The major drawback in using pressure feedback is its negative effect on the static stiffness of the actuator. To overcome this problem, the high pass filters are used in the load pressure feedback loops. The high pass filter removes the negative effect of pressure feedback at low frequency.

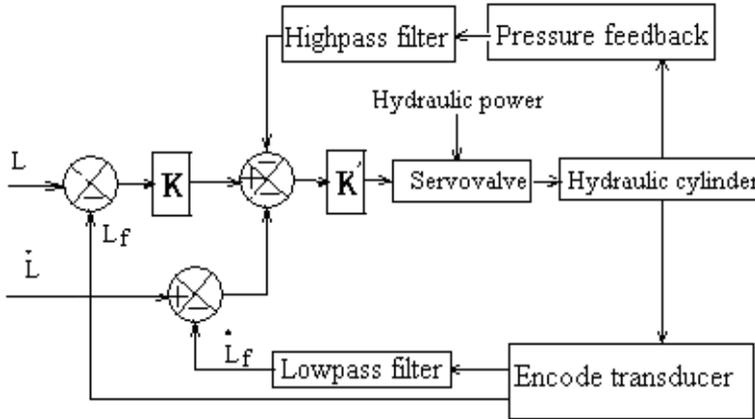


Fig.7 Hydraulic servo position control

4.2 Motor control

Two drive motors contribute effort for the tracking motion of the robot. As the tracking rails are not always straight, the speeds of the two motors are not the same when the robot is moving. The torque control together with a position feedback algorithm is implemented. Fig. 8 shows the control principle.

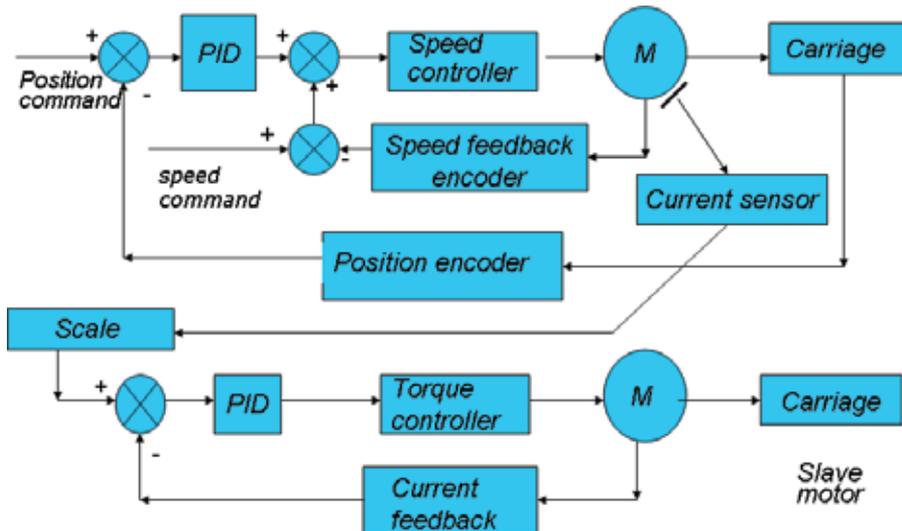


Fig. 8 Tracking motor control

In this method, one motor works as master, and another one works as slave. For the master motor, the position control plus the speed control is applied to guarantee the required speed and position accuracy of the carriage on the tracking rail. For the slave motor, the torque control is applied, which contributes the driving torque for the robot.

4.3 Control of hardware and software

Because there are no commercial controller and software available for the special functions of the parallel robot, an open architecture of hardware and programmable software are being developed. Fig. 9 shows the structure of hardware control system. The controller is an industrial-PC-based motion controller. It provides a reliable and easy-at-use environment for controlling the robot because Earthnet bus is used in the connection of iPC and I/O interfaces.

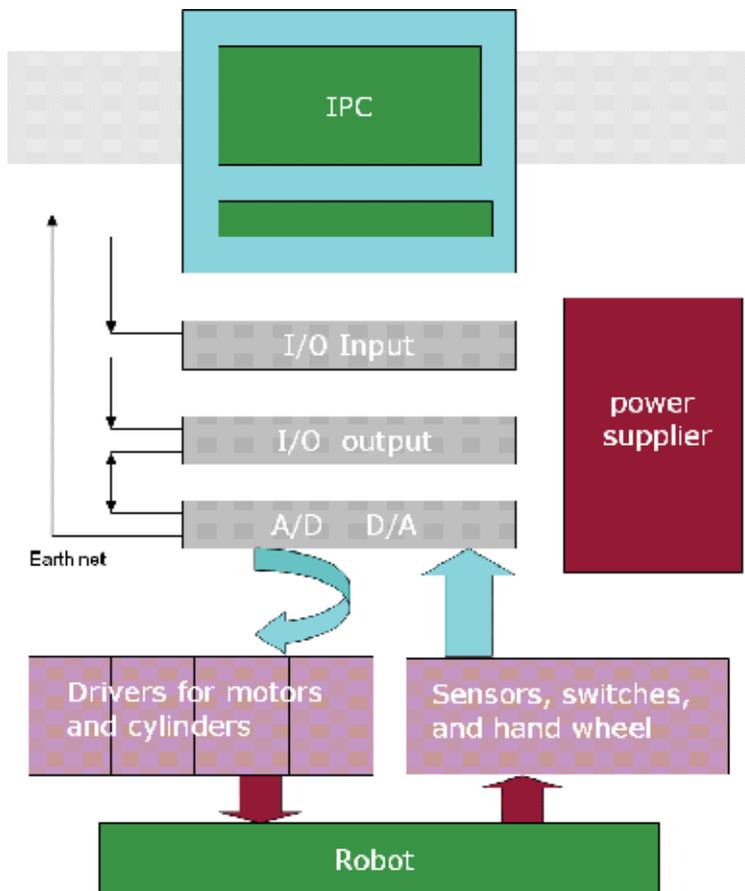


Fig.9 Structure of robot controller

The software is defined in Fig.10, including graphical interface, trajectory planning, forward and inverse kinematics models, interpolator, controller, and I/O interface functions. And

those functions have to be integrated with the program offered by iPC and run completely in real time.

Graphical interface is a high level program, it includes parameter setting, condition monitoring, and graphical visualization functions. User can easily exchange information with this program.

Trajectory planning is also a high level program. As the robot has redundant actuators, the trajectory planning is much more difficult than usual, so an optimization algorithm, which is subjected to minimize the deflection of the robot during motion, has been employed.

Forward and inverse kinematics models and interpolator are real time functions, which generate data for motion controller.

Controller is a real time function including water hydraulic controller and motor controller. As the robot has two tracking motors and the speed of the motors are not always the same at some positions, a master-slave control algorithm has been used.

I/O interface functions are real time functions, which enable transferring data from sensor to controller and from controller to driver.

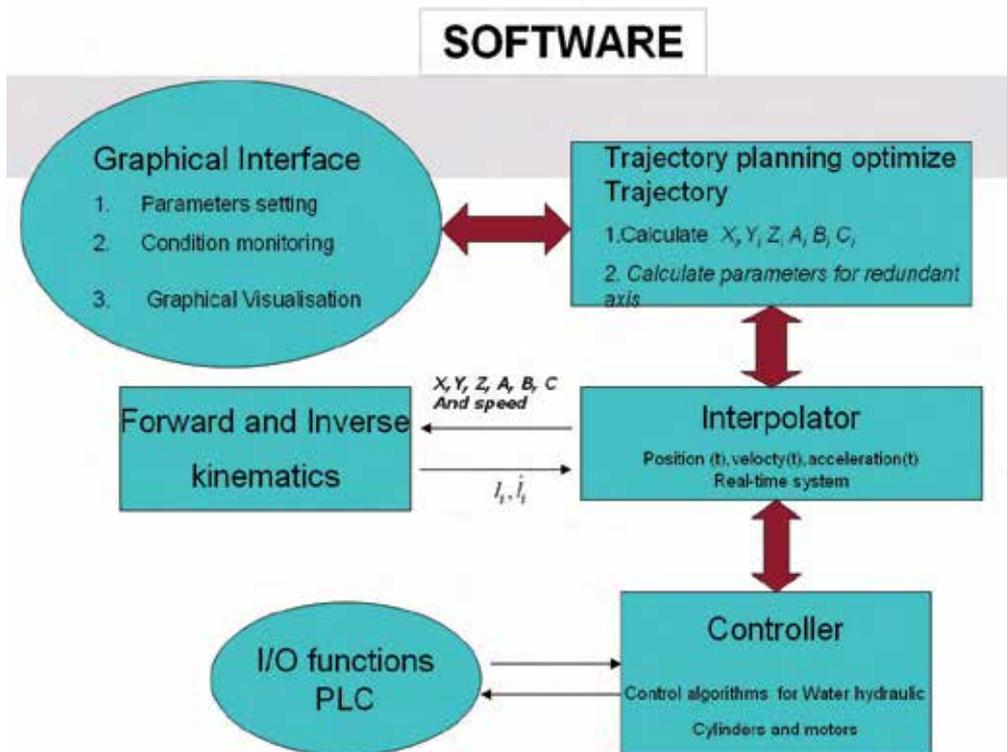


Fig. 10 Structure of software

5. Machining and welding testing mock-up

The parallel robot has been built in Lappeenranta University of Technology and the machining and welding test mock-up is designed shown in Fig.11. The mock-up is one

quarter of a sector built up for testing the machining and welding functions of the robot. Before the mock-up, some welding and machining tests have been carried out with the first prototype of the parallel robot named Penta-WH. The laser welding with seam tracker has been tested in stainless steel and the machine cutting with disc saw has been tested in the stainless steel machining process. The robot performs well in the tests.

5.1 Machining

The cutting test was carried out with stainless steel. The high speed steel cutting tool was used in the test. It is 200mm in diameter and 4mm in thickness. The problem in the experiment was that the high speed cutting tool wore out quickly during the cutting. The carbide tools, which are much more suitable for cutting stainless steel, are suggested to be used in the cutting operations in the ITER.

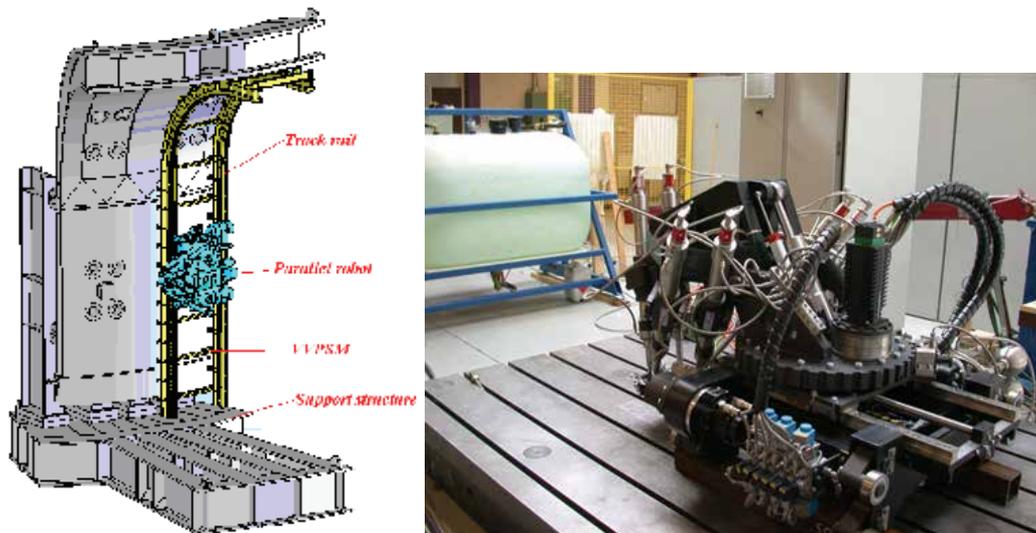


Fig.11 VVPSM mock-up and final version of parallel robot

5.2 Laser welding test

In the ITER assembly, the high quality welds are required to avoid the leak of tritium. They can be achieved by a highly automatic and remotely controlled welding procedure. To guarantee the welding quality, a seam tracker, which guides the robot motions along the center of a welding seam, is employed. With a seam tracker, the parallel robot has the capability to correct the motion trajectory on-line to keep the welding head at the right position and orientation in relation to the welding seam. The position errors of the welding head related to the welding seam caused by the distortion of material and the imprecise track rail are described in Fig.12.

During testing, the laser welding head is mounted on the end-effector of the parallel robot while the seam tracking sensor is mounted in the front of the welding head for guiding the robot welding (Fig.12). The work piece is assembled randomly in the y and z directions during testing. It has approximately a one-degree angle about the Y and Z-axes. The position of seam is unknown for the robot before the seam finding. The maximum output power used in the testing is 3 kW YAG. It has a 200mm focal length resulting in a 0.6 mm diameter focal spot on the work piece. The beam parameter product is 25 mm-mrad. The work piece is made of stainless steel AISI 316LN. It has a 7mm-thickness, 600mm-length, and 0.2mm-root gap for welding.

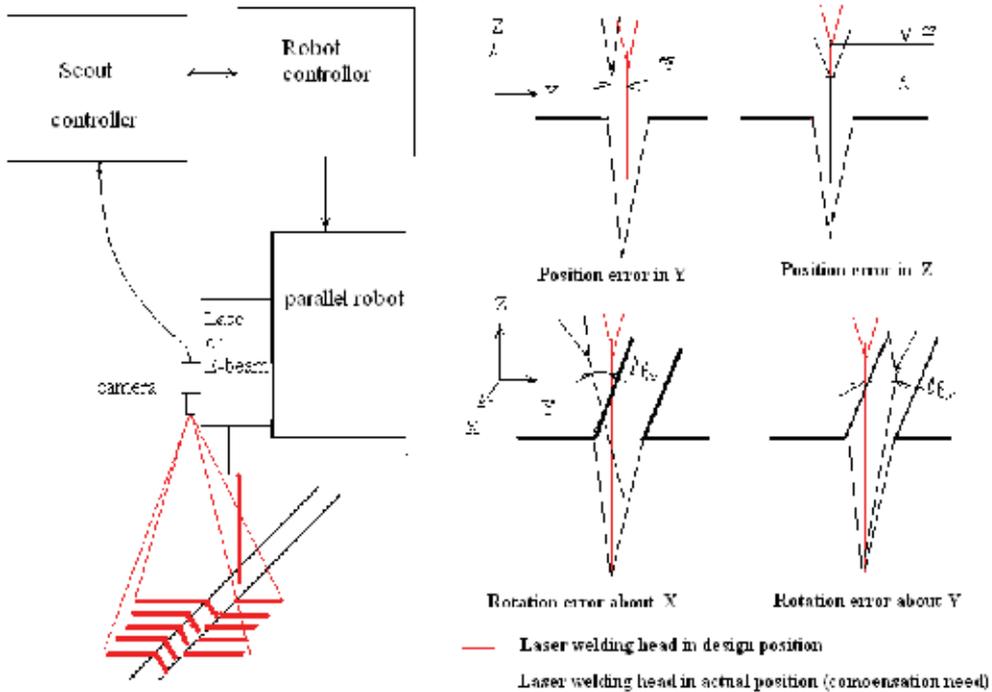


Fig.12 Principle of seam tracker

Two seam tracking algorithms were tested during laser welding trials. One is the off-line teaching algorithm that has two steps: (i) the robot follows the planned trajectory of the seam and records the data from the seam tracker; and after that (ii) the robot compensates the motion trajectory from the data recorded and starts welding following the new motion trajectory. Fig.13 shows the welding results. The second algorithm is the on-line teaching algorithm. In the algorithm, the robot corrects the trajectory on-line using data from the seam tracker during the welding motion. Fig.14 shows the welding results achieved during the tests. From the test results, we can conclude that the on-line teaching algorithm is better than the off-line teaching algorithm, because the on-line teaching algorithm compensates the distortion of the work piece during the welding process.

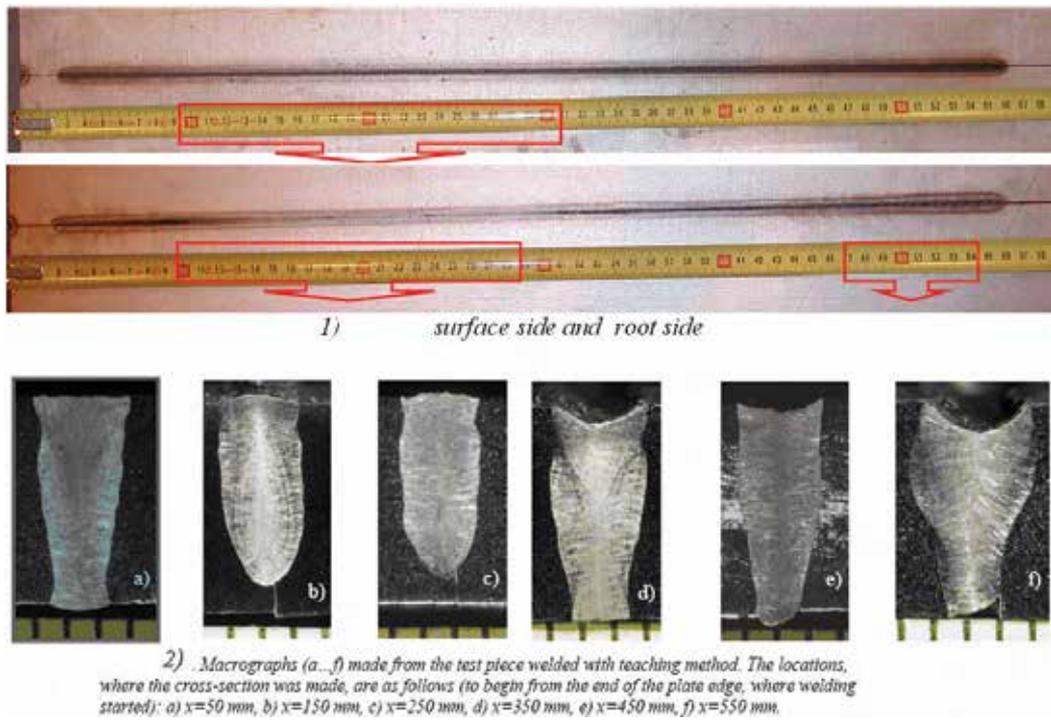


Fig.13 Off-line teaching welding results

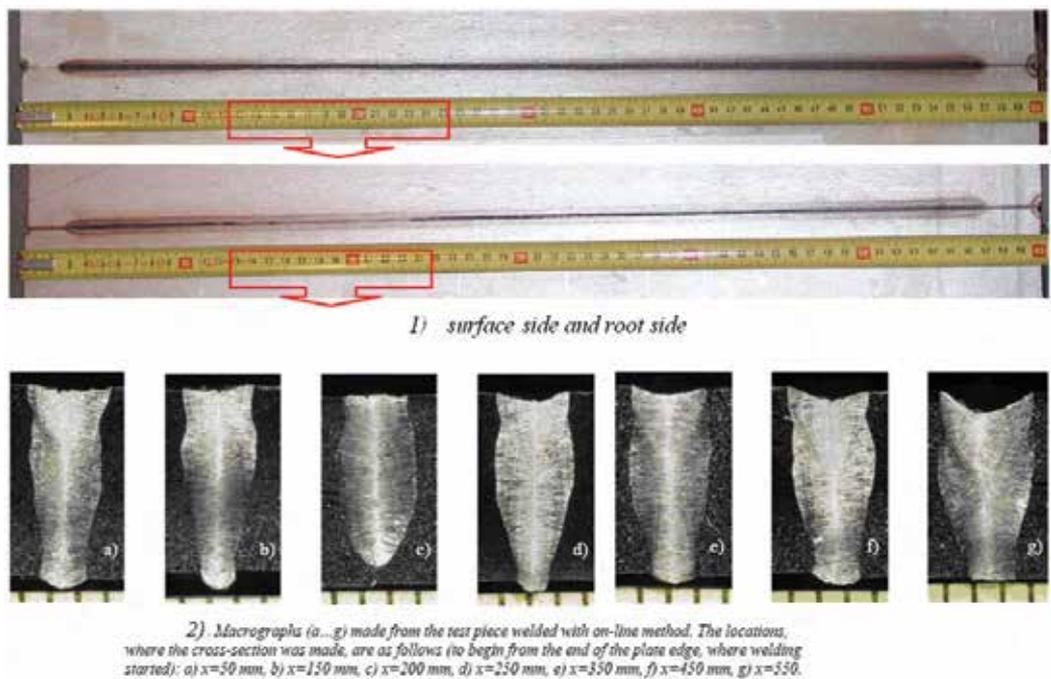


Fig.14 On-line tracking welding results

6. Conclusion

A hybrid parallel robot with four additional serial motion axes is developed for carrying out the necessary machining and welding tasks in the assembling and repairing of the ITER Vacuum Vessel. The robot is capable of holding all necessary machining tools and welding end-effectors in all positions accurately and stably. The kinematics analysis of the robot is presented. The models are complex because of the redundant structure of the robot. The models are separately derived for the Hexa-WH and the carriage mechanism. An optimization algorithm finds the solution in the trajectory planning, ensuring the maximum stiffness during the robot motion. The experiments of the laser welding tests with the seam tracker have been carried out. Both the on-line and off-line teaching algorithms have been developed and the results show that the online teaching algorithm is better. The machining cutting tests with stainless steel have been tested. The entire design and testing process of the robot is a very complex task due to the high specialization of the manufacturing technology needed in the ITER reactor. The results demonstrate the applicability of the proposed solutions quite well.

7. Acknowledgement

The laser welding test is carried out in collaboration with the laser laboratory of VTT in Lappeenranta, Finland, the whole work, supported by the European communities under the contract of Association between EURATOM/Finnish TEKES, was carried out within the framework of the European Fusion Development Agreement, and the views and opinions expressed herein do not necessarily reflect those of the European Commission.

8. References

- D. Stewart (1965), A platform with six degree of freedom, Proc. Inst. Mech. Eng., London, Vol. 180, pp. 371-386.
- T. C. Arai, K. Homma, H. Adachi, and Nakamura (1991), Development of parallel link manipulator for underground excavation task, Proc. Int. Symposium on Advanced Robot Technology, pp. 541-548.
- C. Gosselin and J. Hamel (1994), The agile eye: A high-performance three degree of freedom camera -orienting device, Proc. IEEE Int. Conference on Robotics and Automation, pp. 781-786.
- L. W. Tsai (1999), Robot analysis: The mechanics of serial and parallel manipulators, A Wiley-Interscience Publication, John Wiley & Sons Inc.
- G. Lebret, K. Liu, and L. Lewis (1993), Dynamic analysis and control of a Stewart platform manipulator, J. Robotic Systems, Vol. 5, No. 10, pp. 629-655.
- G. Zheng, L. S. Haynes, J. D. Lee and R. Carroll(1992), On the dynamic model and kinematic analysis of a class of Stewart platforms, Robotics and Autonomous Systems, Vol. 9, pp. 237- 254.
- K. H. Häfele, H. Haffner, and P. Spencer (1992), "Automatic Fetting Cell- An Example for Applying Computer- Aided Robotics", Industrial Robot. Vol.19 No. 5. pp.31-34

- H.Wu, H. Handroos, P. Pessi, J. Kilkki, L. Jones (2005), "Development and control towards a parallel water hydraulic weld/cut robot for machining processes in ITER vacuum vessel", Fusion Engineering and Design.
- H. Wu, H. Handroos (2006), Mechatronics design and development towards a heavy-duty waterhydraulic welding/cutting robot in Book: Mechatronics for Safety, Security and Dependability in a New Era. Elsevier.

Architecture Design and Optimization of an On-the-Fly Reconfigurable Parallel Robot

Allan Daniel Finistauri, Fengfeng (Jeff) Xi and Brian Petz
*Ryerson University, Department of Aerospace Engineering
Canada*

1. Introduction

Throughout the ages, human beings have evolved in various aspects of society including politics, technology, etc. This is true of the age of mechanization in which rudimentary machines were developed and controlled by operators. The age of mechanization then gave down to the age of automation, where the operators were replaced by controllable machines. Humans have since taken on a passive supervisory role and less of an active control of these machines and the mechanized world is dawning on the age of autonomization. In this age, humans will regress further from their involvement with the daily activities of machines as the advancement of embedded computers, smart sensors and intelligent controllers will enable machines to operate autonomously. These include systems embedded with artificial intelligence and those that are able to change their structural configurations. Under different circumstances, such systems would adapt to changes in their surrounding environment by autonomously altering their configuration and function. The advantages of developing these intelligent reconfigurable systems include adaptability, reusability, convertibility, compactness, fault-tolerance and emergency behaviour (Koren et al., 1999).

Research into reconfigurable systems is primarily active in robotics. The main idea of developing reconfigurable systems is based on the use of modular components as building blocks (Yim et al., 2002). Several interesting modular reconfigurable robots have been proposed, and they may be classified into three categories: *manual-configuring*, *self-configuring*, and *self-assembly*.

Manual-configuring robots are often referred to as modular robots. They can only be reconfigured with some form of manual assistance and represented the first steps leading into the age of autonomization. The modular units are usually built with the embedded controllers connected to a host computer. The host computer has the ability to quickly recognize new configurations and automatically generate the kinematic and dynamic equations for control. Examples include the work at Stanford (Yim, 1994), and Carnegie Mellon University (CMU) (Unsal et al., 2000). More recently, a reconfigurable manufacturing cell was developed based on modular robotic components (Chen, 2001). His research team demonstrated that by using modular components they were able to quickly construct a parallel kinematic machine for machining and several serial robots for material handling.

Self-configuring robots cannot perform self-assembly. However, they can perform reconfiguration after the robotic system is assembled with some form of manual assistance, making further progress into the age of autonomization. Michael (Michael, 1995) developed what he called the fractal shape changing robot based on robotic cubes. These cubes are built with embedded active driving mechanisms. Once attached manually, these cubes can slide on each other's faces for reconfiguration. Since the cubes are made in different sizes and can be combined together, the robot is called the fractal shape changing robot.

Self-assembly robots are the robots with the highest level of reconfigurability because they can detach from and attach into a robotic system automatically. For example, a self-assembly robotic system was developed that uses electro-magnetic disks as the basic units that can attract and repel each other through computer control for automatic reconfiguration (Tomita et al., 1996). Reconfigurable systems like these and those with artificial intelligence represent what will define the age of autonomization.

As space flight and exploration is costly, reconfigurable systems for space applications have also been explored to provide more cost effective solutions to an array of problems. A two-dimensional foldable hexapod truss for space deployment was proposed (Onoda et al., 1996). A modular parallel robot, called TETROBOT, was designed based on tetrahedron modules that can be reconfigured for different applications (Hamlin and Sanderson, 1997). Variable geometry truss manipulators (VGTM) have been studied as reconfigurable support structures for space applications (Horner, 1990). Reconfigurable mobile robots have been researched by a research team from MIT, JPL and CMU (Schenker et al., 2000). A light weight modular robot was recently proposed from MIT (Hafez et al., 2003). Space flight and exploration is definitely an area that will prosper greatly from the emergence of autonomization.

This chapter focuses on the development of a reconfigurable parallel robot capable of *on-the-fly* self-reconfiguration. The main idea is to utilize the modularity inherent in the parallel robots, which has previously been overlooked. Since most parallel robots are made of identical kinematic chains arranged in parallel fashion, all the components (mechanical and electrical) used in each individual kinematic chain, hereafter called branch, are the same. This provides a natural base for the development of reconfigurable systems. Also, the capability of on-the-fly self-reconfiguration represents inroads to autonomous parallel robotic systems.

The rest of the chapter is organized as follows. First, the design methodology is presented. A top-down system decomposition method is used to identify the modules required to build the reconfigurable system from the bottom-up. By using this top-down subsystem decomposition approach, the robotic system is decomposed from the system level to separate individual robots, then to subsystems and finally to the component level. To facilitate the design methodology, potential branch configurations are identified using the modular components and a mobility analysis is performed to identify the system constraints. Enumeration rules are established to eliminate the unacceptable branches for each base tripod based on the kinematic constraints required for reconfiguration. A parametric model is established to solve the constraint equations. Individual branch kinematics are established and the loop equations are solved. A workspace analysis is performed and a discrete system optimization yields the optimal parallel robot configuration.

2. Design methodology

The proposed reconfigurable parallel robot is constructed by two base tripods, each being a three degrees-of-freedom (DOF) parallel robot. The first tripod remains fixed to the moving platform, and the links of the second tripod are designed to be detachable from the moving platform. The detachable branches are then reconfigured into 2-DOF serial robotic arms. Fig. 1. shows a 6-DOF parallel robot that has been decomposed into two tripods; one with a typical moving platform, and one which has branches detached from the moving platform. By detaching one, two or all three branches of the second tripod, separately, the parallel robot can be reconfigured on-the-fly from 6-DOF to, 5, 4 or 3-DOF, respectively. Additionally, the detached links can then be used to perform collaborative work with the remaining parallel robot.

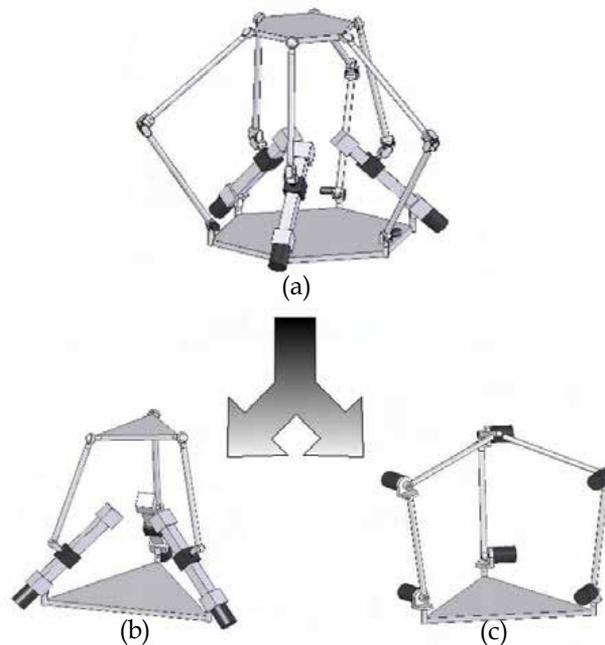


Fig. 1. A (a) 6-DOF parallel robot is decomposed into two base tripods: (b) fixed tripod, (c) detachable tripod

The proposed reconfigurable parallel robot not only provides innovation in autonomous reconfigurable system design but also stimulates new research of parallel robot kinematics. Since traditional parallel robots are not reconfigurable, the kinematics have been studied on a case-by-case basis for the particular parallel robot type, and have generally been restricted to 3-DOF and 6-DOF parallel robots. Also, reconfigurable robotic systems tend to require a halt in operations to allow for the robot to reconfigure itself either by itself or by some type of manual assistance. Here, there is no down-time for reconfiguration as the parallel robot is capable of on-the-fly self-reconfiguration. This enables the robot to quickly adapt to changes in its environmental surroundings as well as changes to task requirements. This single feature is extremely important in distinguishing the final design with other reconfigurable robots in service, where in-task adaptability and reconfiguration are not commonly found.

2.1 Design for reconfiguration

As mentioned in Section 1, the basic idea of designing any reconfigurable system is based on the use of modular components as building blocks. The design methodology used is based on the methodology called *Design for Reconfigurability* (DfR). Based on DfR, the design of a reconfigurable system can be described by the Axiomatic Design Theory (Suh, 1990) such that designing a reconfigurable system is defined as the minimization of the number of modules {Design Parameters} while maximizing its functionality {Function Modes}. Further discussion of this system design methodology can be found in (Chen et al., 2005).

A summary of function modes is summarized in Table 1. These function modes are in fact the design objectives, i.e. the reconfigurable parallel robot must be capable of performing the prescribed functions.

Function Mode	Robot Configuration	
	Parallel Configuration	Auxiliary Configuration
FM1 = Fully Attached	6-DOF	None
FM2 = Partially Detached	5-DOF	1 2-DOF serial arm
FM3 = Partially Detached	4-DOF	2 2-DOF serial arms
FM4 = Fully Detached	3-DOF	3 3-DOF serial arms
FM5 = Coordinated	3-DOF	6-DOF gripper
FM6 = Fully Detached	VGTM	3 3-DOF serial arms
FM7 = Fully Detached	VGTM	6-DOF gripper

Table 1. Function modes and robot configurations

2.2 System decomposition

As a series of modules are required as building blocks for a reconfigurable system, the first step is to decompose the parallel robot into a series of common modules. For serial robots, the decomposition into common modules is generally straightforward as serial robots are a collection of links and joint actuators. These modules are then connected sequentially to build the final robot architecture. For parallel mechanisms, the decomposition of the robot system is not as straightforward due to the physical and mathematical constraints of the system. There does exist though a natural modularity that has generally been overlooked when dealing with parallel robots as will be shown.

From Fig. 1., a 6-DOF parallel manipulator system is separated into two individual 3-DOF tripods subsystems. Each tripod subsystem can further be decomposed into a series of branch subsystems with a common structural subsystem. The 6-DOF robot has only one base and one moving platform, thus the decomposition of the two tripods results in these also having a common base and moving platform. Without losing any generality, the branch subsystems for the two tripods are composed of a collection of link modules, passive joint modules and active joint modules (i.e. actuators). These modules represent the final level of decomposition required to describe a generic parallel robotic system. Further decomposition is also possible and this is where the detailed design and part selection occur and is beyond the scope of this chapter.

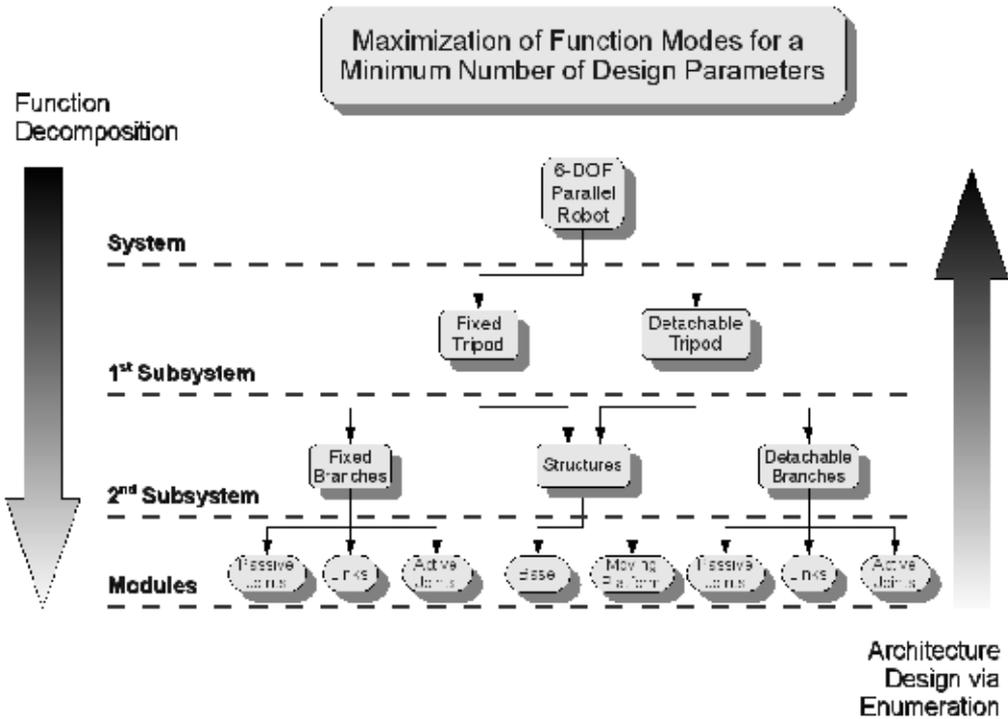


Fig. 2. System design cycle of a reconfigurable parallel robot

This decomposition is further explained in Fig. 2. It is plain to see that the arrow on the left side of the figure indicates the direction taken for the system decomposition. The arrow on the right side of the figure is where the majority of the architecture design occurs. Once the building blocks (modules) of the reconfigurable system have been identified, then we can work our way from the bottom-up to establish the optimal system architecture. This is accomplished by first using the modules to form branch module candidates. A mobility analysis is performed and enumeration rules are used to eliminate those branch candidates that cannot fulfill the design requirements. A kinematic and workspace analysis is performed and then is used to arrive at the final optimal architecture design of the parallel robot. All of this is performed such that the final design can perform all of the function modes identified in Table 1.

We note that each level of decomposition brings an additional level of modularity. The physical modularity was described above. During the architecture design, the modularity inherent in the assembly of reconfigurable robots will be address. We also note that there is modularity in the mathematical computations and control for each system level. The kinematic computations for 6-DOF parallel manipulators, 3-DOF tripod manipulators, open-chain branches (including simple chains consisting of one joint) are well established. This is also true of their subsequent control laws and algorithms. Although this is beyond the scope of this chapter, it is a very important aspect of the advancement of reconfigurable systems.

3. Module identification

The module identification stage is the first and second part of the bottom-up architecture design as seen in Fig. 2. The identification of the components is the first and the identification of the branch configurations is the second. For reconfigurable systems, the larger the cache of building block modules, the larger the solution space with a greater diversity of possible solutions.

3.1 Components

3.1.1 Active joint modules

Active joint modules are the modules that are controllable. Currently, there are numerous commercially available simple actuation devices (having 1-DOF). They are categorized as rotational (revolute), or linear (prismatic). The topographical analysis (Tsai et al., 1998) uses these two categories of actuation devices to enumerate the configurations of some planar and spatial parallel manipulators. The revolute joint was decomposed into the standard rotational joint and a twist joint (Dash et al., 2005). Hereafter we will refer to these joints as transverse revolute joints (R_T), and axial revolute joints (R_A), respectively. We similarly decompose the prismatic joint into a fixed-length actuator (P_F) where a platform slides along a fixed guide track, and a variable length actuator (P_V) as most commonly seen in Gough-Stewart platforms. All four of these actuation devices are commercially available and are included in the identification of feasible branch modules. We also introduce a universal joint (U^*) that has one controllable DOF and one passive DOF as a possible active joint module. Kinematically, it is represented by the presence of two revolute joints whose axes intersect at a point and are orthogonal to each other. Physically one axis is attached to an actuation device.

3.1.2 Passive joint modules

The active revolute joint modules and prismatic joint modules are also identified as passive modules by removing their ability to be controlled. The other common passive joint modules are identified as universal (U), spherical (S), and cylindrical (C).

3.1.3 Link modules

The link modules are simply a means of connecting the active and passive joint modules to each other in series. These can vary in appearance and length depending on the task requirements, but those parameters are left for the detailed design phase, which is beyond the scope of this chapter.

3.1.4 Structural components

The structural components of a parallel manipulator consist of the base and moving platform. The size and shape of these components vary depending on the task requirements but must be designed so that the base supports the various branches and the platform supports the end effector. Again, the specifics are left for the detailed design phase and are beyond the scope of this chapter.

3.2 Branch identification

Using a combinatorial analysis, the branch configurations can be enumerated for their potential feasibility as either a fixed branch or a detachable branch or both. In general each

branch in a spatial parallel manipulator must consist of at least two links and three joints. Branches can consist of any number of joints and links such that the total branch DOF meets the mobility requirements. For a 6-DOF parallel manipulator with six branches, the branch DOF must be equal to six (more information on this is covered in the mobility analysis). The combinatorial analysis is limited to those branches that have two links and three joints for the following reasons:

- Smaller branches (those with fewer joints and links) are easier to evaluate mathematically. With additional joints, there exists the possibility of multiple solutions for the forward and inverse kinematics of the active and passive joint variables. This situation is less likely, and sometimes impossible, for two-link, three-joint branches.
- In both attached and detached configurations, they provide the minimal amount of joint-link combinations to maintain functionality. This will become more apparent during the architecture design phase.
- Branches with a large numbers of links and joints require more physical constraints when converting from attached to detached configurations, thus making the structure itself more physically complicated. This is especially true in the case of individual detached arms. This is a direct result of the configurations presented in Table 1 and will also become apparent during the architecture design phase.
- The fewer number of joints within the individual branches leads to a lesser chance of collision between the branches.

Using the five active joint modules and the seven passive joint modules a total of 78 branch configurations are identified as being theoretically possible. The only restriction placed on joint sequence is for the fixed-length prismatic joint in that it must either be placed at the base or platform position due to the structural advantages of having a rigid connection of the track. If it were to be placed as the middle joint, then it would act as a variable length prismatic joint and lose all of its structural advantages. Using the notation stated above, Table 2. summarizes the various configurations.

Active Joint	Configurations
R_T	$R_TUS, R_TSU, UR_T S, SR_T U, USR_T, SUR_T, R_TCS, R_TSC, CR_T S, SR_T C, CSR_T, SCR_T$
R_A	$R_AUS, R_ASU, UR_A S, SR_A U, USR_A, SUR_A, R_ACS, R_ASC, CR_A S, SR_A C, CSR_A, SCR_A$
P_F	$P_FUS, P_FSU, USP_F, SUP_F, P_FCS, P_FSC, CSP_F, SCP_F$
P_V	$P_VUS, P_VSU, UP_V S, SP_V U, USP_V, SUP_V, P_VCS, P_VSC, CP_V S, SP_V C, CSP_V, SCP_V$
U^*	$U^*UU, UU^*U, UUU^*, U^*R_T S, U^*SR_T, R_TU^*S, SU^*R_T, R_TSU^*, SR_TU^*, U^*R_AS, U^*SR_A, R_AU^*S, SU^*R_A, R_TSU^*, SR_AU^*, P_FU^*S, P_FSU^*, SU^*P_F, U^*SP_F, U^*CU, U^*UC, CU^*U, UU^*C, CUU^*, UCU^*, U^*P_V S, U^*SP_V, P_VU^*S, PU^*P_V, P_VSU^*, SP_VU^*, U^*CC, CU^*C, CCU^*$

Table 2. Branch configurations

4. Architecture design

The enumeration part of the design serves the purpose of defining what is deemed acceptable candidates for the fixed and detachable tripods. A mobility analysis is done to provide a link between the identified branches and the mobility requirements of both tripods and is important for the formation of many of the enumeration criteria.

4.1 Mobility analysis

From Fig. 1. and Table 1., it can be seen that the reconfiguration of the robot will change the robot constraints. For example, going from an attached to detached configuration, the robot must change its constraints in order to constrain the freedom released by the detached branch(es). Otherwise, the robot would be loose and uncontrollable. Hence, in order to understand how the robot constraints change during reconfiguration, a mobility analysis is required. As will be explained later on, solving the constraint equations is a priori to solving the inverse kinematics.

In general, the reconfigurable parallel robot under study can be categorized to have attached and detached configurations. The mobility requirements are thus different for different configurations. In the attached configuration, the parallel robot is a 6-DOF parallel robot. The mobility of a system is given by the following equation

$$F = \lambda(n_l - n_j - 1) + \sum_{i=1}^{n_j} f_i \quad (1)$$

where F denotes the mobility or the effective DOF of a parallel mechanism, λ is the order of the system ($\lambda = 3$ for planar motion, and $\lambda = 6$ for spatial motion), n_l is the total number of the links, n_j is the total number of the joints, and f_i is the number of DOF for the i^{th} joint.

For a parallel manipulator, the branch connectivity can be calculated using Euler's equation. Through some mathematical manipulation it can be shown that the sum of the connectivity, C_k , of the k^{th} branch is equal to the total DOF of the system

$$\sum_{k=1}^{n_b} C_k = \sum_{i=1}^{n_j} f_i = F - \lambda(n_l - n_j - 1) \quad (2)$$

where n_b is the number of attached branches. Further manipulation shows that the connectivity of each branch must be less than or equal to the order of the system, and it must be greater than or equal to the mobility of the moving platform

$$\lambda \geq C_k \geq F \quad (3)$$

The full derivation has previously been derived and can be found in (Tsai, 1998). Table 3. shows a summary the mobility analysis for the various robot configurations, including the connectivity of the k^{th} branch.

Variable	Symbol	Parallel Robot Configuration			
		3-DOF	4-DOF	5-DOF	6-DOF
System Order	λ	6	6	6	6
Degrees-of-Freedom	F	3	4	5	6
Number of Links	n_l	8	10	12	14
Number of Joints	n_j	9	12	15	18
Branch Connectivity	C_k	5, 5, 5	6, 6, 5, 5	6, 6, 6, 6, 5	6, 6, 6, 6, 6, 6
Number of Constraints	m	3	2	1	0

Table 3. Mobility analysis summary

4.2 Enumeration criteria

With the branch configurations identified and connectivity constraints established, the enumeration process can now be performed to eliminate some of the branch configurations. Since there are two tripods which are functionally different, there are two sets of enumeration criterion for the elimination of branch configuration. There is some overlap in branch elimination criteria between the two tripods and these are addressed first followed by the tripod-specific enumeration rules.

4.2.1 Fixed and detachable tripod enumeration criteria

The active joint must be placed on, or near the base. This requirement is what generally gives parallel robots their payload-to-weight advantages. If the active joints (i.e. motors) are placed at or near the base, then the majority of mass/inertia to be driven is in the platform and end effector. All configurations with the active joint at the platform are eliminated.

A spherical joint must be located at the moving platform. As will be shown later, the presence of a spherical joint in the branch is most advantageous if it is located at the moving platform. It provides a natural pivot point for the moving platform. Thus the elimination of all branches without a spherical joint, and those with spherical joint modules at the base or middle position is necessary.

In the fully connected configuration, the motion profile for all branches must be spatial. In the fully detached configuration, the motion profile for both the individual fixed and detached tripod branches must be planar. Although these may seem obvious, it helps in the elimination of some of the branch configurations that are not capable of these mobility requirements. For the fully detached configuration and those branches with kinematic constraints in the partially detached configurations, the plane of motion of the branch must orthogonal to the base and parallel to a plane passing through the joint at the base, and the base joint directly opposite to it. This eliminates all branch configurations with an active or passive axial revolute joint module.

After these enumeration criteria are applied, a total of 15 configurations remain as possibilities for the fixed and detachable tripod branches which are summarized in Table 4.

Active Joint	Configurations
R_T	$R_TUS, UR_T S, R_TCS, CR_T S$
P_F	P_FUS, P_FCS
P_V	$P_VUS, UP_V S, P_VCS, CP_V S$
U^*	$U^*R_T S, R_TU^* S, P_FU^* S, U^*P_V S, P_VU^* S$

Table 4. Acceptable fixed and detachable branch configurations after applying initial enumeration criteria

4.2.2 Fixed branch enumeration criteria

Fixed branches must have one lockable DOF. As seen in Table 3., the connectivity requirements for the fixed branches change according to the number of the branches that are either attached or detached from the moving platform. A fully attached parallel robot configuration requires each branch to have a connectivity of 6-DOF and a fully detached parallel robot configuration requires each branch to have a connectivity of 5. Thus it is

required that there exists a joint that has a lockable DOF. The lockable DOF must exist on a joint with 2-DOF for the following reasons:

- If a single DOF joint is locked, it then forms a rigid bond between the two link modules that it is attached to, thus reducing the number of links in the branch from two to one. One link does not allow for proper articulation of the moving platform and therefore single DOF joints cannot be locked.
- For the 3-DOF spherical joint, it is possible to lock out one of the DOF, but is not necessarily easy. Since, the spherical joint is positioned at the moving platform, locking one of these DOFs will cause the branch to have spatial motion, which as previously mentioned as unacceptable.

From this, there are three possible joint modules that are candidates for a lockable DOF; one axis of the passive universal joint module; the revolute axis of the passive cylindrical joint module, or; the passive axis of the 1 DOF controllable universal joint module. Although this rule does nothing to eliminate branch configurations, it is important to establish this criterion when it comes to the physical design of the robot itself.

Fixed-length vs. variable length prismatic joints. For structural considerations, having a fixed-length prismatic joint at the base is more advantageous than having a variable length prismatic joint. We thus eliminate the P_VUS , P_VCS , and P_VU^*S branches.

Branches with identical modules, but different sequences. One of the previous enumeration criteria was that the active joint module and thus motor should be placed at the base or close to it (i.e. the second joint position). There are several remaining branch configurations that have the same joint modules, but vary in sequence. Again, the advantages of keeping the motor on the base itself as opposed to at the second joint enables the elimination of those branch configurations that have identical modules and the active joint module in the middle. Thus the $UR_T S$, $CR_T S$, and the $R_T U^* S$ configurations.

As seen in Table 5., this does not eliminate all of the configurations with an active module in the middle joint position, rather just the ones that are less advantageous. A total of nine branch modules remain as candidates for the fixed branch tripod. Also shown is the configuration required for the branch(es) after reconfiguration into the 3, 4 or 5-DOF configurations. It is seen that there are six unique configurations after reconfiguration.

Active Joint	Configurations		
R_T	$R_T U S \rightarrow R_T R_T S$		$R_T C S \rightarrow R_T P_V S$
P_F	$P_F U S \rightarrow P_F R_T S$		$P_F C S \rightarrow P_F P_V S$
P_V	$U P_V S \rightarrow R_T P_V S$		$C P_V S \rightarrow P_V P_V S$
U^*	$U^* R_T S \rightarrow R_T R_T S$	$P_F U^* S \rightarrow P_F R_T S$	$U^* P_V S \rightarrow R_T P_V S$

Table 5. Potential fixed tripod branch module configurations

4.2.3 Detachable branch enumeration criteria

Detachable branches must transform from a closed loop 6-DOF connected arm, to a 2-DOF, serial arm. To maintain usability of the detached arms, and maintain the requirement of planar motion in the fully detached or partially detached configurations, there must be two controllable axes. Since the arm will detach from the spherical joint module connection, there is still a total of 3-DOF and two links. One of these DOF is already controllable, so to satisfy the requirements, one of the other axes must be controllable, and the other lockable.

For proper articulation, the control must be present at each joint location. This is summarized in Table 6. where the reconfiguration of the detachable arms are shown. It is seen that after reconfiguration, several of the branches are kinematically identical, but are not physically identical. The reconfiguration requires that passive universal joints become active transverse revolute joints, passive revolute joints become active while the passive axis on the 1-DOF controllable universal joint locks, and the passive cylindrical joint becomes an active variable length prismatic joint. Although, this enumeration criterion does not eliminate any branch modules, it is important as it establishes the kinematic and physical requirements that each branch must adhere to after reconfiguration.

Initial Active Joint	Configurations
R_T	$R_TUS \rightarrow R_TR_T, UR_T S \rightarrow R_TR_T, R_TCS \rightarrow R_TP_V, CR_T S \rightarrow P_VR_T$
P_F	$P_FUS \rightarrow P_FR_T$
P_V	$UP_V S \rightarrow R_TP_V$
U^*	$U^*R_T S \rightarrow R_TR_T, R_TU^* S \rightarrow R_TR_T, P_FU^* S \rightarrow P_FR_T, U^*P_V S \rightarrow R_TP_V$

Table 6. Reconfiguration of the detachable tripod branch module configurations

Detachables branches must have acceptable reach beyond the height of the moving platform. It is obvious that the detachable branch must be able to reach the moving platform, but here we require that they extend beyond the position of the platform for greater usability. Although this requirement is ambiguous and there is no clear definition of what is acceptable, we eliminate those branches that have prismatic actuation after disconnection. It is clear to see that a 2-DOF robotic arm with two revolute joints has a larger potential reach than those with prismatic joints. The notion of potential reach is based on the length of the links and those links connected to revolute joints are traditionally longer in parallel manipulators than their prismatic counterparts.

Branches with identical modules, but different sequences. The four branches that reconfigure into the R_TR_T configuration are acceptable as candidates for the detachable branches, and all four cases require the second joint to be independently actuated. In the attached configuration however, only one joint is driven and therefore in this configuration it is beneficial to drive the joint at the base. After elimination, the only branch configurations that are candidates for the detachable tripod are the R_TUS and the $U^*R_T S$.

Motor placement. With only two branch configurations remaining, the placement of the joint motors is the final enumeration criteria. Previously, we required that the motor be placed at or near the base for the primary driven motor. In this case, there is a requirement that the first and second joint be driven when detached from the platform. In the case of the R_TUS branch, the motion profile of the middle universal joint is always planar. This allows for the second joint to be driven remotely, i.e. belt driven with the motor placed on the base as well. Having both motors placed on the base requires that the first motor drive only the mass/inertia of the links and not the second motor. Since the motion profile of the second joint of the $U^*R_T S$ branch is not planar, it becomes much more difficult to drive the second joint motor remotely. It is for this reason that then we select the R_TUS branch configuration with a remotely driven second joint as the branch configuration for the detachable branches, and no further enumeration and elimination is required.

4.3 Tripod configurations

Since we can pair each of the fixed-tripod branches with the detachable R_1US branches, there are a total of nine possible parallel robotic systems after enumeration. To further evaluate the possible configurations, a workspace analysis and comparison is used. To calculate the workspace of a parallel robot, the inverse kinematics model is used to search for reachable points. The next section deals with the development of the kinematics of each branch configuration as well as dealing with the constraint equations for the 3, 4 and 5-DOF configurations.

5. Robot kinematics

5.1 Parametric kinematic model

Parallel robots have inverse kinematics that can generally be solved geometrically. That is for a given end effector position and orientation, the joint variables can be solved directly without numerical methods. A hypothetical parallel robot is presented in Fig. 3. that shows the extensible branch model. From this model, solutions to all other branch configurations can be derived.

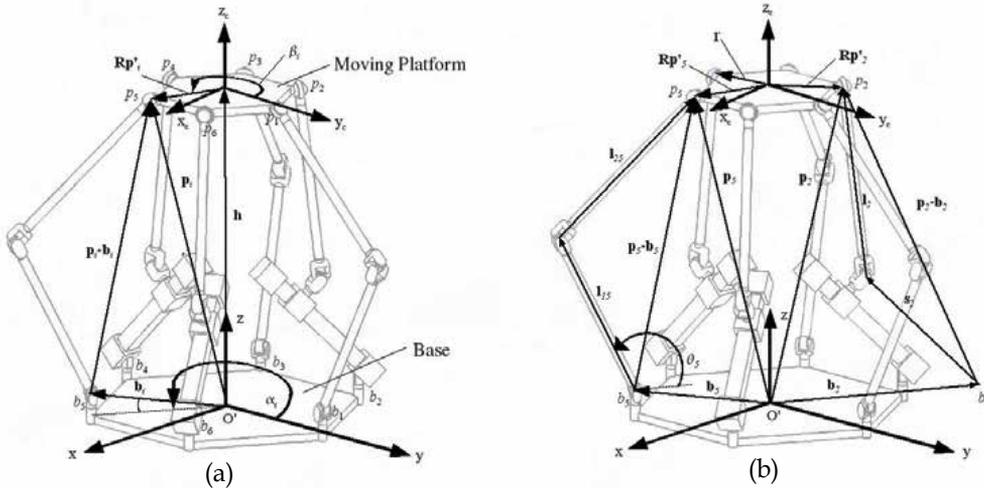


Fig. 3. Kinematic model of a parallel robot with extensible branches: (a) Kinematic model of a generic extensible leg, (b) Kinematic model of the fixed and connected detachable legs

According to the coordinate systems defined in Fig. 3, the position of the i^{th} spherical joint attached to the moving platform can be given as

$$p_i = h + R p'_i \tag{4}$$

where $p_i = [p_{ix} \ p_{iy} \ p_{iz}]^T$ is the position of the i^{th} joint expressed in the global joint expressed in the global coordinate frame $O-xyz$, p'_i presents the same point in the local coordinates $O'-x'y'z'$ attached to the moving platform, $h = [x_c \ y_c \ z_c]^T$ is the vector representing the position of the moving platform, and R is the rotation matrix of the moving platform.

Now let m be the number of the constrained branches, a complete set of the branch constraint equations may be presented as

$$\mathbf{p}_y = \mathbf{T}\mathbf{p}_x \tag{5}$$

where \mathbf{p}_x is the vector containing p_{ix} components of m constrained branches, \mathbf{p}_y is the vector containing p_{iy} components, and $\mathbf{T} = \text{diag}(\tan \alpha_i)$

Equation (5) represents a parametric model in terms of a_i , p_{ix} and p_{iy} that can be used to describe the branch constraint equations for all configurations of the reconfigurable robot. Depending on the robot configuration, constraint equations must be solved in order to define the motion of the moving platform. Table 7. describes which moving platform motions are constrained for each configuration. Note that the number of constraint equations required is identical to those listed in Table 3. A complete derivation of the constraint equations can be found in (Xi et al., 2006).

Mobility (DOF)	Independent motion variables	Constrained motion variables
6	$x_c, y_c, z_c, \theta_x, \theta_y, \theta_z$	N/A
5	$x_c, y_c, z_c, \theta_x, \theta_y$	θ_z
4	x_c, y_c, z_c, θ_z	θ_x, θ_y
3	z_c, θ_x, θ_y	x_c, y_c, θ_z

Table 7. Motion constraints of the reconfigurable parallel robot

5.2 Branch module inverse kinematics

With the position of the i^{th} spherical joint known as defined by the constraint equations (if any), then the inverse kinematics for the i^{th} branch can be solved. As listed in the enumeration criteria, the motion of the i^{th} spherical joint must be planar for the constrained branch configuration and spatial for the unconstrained branch configuration. This gives rise to a planar and spatial solution to the branch kinematics.

The solution to the i^{th} branch kinematics is generally solved by the use of loop equations. That is, a loop of vectors that describes the links, base and platform is established in an effort to eliminate specific unknown information and create an algebraic solution to the joint variable. These solutions are derived such that it allows for the joint variable solution regardless of the configuration of the parallel robot. Thus there is no need to reform the kinematic equations when the robot is reconfigured from one configuration to another.

The following is a description of the loop equations for fixed-tripod branch configurations described in Table 5. As seen in Table 5., after reconfiguration there are six different reduced DOF branch configurations. Since the kinematics are applicable to the branch regardless of the robot configuration, the solutions presented cover all nine potential fixed-tripod branch configurations. Also, the detachable branch configuration is also a potential fixed branch configuration, thus the kinematics are automatically covered.

5.2.1 R_TUS and U^{*}R_TS branch kinematics

The first branch configuration is that which takes the form of the R_TR_TS when it is in a reduced DOF form. This includes the R_TUS and U^{*}R_TS branch configurations. As shown in Fig. 4., the R_TR_TS branch module can be related to the extensible model using the following relation

$$\mathbf{h} + \mathbf{R}\mathbf{p}'_i = \mathbf{b}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{6a}$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{6b}$$

If the first joint is a revolute joint, then the direction vector of the upper arm, denoted by \mathbf{u}_{1i}^l , can be defined as

$$\mathbf{u}_{1i}^l = \begin{Bmatrix} \cos \alpha_i \cos \theta_i \\ \sin \alpha_i \cos \theta_i \\ \sin \theta_i \end{Bmatrix} \tag{7}$$

where θ_i is the driving angle of the \mathbf{l}_{2i} arm, and a_i is the same as defined in Fig. 3. By applying the constraint that the length of the \mathbf{l}_{1i} arm is constant, the following equation is obtained

$$|\mathbf{p}_i - \mathbf{b}_i - l_{1i}\mathbf{u}_{1i}^l| = l_{2i} \tag{8a}$$

Alternatively, using \mathbf{d}_i

$$|\mathbf{d}_i - l_{1i}\mathbf{u}_{1i}^l| = l_{2i} \tag{8b}$$

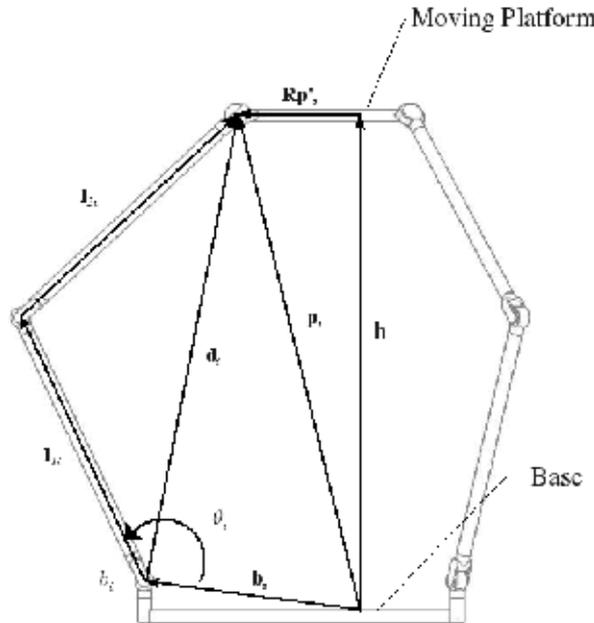


Fig. 4. Relation of the R_TUS and $U^*R_T S$ branch configurations to the extensible module

Using Equation (8a) or (8b), the solution to the inverse kinematics can be solved for the planar cases ($R_T R_T S$). It can also be used for the $R_T U S$ configuration. For the $U^* R_T S$ configuration, the solution to Equations (8a) and (8b) can only be solved if the passive rotation axis of the U^* joint is axial in nature. In other words, so long as the motion profile of the middle R_T joint is planar, then the direction vector \mathbf{u}_{1i}^l is solvable, and the passive joint variable is naturally eliminated from the loop equation. The solution to the case where the passive axis of the U^* joint is not axial can be viewed in (Sabater et al., 2005).

5.2.2 $P_F U S$ and $P_F U^* S$ branch kinematics

The second branch configuration is that which takes the form of the $P_F R_T S$ when it is in a reduced DOF form. This includes the $P_F U S$ and $P_F U^* S$ branch configurations. As shown in Fig. 5., the extensible leg can be related to the $P_F R_T S$ branch module by following loop equation

$$\mathbf{h} + R\mathbf{p}_i^l = \mathbf{b}_i + \mathbf{s}_i + \mathbf{l}_i \tag{9a}$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{s}_i + \mathbf{l}_i \tag{9b}$$

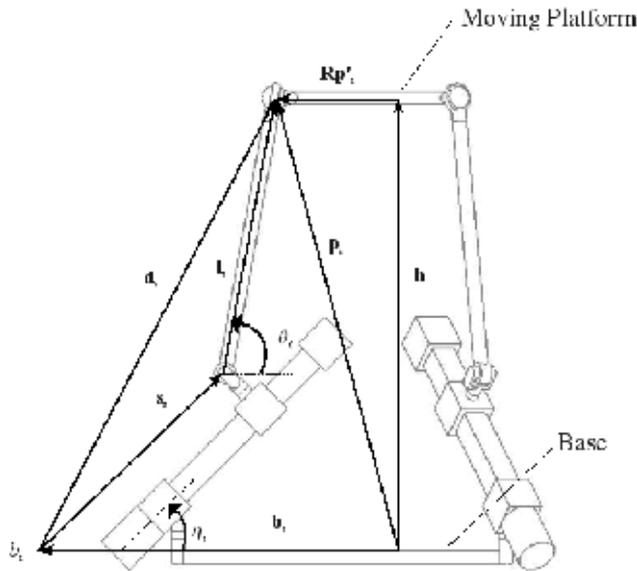


Fig. 5. Relation of the $P_F U S$ and $P_F U^* S$ branch configurations to the extensible module

Here \mathbf{s}_i is the vector representing the track platform traveling displacement parallel to the guide way, noting that \mathbf{b}_i does not necessarily point to a physical point on the robot. If \mathbf{b}_i were to end at the connection of the base and track, then the vector \mathbf{s}_i does not only change in length, but also direction, which is not desirable. Furthermore, \mathbf{l}_i is the vector representing the slide in space. Since the track is fixed and \mathbf{s}_i acts parallel to the track, its

direction vector, denoted \mathbf{u}_i^s , is specified. By applying the constraint that the length of the track is constant, the traveling distance s_i can be solved from the following equation

$$\left| \mathbf{p}_i - \mathbf{b}_i - s_i \mathbf{u}_i^s \right| = l_i \quad (10a)$$

which may be expressed in terms of \mathbf{d}_i as

$$\left| \mathbf{d}_i - s_i \mathbf{u}_i^s \right| = l_i \quad (10b)$$

With the length of the guide way solved, if the branch is of the P_FU^{*}S configuration, then the joint variable θ_i can be solved by solving Equation (9a) or (9b) for \mathbf{l}_i and then using a four-quadrant arctangent

$$\theta_i = \pi - \arctan 2 \left(s_{iz}, s_{iy} \cos \alpha_i - s_{ix} \sin \alpha_i \right) \quad (11)$$

The loops Equations (10a) and (10b) can be solved for both the planar and spatial case since the direction the sliding arm is eliminated from the equations.

5.2.3 UP_VS and U^{*}P_VS branch kinematics

The third branch configuration is that which takes the form of the R_TP_VS when it is in a reduced DOF form. This includes the UP_VS and U^{*}P_VS branch configurations. As shown in Fig. 6., the extensible leg model is the exact solution of the kinematics, thus

$$\mathbf{h} + \mathbf{R}\mathbf{p}_i = \mathbf{b}_i + \mathbf{s}_i \quad (12a)$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{s}_i \quad (12b)$$

Thus the solution to the extensible leg is simply

$$\left| \mathbf{p}_i - \mathbf{b}_i \right| = s_i \quad (13a)$$

which may be expressed in terms of \mathbf{d}_i as

$$\left| \mathbf{d}_i \right| = s_i \quad (13b)$$

If, the active joint variable is rotational instead of linear, then, a four-quadrant arctangent can be used to solve for θ_i . We note that $\mathbf{s}_i = [s_{ix} \ s_{iy} \ s_{iy}]^T$, then

$$\theta_i = \arctan 2 \left(s_{iz}, \sqrt{s_{ix}^2 + s_{iy}^2} \right) \quad (14)$$

Again, we note that the solution to the loop Equations (12a) and (12b) will hold for both the planar and spatial cases.

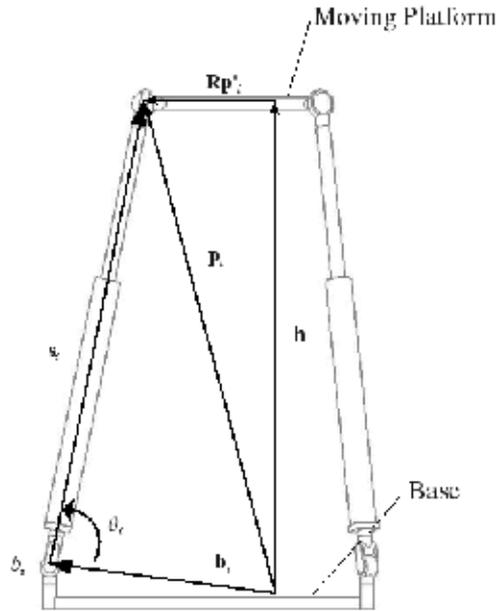


Fig. 6. Relation of the UP_vS and U*P_vS branch configurations to the extensible module

5.2.4 P_FCS branch kinematics

The fourth branch configuration is that which takes the form of the P_FP_vS when it is in a reduced DOF form. This includes the P_FCS branch configuration. As shown in Fig. 7., the P_FP_vS branch module can be related to the extensible model using the following relation

$$\mathbf{h} + \mathbf{R}\mathbf{p}_i^l = \mathbf{b}_i + \mathbf{s}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{15a}$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{s}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{15b}$$

The in plane the angle φ_i is always known since the angles η_i, ξ_i, and γ_i are physical parameters that are known constants. The direction vector of the upper arm, denoted by \mathbf{u}_{2i}^l , can be defined as

$$\mathbf{u}_{2i}^l = \begin{Bmatrix} \cos \alpha_i \cos \phi_i \\ \sin \alpha_i \cos \phi_i \\ \sin \phi_i \end{Bmatrix} \tag{16}$$

yielding $\mathbf{l}_{2i} = l_{2i} \mathbf{u}_{2i}^l$. From this, the vector \mathbf{e}_i can be found from the modified loop equation $\mathbf{e}_i = \mathbf{d}_i - \mathbf{l}_{2i}$. The angle ν_i between the vector \mathbf{e}_i and the guide way can be found using a four-quadrant arctangent

$$\nu_i = \eta_i - \arctan 2 \left(e_{iz}, \sqrt{e_{ix}^2 + e_{iy}^2} \right) \tag{17}$$

The Sine Law can then be used to find the distance of the track along the guide way and the length of the cylindrical joint along it's guide way if necessary.

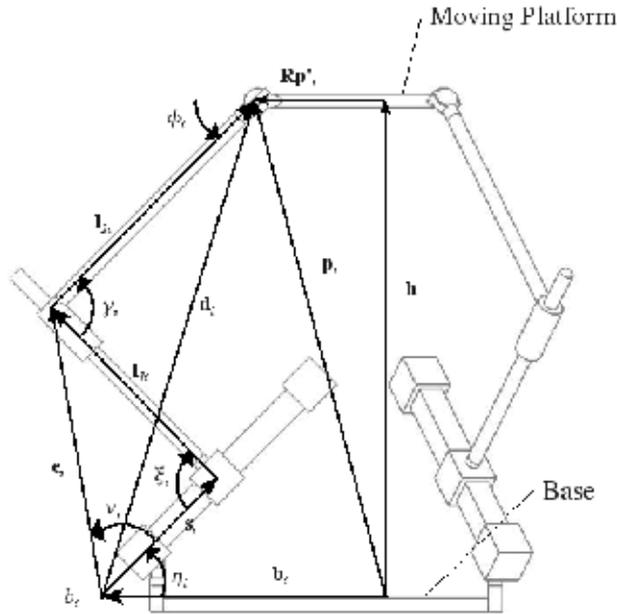


Fig. 7. Relation of the P_FCS branch configuration to the extensible module

Equation (16) only holds for the planar case. The solution to the kinematics for the spatial case must be solved numerically. In Equations (15a) and (15b), there are three unknowns, the lengths of s_i and l_{1i} , and the direction of l_{2i} , and there is no way to eliminate two of these unknowns to provide the means to solve the loop equation. It is thus not optimal to use the P_FCS branch configuration over those with analytical solutions for all kinematic cases and we therefore eliminate the P_FCS branch as a potential fixed tripod branch configuration.

5.2.5 CP_VS branch kinematics

The fifth branch configuration is that which takes the form of the P_VP_VS when it is in a reduced DOF form. This includes the CP_VS branch configuration. As shown in Fig. 8., the P_VP_VS branch module can be related to the extensible model using the following relation

$$\mathbf{h} + \mathbf{R}\mathbf{p}_i = \mathbf{b}_i + \mathbf{l}_i + \mathbf{s}_i \tag{18a}$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{l}_i + \mathbf{s}_i \tag{18b}$$

The angles θ_i and γ_i are known constants. Since the first joint is a cylindrical joint, the direction vector of the upper arm, denoted by \mathbf{u}_{1i}^l , can be defined using Equation (7). From this, the angle ζ_i between the cylindrical joint guide way and the vector \mathbf{d}_i can be calculated

$$\zeta_i = \arccos \frac{\mathbf{u}_{1i}^l \bullet \mathbf{d}_i}{|\mathbf{d}_i|} \tag{19}$$

The loop equation formed by $\mathbf{d}_i = \mathbf{l}_i + \mathbf{s}_i$ forms a triangle in space. Since the length of \mathbf{d}_i is known, and the two interior angles γ_i and ζ_i are also known, then the joint variable s_i can be solved for all planar and spatial cases by using the Sine Law.

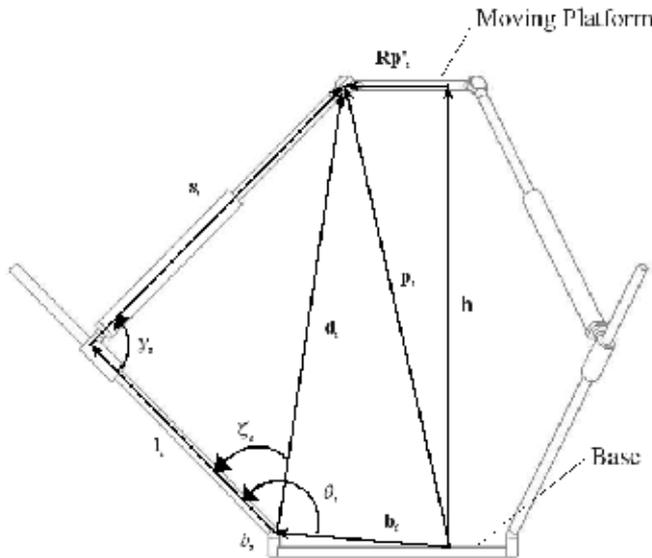


Fig. 8. Relation of the P_VP_VS branch configuration to the extensible model

5.2.6 R_TCS branch kinematics

The sixth branch configuration is that which takes the form of the R_TP_VS when it is in a reduced DOF form. This includes the R_TCS branch configuration. As shown in Fig. 9., the second R_TP_VS branch module can be related to the extensible model using the following relation

$$\mathbf{h} + R\mathbf{p}_i^l = \mathbf{b}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{20a}$$

or

$$\mathbf{p}_i = \mathbf{b}_i + \mathbf{l}_{1i} + \mathbf{l}_{2i} \tag{20b}$$

The angle γ_i , is a known constant, thus the Sine Law can then be used to find the angle between guide way and the vector \mathbf{d}_i , as well as the length of the cylindrical joint along its guide way.

With the length of both legs known, the loop Equations (20a) and (20b) take on a form similar to the loop Equations (6a) and (6b). Equation (7) is then used to define \mathbf{u}_{1i}^l and the

design phase depending on the requirements of the system. In order to evaluate the various configurations, the physical parameters for the base and moving platform are uniform throughout. The radius of the base, b_i , is 100 mm and the radius of the moving platform, r , is 50 mm. In the 6-DOF case, the branches are spaced at 60° intervals. The spacing increases as necessary when the detachable branches are disconnected from the moving platform. The physical constraints for each branch configuration are described in Table 8.

Branch Configuration	Variable	Symbol	Value
R_TUS and U*_RT_S Fixed Branches¹	Lower arm length	l_{1i}	125 mm
	Upper arm length	l_{2i}	125 mm
	Joint variable range	θ_i	$0^\circ - 180^\circ$
P_FUS and P_FU*_S Fixed Branches	Guide way inclination	η_i	45°
	Arm length	l_i	100 mm
	Joint variable range	s_i	0 mm - 100 mm
U_PV_S and U*_PV_S Fixed Branches	Joint variable range	s_i	125mm - 200 mm
CP_VS Fixed Branches	Guide way inclination	θ_i	135°
	Joint variable range	s_i	125 mm - 200 mm
	Cylindrical joint linear range	l_i	0 mm - 150 mm
R_TCS Fixed Branches	Arm length	l_{2i}	150 mm
	Cylindrical joint inclination angle	γ_i	90°
	Joint variable range	θ_i	$0^\circ - 180^\circ$
	Cylindrical joint linear range	l_i	0 mm - 150 mm
Detachables Branches^{1, 2}	Lower arm length	l_{1i}	150 mm
	Upper arm length	l_{2i}	150 mm
	Joint variable range	θ_i	$0^\circ - 180^\circ$

Table 8. Branch physical constraints

Finally, we ignore the physical constraint that would be imposed due to the presence of the spherical joints. In reality, contact between the edge of the socket and the extension of the ball would impede further motion and thus affect the shape of the workspace. The

1 To avoid singularities, the z -coordinate of the i^{th} middle joint must be less than the z -coordinate of the i^{th} \mathbf{d}_i vector.

2 The lengths of the detachable branch arms are consistent for all fixed branch configurations except the R_TCS configuration in which both arms lengths are 250 mm. The change in length was to provide and an acceptable workspace boundary.

alignment of the spherical joints is left to the detailed design phase, as careful design and alignment can alleviate this impedance.

6.2 Workspace

With the kinematic equations established and the physical parameters defined, the position workspace can finally be calculated. Table 9. shows the summary of workspace.

Fixed Branch Configuration	Robot Workspace Volume [mm ³]			
	3-DOF	4-DOF	5-DOF	6-DOF
R_TUS and U^*R_TS	7549	13, 778	15, 595	29, 463
P_FUS and P_FU^*S	15, 368	18, 224	14, 309	17, 265
UP_VS and U^*P_VS	13, 263	9386	6275	6813
CP_VS	9619	4634	1898	2244
R_TP_VS	43	161	2928	6021

Table 9. Workspace volume of the reconfigurable parallel robot

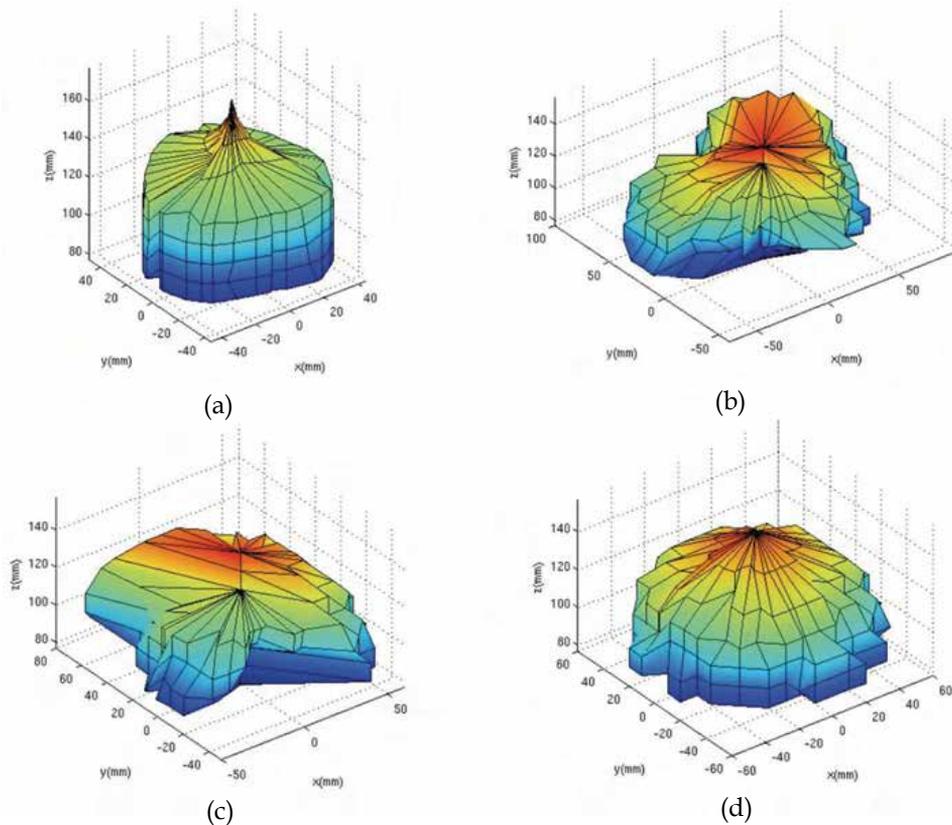


Fig. 10. Workspace boundaries of the P_FUS and P_FU^*S fixed branch configuration: (a) 3-DOF, (b) 4-DOF, (c) 5-DOF, (d) 6-DOF

Note that the increase in DOF does not necessarily result in an increase in workspace volume. This is due to the constraints of the branches and not the kinematic constraints of the position and orientation of the moving platform. Also, in the 4 and 5 DOF cases, the spacing of the branches is not uniform and results in very odd shaped workspace boundaries. An example of the P_FUS and P_FU^*S workspace is shown in Fig. 10.

7. Optimal configuration

The optimal configuration depends heavily on the type of task the robot is required to perform. Some examples for parallel robotics include:

- Flight simulation test beds use 6-DOF hydraulically actuated UP_VS branches.
- Machining tool applications use the P_FUS branch configuration due to their structural stiffness.
- Pick and place automation use planar Delta robots (specialized R_TUS branches).

We further note that the optimization here is not the optimization of a continuous system. Here, there are five discrete systems that require some form of comparison to evaluate their strengths and weaknesses. There are many methods of evaluating the merits of discrete systems. These usually include some form of design decision matrix and there are many works available that covers this topic. All of these methods require a certain degree of designer input and different designers will form their decision matrices differently. Once an architecture is chosen, then continuous optimization algorithms can be used.

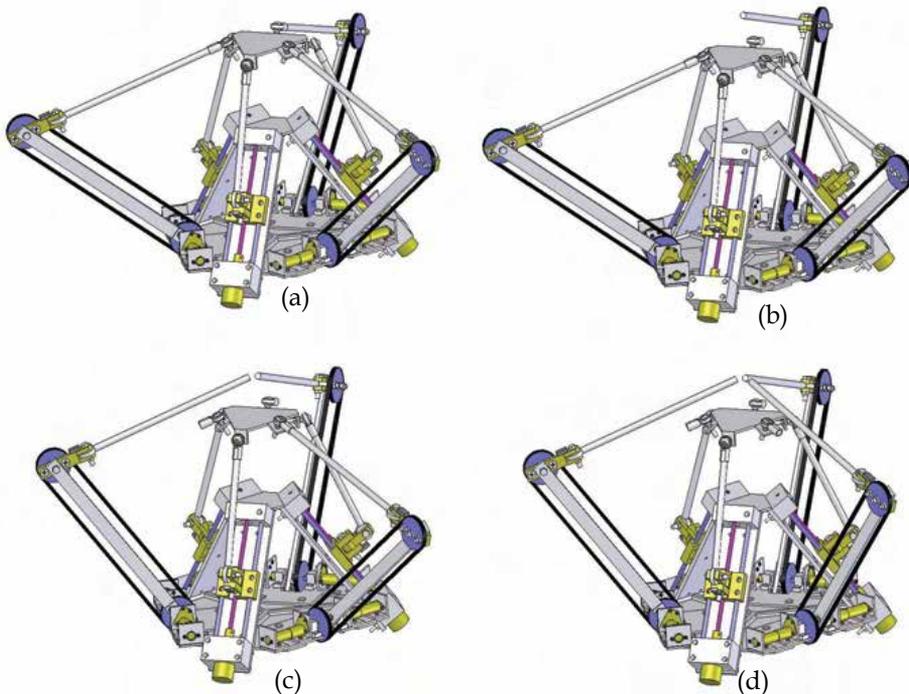


Fig. 11. Reconfigurable parallel robot currently being developed at Ryerson University: (a) 6-DOF, (b) 5-DOF, (c) 4-DOF, (d) 3-DOF

A very simple pair-wise comparison of the discrete systems against each of for the specified functional requirements is used to arrive at the optimal configuration (Salustri, 2008). Using this design decision method, the optimal architecture of the parallel robot is the P_FUS configuration. This robot configuration is currently being developed at Ryerson University and is shown in Fig. 11 and the actual prototype is shown in Fig. 12. Also in development, is a universal joint capable of locking one DOF in order to satisfy the branch constraint requirements previously described.

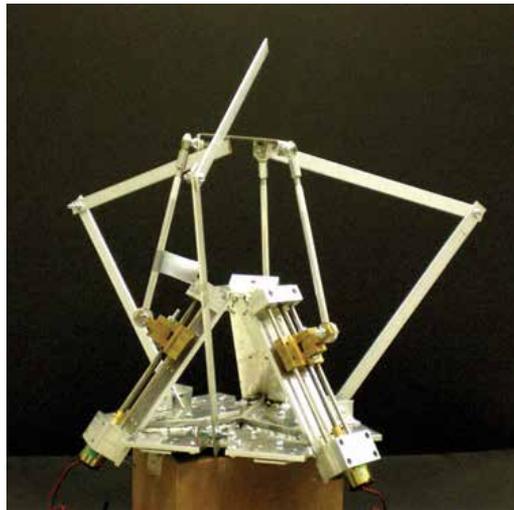


Fig. 12. The prototype of the proposed reconfigurable parallel robot

8. Conclusion

A novel method for the architecture design of a reconfigurable parallel robot is presented based on common actuation devices. System design techniques are used to classify parallel robot modules and enumeration rules are established to determine the feasible robot architectures. Branch kinematics are developed and a workspace analysis is performed. An optimal design is selected from the remaining discrete robot configurations. The final design is a self-reconfigurable parallel robot that has the ability to perform on-the-fly reconfiguration. The proposed reconfigurable parallel robot not only provides innovation in reconfigurable system design but also stimulates new research into parallel robot kinematics.

9. References

- Chen, I.-M. (2001). Rapid response manufacturing through a rapidly re-configurable robot workcell. *Robotics and Computer-Integrated Manufacturing*, 17, 3, (June 2001) pp. 199-213, 0736-5845.
- Chen, L., Xi, F., Macwan, A. (2005). Optimal module selection for designing reconfigurable machining systems. *ASME Journal of Manufacturing Science and Engineering*, 127, 1, (February 2005) pp. 104-115, 1087-1357.

- Dash, A.K., Chen, I.-M., Yeo, S.H., Yang, G. (2005). Task-oriented configuration design for reconfigurable parallel manipulator systems. *International Journal of Computer Integrated Manufacturing*, 18, 7, (October-November 2005) pp. 615-634, 0951-192X.
- Hafez, M., Lichter, M.D., Dubowsky, S. (2003). Optimized binary modular reconfigurable robotic devices. *IEEE/ASME Transaction on Mechatronics*, 8, 1, (March 2003) pp. 18-25, 1083-4435.
- Hamlin, G.J., Sanderson, A.C. (1997). TETRABOT: a modular approach to parallel robotics. *IEEE Robotics and Automation Magazine*, 4, 1, (March 1997) pp. 42-50, 1070-9932.
- Horner, C.G., (1990). Adaptive truss structure. *US/Japan Workshop on Smart/Intelligent Materials and Systems*, Honolulu, March 1990, 19-23.
- Salustri, Filipino A., (2008) http://deseng.ryerson.ca/xiki/Learning/Main:Web_home
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pristchow, G., Ulsoy, G., Van Brussel, H. (1999). Reconfigurable manufacturing systems. *Annals of the CIRP*, 48, 2, pp. 527-540, 0007-8506.
- Michael, J. (1995). Fractal shape changing robot construction theory and application note. *Robodyne Cybernetics Ltd.*
- Montana, D.J. (1998). The kinematics of contact and grasp. *International Journal of Robotics Research*, 7, 3, (June 1998) pp. 17-32, 0278-3649.
- Onoda, J., Fu, D.-Y., Minesugi, K. (1996). Two-dimensional deployable hexapod truss. *Journal of Spacecraft and Rockets*, 33, 3 (May-June 1996) pp. 416-421, 0022-4650.
- Sabater, J.M., Saltarén, R.J, Aracil, R. (2005). Design, modelling and implementation of a 6 URS parallel haptic device. *Robotics and Autonomous Systems*, 47, 1, pp. 1-10, 0921-8890.
- Schenker, P., Pirjanian, P., Huntsberger, T., Aghazarian, H., Baumgartner, E., Iagnemma, K., Rzepniewski, A. (2000). Reconfigurable robots for all-terrain exploration. *Proceedings of the SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems*, pp. 454-468, 0277-786X, Boston, September 2000, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, USA.
- Suh, N.P. (1990). The principles of design. Oxford University Press, New York.
- Tomita, K., Murata, S., Yoshida, E., Kurokawa, H., Kokaji, S. (1996). Reconfiguration method for a distributed mechanical system, In: *Distributed Autonomous Robotic Systems 2*. Asama, H., Fukuda, T., Arai, T., Endo, I. (Ed), pp. 17-25, Springer-Verlang New York Inc., 4431701907, Secaucus, NJ, USA.
- Tsai, Lung-Wen. (1998). Systematic enumeration of parallel manipulators. *Department of Mechanical Engineering and Institute for Systems Research*, Technical Report, pp. 1-11.
- Unsal, C., Kiliccote, H., Patton, M., Khosla, P. (2000). Motion planning for a modular self-reconfiguring robotic system. *Distributed Autonomous Robotic Systems*, 4, pp. 1-10.
- Xi, F., Verner, M., Ross, A. (2000). A reconfigurable hexapod system – preliminary results. *Proceedings of the 2000 Japan-USA Symposium, Special Session on Modular and Reconfigurable Controller for Flexible Automation*, University of Michigan, July 2000.
- Xi, Fengeng, Xu, Yuonan, Xiong, Guolian. (2006). Design and analysis of a re-configurable parallel robot. *Mechanisms and Machine Theory*, 41, 2, (February 2006) pp. 191-211, 0094-114X.

- Yim, M. (1994). Locomotion with a unit-modular re-configurable robot. *Ph.D. Thesis*, Stanford University.
- Yim, M., Zhang, Y., Duff, D. (2002). Modular robots. *IEEE Spectrum*, 39, 4 (February 2002) pp. 30-34, 0018-9235.

A Novel 4-DOF Parallel Manipulator H4

Jinbo Wu¹ and Zhouping Yin²

¹Traffic Science & Engineering College

²State Key Laboratory of Digital Manufacturing Equipment and Technology
Huazhong University of Science & Technology

China

1. Introduction

Parallel manipulators have the advantages of high stiffness and low inertia compared to serial mechanisms. Based on the Stewart-Gough platform architecture, a lot of 6-DOF mechanical devices have been proposed. The 6-DOF parallel manipulators suffer from a small workspace, complex mechanical design, and difficult motion generation and control due to their complex kinematic analysis. To overcome these shortcomings, the *limited-DOF manipulator*, which has fewer than 6 DOFs, can be found in many production lines. It is clear today that most attention has been paid to 3-DOF family among the limited-DOF parallel manipulators (Carretero, 2000). However, in many industrial situations, there is a need for equipment providing more than 3-DOFs. For example, for most pick-and-place applications in semiconductor manufacturing, at least 4 DOFs are required (3 translation to move the carried die from one point to the other, 1 rotation to adjust the orientation in its final location). A new family of 4-dof parallel manipulators called H4 that could be useful for high-speed pick-and-place applications is proposed by Pierrot and Company (Pierrot, 1999). The H4 manipulator offers 3 DOFs in translation and 1 DOF in rotation about a given axis. The H4 manipulator is useful for high-speed handling in robotics and milling in machine-tool industry since it is a fully-parallel mechanism with no passive chain and able to provide high performance in terms of speed and acceleration.

This chapter discusses the kinematic analysis of the H4 manipulator. In section 2, synthesis methods for designing H4 are presented, and various possible mechanical architectures of the parallel manipulator are exposed. Section 3 discusses the inverse and forward kinematics problem of H4. Section 4 deals with singularity analysis of H4 utilizing line geometry tools and screw theory. Section 5 concludes this chapter by providing the development tendency of the parallel manipulators.

2. Structural synthesis and architectures

2.1 General concept of H4

Parallel manipulators are constituted of a moving platform that is connected to a fixed base by several chains (limbs). Generally, the number of limbs is equal to the degrees of freedom (DOF) of the moving platform such that each limb is driven by no more than one actuator and all actuators can be mounted on or near the fixed base. By acting on the limbs the

platform pose (position and orientation) is controlled. Moreover, if the actuators are locked, the manipulator will become an isostatic structure in which all the legs carry the external loads applied to the platform. This feature makes the parallel manipulators with high stiffness possible throughout the whole space. From the first ideas proposed by Gough (Gough, 1956) or Steward (Steward, 1965), a lot of interesting mechanical devices or design methods have been extensively studied. In the beginning, many structures were based on the ingenuity of the researchers and not on a systematic approach. Subsequently, a new research domain called *structure (or type) synthesis* was proposed, in which various methodologies were tried to generate all the structures that have a desired kinematic performance. The most widely used synthesis approaches (and their variants) are *graph theory*, *group theory* and *screw theory* (Merlet, 2006). In this section, we will not discuss these theoretical problems, but focus on the structure generation of H4.

The 6-DOF parallel manipulators generally suffer from a small workspace, complex mechanical design and difficult motion generation and control due to their complex kinematic analysis. To overcome these shortcomings, new structures for parallel manipulators having less than 6-DOF (which are called limited-DOF manipulators in this chapter, although many of these new structures were known well before) are explored. There is an overriding motivation behind such efforts: limited-DOF manipulators may be needed for many applications. For example, parallel wrists need only three rotational DOFs. It is clear today that most attention has been paid to 3-DOF family among the limited-DOF parallel manipulators (Gosselin & Angeles, 1989; Agrawal et al., 1995; Gosselin & St-Pierre, 1997; Fattah & Kasaei, 2000; Gregorio, 2001). However, in many industrial situations, there is a need for equipment providing more than 3 DOFs arranged in parallel and based on simpler arrangements than 6-DOF structures. The reference (Cheung et al., 2002) developed a 4-DOF parallel manipulator E4 which could be used in a semiconductor packaging system. It is in the early 80's when Raymond Clavel comes up with the brilliant idea of using parallelograms to build a parallel robot with three translational and one rotational degree of freedom (Bonev, 2001). Contrary to opinions published elsewhere, his inspiration was truly original and does not come from any parallel mechanism patents. Professor Clavel called his creation the Delta robot. The new family of 4-DOF parallel manipulators called H4 that could be useful for high-speed pick-and-place applications is just based on the idea of Delta structure (Pierrot & Company, 1999; Company & Pierrot, 1999). The prototype built in the Robotics Department of LIRMM can reach 10g accelerations and velocities higher than 5m/s (Robotics Department of LIRMM).

2.1.1 Delta structure

Even if an incredibly large number of different structures have been proposed by academic researchers in the last 30 years, most of these that are used widely in industry can be classified into two basic types: the Delta structure and the so-called "hexapod" with 6 U-P-S chains in parallel (U-P-S: Universal-Prismatic-Spherical). This may be a result of either the exceptional simplicity of the Delta 3-DOF solution, or the enormous research effort dedicated to "hexapod" (Company, 1999).

The basic idea behind the Delta parallel robot design is the use of parallelograms. A parallelogram allows an output link to remain at a fixed orientation with respect to an input link. The use of three such parallelograms restrain completely the orientation of the mobile platform which remains only with three purely translational degrees of freedom. The input

links of the three parallelograms are mounted on rotating levers via revolute joints. The revolute joints of the rotating levers are actuated in two different ways: with rotational (DC or AC servo) motors or with linear actuators. Finally, a fourth leg is used to transmit rotary motion from the base to an end-effector mounted on the mobile platform. Fig.1 shows the most famous Delta robot with three translation degrees of freedom, which was initially developed at École Polytechnique from Lausanne by Clavel (Clavel, 1988). All the kinematic chains of this robot are of the $RRPaR$ type: a motor makes a revolute joint rotate about an axis \mathbf{w} . On this joint is a lever, at the end of which another joint of the R type is set, with axis parallel to \mathbf{w} . A parallelogram Pa is fixed to this joint, and allows translation in the direction parallel to \mathbf{w} . At the end of this parallelogram is a joint of the R type, with axis parallel to \mathbf{w} , and which is linked to the end effector.

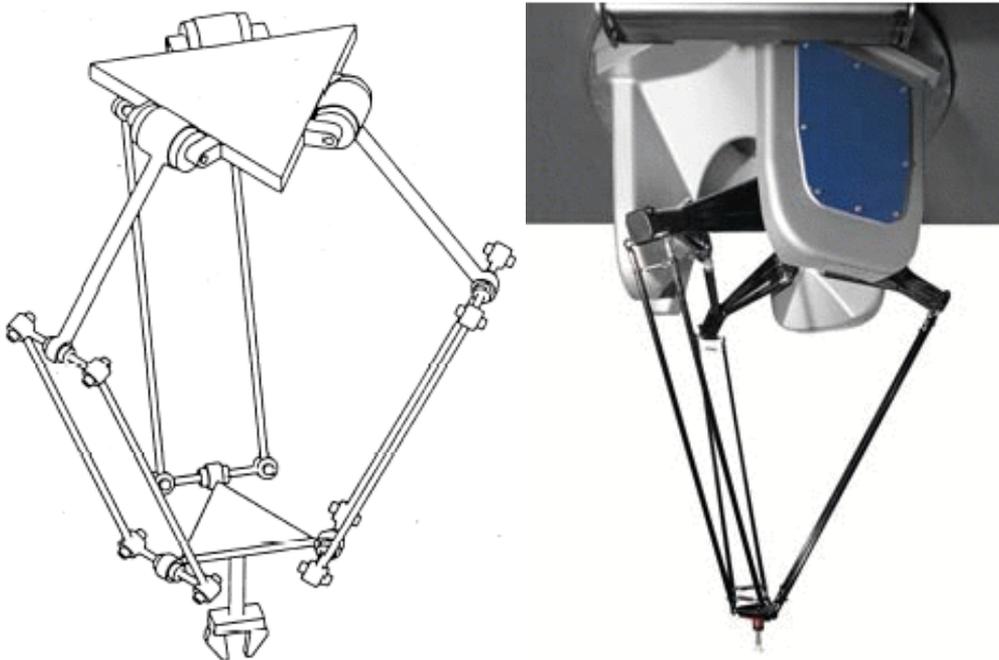


Fig.1. The Delta structure proposed by Prof. Clavel and one of its industrial version, the CE33 (courtesy of SIG Pack Systems)

The Delta robot is firstly marked by the two Swiss brothers Marc-Olivier and Pascal Demaurex who created the Demaurex company (Bonev, 2001). The joint-and-loop graph of the Delta robot and one of its equivalent structures are shown in Fig.2, where P, R and S represent prismatic, revolute and spherical joint respectively. The displacement of the end-effector of the Delta robot is the result of the movement of the three articulated arms mounted on the base, each of which are connected to a pair of parallel rods. The three orientations are eliminated by joining the rods in a common termination and the three parallelograms ensure the stability of the end-effector. It is noted that the rotary actuator and lever part of the Delta could be replaced by a linear actuator, as suggested by Clavel itself and Zobel (Zobel, 1996). This type of Delta is sometime called a Linapod or a linear Delta.

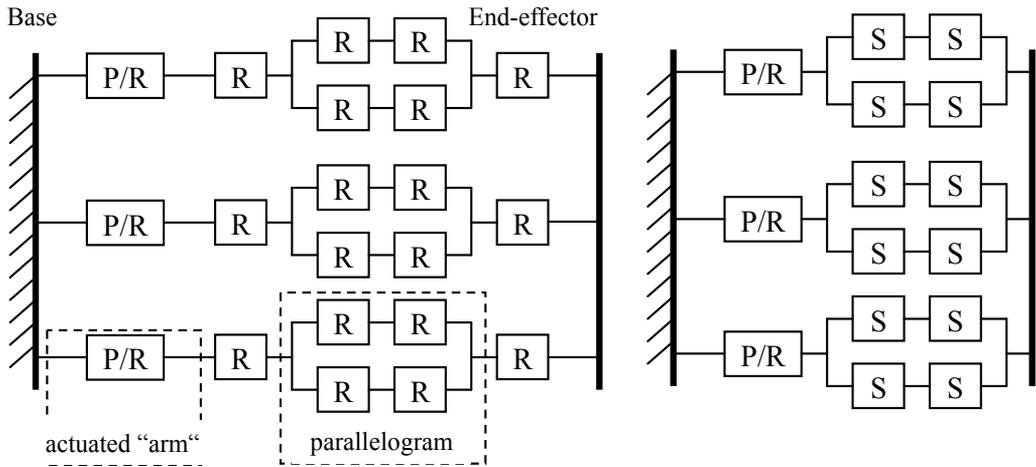


Fig.2. Joint-and-loop graph of the Delta robot and one of its equivalent structure

2.1.2 Structure derivation of H4

H4 is based on 4 independent kinematic chains between the base and the nacelle, each chain is actuated where each actuator is fixed on the base (Pierrot & Company, 1999; Company & Pierrot, 1999). Such technological and conceptual ideas have already proven their efficiency on high-speed equipment for the Delta robot. So, knowing the advantages of this family of mechanisms, it is interesting to recall a few important design features of the Delta robot. The Delta robot is based on three actuated linear or rotational joints, and three pairs of rods equipped with ball joints at each end. As a matter of fact, two ball joints on each rod introduce an internal DOF for the rod which can rotate about its own axis. An arrangement such as the one depicted in Fig.3 suppresses these internal DOF and keeps the same global behaviour. To go a little further, it is possible to consider each pair of rods (that is: two U-S chains parallel one to the other) as equivalent to a unique rod equipped with universal joints at each extremity. The admissible motions for both chains are not exactly the same, but they can be regarded as “equivalent” in terms of number and type of degrees of motion.

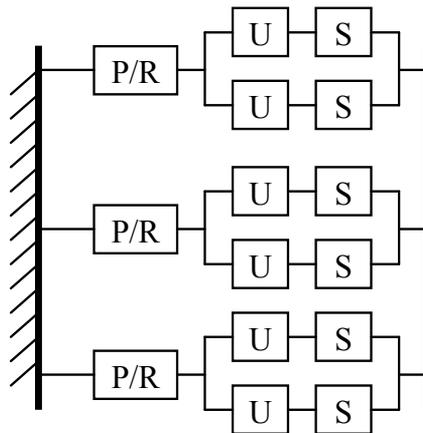


Fig.3. The structure of a Delta robot with no internal DOF

In order to define the basic principle of a 4-DOF mechanism, the classical kinematics formulas can be used. For a mechanism with a “closed loop” (a chain going from ground to the nacelle, and then back to ground), its mobility is given by:

$$m = \sum_{i=1}^{N_l} dof_i - 6L \tag{1}$$

where L is the number of loops; dof_i is the number of DOF of the i th link; N_l is the number of links.

For a fully-parallel 4-DOF mechanisms involving no passive chain, $L = 3$ and $m = 4$. Thus:

$$\sum_{i=1}^{N_l} dof_i = 4 + (6 \times 3) = 22$$

The four P-U-U chains provide: $4 \times 5 = 20dof$. Thus each additional joint must be only 1 DOF joint. Two rotational joints can be used because such joints are extremely easy to manufacture with good accuracy at low cost. So the arrangement shown in Fig.3 are created, each pair is connected to the nacelle by an R joint. Note that the actuated prismatic joints can be replaced by actuated rotational joints, and that the U-U chains can be replaced by $(U-S)_2$ chains as well (shown in Fig.3(b)). To date, the conditions for a 4-DOF machine have been set up. The conditions to be fulfilled for obtaining 3 translations and 1 rotation about a given axis should be provided.

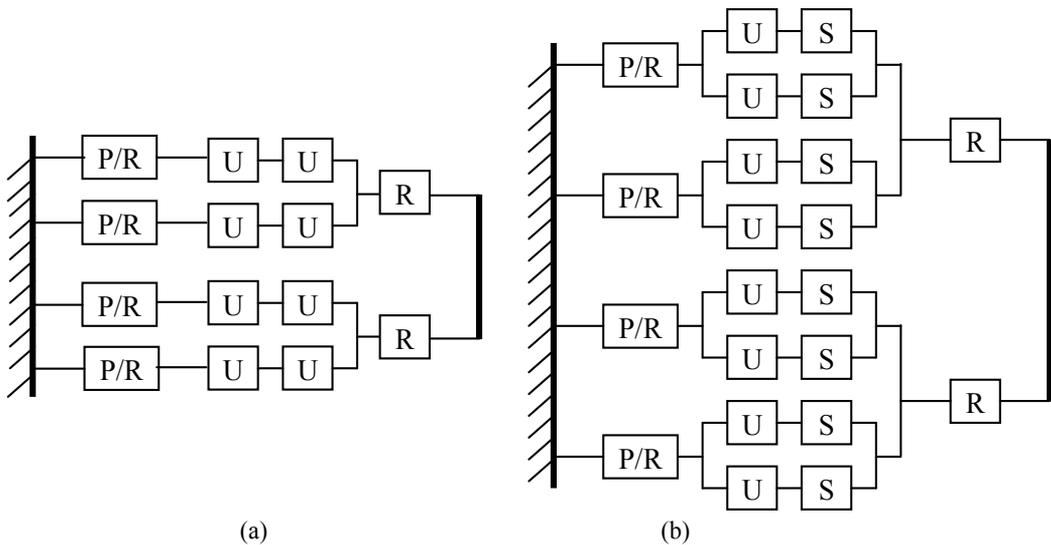


Fig.4. Architecture scheme of H4

The demonstration of “quasi-equivalence” between P-U-U and P-(U-S)₂ is proposed in (Company et al., 2003). The following discusses the possible motion of H4 (Pierrot & Company, 1999; Company & Pierrot, 1999).

Firstly, assume that the two rods of the $(U-S)_2$ chains are parallel to each other. Fig.5 shows a simple scheme of a possible H4 structure. A_i and B_i are the centers of segments $A_{i1}A_{i2}$ and

placed on the nacelle have the same direction represented by a unit vector \mathbf{v} , the existence of a rotational motion about \mathbf{v} can be written as follows:

$$\exists \alpha, \quad \alpha \neq 0 \quad [(\mathbf{w}_1 \times \mathbf{w}_2) \times \mathbf{v}] \times [(\mathbf{w}_3 \times \mathbf{w}_4) \times \mathbf{v}] = \alpha \mathbf{v} \quad (4)$$

Where $\mathbf{w}_i = \mathbf{A}_i \mathbf{B}_i \times \mathbf{u}_i$. Equation (4) is equivalent to:

$$\{[(\mathbf{w}_1 \times \mathbf{w}_2) \times \mathbf{v}] \times [(\mathbf{w}_3 \times \mathbf{w}_4) \times \mathbf{v}]\} \cdot \mathbf{v} = \alpha \mathbf{v} \cdot \mathbf{v}$$

Since \mathbf{v} is a unit vector, then we can obtain

$$\{[(\mathbf{w}_1 \times \mathbf{w}_2) \times \mathbf{v}] \times [(\mathbf{w}_3 \times \mathbf{w}_4) \times \mathbf{v}]\} = \alpha \neq 0$$

Thus:

$$(\mathbf{w}_3 \times \mathbf{w}_4) \cdot [(\mathbf{w}_1 \times \mathbf{w}_2) \times \mathbf{v}] = -[(\mathbf{w}_1 \times \mathbf{w}_2) \times (\mathbf{w}_3 \times \mathbf{w}_4)] \cdot \mathbf{v} = \alpha \neq 0$$

So, the necessary condition that the nacelle can rotate about a given axis is:

$$[(\mathbf{w}_1 \times \mathbf{w}_2) \times (\mathbf{w}_3 \times \mathbf{w}_4)] \cdot \mathbf{v} \neq 0 \quad (5)$$

It should be noted that the previous condition is only a necessary condition and any singular configuration is not taken into account. The following should demonstrate that the two rods of the P-(U-S)₂ chain are parallel to each other when the nacelle moves, so the necessary condition (7) is also satisfied.

The rods of the P-(U-S)₂ chain will stay parallel to each other if both links B₁B₂ and B₃B₄ move only in translation with respect to the ground. As a matter of fact, the possible motions of the nacelle imply that the two pivots fixed on the nacelle along the direction of vector \mathbf{v} will keep this direction. Thus the only possible rotation for B₁B₂ and B₃B₄ is about vector \mathbf{v} . But equation (2) indicates that B₁B₂ can only rotate about $\mathbf{w}_1 \times \mathbf{w}_2$. So, as long as $\mathbf{w}_1 \times \mathbf{w}_2$ and \mathbf{v} are not collinear, the link B₁B₂ has no possible rotation. A similar remark can be made for B₃B₄, leading to the two following conditions:

$$\begin{aligned} \mathbf{w}_1 \times \mathbf{w}_2 &\neq \mathbf{v} \\ \mathbf{w}_3 \times \mathbf{w}_4 &\neq \mathbf{v} \end{aligned} \quad (6)$$

In conclusion, if conditions (6) are fulfilled, the links B₁B₂ and B₃B₄ move only in translation, and the rods in a pair will stay parallel to each other.

2.1.3 H4 mechanism with symmetrical design

In this chapter, the symmetrical design of H4 refers to the structure scheme based on four identical P-U-U chains or P-(U-S)₂ chains. The four identical chains provide 4×5=20 DOFs. Thus, two additional 1-DOF joints are needed. H4 uses two rotational joints because they are easy to manufacture with good accuracy at low cost. Two types of typical architecture scheme of H4 are shown in Fig.4. Note that the U-U chains can be replaced by (U-S)₂ chains. Moreover, the actuated prismatic joints could be replaced by actuated rotational joints as well. A symmetrical mechanism with prismatic actuators is shown in Fig.6 (a), while an equivalent mechanism with rotational actuators is shown in Fig.6 (b).



Fig.6. Symmetrical H4 with actuated prismatic joints and rotational joints

2.1.3 H4 mechanism with asymmetrical design

The link B_1B_2 has three translations and a possible rotation about a vector given by equation (2). It is possible to create more advanced mechanisms based on a particular case of equation (2). For example, if $u_1 = u_2$, then equation (2) becomes:

$$(A_1B_1 \times u_1) \times (A_2B_2 \times u_1) = \beta u_1$$

As long as $A_1B_1 \neq A_2B_2$, then $\beta \neq 0$. So the only possible rotation of this link is about vector u_1 . Thus, the first “meta-chain” already fulfilled our requirements: 3 translations and 1 rotation about a given axis. However, this “meta-chain” is only equipped with 2 actuators; thus 2 other chains (with one actuator each) are needed in parallel to lead to a fully-parallel manipulator. If the first “meta-chain” is constituted of two P-U-U chains, it provides $2 \times 5 = 10$ DOFs. So the two additional chains that constitute the second “meta-chain” should be 6

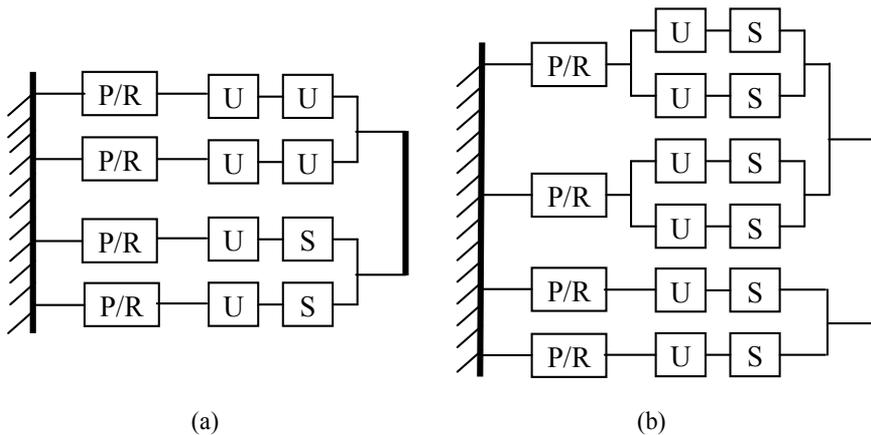


Fig.7 Asymmetrical design of H4

DOFs, which are obviously compatible with the motions offered by the first “meta-chain”. Such mechanical structures are named *asymmetrical H4* (Pierrot & Company, 1999; Company & Pierrot, 1999). An asymmetrical design of H4 using two P-U-Us and two P-U-S’s is shown in Fig.7 (a). The P-U-U’s can be replaced by P-(U-S)₂’s, as shown in Fig.7 (b).

2.2 Evolution of H4

A new trend in the researches on parallel robotics is the development of lower mobility manipulators. Indeed, a lot of applications do not need six DOFs. A classification proposed by Brogardh (Brogardh, 2002) gives the necessary number of DOF for different industrial applications. He shows that applications such as pick-and-place, assembly, cutting, measurement, etc. just need from three up to five DOFs.

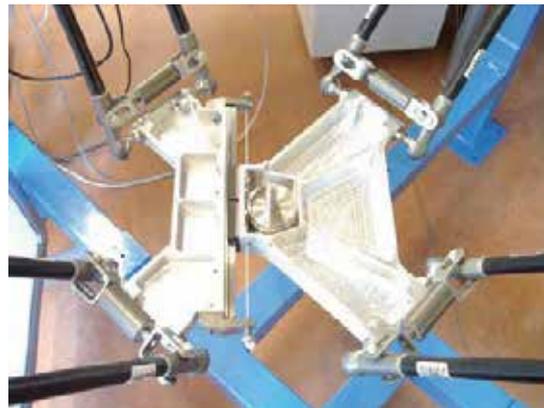
The parallel manipulator H4 can provide three translations and one rotation about a fixed axis, which are also called Schonflies motions (Hervé, 1999). SCARA robots were the first manipulators developed to produce these movements. Due their serial architecture, these robots involve high moving masses which are not suitable for high dynamics. Parallel architecture can overcome this problem. Delta robot is first introduced by Clavel to execute Schonflies motions (Clavel, 1988). This architecture is able to produce three translations and one rotational motion is obtained using a central “telescopic” leg built with universal and prismatic joints. This RUPUR chain suffers from a short service life, and involves a bad stiffness of the rotation motion. Orthoglide (Chablat, 2002) is another parallel robot that can provide Schonflies motions using linear actuators. Its particularity is to be isotropic at the center of its workspace. The machine tool HITA STT has been proposed to produce Schonflies motions (Thurneysen et al, 2002). The particularity of this architecture is to use additional parts in the traveling plate in order to amplify the rotational motion. Other Schonflies motion generators include Gross Manipulator (Angeles et al., 2000), SMG (Angeles, 2005), Kanuk and Manta (Rolland et al., 1999) robots.

In order to avoid the central telescopic leg of the Delta robot, in 1999, the Montpellier Laboratory of Computer Science, Robotics, and Microelectronics (LIRMM) in French invented the parallel manipulator H4 which used the concept of articulated traveling plate. The rotational movement is obtained using an internal mobility on the traveling plate, and a transforming device gives the desired range of rotational motion. Because placing actuators in a symmetrical way (*i.e.* at 90° one relatively to each other) involves “internal singularities” (Pierrot & Company, 1999), the robot has to be built using a particular arrangement of motors. This non-symmetrical arrangement entails a non-homogeneous behavior in the workspace and a limited stiffness of the robot (Company et al., 2005). Later, based on the concept of H4, an improved mechanical structure called I4 is proposed by LIRMM (Krut et al., 2004; Krut et al., 2003). The internal mobility of I4 is obtained with a prismatic joint (Fig. 8). The advantage of this architecture is to authorize a symmetrical arrangement of the actuators. As demonstrates by Krut (Krut et al., 2004), it is possible to place the actuators at 90° one relatively to each other. However, this architecture is more adapted to machine-tool application than to high speed pick-and-place. Indeed, commercial prismatic joints are not suitable for high speed and high accelerations, and have a short service life under such conditions. This inconvenient is due to the high pressure exerted on the balls of these elements at high acceleration conditions. Based on H4 and I4 architecture, an evolution of these mechanisms, named Part 4, has been developed with the wish of reaching high speeds

and accelerations (Nabat et al., 2005). A paper written by Nabat (Nabat et al., 2006) presented experimental results showing that Part 4 was able to reach an acceleration of 15G. Part 4 is a parallel manipulator composed of four closed kinematic chains and an articulated traveling plate. The kinematic chains are similar for Delta, H4 and I4, each of which is composed of an arm and a spatial parallelogram (forearm) linked with spherical joints. The traveling plate is composed of four parts: two main parts (1, 2) linked by two bars (3, 4) with revolute joints (see Fig.9). Thus, its shape is a planar parallelogram and the internal mobility of this traveling plate is a circular translation. Generally, the “natural” range of the rotational operational motion is small, in order to make a complete turn: $[-\pi; \pi]$, an amplification system has to be added on the traveling plate. Several options are available for this amplification such as gears or belt/pulleys. The prototype shown in Fig.9 (b) has been built using belt/pulleys system, with the first pulley fixed on one half traveling plate, and the second one is linked with a revolute joint to the second half traveling plate. Due to its traveling plate having the shape of a planar parallelogram, Part 4 has all the advantages of the previous robots, without their drawbacks. The traveling plate of this robot is articulated and is exclusively realized with revolute joints. Nabat (Nabat et al., 2006) has demonstrated that it is possible to have the same arrangements of the actuators as I4. Thus, this robot is well suited to reach high dynamics and, at the same time, to have a good stiffness and a homogenous behavior in the workspace.



(a) Overview of I4 prototype



(b) Articulated traveling plate

Fig.8. I4 prototype (courtesy of LIRMM)

The use of base-mounted actuators and low-mass links allows the traveling plate to achieve great accelerations. This makes this type of parallel manipulator a perfect candidate for pick and place operations of light objects. Especially for semiconductor end-package equipments, which need high-speed and high-precision pick-and-place movement, these parallel manipulators provide a novel solution scheme.

In this chapter, only H4 is considered, because it firstly provides the most important concept of traveling plate. Based on the analytical method of H4, other types of evolution structure can be analyzed conveniently.



(a) Overview of Part4 prototype



(b) Articulated traveling plate

Fig.9. Part4 prototype (courtesy of LIRMM)

3. Kinematic analysis

This section will examine the relations between the actuated joint coordinates of H4 and the nacelle (or traveling plate) pose.

3.1 Inverse kinematics

The relation giving the actuated joint coordinates for a given pose of the end-effector is called the *inverse kinematics*, which is simple for parallel robots. The inverse kinematics consists in establishing the value of the joint coordinates corresponding to the end-effector configuration. Establishing the inverse kinematics is essential for the position control of parallel robots. There are multiple ways to represent the pose of a rigid body through a set of parameters \mathbf{X} . The most classical way is to use the coordinates in a reference frame of a given point C of the body, and three angles to represent its orientation. But there are other ways such as *kinematic mapping* which maps the displacement to a 6-dimensional hyperquadric, the *Study quadric*, in a seventh-dimensional projective space. The kinematic mapping may have an interest as equations involving displacement are algebraic (and the structure of algebraic varieties is better understood than other non-linear structures) and may have interesting properties, for example, stating that a point submitted to a displacement has to lie on a given sphere is easily written as a quadric equation using Study coordinates (Merlet, 2006).

3.1.1 Analytic method

For a fully-parallel mechanism, each of the chains link the base to the moving platform. If \mathbf{A} represents the end of the chain that is linked to the base, and \mathbf{B} the end of the chain that is linked to the moving platform. By construction the coordinates of \mathbf{A} are known in a fixed reference frame, while the coordinates of \mathbf{B} may be determined from the moving platform position and orientation. Hence the vector \mathbf{AB} is fundamental data for the inverse kinematic problem, this is why it plays a crucial role in the solution.

In order to write the position relationship of H4, consider the robotic structure (see Fig.10, chain #4 is not plotted for sake of simplicity). The geometrical parameters of the manipulator at hand are defined as follows:

Where $Rot(\mathbf{v}, \theta)$ denotes the rotation matrix of nacelle with respect to reference frame. The position of each point B_i is given by:

$$\begin{aligned} \mathbf{B}_1 &= \mathbf{C}_1 + \mathbf{C}_1 \mathbf{B}_1 & \mathbf{B}_2 &= \mathbf{C}_1 + \mathbf{C}_1 \mathbf{B}_2 \\ \mathbf{B}_3 &= \mathbf{C}_2 + \mathbf{C}_2 \mathbf{B}_3 & \mathbf{B}_4 &= \mathbf{C}_2 + \mathbf{C}_2 \mathbf{B}_4 \end{aligned} \quad (8)$$

The position relationship can then be written as:

$$\|\mathbf{A}_i \mathbf{B}_i\|^2 = M_i^2 \quad (9)$$

Then for the first axis:

$$\begin{aligned} \mathbf{A}_1 \mathbf{B}_1 &= [\mathbf{B} + Rot(\mathbf{v}, \theta)(\mathbf{B}\mathbf{C}_1) + \mathbf{C}_1 \mathbf{B}_1 - \mathbf{P}_1] - q_1 \mathbf{z}_1 \\ (\mathbf{A}_1 \mathbf{B}_1)^2 &= q_1^2 - 2q_1 \mathbf{d}_1 \cdot \mathbf{z}_1 + \mathbf{d}_1^2 \end{aligned} \quad (10)$$

where $\mathbf{d}_1 = \mathbf{B} + Rot(\mathbf{v}, \theta)(\mathbf{B}\mathbf{C}_1) + \mathbf{C}_1 \mathbf{B}_1 - \mathbf{P}_1$. Finally, the two solutions are given by:

$$q_1 = \mathbf{d}_1 \cdot \mathbf{z}_1 \pm \sqrt{(\mathbf{d}_1 \cdot \mathbf{z}_1)^2 + M_1^2 - \mathbf{d}_1^2} \quad (11)$$

Similar derivations give the solution for q_2 , q_3 and q_4 .

3.1.2 Geometrical method

The geometrical approach to the inverse kinematics problem is to consider that the extremities A , B of each chain have a known position in 3D space. The leg can be cut at a point M and two different mechanisms M_A , M_B constituted of the chain between A , M and the chain between B , M can be gotten. The free motion of the joints in these two chains will be such that point M , considered as a member of M_A , will lie on a variety V_A , while considered as a member of M_B it will lie on a variety V_B . If assume the mechanism have only classical lower pairs, these varieties will be algebraic with dimensions d_A , d_B . In the 3D space, a variety of dimension d is defined through a set of $3-d$ independent equations, and hence V_A , V_B will be defined by $3-d_A$ and $3-d_B$ equations. The solutions of the inverse kinematic problem lie at the intersection of these varieties. As the number of solutions must be finite (otherwise the robot cannot be controlled), the rank of the intersection variety must be 0. In other words, in order to determine the 3 coordinates of the points, $3-d_A+3-d_B$ should equal to 3 or d_A+d_B equal to 3 (Merlet, 2006).

The key problem about the geometrical method rests with the choice of the cutting point. For the H4 structure shown in Fig.10, B_i are chosen as the cutting points. Points B_i has to lie on a sphere centered at A_i with radius M_i , while for the nacelle, the coordinates of Points B_i can be described as (8). Hence to obtain the intersection of these 2 varieties the known distance between A , B of should be equal to M_i : this equation will give the joint coordinates q_i . The same results can be obtained as described as (11).

3.2 Direct kinematics

Direct kinematics addresses the problem of determining the pose of the end-effector of a parallel manipulator from its actuated joint coordinates. This relation has a clear practical interest for the control of the pose of the manipulator, but also for the velocity control of the end-effector.

In general, the solutions for determining the pose of the end-effector from measurements of the minimal set of joint coordinates that are necessary for control purposes is not unique. There are several ways of assembling a parallel manipulator with given actuated joint coordinates, and generally the direct kinematic relationship cannot be expressed in an analytical manner. Although various methods have been presented for finding all the solutions for this problem and their computation times are decreasing (Faugère, 1995; Hesselbach & Kerle, 1995; Gosselin, 1996; Merlet, 2004), it is still difficult to use these algorithms in a real time context. Furthermore, there is no known algorithm that allows the determination of the current pose of the platform among the set of solutions. The numerical methods using a-priori information on the current pose are more compatible with a real-time context, while their convergence and robustness is an important issue. As direct kinematics is an important issue for control, it may be necessary and interesting to investigate how to improve the computation time. Besides the progress in algorithms and processor speed, one possible approach to solve these problem is to add sensors (i.e. to have more than n sensors for a n -DOF manipulator) to obtain information, allowing a faster calculation of the current pose of the platform, at the cost of more complex hardware (Bonev et al., 2001; Chui & Perng; 2001; Parenti-Castelli, 2001). Merlet has pointed out several unsolved problems about the parallel robot's direct kinematics (Merlet, 2000). Especially the *algebraic geometry* and *computational kinematics* should bring about enough attention.

The first step of solving direct kinematics is to determine a bound on its maximal number of solution. Then, the equations are reduced for the system to obtain the solution of a univariate polynomial whose degree should be determined in the previous analysis. Many researchers have tried to obtained closed-form solutions of parallel robots in a univariate polynomial equation. Especially, the 3- and 6- DOFs parallel robots, e.g., planar, Delta, Steward platform and Hexapod robots, have been mainly considered. This section provides a univariate polynomial equation for H4. It is shown that the solutions of the forward kinematics yield a 16th degree polynomial in a single variable (Choi et al., 2003).

3.2.1 Forward kinematics formulation

The forward kinematics of the H4 is the problem of computing the position and orientation of the nacelle (traveling plate) form the motor angles or the moving distance of the actuator. To get a closed-form solution, a univariate polynomial equation needs to be solved. The following method is extracted from the paper written by Choi (Choi et al., 2003).

The kinematic structure of H4 is shown in Fig.10 and design parameters are shown in Fig.11. The nacelle is composed of three parts (two lateral bars and one central bar). The four points B_1 , B_2 , B_3 and B_4 form a parallelogram. Let ${}^A B_i$ and ${}^A A_i$ represent the homogeneous coordinates of the points B_i in $\{\mathbf{A}\}$ and A_i in $\{\mathbf{A}\}$ respectively. Then the homogeneous coordinates of ${}^A B_i$ and ${}^A A_i$ can be written as:

$${}^A B_i = \begin{bmatrix} x + hE_{1i} \cos \theta \\ y + hE_{1i} \sin \theta + dE_{2i} \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} P_i x + x \\ P_i y + y + Q_i \\ z \\ 1 \end{bmatrix} \quad (12)$$

$${}^A A_i = [a_i \quad b_i \quad q_i \quad 1]^T$$

where

$$i_x = \cos \theta, \quad i_y = \sin \theta, \quad P_i = hE_{1i}, \quad Q_i = dE_{2i}, \quad E_{11} = E_{12} = -1, \quad E_{13} = E_{14} = 1, \quad E_{21} = E_{24} = -1, \\ E_{22} = E_{23} = 1.$$

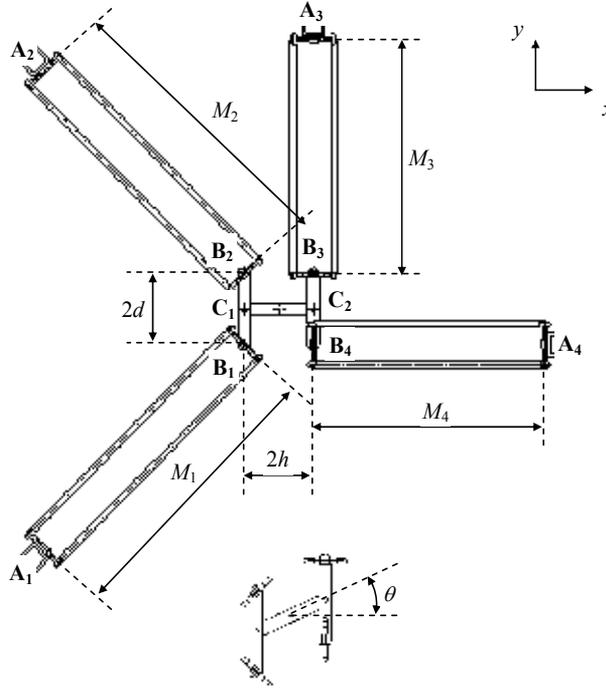


Fig.11. Design parameters of H4

The kinematic closure of each elementary chain can be written as:

$$\| {}^A \mathbf{A}_i \quad {}^A \mathbf{B}_i \|^2 = M_i^2 \tag{13}$$

Consequently, the each chain provides us with the following equation:

$$(P_i i_x + x - a_i)^2 + (P_i i_y + y + Q_i - b_i)^2 + (z - q_i)^2 = M_i^2 \tag{14}$$

Expanding and rearranging equation (14), the following equation is obtained:

$$\tilde{A}_i x + \tilde{B}_i y + \tilde{C}_i z + \tilde{D}_i + \tilde{E}_i = 0 \tag{15}$$

where,

$$\tilde{A}_i = 2(P_i i_x - a_i), \quad \tilde{B}_i = 2(P_i i_y + Q_i - b_i), \quad \tilde{C}_i = -2q_i, \\ \tilde{D}_i = a_i^2 + b_i^2 + q_i^2 - 2P_i(a_i i_x + b_i i_y - Q_i i_y) - 2Q_i b_i + P_i^2 + Q_i^2 - M_i^2, \quad \tilde{E}_i = x^2 + y^2 + z^2$$

From equation (15), the four equations become:

$$\tilde{A}_1 x + \tilde{B}_1 y + \tilde{C}_1 z + \tilde{D}_1 + \tilde{E}_1 = 0 \tag{16}$$

$$\tilde{A}_2x + \tilde{B}_2y + \tilde{C}_2z + \tilde{D}_2 + \tilde{E}_2 = 0 \quad (17)$$

$$\tilde{A}_3x + \tilde{B}_3y + \tilde{C}_3z + \tilde{D}_3 + \tilde{E}_3 = 0 \quad (18)$$

$$\tilde{A}_4x + \tilde{B}_4y + \tilde{C}_4z + \tilde{D}_4 + \tilde{E}_4 = 0 \quad (19)$$

Choosing any three from the above four equations, we can eliminate x^2 , y^2 and z^2 simultaneously and obtain an equation with variables x , y and z . Subtracting equation (18) from equation (17) and equation (19) from equation (17) respectively, the following two equations are obtained, which are used to eliminate x and y to obtain a polynomial in a single variable z .

$$\Delta A_{23}x + \Delta B_{23}y + \Delta C_{23}z + \Delta D_{23} = 0 \quad (20)$$

$$\Delta A_{24}x + \Delta B_{24}y + \Delta C_{24}z + \Delta D_{24} = 0 \quad (21)$$

where, $\Delta A_{23} = (\tilde{A}_2 - \tilde{A}_3)$, $\Delta B_{23} = (\tilde{B}_2 - \tilde{B}_3)$, $\Delta C_{23} = (\tilde{C}_2 - \tilde{C}_3)$, $\Delta D_{23} = (\tilde{D}_2 - \tilde{D}_3)$, $\Delta A_{24} = (\tilde{A}_2 - \tilde{A}_4)$, $\Delta B_{24} = (\tilde{B}_2 - \tilde{B}_4)$, $\Delta C_{24} = (\tilde{C}_2 - \tilde{C}_4)$, $\Delta D_{24} = (\tilde{D}_2 - \tilde{D}_4)$.

From equation (20) and (21), x and y are solved as:

$$x = \frac{\Delta_{11}z + \Delta_{12}}{\Delta_0} = e_1z + e_2 \quad (22)$$

$$y = \frac{\Delta_{21}z + \Delta_{22}}{\Delta_0} = e_3z + e_4 \quad (23)$$

where, $e_1 = \frac{\Delta_{11}}{\Delta_0}$, $e_2 = \frac{\Delta_{12}}{\Delta_0}$, $e_3 = \frac{\Delta_{21}}{\Delta_0}$, $e_4 = \frac{\Delta_{22}}{\Delta_0}$, with $\Delta_0 = \Delta A_{23}\Delta B_{24} - \Delta A_{24}\Delta B_{23}$,

$\Delta_{11} = \Delta B_{23}\Delta C_{24} - \Delta B_{24}\Delta C_{23}$, $\Delta_{12} = \Delta B_{23}\Delta D_{24} - \Delta B_{24}\Delta D_{23}$, $\Delta_{21} = \Delta A_{24}\Delta C_{23} - \Delta A_{23}\Delta C_{24}$,
 $\Delta_{22} = \Delta A_{24}\Delta D_{23} - \Delta A_{23}\Delta D_{24}$.

Substituting equations (22) and (23) into equation (17), the following quadratic equation is obtained:

$$\lambda_0z^2 + \lambda_1z + \lambda_2 = 0 \quad (24)$$

where, $\lambda_0 = e_1^2 + e_3^2 + 1$, $\lambda_1 = 2e_1e_2 + 2e_3e_4 + \tilde{A}_2e_1 + \tilde{B}_2e_3 + \tilde{C}_2$, $\lambda_2 = e_2^2 + e_4^2 + \tilde{A}_2e_2 + \tilde{B}_2e_4 + \tilde{D}_2$.

From equation (24), z can be calculated as:

$$z = \frac{-\lambda_1 + \rho}{2\lambda_0} \quad (25)$$

where,

$$\rho = \pm\sqrt{\lambda_1^2 - 4\lambda_0\lambda_2} \quad (26)$$

$$\mathbf{A}_3 = [6 \ 48 \ 0], \mathbf{B}_3 = [6 \ 6 \ -42], \mathbf{A}_4 = [48 \ -6 \ 0], \mathbf{B}_4 = [6 \ -6 \ -42],$$

$$\mathbf{u} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3 \ \mathbf{u}_4] = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ where } \mathbf{u}_i = \mathbf{B}_{i1} \mathbf{B}_{i2}.$$

When the position and orientation of the nacelle are set $x=5$, $y=-5$, $z=-30$, $\theta=\pi/6$ (rad), four actuators' position which is calculated by inverse kinematics are given by:

$$q_1=13.021, q_2=-7.488, q_3=9.679, q_4=15.770 \quad (33)$$

For the actuators' position given by equation (33), Matlab Symbolic Math Toolbox is used to solve the polynomial equation with variable T . The results indicate that there are only twelve solutions, the author believes that there are several repeated roots. This result needs further study. The two real roots are $T=0.268$ and $T=-2.882$ respectively. When set $T=0.268$, the configuration of the nacelle is calculated as: $x=5\text{mm}$, $y=-5\text{mm}$, $z=-30\text{mm}$, $\theta=0.524$ (rad), which is the expected solutions. When set $T=-2.882$, the configuration of the nacelle is calculated as: $x=3\text{mm}$, $y=-7.95\text{mm}$, $z=-14.58\text{mm}$, $\theta=-2.47$ (rad). This configuration cannot be realized in practice because of the mutual interference of the parallelograms as shown in Fig.11.

3.3 Jacobian matrix

Jacobian matrix relates the actuated joint velocities to the end-effector cartesian velocities, and is essential for the velocity and trajectory control of parallel robots. For parallel manipulators, their inverse Jacobian matrix can be established without a very high complexity, but their Jacobian matrix cannot be obtained directly, even with the help of symbolic computation, except in some particular cases (Bruyninckx, 1997; Pennock & Kassner, 1990). Theoretical analytic formulations of jacobians have been proposed, but require complicated matrix inversions (Dutr e et al., 1997; Kim et al., 2000). Generally, the difficulty of the inversion does not lie in the complexity of the algorithm but in the sheer size of the result (Merlet, 2006).

The velocity of the nacelle can be defined by resorting to a velocity for the translation, $\mathbf{v}_B = [\dot{x} \ \dot{y} \ \dot{z}]$ and a scalar for the rotation about \mathbf{v} , $\dot{\theta}$. Thus, the velocity of points C_1 and C_2 can be written as follows (Pierrot & Company, 1999):

$$\mathbf{v}_{C_1} = \mathbf{v}_B + \dot{\theta} \mathbf{v} \times \mathbf{BC}_1 \quad \mathbf{v}_{C_2} = \mathbf{v}_B + \dot{\theta} \mathbf{v} \times \mathbf{BC}_2$$

Moreover, since the links B_1B_2 and B_3B_4 move only in translation, the following relations hold:

$$\mathbf{v}_{B_1} = \mathbf{v}_{B_2} = \mathbf{v}_{C_1} \quad \mathbf{v}_{B_3} = \mathbf{v}_{B_4} = \mathbf{v}_{C_2}$$

On the other hand, velocity of points A_i is given by:

$$\mathbf{v}_{A_i} = \dot{q}_i \mathbf{z}_i$$

where \mathbf{z}_i is the unit vector along the direction of prismatic. The velocity relationship can then be written thanks to the classical property:

$$\mathbf{v}_{A_i} \cdot \mathbf{A}_i \mathbf{B}_i = \mathbf{v}_{B_i} \cdot \mathbf{A}_i \mathbf{B}_i \quad (34)$$

Equation (34) can be written for $i=1, \dots, 4$ and the results grouped in a matrix form, such as:

$$\mathbf{J}_q \dot{\mathbf{q}} = \mathbf{J}_x \dot{\mathbf{x}}$$

where

$$\mathbf{J}_q = \begin{bmatrix} (\mathbf{A}_1 \mathbf{B}_1 \cdot \mathbf{z}_1) & 0 & 0 & 0 \\ 0 & (\mathbf{A}_2 \mathbf{B}_2 \cdot \mathbf{z}_2) & 0 & 0 \\ 0 & 0 & (\mathbf{A}_3 \mathbf{B}_3 \cdot \mathbf{z}_3) & 0 \\ 0 & 0 & 0 & (\mathbf{A}_4 \mathbf{B}_4 \cdot \mathbf{z}_4) \end{bmatrix}$$

$$\dot{\mathbf{q}} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4]^T$$

$$\mathbf{J}_x = \begin{bmatrix} (\mathbf{A}_1 \mathbf{B}_1)_x & (\mathbf{A}_1 \mathbf{B}_1)_y & (\mathbf{A}_1 \mathbf{B}_1)_z & (\mathbf{A}_1 \mathbf{B}_1 \times \mathbf{BC}_1) \cdot \mathbf{v} \\ (\mathbf{A}_2 \mathbf{B}_2)_x & (\mathbf{A}_2 \mathbf{B}_2)_y & (\mathbf{A}_2 \mathbf{B}_2)_z & (\mathbf{A}_2 \mathbf{B}_2 \times \mathbf{BC}_1) \cdot \mathbf{v} \\ (\mathbf{A}_3 \mathbf{B}_3)_x & (\mathbf{A}_3 \mathbf{B}_3)_y & (\mathbf{A}_3 \mathbf{B}_3)_z & (\mathbf{A}_3 \mathbf{B}_3 \times \mathbf{BC}_2) \cdot \mathbf{v} \\ (\mathbf{A}_4 \mathbf{B}_4)_x & (\mathbf{A}_4 \mathbf{B}_4)_y & (\mathbf{A}_4 \mathbf{B}_4)_z & (\mathbf{A}_4 \mathbf{B}_4 \times \mathbf{BC}_2) \cdot \mathbf{v} \end{bmatrix}$$

$$\dot{\mathbf{x}} = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\theta}]$$

If the mechanism is not in a singular configuration, the inverse Jacobian matrix is described as:

$$\mathbf{J}^{-1} = \mathbf{J}_q^{-1} \mathbf{J}_x \quad (35)$$

And the Jacobian matrix is:

$$\mathbf{J} = \mathbf{J}_x^{-1} \mathbf{J}_q \quad (36)$$

When the determinant of inverse Jacobian matrix equals to zero, the parallel manipulator is in a singular configuration, which is a general method for identifying the singular configurations of parallel manipulators. But for a manipulator with $n < 6$ DOF, it should be noted that it may not be efficient to identify all the singular configurations by determining only the $n \times n$ inverse kinematic Jacobian that relates the actuated joint velocities to the possible DOF velocities. Next section will discuss an inverse kinematic Jacobian that involves actuated joints velocities and the full twist of the end-effector, which may be essential for singularity analysis. This type of matrix is coined as *overall jacobian* by Joshi (Joshi & Tsai, 2002), while Merlet called it a *full inverse kinematic jacobian* (Merlet, 2006).

3.4 Determination of the joint velocities and twist

The purpose of this section is to calculate the actuated joint (or end-effector) velocities, being given the end-effector (or actuated joint) velocities.

In the previous section, equation (35) can be applied to compute the actuated joint velocities directly. For example, the structure and design parameters of H4 is described in Fig.10 and

Fig.11, where the following parameters are chosen as described in section 3.2.2. So, according to equation (35), the inverse Jacobian matrix can be expressed as:

$$\mathbf{J}^{-1} = \begin{bmatrix} -1 & -1 & 1 & -6 \\ -1 & 1 & 1 & 6 \\ 0 & 1 & 1 & -6 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

This inverse Jacobian matrix can be used to calculate the actuated joint velocity directly when the pose of the nacelle is shown in Fig.11.

It is usually difficult, even for robots with less than 6-DOFs robots, to invert \mathbf{J}^{-1} analytically. So a numerical procedure will be needed to calculate the twist of the nacelle from the joint velocities. For a given pose of the end-effector, there are two methods to determine the Jacobian matrix, namely, a numerical inversion algorithm and the quasi-Newton scheme (Reboulet, 1985; Merlet, 2006).

$$\mathbf{J}_{k+1} = \mathbf{J}_k + \mathbf{J}_0(\dot{\mathbf{q}} - \mathbf{J}^{-1}\mathbf{J}_k) \quad (37)$$

where $\dot{\mathbf{q}}$ is the actuated joint velocity; \mathbf{J}_0 denotes the kinematic jacobian matrix calculated for a given pose. The algorithm stops when the differences between the joint velocities and those that are calculated from the twist are lower than a fixed threshold.

For the H4 structure and design parameters described above, when the actuated joint velocities is $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4] = [1 \ -1 \ 1 \ 1]$, the speed of the nacelle can reach $\dot{\mathbf{x}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\theta}] = [0.5 \ -0.25 \ 0.5 \ -0.125]$.

3.5 Accelerations analysis

This section will determine what are the relations between the actuated joint accelerations and the cartesian and angular accelerations of the end-effector. H4 presents excellent characteristics as to acceleration which can reach 10g. For parallel robots it is generally easy to obtain these relations directly. From the equation $\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$, the following can be obtained by differentiation

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}\ddot{\mathbf{x}} + \dot{\mathbf{J}}^{-1}\dot{\mathbf{x}} \quad (38)$$

For the various categories of parallel manipulators, the determination of the acceleration equations thus amounts to the determination of the derivative of the inverse kinematic jacobian matrix (Merlet, 2006).

But for H4, the method described above is not intuitionistic. So we derive the relations between the actuated joint accelerations and the accelerations of the end-effector using geometric method. From equation (12), accelerations of the point B_i and A_i in $\{\mathbf{A}\}$ can be obtained as:

$$\begin{aligned} a_{B_i} &= [\ddot{x} - hE_{i1} \cos\theta \cdot \ddot{\theta} \quad \ddot{y} - hE_{i1} \sin\theta \cdot \ddot{\theta} \quad \ddot{z}]^T \\ a_{A_i} &= [0 \quad 0 \quad \ddot{q}_i]^T \end{aligned} \quad (39)$$

The acceleration projection of points B_i and A_i on the line A_iB_i is equal. So the following equation comes into existence:

$$a_{A_i} \cdot \mathbf{A}_i \mathbf{B}_i = a_{B_i} \cdot \mathbf{A}_i \mathbf{B}_i \quad (40)$$

Equation (40) can be written for $i=1, \dots, 4$ and the results grouped in a matrix form, such as:

$$\mathbf{J}_q \ddot{\mathbf{q}} = \mathbf{J}_a \ddot{\mathbf{x}} \quad (41)$$

where

$$\ddot{\mathbf{q}} = [\ddot{q}_1 \quad \ddot{q}_2 \quad \ddot{q}_3 \quad \ddot{q}_4]^T$$

$$\mathbf{J}_a = \begin{bmatrix} (\mathbf{A}_1 \mathbf{B}_1)_x & (\mathbf{A}_1 \mathbf{B}_1)_y & (\mathbf{A}_1 \mathbf{B}_1)_z & -((\mathbf{A}_1 \mathbf{B}_1)_x \cos \theta + (\mathbf{A}_1 \mathbf{B}_1)_y \sin \theta) \cdot hE_{11} \\ (\mathbf{A}_2 \mathbf{B}_2)_x & (\mathbf{A}_2 \mathbf{B}_2)_y & (\mathbf{A}_2 \mathbf{B}_2)_z & -((\mathbf{A}_2 \mathbf{B}_2)_x \cos \theta + (\mathbf{A}_2 \mathbf{B}_2)_y \sin \theta) \cdot hE_{12} \\ (\mathbf{A}_3 \mathbf{B}_3)_x & (\mathbf{A}_3 \mathbf{B}_3)_y & (\mathbf{A}_3 \mathbf{B}_3)_z & -((\mathbf{A}_3 \mathbf{B}_3)_x \cos \theta + (\mathbf{A}_3 \mathbf{B}_3)_y \sin \theta) \cdot hE_{13} \\ (\mathbf{A}_4 \mathbf{B}_4)_x & (\mathbf{A}_4 \mathbf{B}_4)_y & (\mathbf{A}_4 \mathbf{B}_4)_z & -((\mathbf{A}_4 \mathbf{B}_4)_x \cos \theta + (\mathbf{A}_4 \mathbf{B}_4)_y \sin \theta) \cdot hE_{14} \end{bmatrix}$$

$$\ddot{\mathbf{x}} = [\ddot{x} \quad \ddot{y} \quad \ddot{z} \quad \ddot{\theta}]$$

For the H4 structure and design parameters described above, when the nacelle is in the origin pose and the actuated joint accelerations are $\ddot{\mathbf{q}} = [10 \quad -10 \quad 10 \quad 10] \text{m/s}^2$, the linear acceleration of the nacelle can reach 23m/s^2 and the angular acceleration can reach 2.5rad/s^2 .

3.6 Kinetostatic performance indices

3.6.1 Manipulability

The inverse kinematic jacobian matrix reflects a linear relation between the manipulator accuracy $\Delta \mathbf{x}$ and the measurement errors $\Delta \mathbf{q}$ on \mathbf{q} . If the measurement errors are bounded, then the hyper-sphere in the joint error space can be mapped into an ellipsoid in the generalized Cartesian error space with using the Euclidean norm. This ellipsoid is usually called the *manipulability ellipsoid*. If using the infinity norm, the joint errors are restricted to lie in a hyper-cube in the joint error space. The hyper-cube in the joint error space can be mapped into the *kinematic polyhedron* in the positioning errors space (Merlet, 2006). The shape and volume of the manipulability ellipsoid and the kinematic polyhedron characterize the manipulator dexterity. It is necessary to set up some kinetostatic performance indices that are used to quantify the dexterity of a robot. The famous Yoshikawa's *manipularity index* $\sqrt{\mathbf{J}\mathbf{J}^T}$ is used for serial robots for a long time. Another index is the *condition number* that is often used for parallel robots.

$$\kappa = \|\mathbf{J}^{-1}\| \|\mathbf{J}\| \quad (42)$$

The condition number expresses how a relative error in joint space gets multiplied and leads to a relative error in work space. The condition number is dependent on the choice of the metric norm and the most used norms are the 2-norm and the Euclidean norm. The condition number is mentioned as the main index for characterizing the accuracy of parallel robots. But there is major drawback to the condition number for H4, since the elements of the matrix corresponding to translations are dimensionless, whereas those corresponding to the rotations are lengths. A direct consequence is that the condition number has no clear

physical meaning, as the rotations are transformed arbitrarily into “equivalent” translations (Merlet, 2006). There are various proposals that have been made to avoid this drawback (Gosselin, 1990; Kim & Ryu, 2003). But all these proposals have their own special drawbacks. How to design a general law for parallel robots’ manipulability is still an open problem.

3.6.2 Isotropy

Poses with a condition number of 1 are called *isotropic poses*, and robots having only such type of poses are called isotropic robots (Merlet, 2006). At the given pose as shown in Fig.11, the condition number of H4 is about 8.56 and there is no any isotropic poses in the workspace of H4 (Company et al., 2005). While Part4 robots which are the evolutionary structure of H4 are isotropic at their original poses.

There are some other manipulability and accuracy indices, such as global conditioning index, uniformity of manipulability and maximal positioning error index et al. All the above definitions of the kinetostatic indices do not take into account other factor affecting the accuracy of parallel robots, such as manufacturing tolerances, clearance and friction in the joints. In order to include these factors in accuracy analysis, Monte Carlo statistical simulation technique can be used to evaluate the accuracy of parallel manipulators (Ryu & Cha, 2003).

4. Singularity analysis

This section will identify all the singular configurations of the H4 and analyze the manipulator’s self-motion when in or closed to a singular configuration. Generally, singular configurations refer to particular poses of the end-effector, for which parallel robots lose their inherent infinite rigidity, and in which the end-effector will have uncontrollable degrees of freedom. But for this new family of parallel manipulator H4, singularities are associated with either loss or gain of DOF. This section utilizes line geometry tools and screw theory to deal with singularity analysis of H4. Firstly, the basic theory including in the process of singularity analysis is introduced briefly. Then the static equilibrium condition of the end-effector is derived to obtain the *full inverse kinematic jacobian* 6×6 matrix, which is set of governing lines of the manipulator. Based on linear dependency of these lines, the singular configurations of the manipulator can be identified. Moreover, in order to deal with singularities associated with loss of DOFs (serial singularity), the static equilibrium of the actuators is also defined. Secondly, architecture and constraint singularities associated with gain of DOFs (parallel singularity) are defined and analyzed using linear complex approximation algorithm (LCAA), which is employed to obtain the closest linear complex, presented by its screw coordinates, to the set of governing lines. The linear complex axis and pitch provide additional information and a better physical understanding of the manipulator’s self-motion when in or closed to a singular configuration. Lastly, various singularities of an example H4 manipulator are presented and analyzed using the proposed methods (Wu et al., 2006).

4.1 Basic theory

In the context of designing a parallel manipulator, understanding the intrinsic nature of singularities and their relations with the kinematic parameters and the configuration spaces

is of great importance. The phenomena of singularity in parallel manipulators have been approached from different points of view. One way to introduce singular configurations is to examine the relations obtained for inverse kinematics. Singularities correspond to the roots of the parallel manipulator Jacobian's determinant. Based on the rank deficiency associated with the Jacobian matrices, singularities of parallel manipulators have been explained (Gosselin & Angeles, 1990; Zlatanov et al., 1994). Using the linear decomposition method and co-factor expansion, St-Onge and Gosselin (St-Onge & Gosselin, 2000) studied the singularity loci of the Gough-Stewart platform, and obtained a graphical representation of these loci in the manipulator workspace. Due to the complexity of the kinematic model, several authors proposed numerical methods to analyze the singularities (Feng-Cheng & Haug, 1994; Funabashi & Takeda, 1995). However, even with symbolic computation software, the calculation of the determinant of Jacobian matrix is a complicated task. Moreover, identifying kinematic singularity through matrices does not provide physical insight into the nature of singular configuration of parallel manipulators.

A parallel manipulator is naturally associated with a set of constraint functions defined by its closure constraints (geometric relations of the closed-chain mechanism). The differential forms arising from these constraint functions completely characterize the geometric properties of the manipulator. So Liu et al. (Liu et al., 2003) used differential geometric tools to study singularities of parallel mechanisms and provided a finer classification of singularities. In their works, they classified singularity into configuration space singularities and parametrization singularities which include actuator and end-effector singularities as their special cases. But the method is too abstract and complicated when dealing with singularities of spatial parallel manipulators.

Another approach to analyze the parallel manipulators' singularity is based on line geometry and screw theory. Merlet (Merlet, 1992) considered the Jacobian matrix of the Gough-Stewart platform as the Plücker line coordinates of the robot's actuators and identified the singular configurations of the robot. In (Romdhane et al., 2002), a mixed geometric and vector formulation is used to investigate the singularities of a 3-translational-DOF parallel manipulator. In (Collins, 1995), singularities of an in-parallel hand controller for force-reflected teleoperation were analyzed, the six columns of the Jacobian matrix were viewed as zero-pitch wrenches (lines) acting on the top platform, then line geometry and rank determining geometric constructions were used to obtain all configuration singularities. Basu and Ghosal (Basu & Ghosal, 1997) presented a geometric condition to deal with the singularity analysis associated with gain of DOF in a class of platform-type, multi-loop spatial manipulators. Joshi and Tsai (Joshi & Tsai, 2002) developed a methodology for the Jacobian analysis of limited-DOF parallel manipulators by making use of the theory of reciprocal screws. A 6×6 Jacobian matrix so derived provided information about both architecture and constraint singularities.

The present investigation identifies all the singular configurations of the H4 robot and provides physical insight into the nature of these singular configurations from the view point of geometry. Moreover, the behavior in these singular configurations or in the neighboring ones is also determined using linear complex approximation. All these analyses need some basic theories, include Grassmann geometry, Line geometry and Screw theory etc. The following gives a brief presentation of these theories.

4.1.1 Line geometry and Plücker coordinates

Line geometry investigates the set of lines in three-space. The ambient space can be a real projective, affine, Euclidean or a non-Euclidean space. Line geometry possesses a close relation to spatial kinematics (Bottema & Roth, 1990) and has therefore found applications in mechanism design and robot kinematics. A parallel manipulator has a singular position if the axes of its hydraulic cylindrical legs lie in a linear complex, a special three-parameter set of lines. In practice, several sources for errors (manufacturing, material properties, computing, etc.) can hardly be avoided. Thus, the question is whether the lines on the objects near their realization are close – within some tolerance – to the lines of a linear complex. This is an approximation or regression problem in line space (Helmut et al., 1999). In real Euclidean space E^3 , a Cartesian coordinate system is used and the points are represented by their coordinate vectors \mathbf{M} . A straight line L can be determined by a point $\mathbf{p} \in L$ and a normalized direction vector \mathbf{l} of L , i.e. $\|\mathbf{l}\|=1$. To obtain coordinates for L , one forms the moment vector

$$\bar{\mathbf{l}} := \mathbf{p} \times \mathbf{l} \quad (43)$$

with respect to the origin. If \mathbf{p} is substituted by any point $\mathbf{q} = \mathbf{p} + \lambda \mathbf{l}$ on L , equation (43) implies that $\bar{\mathbf{l}}$ is independent of \mathbf{p} on L . The six coordinates $(\mathbf{l}, \bar{\mathbf{l}})$ with

$$\mathbf{l} = (l_1, l_2, l_3), \text{ and } \bar{\mathbf{l}} = (l_4, l_5, l_6)$$

are called the *normalized Plücker coordinates* of L . The set of lines in E^3 is a four-dimensional manifold and accordingly the six Plücker coordinates satisfy two relations. One is the normalization and the other is, by equation (43), the so-called *Plücker relation*

$$\mathbf{l} \cdot \bar{\mathbf{l}} = 0 \quad (44)$$

which expresses the orthogonality between \mathbf{l} and $\bar{\mathbf{l}}$. Conversely, any six-tuple $(\mathbf{l}, \bar{\mathbf{l}})$ with $\|\mathbf{l}\|=1$, which satisfies the Plücker relation $\mathbf{l} \cdot \bar{\mathbf{l}} = 0$ represents a line in E^3 . As the orientation are not concerned, $(\mathbf{l}, \bar{\mathbf{l}})$ and $(-\mathbf{l}, -\bar{\mathbf{l}})$ describe the same line L .

The topic about the Klein mapping and special sets of lines can refer to the paper written by Pottmann et al. (Pottmann et al., 1999).

4.1.2 Grassmann geometry

As shown in the previous section, two lines with Plücker coordinates $\mathbf{l}_1 = [\mathbf{p}_1, \mathbf{q}_1]$, $\mathbf{l}_2 = [\mathbf{p}_2, \mathbf{q}_2]$ intersect if and only if $\mathbf{p}_1 \cdot \mathbf{q}_1 + \mathbf{q}_2 \cdot \mathbf{p}_2 = 0$. Plücker vectors with $\mathbf{p} = 0$ do not represent real lines and are associated with a line at infinity. All lines at infinity belong to a plane, the plane at infinity. A point may also be represented by the Plücker coordinates (α, \mathbf{r}) so that its coordinates are \mathbf{r}/α . If $\alpha=0$, then the point is at infinity, and a point $(0, \mathbf{r})$ at infinity is on the line at infinity $(0, \mathbf{q})$ if and only if $\mathbf{r} \cdot \mathbf{q} = 0$. Consequently a point at infinity that belongs to the two lines at infinity $(0, \mathbf{s}_1)$, $(0, \mathbf{s}_2)$ has coordinates $(0, \mathbf{s}_1 \times \mathbf{s}_2)$.

The columns of the full inverse kinematic jacobian matrices of most parallel robots are constructed from the Plücker vectors of lines associated with links of the manipulator. The singularity of this matrix therefore means that there will be a linear dependence between these vectors. Grassmann showed that linear dependence of Plücker vectors induced

geometric relations between the associated lines, so that a set of n Plücker vectors creates a variety with dimension $m < n$. A thorough introduction to Grassmann geometry please refers to (Pottmann et al., 1999). The applications of Grassmann geometry can be found in references (Collins & Long, 1995; Hao & McCarthy, 1998; Wu et al., 2006).

Using the Grassmann's geometrical conditions, an algorithm for finding the singular configurations of any type of parallel robot whose full inverse kinematic jacobian consists of Plücker vectors should be designed. Firstly consider all sets of n lines that are associated with the Plücker vectors constructed from full inverse kinematic jacobian matrices, and then the pose of the moving platform is determined so that the n lines satisfy one of the geometrical conditions which ensure that they span a variety of dimension $n-1$, thereby leading to a singularity of the robot. The following list the geometric conditions that ensure that the dimension of the variety spanned by a set of $n+1$ Plücker vectors is n , for each possible dimension n of the variety (Merlet, 2006). It should be noted that the notation of intersecting lines also include parallel lines which intersect at infinity. To consider the reading convenience, the following results is excerpted from the monograph written by Merlet (Merlet, 2006).

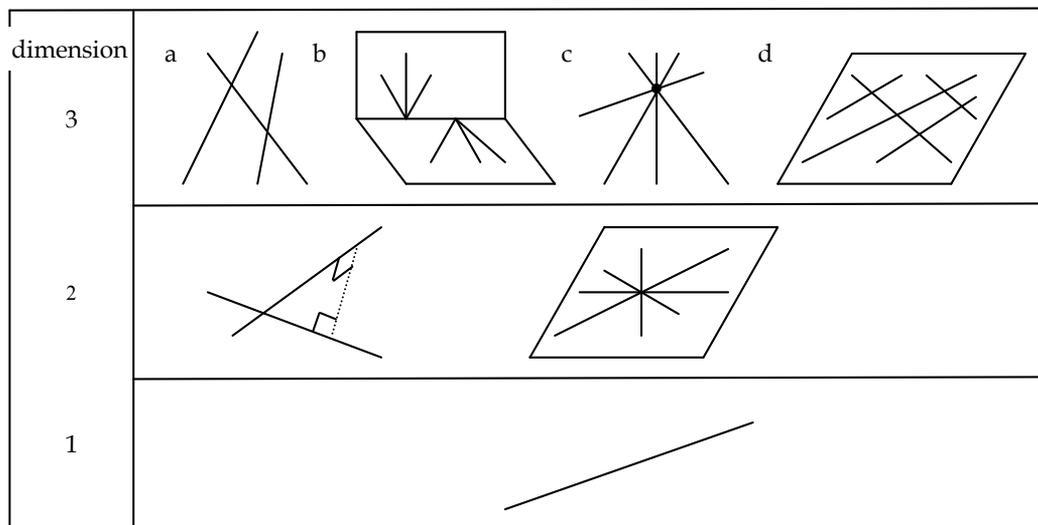


Fig.12. Geometrical conditions that characterize the varieties of dimension 1, 2 and 3 (Merlet, 2006)

For the variety of dimension 1 (called a point) there is just one Plücker vector and one line. A variety of dimension 2, called a line, may be constituted either by two Plücker vectors for which the associated lines are skew, i.e. they do not intersect and they are not parallel, or be spanned by more than two Plücker vectors if the lines that are associated with the vectors form a planar pencil of lines, i.e. they are coplanar and possess a common point (possibly at infinity, to cover the case of coplanar parallel lines). A variety of dimension 3, called a plane, is the set of lines F that are dependent on 3 lines F_1, F_2, F_3 . It is possible to show that all the points belonging to the lines F lie on a quadric surface Q . This quadric degenerates to a pair of planes P_1, P_2 if any two of the three lines intersect.

- condition 3d: all the lines are coplanar, but do not constitute a planar pencil of lines; F_1, F_2, F_3 are coplanar and P_1, P_2 are coincident.

- condition 3c: all the lines possess a common point, but they are not coplanar; F_1, F_2, F_3 intersect at the same point, possibly at infinity (this covers the case of parallel lines).
- condition 3b: all the lines belong to the union of two planar pencils of non coplanar lines that have a line L in common; two of the lines intersect at a point \mathbf{p} , and L intersect the last line at \mathbf{a} . Two different cases may occur:
 - P_1, P_2 are distinct and intersect along the line L . The set of dependent lines are the lines in P_1 that go through \mathbf{a} , and the lines in P_2 that go through \mathbf{p}
 - P_1, P_2 are distinct and parallel. This occurs if two of the lines F_i are parallel; L is a line at infinity, and the set of dependent lines are two planes of parallel lines
- condition 3a: all the lines belong to a regulus; F_1, F_2, F_3 are skew.

A variety of dimension 4, called a congruence, corresponds to a set of lines which satisfies one of the following 4 conditions:

- condition 4d: all the lines lie in a plane or meet a common point that lies within this plane. This is a degenerate congruence.
- condition 4c: all the lines belong to the union of three planar pencils of lines, in different planes, but which have a common line. This is a parabolic congruence.
- condition 4b: all the lines intersect two given skew lines. This is a hyperbolic congruence.
- condition 4a: the variety is spanned by 4 skew lines such that none of these lines intersects the regulus that is generated by the other three. This is an elliptic congruence.

A variety of dimension 5, called a linear complex, is defined by two 3-dimensional vectors $(\mathbf{c}, \bar{\mathbf{c}})$ as the set of lines L with Plücker coordinates $(\mathbf{l}, \bar{\mathbf{l}})$ such that $\bar{\mathbf{c}} \cdot \mathbf{l} + \mathbf{c} \cdot \bar{\mathbf{l}} = 0$. The complex may be:

- condition 5b: all the lines of the complex intersect the line with Plücker coordinates $(\mathbf{c}, \bar{\mathbf{c}})$, namely, $\mathbf{c} \cdot \bar{\mathbf{c}} = 0$. This is a singular configuration.
- condition 5a: $\mathbf{c} \cdot \bar{\mathbf{c}} \neq 0$. This is a general or non singular configuration.

The degree of freedom associated with a linear complex is a screw motion with axis defined by the line with Plücker vector $(\mathbf{c}, \bar{\mathbf{c}} - p\mathbf{c})/\|\mathbf{c}\|$, where $p = \mathbf{c} \cdot \bar{\mathbf{c}}/\|\mathbf{c}\|^2$ is the pitch of the motion.

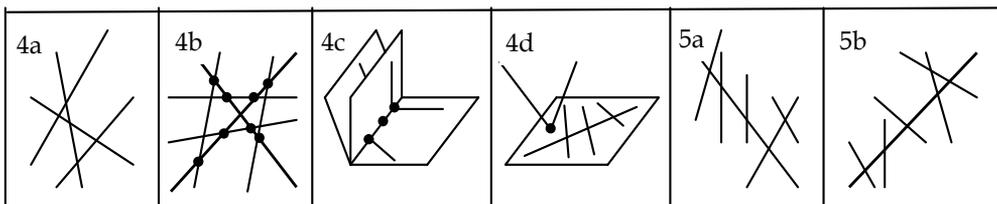


Fig.13. Geometrical conditions that characterize the varieties of dimension 4 and 5 (Merlet, 2006)

4.1.3 Screw theory

According to screw theory, the instantaneous velocities of a rigid body can be described by a 6-dimensional vector of the form $(\boldsymbol{\omega}, \mathbf{v})$, where $\boldsymbol{\omega}$ is the angular velocity of the rigid body, and \mathbf{v} is its translational velocity. These elements are called *velocity twists* or *screws*. Forces and torques exerted on the rigid body are important for motion and may be represented as a

couple of 3-dimensional vectors (\mathbf{F}, \mathbf{M}) called a *wrench*. A twist and a wrench will be said to be reciprocal if

$$\mathbf{F} \cdot \mathbf{v} + \mathbf{M} \cdot \boldsymbol{\omega} = 0 \quad (45)$$

When a kinematic chain is connected to a rigid body the key point is that the possible instantaneous twists for the rigid body are reciprocal to the wrenches imposed by the kinematic chains (called the constraint wrenches). In other words, the DOF of the rigid body are determined by the constraint wrenches. So based on linear dependency of the constraint wrenches, the singular configurations can be identified. Another function of the screw theory is the mobility analysis (Li & Huang, 2003).

4.2 Static analysis of the H4

When deriving the Jacobian matrix of the H4 using the velocity-equation formulation described in section 3.3, the result is a 4×4 Jacobian matrix because of the 4-DOF of the manipulator. However, it is not clear as to this is the best way to express the Jacobian of this type of limited-DOF parallel manipulator when singular analysis is processed. Because this approach is valid for general-purpose planar or spatial parallel manipulators, for which the connectivity of each serial chain (limb) is equal to the mobility of the end effector, it is not necessary true for parallel manipulators with less than 6-DOF (Joshi & Tsai, 2002). For example, this approach leads to a 3×3 Jacobian matrix for the 3-UPU parallel manipulator assembled for pure translation (Tsai, 1996; Tsai & Joshi, 2000). However, such a 3×3 Jacobian matrix cannot predict all possible singularities. Di Gregorio and Parenti-Castelli (Di Gregorio & Parenti-Castelli, 1999) analyzed the singularities of the 3-UPU translational platform. They derived the conditions for which the actuators cannot control the linear velocity of the moving platform, generally known as *architecture singularities*. Bonev and Zlatanov also found that under certain configurations the 3-UPU translational platform would gain rotational DOFs, which was called *constraint singularities* (Bonev and Zlatanov, 2001). Constraint singularities occur when the limbs of a limited-DOF parallel manipulator lose their ability to constrain the moving platform to the intended motion.

The 4×4 Jacobian matrix of the H4 cannot predict all possible singularities. Under certain configurations the chains of the H4 parallel manipulator lose their ability to constrain the moving platform to the intended motion, and the nacelle will gain additional DOF(s). So it is necessary to derive a full 6×6 matrix of the robot. In order to obtain the full 6×6 matrix, including the moments of constraints, one can express the set of static equilibrium conditions of the nacelle. These expressions result in a 6×8 matrix that maps the external wrench acting on the nacelle to the moments of the internal forces that are generated by the $(S-S)_2$ chains. Taking into account the constraint relations between the $(S-S)_2$ chains, we can derive a 6×6 matrix. This matrix's rows describe the six governing lines, from which the manipulator's singularities are obtained.

The structure and the geometrical parameters of the manipulator are shown in Fig.10. The static analysis of H4 manipulator can be divided into two parts. One regards the static equilibrium of the nacelle with B_1B_2 and B_3B_4 . The second deals with the static equilibrium conditions of the parallelograms which are composed of P-(S-S)₂ chains. For a P-(S-S)₂ chain, $\mathbf{A}_{i1}\mathbf{B}_{i1} = \mathbf{A}_{i2}\mathbf{B}_{i2} = \mathbf{A}_i\mathbf{B}_i$, \mathbf{A}_i and \mathbf{B}_i are centers of segments $\mathbf{A}_{i1}\mathbf{B}_{i1}$ and $\mathbf{A}_{i2}\mathbf{B}_{i2}$. Every chain has 5 DOFs, and provides the same type: 3 translations and 2 rotations. Since both rods

$A_{i1}B_{i1}$ and $A_{i2}B_{i2}$ are considered as plain solids, the impossible motion is the rotation about the vector: $A_iB_i \times u_i$, where $u_i = B_{i1}B_{i2}$. Observing the structure of the parallel part of the (S-S)₂ chain in Fig.2, one can detect that due to the spherical joint, there are moments exerted by the rods to B_i (the direction is along $A_iB_i \times u_i$) and forces to B_i (the direction is along B_iA_i), the static equilibrium for the i th chain is given by (see Fig.14):

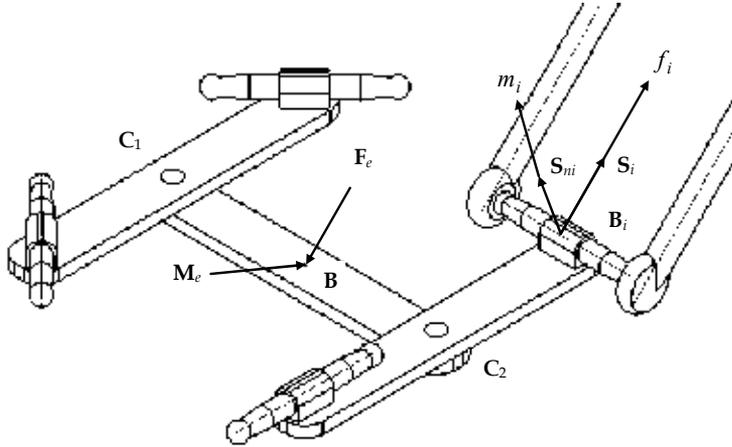


Fig.14. Forces and moments transmitted to the nacelle

$$\sum_{i=1}^4 f_i S_i - F_e = 0 \tag{46}$$

$$\sum_{i=1}^4 {}^A R_B r_i \times S_i \cdot f_i + \sum_{i=1}^4 m_i S_{ni} - M_e = 0$$

where ${}^A R_B$ is a rotation matrix, which expresses the orientation of the nacelle with respect to the fixed reference frame; r_i is a vector connecting B to B_i ; S_i and f_i are the unit direction vectors and the internal forces acting on B_i in the direction of B_iA_i respectively; S_{ni} and m_i are the unit direction vectors and the internal moments acting on the B_i in the direction of $A_iB_i \times u_i$. Writing equation (46) in a matrix form yields:

$$\begin{bmatrix} S_1 & S_2 & S_3 & S_4 & 0 & 0 & 0 & 0 \\ {}^A R_B r_1 \times S_1 & {}^A R_B r_2 \times S_2 & {}^A R_B r_3 \times S_3 & {}^A R_B r_4 \times S_4 & S_{n1} & S_{n2} & S_{n3} & S_{n4} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} F_e \\ M_e \end{bmatrix} \tag{47}$$

the matrix in the left of equation (47) is a 6×8 matrix. In order to obtain the 6×6 matrix, the constraint relations between $f_i, m_i (i = 1,2,3,4)$ should be analyzed.

Link C_1C_2 is connected, on one side, to B_1B_2 by a revolute joint whose direction is represented by vector \mathbf{v}_1 , on the other side, to B_3B_4 by a revolute joint whose direction is represented by vector \mathbf{v}_2 . Thus, we can consider two 5-DOF “meta-chains” on each side of the nacelle. Those “meta-chains” have the following properties (Pierrot & Company, 1999):

- First “meta-chain”: the rotation about $(\mathbf{S}_{n1} \times \mathbf{S}_{n2}) \times \mathbf{v}_1$ is impossible.
- Second “meta-chain”: the rotation about $(\mathbf{S}_{n3} \times \mathbf{S}_{n4}) \times \mathbf{v}_2$ is impossible.

So, the moment’s direction exerted by B_1B_2 to C_1C_2 is along $(\mathbf{S}_{n1} \times \mathbf{S}_{n2}) \times \mathbf{v}_1$, and the moment’s direction exerted by B_3B_4 to C_1C_2 is along $(\mathbf{S}_{n3} \times \mathbf{S}_{n4}) \times \mathbf{v}_2$. The static equilibrium of C_1C_2 is given by (see Fig.15):

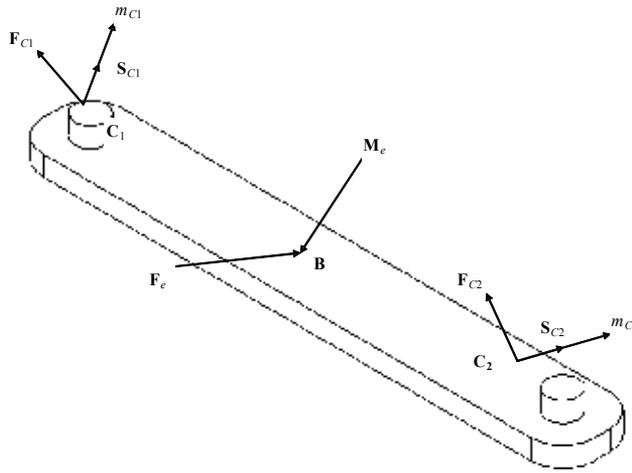


Fig.15. Forces and moments transmitted to C_1C_2

$$\begin{aligned} \mathbf{F}_{C1} + \mathbf{F}_{C2} &= \mathbf{F}_e \\ \mathbf{BC}_1 \times \mathbf{F}_{C1} + \mathbf{BC}_2 \times \mathbf{F}_{C2} + m_{C1} \mathbf{S}_{C1} + m_{C2} \mathbf{S}_{C2} &= \mathbf{M}_e \end{aligned} \tag{48}$$

where

$$\begin{aligned} \mathbf{F}_{C1} &= f_1 \mathbf{S}_1 + f_2 \mathbf{S}_2 \\ \mathbf{F}_{C2} &= f_3 \mathbf{S}_3 + f_4 \mathbf{S}_4 \end{aligned} \tag{49}$$

are the internal force vectors acting on the points C_1 and C_2 , m_{C1} and m_{C2} are moments transmitted to C_1C_2 ; \mathbf{S}_{C1} and \mathbf{S}_{C2} are the unit vectors along the direction of $(\mathbf{S}_{n1} \times \mathbf{S}_{n2}) \times \mathbf{v}_1$ and $(\mathbf{S}_{n3} \times \mathbf{S}_{n4}) \times \mathbf{v}_2$ respectively.

Introducing expressions (49) into equation (48) becomes as follows:

$$\begin{aligned} \sum_{i=1}^4 f_i \mathbf{S}_i - \mathbf{F}_e &= 0 \\ {}^A \mathbf{R}_B \mathbf{BC}_1 \times \left(\sum_{i=1}^2 \mathbf{S}_i f_i \right) + {}^A \mathbf{R}_B \mathbf{BC}_2 \times \left(\sum_{i=3}^4 \mathbf{S}_i f_i \right) + \sum_{i=1}^2 m_{C1} \mathbf{S}_{C1} - \mathbf{M}_e &= 0 \end{aligned} \tag{50}$$

Writing equation (50) in a matrix form yields:

$$\begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 & \mathbf{S}_4 & \mathbf{0} & \mathbf{0} \\ {}^A\mathbf{R}_{B1} \mathbf{l}_1 \times \mathbf{S}_1 & {}^A\mathbf{R}_{B1} \mathbf{l}_1 \times \mathbf{S}_2 & {}^A\mathbf{R}_{B2} \mathbf{l}_2 \times \mathbf{S}_3 & {}^A\mathbf{R}_{B2} \mathbf{l}_2 \times \mathbf{S}_4 & \mathbf{S}_{C1} & \mathbf{S}_{C2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ m_{C1} \\ m_{C2} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_e \\ \mathbf{M}_e \end{bmatrix} \quad (51)$$

where $\mathbf{l}_1 = \mathbf{BC}_1, \mathbf{l}_2 = \mathbf{BC}_2$. The forces and moments at the robot joints due to external load are given by:

$$\mathbf{J}_S^{-1} \begin{bmatrix} \mathbf{F}_e \\ \mathbf{M}_e \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ m_{C1} \\ m_{C2} \end{bmatrix} \quad (52)$$

Investigating the transformation matrix \mathbf{J}_S given in equation (52), one can analyze the singularity of the parallel part. The columns of \mathbf{J}_S are lines lying on the *Klein quadric* M_2^4 , as they satisfy Klein's equation (Hunt, 1978):

$$l_1 l_4 + l_2 l_5 + l_3 l_6 = 0 \quad (53)$$

where each column of equation (53) is represented by its Plücker line coordinates given by $(l_1, l_2, l_3, l_4, l_5, l_6)^T$.

Moreover, considering the forces and moments exerted on the actuator, one can obtain (see Fig.16).

$$\tau_i \mathbf{k}_z + \sigma_i \mathbf{k}_{pi} = f_i \mathbf{S}_i \quad (54.1)$$

$$f_i \cos \theta_i = \tau_i \quad (54.2)$$

$$f_i \sin \theta_i = \sigma_i \quad (54.3)$$

$$\cos \theta_i = \mathbf{k}_z \cdot \mathbf{S}_i \quad (54.4)$$

where \mathbf{k}_{pi} is the unit vector along the direction $(\mathbf{k}_z \times \mathbf{u}_i)$. Equation (54.2) can be written for $i=1,2,3,4$, and the results grouped in a matrix form, τ_i can be written as:

$$\mathbf{J}_P \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (55)$$

where

$$\mathbf{J}_P = \begin{bmatrix} \cos\theta_1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & \cos\theta_3 & 0 \\ 0 & 0 & 0 & \cos\theta_4 \end{bmatrix} \quad (56)$$

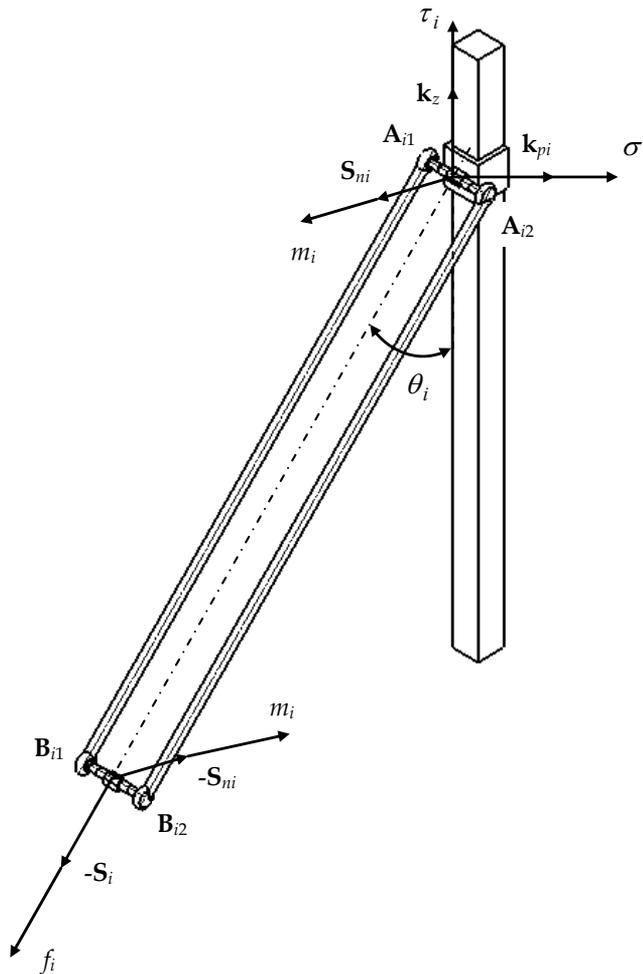


Fig.16. Parallelogram static analysis

\mathbf{J}_S (equation (52)) and \mathbf{J}_P (equation (56)) provide the information about the singularities of the H4 manipulator. The analysis of \mathbf{J}_S provides the singularity of the parallel part of H4 robot known as the “parallel singularity” and \mathbf{J}_P provides the singularity due to the parallelogram structure, known as the “serial singularity”. In (Liu et al., 2003), the singularity caused by \mathbf{J}_P is also called as “actuator singularity”.

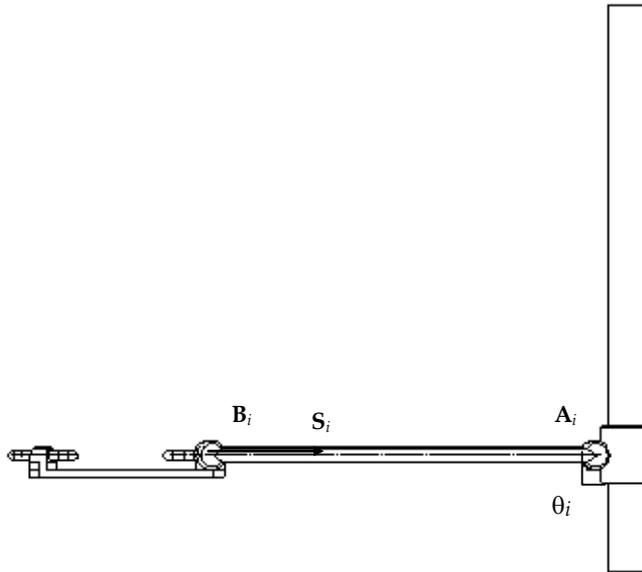


Fig.17. Serial singularity of H4 manipulator

Observing \mathbf{J}_p one can see that singularity occurs whenever $\theta_i = \pi/2 + \pi k$ ($k=1, \dots, n$, $i=1, 2, 3, 4$). In this case one or more columns of \mathbf{J}_p are zero and the nacelle can't move in the S_i direction, namely, the manipulator loses a DOF. Fig.17 shows the serial singularity of H4 manipulator.

4.3 The behavior of the H4 manipulator in parallel singular configurations

The parallel singular configurations of the H4 are identified by investigating matrix \mathbf{J}_s . The first four columns of \mathbf{J}_s define the four internal forces of parallelogram which are along the direction $\mathbf{B}_i\mathbf{A}_i$ (see Fig.53 and 55). Utilizing Grassmann geometry, we can obtain all singularities of \mathbf{J}_s .

Based on H4's kinematic architecture, the first four columns of \mathbf{J}_s can form only the following linear varieties:

- Type I: a line bundle where all lines intersect a common point, but they are not coplanar. This is the condition 3c described in section 4.1.2.
- Type II: all the lines belong to the union of two planar pencils of non coplanar lines that have a line in common. This is the condition 3b described in section 4.1.2.

Type I can be considered as the special condition of Type II. When the first four columns of \mathbf{J}_s are linear dependent, the actuators can't control the linear velocity of the nacelle, generally known as "architecture singularity" (Joshi & Tsai, 2002).

The last two columns of \mathbf{J}_s define the two moments of constraint which are along the direction \mathbf{S}_{C_i} . We make here the assumption that the two revolute joints placed on the nacelle have the same direction represented by vector \mathbf{v} . This choice may be not the only one, but it worth noting that placing two revolute joints parallel to each other on the nacelle

is a good choice for practical matters. The existence of a rotational motion about \mathbf{v} can be written as follows (Pierrot & Company, 1999):

$$\mathbf{S}_{C1} \times \mathbf{S}_{C2} \neq \mathbf{0} \quad (57)$$

When $\mathbf{S}_{C1} \times \mathbf{S}_{C2} = \mathbf{0}$, the vectors \mathbf{S}_{C1} and \mathbf{S}_{C2} are linearly dependent. The constraint moments exerted on the nacelle are along the same direction, and the robot would gain additional DOF(s). This singularity arises from deficiency of constraints, so we call this singularity as “*constraint singularity*”.

In order to investigate the instantaneous motion the robot tends to perform while in singular configurations, the LCAA which was presented at (Pottmann et al., 1999) and further used for robotic analysis in (Wolf & Shoham, 2003) is adopted. The LCAA is based on determining the linear complex $\mathbf{C} = (\mathbf{c}, \bar{\mathbf{c}})$, which is the closest one to the given set of lines $\mathbf{L}_i = (\mathbf{l}_i, \bar{\mathbf{l}}_i)$. The moment of \mathbf{L}_i with respect to a linear complex \mathbf{C} is given by:

$$m(\mathbf{L}, \mathbf{C}) = \frac{|\bar{\mathbf{c}} \cdot \mathbf{l}_i + \mathbf{c} \cdot \bar{\mathbf{l}}_i|}{\|\mathbf{c}\|} \quad (58)$$

Hence, given a set of lines \mathbf{L}_i , the closest linear complex among all linear complexes χ can be given by the minimization of:

$$\sum_{i=1}^k m(\mathbf{L}_i, \chi) \quad (59)$$

where χ is given by $\chi = (\mathbf{x}, \bar{\mathbf{x}}) \in \mathfrak{R}^6$.

Equation (59) is equivalent to minimizing the positive semi-definite quadratic form given by (Pottmann et al., 1999):

$$F(\mathbf{X}) = \sum_{i=1}^k (\bar{\mathbf{x}} \cdot \mathbf{l}_i + \mathbf{x} \cdot \bar{\mathbf{l}}_i)^2 = \chi^T \mathbf{M} \chi \quad (60)$$

under the normalization condition $1 = \|\mathbf{x}\|^2 = \chi^T \mathbf{D} \chi$, where $\mathbf{D} = \text{diag}(1, 1, 1, 0, 0, 0)$, χ presents the set of all linear complexes given by $\chi = (\mathbf{x}, \bar{\mathbf{x}}) \in \mathfrak{R}^6$, and \mathbf{M} is the Gramian matrix of \mathbf{J}_S .

Equation (60) is a general eigenvalue problem and the solution, \mathbf{C} , is the eigenvector χ_i corresponding to the smallest eigenvalue. Moreover, given the closest linear complex found by the algorithm, the axis, \mathbf{A} , of the linear complex and its pitch can also be revealed. In fact, the minimization procedure shown in above minimizes the work of the wrench applied on the moving platform and the twists. When the smallest eigenvalue is zero, there is no work generated by the set of wrenches when the platform moves in the twist direction given by χ . So we can use LCAA to investigate the behavior of H4 manipulator in singular configurations.

4.4 Simulation and results

In this section, the parallel singularities of an example H4 manipulator are presented and the LCAA method is used to investigate the instantaneous motion the manipulator tends to

perform while in these singular configurations. For simulation, the design parameters we have selected is described in Fig.11, where the parameters have been chosen as described in section 3.2.2.

Architecture singularities

Architecture singular configurations of the H4 manipulator are identified using line geometry (Merlet, 2006). Note the first four columns of \mathbf{J}_s as:

$$\begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 & \mathbf{S}_4 \\ {}^B\mathbf{R}_D\mathbf{I}_1 \times \mathbf{S}_1 & {}^B\mathbf{R}_D\mathbf{I}_1 \times \mathbf{S}_2 & {}^B\mathbf{R}_D\mathbf{I}_2 \times \mathbf{S}_3 & {}^B\mathbf{R}_D\mathbf{I}_2 \times \mathbf{S}_4 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ \bar{a}_1 & \bar{a}_2 & \bar{a}_3 & \bar{a}_4 \end{bmatrix} = [\mathbf{A}_1 \ \mathbf{A}_2 \ \mathbf{A}_3 \ \mathbf{A}_4] \quad (61)$$

where $\|a_i\|=1$ and $a_i \cdot \bar{a}_i = 0$, so the six coordinates with (a_i, \bar{a}_i) are called the normalized Plücker coordinates of \mathbf{A}_i ($i=1,2,3,4$). Deriving from the geometric conditions (Type I and Type II) in section 4.3, we can find that when the coordinates of point D satisfy the following condition:

$$\begin{aligned} x &= -9.5812 \\ y &= -9.4130 \\ \theta &= -1.5080 \end{aligned} \quad (62)$$

singularity occurs. The closest linear complex's axis found by the algorithm is:

$$\mathbf{A} = [0 \ 0 \ 1 \ -5.2937 \ -0.3331 \ 0]$$

The standard deviation of the lines form \mathbf{C} is given by: $\sigma = \sqrt{\frac{\lambda_i}{k-5}} = 0$, and the linear complex's pitch is given by: $p = \frac{\mathbf{c} \cdot \bar{\mathbf{c}}}{\mathbf{c}^2} = -9.4047$. Fig.18 demonstrated the result of the LCAA.

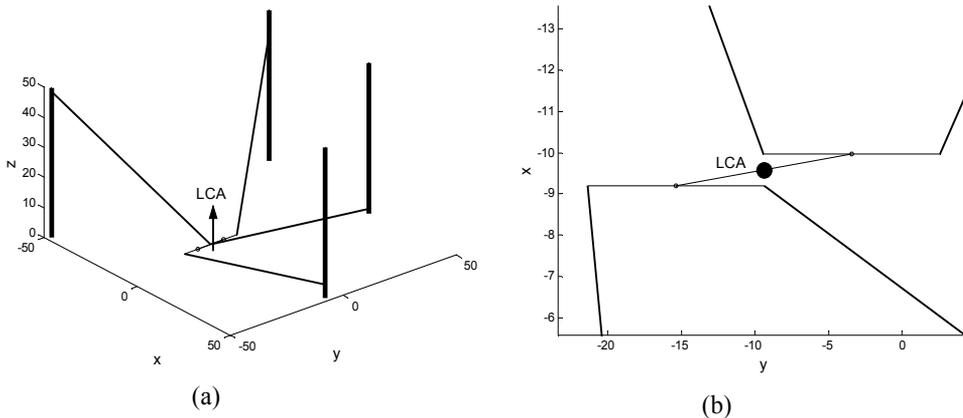


Fig.18. H4 robot with linear complexes (a): 3D view (b): top view

Corresponding to the non-zero pitch value of the linear complex, at this singularity the robot performs a screw motion of rotation and translation around the linear complex axis \mathbf{A} .

Constraint singularities

From the discussion in section 4.3, we know that when $S_{C1} \times S_{C2} = 0$, the robot is in a constraint singular configuration. If we select the structure parameters as Fig.11, when:

$$\begin{aligned} x &= -3.6683 \\ y &= 3.3709 \\ \theta &= -2.3905 \end{aligned} \tag{63}$$

the robot reaches singular configuration.

Constraint singularity occurs in limited-DOF manipulator when the screw system, formed by the constraint wrenches in all chains, loses rank. In this section, the geometric explanation of constraint singularity is given, then LCAA is applied to investigate the behavior of H4 manipulator in constraint singular configuration. From the viewpoint in (Zlatanov et al., 2002), the system of the constraining wrenches of H4, \mathcal{W} , contains as a subspace the 2-system of all pure moments. Then its reciprocal system, the freedoms system \mathcal{T} , must contain 3 translations and 1 rotation. In a H4 manipulator, the two “meta-chain” constrain system \mathcal{W}_p contains two pure moments, one’s direction is along S_{C1} , and the other’s direction is along S_{C2} . When the two moments of the two “meta-chain” are linearly independent, two pure moments will be in \mathcal{W} .

However, in the constraint singular configuration, S_{C1} and S_{C2} are parallel. The constraining moments of two “meta-chain” are identical and the system \mathcal{W}_p are the same one-system. Then, \mathcal{W} consists of only one screw. Hence, the twist system of the robot is a five-system and the mechanism gains new additional freedom. Applying the method of the closest linear complex on the H4 robot in its constraint singular configuration, we can find the closest linear complex’s axis is: $A = [0.9137 \ 0.3524 \ -0.2024 \ 0.3208 \ -0.2064 \ 1.0887]$

and the linear complex’s pitch is: $p = \frac{c \cdot \bar{c}}{c^2} = 1.0469 \cdot S_{C1} = -S_{C2} = [0.3599 \ -0.9330 \ 0]^T$ means

that this configuration is singular and the robot is able to execute a screw motion of rotation and translation around the linear complex axis. The direction of the linear complex axis is perpendicular to S_{C1} (and S_{C2}) because $a \cdot S_{C1} = 0$ (and $a \cdot S_{C2} = 0$). This corroborates the results deduced from the viewpoint of constraint singularities. Fig.19 demonstrates the result of the LCAA in constraint singular configurations.

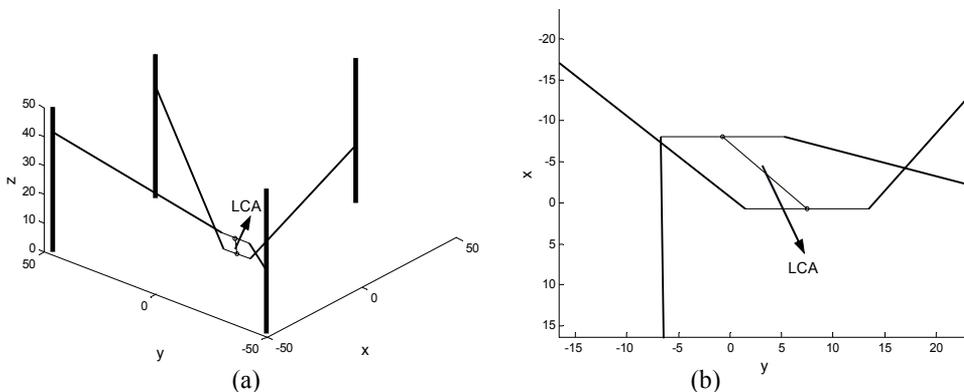


Fig.19. H4 robot with linear complexes (a): 3D view (b): top view

Discussion: Simulation Results of the H4 robot

From above discussion, only one architecture singular configuration can be found. In this configuration, the intersection point of $\mathbf{A}_1, \mathbf{A}_2$ and the intersection point of $\mathbf{A}_3, \mathbf{A}_4$ (see equation (61)) are superposed, which satisfies the geometric condition of Type I (in section 4.3). However, for constraint singularities, as long as $\mathbf{S}_{C1} \times \mathbf{S}_{C2} = \mathbf{0}$ is satisfied, the manipulator would reach singular configurations. Four parameters x, y, z, θ are used to describe the configuration of a H4 manipulator. z is independent of the singularities. The singular equation $\mathbf{S}_{C1} \times \mathbf{S}_{C2} = \mathbf{0}$ provides only one constraint equation, so the three parameters x, y, θ constitute a three dimension curved surface. All the points in the curved surface satisfy the equation $\mathbf{S}_{C1} \times \mathbf{S}_{C2} = \mathbf{0}$. We call this surface as constraint singular surface. The equation of the constraint singular surface can be written as:

$$(\mathbf{S}_{C1} \times \mathbf{S}_{C2}) \cdot \mathbf{v} = 0 \quad (64)$$

Fig.20 depicts the constraint singular surface.

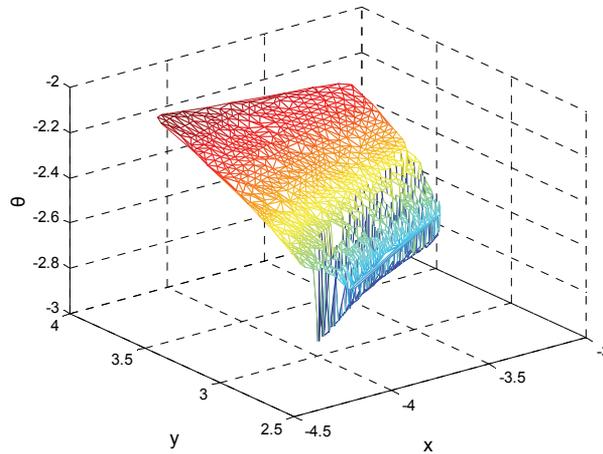


Fig.20. The constraint singular surface

As we can see from the reference (Wolf & Shoham, 2003):

$$F(\mathbf{X}) = \sum_{i=1}^k (\bar{\mathbf{x}} \cdot \mathbf{l}_i + \mathbf{x} \cdot \bar{\mathbf{l}}_i)^2 = \boldsymbol{\chi}^T \mathbf{M} \boldsymbol{\chi} = \lambda$$

λ has the meaning of the sum of the square of the virtual work generated by the structure to the nacelle, when the latter is moving instantaneously in the twist direction given by $\boldsymbol{\chi}$. Hence, if $\lambda = 0$, the chains would lose the constraints on the nacelle and the nacelle could move instantaneously in the twist direction given by $\boldsymbol{\chi}$.

The standard deviation of the lines \mathbf{L}_i from C, may be written as:

$$\sigma = \sqrt{\frac{\lambda}{k-5}} = \sqrt{\frac{\sum_{i=1}^k m(\mathbf{L}_i, \mathbf{c})^2}{k-5}}$$

which is the square of error from \mathbf{L}_i to \mathbf{C} . In case of a sufficiently small deviation σ , the given lines \mathbf{L}_i can be well approximated by lines of the linear complex \mathbf{C} , which means that the current configuration is close to the singular configuration. When $\sigma = 0$, the lines \mathbf{L}_i lie on the linear complex \mathbf{C} , which means that the manipulator is in a singular configuration and the nacelle can move in the twist direction of the linear complex axis. In order to investigate the robot's behavior in the vicinity of the singular configuration, a simulation is performed in the neighborhood of the architecture singular configuration (Fig.21-Fig.22).

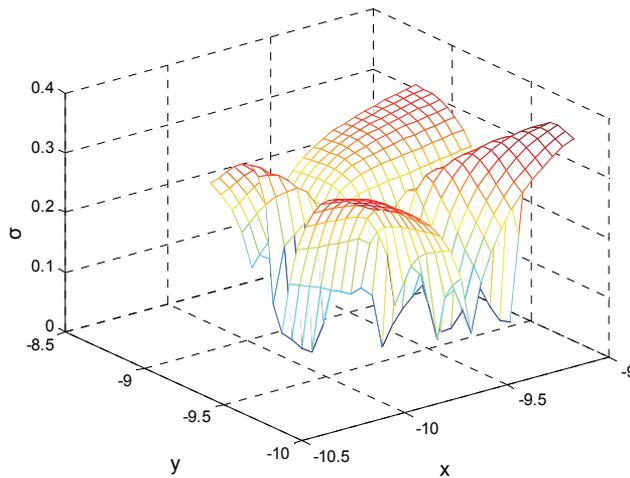


Fig.21. Minimum σ in the vicinity of the architecture singular configuration

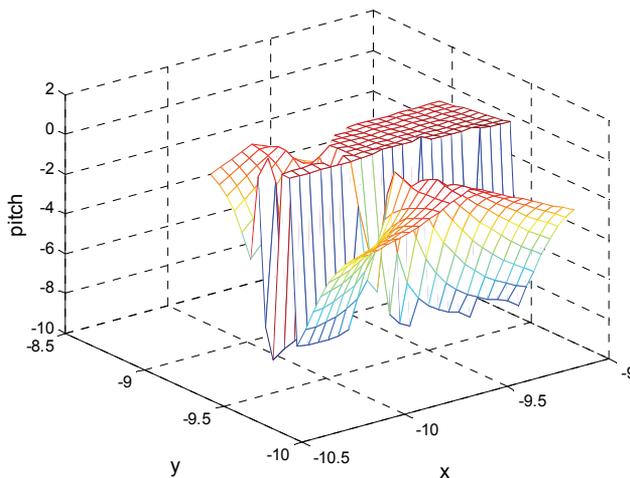


Fig.22 Pitch of linear complex with minimum σ in the vicinity of the architecture singular configuration

All the simulations are in the adjacent range of the singular configuration: $x \in [-10.1 \quad -9.1]$, $y \in [-9.9 \quad -8.9]$, $z = 0$, $\theta = -1.5080$.

It can be seen that the manipulator is no longer in a singular position. However, when manufacturing tolerances are greater than the value σ , there is still possibility of an uncontrolled motion of the nacelle according to the interpretation of σ .

5. Conclusions

This chapter mainly discusses the architectures and some kinematic problems about the 4-DOF parallel manipulator H4 which is suitable for pick and place operations. The geometric methods used to analyze the singularity poses are introduced in detail. As this type of parallel manipulator generally performs tasks implying significant accuracy, velocities and accelerations, there are still numerous topics need further study.

It is the complexity of the parallel manipulator's implantation, of their control and of their design that hinders their widely use. Control efficiency is related to kinematics calculation. Although the analytical formulation of the direct kinematics is deduced in section 3.2.1, solving the 16th degree univariate polynomial equation is not trivial and efficient control still requires very fast algorithms which give all of the solutions.

The motions of H4 are mainly restricted by mechanical limits on passive joints, interference links and limitations due to the actuators. Because of the coupling between translations and rotations, it is generally complex to determine the workspace of H4. There are various methods can be used to calculate the workspace of H4, such as the discretization approach, inequality constraints approach (Jo, 1989) and algebraic approach (Husty, 1996). Merlet (Merlet, 2006) has pointed out that the geometrical approaches are very efficient for the determination of various types of workspace and algebraic geometry should also play an important role in some of the algorithms which are used to calculate the workspace of parallel manipulators.

Singularity analysis of parallel manipulators is crucial for practical applications. Besides the geometric method suggested in section 4, some new ideas are provided to analyze singularities of parallel manipulators by using different mathematic tools. Differential geometric is a powerful tool for general analysis of parallel manipulators. From the geometric properties of the parallel manipulator, Liu et al. presented a framework for analyzing the singularities of a parallel manipulator (Liu et al., 2003). Topological structure of these singularities and their relations with the kinematic parameters of the system were investigated systematically using Morse function theory. An intrinsic definition of parameterization singularities was given in this chapter. Due to the profound explanation of the parallel manipulators' characteristics, although the differential geometric tool is abstract, its use for general analysis and extension to geometric control of parallel manipulator is still worthy of study. In some situations, it is only necessary to seek singular configurations in the workspace of parallel manipulators. Su et al. (Su et al., 2003) proposed a new singularity analysis method for Stewart platform using genetic algorithm (GA). The square of determinant of the Jacobian matrix is selected as the objective function, and the minimal of this objective function is searched in the workspace of Stewart platform by the GA. The singularity of Stewart platform depends on this minimal objective function: if this value is

zero, the singularity of Steward platform will take place, otherwise, the Sterward platform is singularity-free. This method can be used to determine the dimensioning of a H4 so that it is singularity-free, at least in a given workspace. Of course, the effectiveness and speediness of the GA are the chief problems which should be considered.

The study of the novel parallel manipulator H4 and its various reformative structures has become increasingly important during recent years for its excellent performance of velocity and acceleration. But numerous problems still remain open, especially the positioning accuracy of the end-effector. From the paper written by Renaud et al. (Renaud et al., 2003), the positioning accuracy of a H4 can only reach lower than 0.5mm, which cannot meet the requirements (generally lower than 0.05mm) of the semiconductor end-package equipments. We hope that this chapter will arise some attention to extend this type of parallel manipulators to semiconductor industry.

6. Acknowledgements

The work is supported by the Natural Science Fund of China (NSFC) (Project No. 50625516) and the National Fundamental Research Program (973) (Project No. 2003CB716207 and 2007CB714000).

7. References

- Agrawal, S.K.; Desmier, G. & Li, S. (1995). Fabrication and analysis of a novel 3 DOF parallel wrist mechanism. *ASME Journal of Mechanical Design*, Vol. 117, June, pp. 343-345
- Angeles, J.; Morozov, A. & Navarro, O. (2000). A novel manipulator architecture for the production of scara motions, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2370-2375, San Francisco, USA, April 2000
- Angeles, J. (2005). The degree of freedom of parallel robots: a group-theoretic approach, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1017-1024, Barcelona, Spain, April 2005
- Basu, D. & Ghosal, A. (1997). Singularity analysis of platform-type multi-loop spatial mechanisms. *Mechanism and Machine Theory*, Vol. 32, No. 3, pp. 375-389
- Benov, I. (2001). Delta parallel robot - the story of success. May, 6, <http://www.parallemic.org/Reviews/Review002.html>
- Bonev, L. & Zlatanov, D. (2001). The mystery of the singular SUN translational parallel robot. June, 12, 2001, <http://www.parallemic.org/Reviews/Review004.html>
- Bonev, I.A. et al. (2001). A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors. *IEEE Transaction on Robotics and Automation*, Vol. 17, No. 2, pp. 148-156
- Bottema, O. & Roth, B. (1990). *Theoretical kinematics*. New York: Dover Publications
- Brogardh, T. PKM research - important issues, as seen from a product development perspective at ABB robotics, *Workshop on Fundamental Issues and Future Research Directions for parallel Mechanisms and Manipulators*, Quebec, Canada, 2002

- Bruyninckx, H. (1997). The 321-hexa: a fully parallel manipulator with closed-form position and velocity kinematics, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 2657-2662, Albuquerque, April, 1997
- Carretero, J.A.; Podhorodeski, R.P.; Nahon, M.A. & Gosselin, C.M. (2000). Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. *ASME Journal of Mechanical Design*, Vol. 122, pp. 17-24
- Chablot D, D.; Wenger, P. Design of a three-axis isotropic parallel manipulator for machining applications: the orthoglide, *Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Quebec, Canada, Oct., 2002
- Cheung, J.W.F.; Hung, Y.S. & Widdowson, G.P. (2002). Design and analysis of a novel 4-dof parallel manipulator for semiconductor applications, *Proceedings of the 8th Mechatronics Forum International Conference*, pp. 1358-1366, Twente, Netherlands: Drebbe Institute for Mechatronics
- Chiu, Y.J. & Perng, M-H. (2001). Forward kinematics of a general fully parallel manipulator with auxiliary sensors. *International Journal of Robotics Research*, Vol. 20, No. 5, pp. 401-414
- Choi, H.-B.; Konno, A. & Uchiyama, M. (2003). Closed-form solutions for the forward kinematics of a 4-DOFs parallel robot H4, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3312-3317, Las Vegas, USA, October, 2003
- Clavel, R. (1988). DELTA, a fast robot with parallel geometry, *Proceedings of the 18th International Symposium on Industrial Robot*, pp. 91-100, April, 1988, Lausanne
- Collins, C.L. & Long, G.L. (1995). The singularity analysis of an in-parallel hand controller for force-reflected teleoperation. *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, pp. 661-669
- Company, O. & Pierrot, F. (1999). A new 3T-1R parallel robot, *Proceedings of the 9th International Conference on Advanced Robotics*, pp. 557-562, October, 1999, Tokyo, Japan
- Company, O.; Marquet, F. & Pierrot F. (2003). A new high-speed 4-DOF parallel robot synthesis and modeling issues. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 3, pp. 411-420
- Company, O.; Pierrot, F. & Fouroux, J.C. (2005). A method for modeling analytical stiffness of a lower mobility parallel manipulator, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 18-22, Barcelona, Spain, April, 2005, Milan, September, 1995
- Di Gregorio, R. & Parenti-Castelli, V. (1999). Mobility analysis of the 3-UPU parallel mechanism assembled for a pure translational motion, *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 520-525, Atlanta, GA, September, 1999
- Fattah, A. & Kasaei, G. (2000). Kinematics and dynamics of a parallel manipulator with a new architecture. *Robotica*, Vol. 18, pp. 535-543

- Faugère, J.C. & Lazard, D. (1995). The combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, Vol. 30, No. 6, pp. 765-776
- Feng-Cheng, Y. & Haug, E.J. (1994). Numerical analysis of the kinematic working capability of mechanism. *ASME Journal of Mechanical Design*, Vol. 116, pp. 111-117
- Funabashi, H. & Takeda, Y. (1995). Determination of singular points and their vicinity in parallel manipulators based on the transmission index, *Proceedings of the Ninth World Congress on the Theory of Machine and Mechanism*, pp. 1977-1981,
- Gosselin, C.M. & Angeles, J. (1989). The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *ASME Journal of Mechanisms, Transmission and Automation in Design*, Vol.111, No.2, pp. 202-207
- Gosselin, C. & Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains. *IEEE Transaction on Robotics and Automation*, Vol. 6, No. 3, pp. 281-290
- Gosselin, C. (1990). Dexterity indices for planar and spatial robotic manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 650-655, Cincinnati, May, 1990
- Gosselin, C. (1996). Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 118, No. 1, pp. 22-28
- Gosselin, C.M. & St-Pierre, É. (1997). Development and experimentation of a fast 3-DOF camera-orienting device. *The International Journal of Robotics Research*, Vol.16, No.5, pp. 619-629
- Gough, V.E. (1956). Contribution to discussion of papers on research in automobile stability, control and tyre performance, *Proceedings of the Automotive Division of the Institution of Mechanical Engineers, Part D*, pp. 392-394
- Gregorio, R. D. (2001). A new parallel wrist using only revolute pairs: the 3-RUU wrist. *Robotica*, Vol. 19, pp. 305-309
- Hao, F. McCarthy, J.M. (1998). Conditions for line-based singularities in spatial platform manipulators. *Journal of Robotic Systems*, Vol. 15, No. 1, pp. 43-55
- Hesselbach, J. & Kerle, H. (1995). Structurally adapted kinematic algorithms for parallel robots up to six degrees of freedom (dof), *Proceedings of the 9th World Congress on the Theory of Machines and Mechanisms*, pp. 1935-1939, Milan, Italy, August, 1995
- Hunt, K.H. (1978). Kinematic geometry of mechanism. Department of Mechanical Engineering, Monash University, Clayton, Victoria, Australia
- Husty, M.L. (1996). On the workspace of planar three-legged platforms. *Proceedings of the World Automation Congress*, Vol. 3, pp. 339-344, Montpellier, May, 1996
- Jo, D.Y. & Haug, E.J. (1989). Workspace analysis of closed loop mechanism with unilateral constraints. *Proceedings of the ASME Design Automation Conference*, pp. 53-60, Montreal, September, 1989
- Joshi, S.A. & Tsai L.-W. (2002) Jacobian analysis of limited-dof parallel manipulators. *ASME Journal of Mechanical Design*, Vol. 124, No. 2, pp. 254-258

- Kim, D.; Chung, W. & Youm, Y. (2000). Analytic jacobian of in-parallel manipulators. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2376-2381, San Francisco, April, 2000
- Kim, S.-G. & Ryu, J. (2003). New dimensionally homogeneous jacobian matrix formulation by three end-effector points for optimal design of parallel manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 4, pp. 731-736
- Krut, S.; Company, O. & Benoit, M. et al. (2003). I4: a new parallel mechanism for scara motions, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1875-1880, Taipei, Taiwan, September, 2003
- Krut, S.; Nabat, V.; Company, O. & Pierrot, F. (2004). A high-speed parallel robot for scara motions, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4109-4115, New Orleans, USA, April, 2004
- Li, Q. & Huang, Z. (2003). Mobility analysis of lower-mobility parallel manipulators based on screw theory, *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 1179-1184, Taipei, Taiwan, September, 2003
- Liu, G.F.; Lou, Y.J. & Li, Z.X. (2003). Singularities of parallel manipulators: a geometric treatment. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 4, pp. 579-594
- Merlet, J.P. (1992). On the infinitesimal motion of a parallel manipulator in singular configurations, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 320-325, Nice, France, May, 1992
- Merlet, J.-P. (2000). Kinematics' not dead, *Proceedings of IEEE International Conference on Robotics & Automation*, pp. 1-6, San Francisco, CA, April, 2000
- Merlet, J.-P. & Daney, D. (2005). Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 19-22, Barcelona, Spain, April, 2005
- Merlet, J.-P. (2006). *Parallel Robots (Second Edition)*, Springer, Dordrecht, The Netherlands
- Nabat, V.; Company, O. & Krut, M. et al. (2005). Part 4: very high speed parallel robot for pick-and-place, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 553-558, Edmonton, Canada, August, 2005
- Nabat, V.; Company, O. & Pierrot, F. (2006). Dynamic modeling and identification of Part 4, a very high speed parallel manipulator, *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 496-501, Beijing, China, October, 2006
- Parenti-Castelli, P. & Di Gregorio, P. (20010). Real-time actual pose determination of the general fully parallel spherical wrist, using only one extra sensor. *Journal of Robotic Systems*, Vol. 18, No. 12, pp. 723-729
- Pennock, G.R. & Kassner, D.J. (1990). Kinematic analysis of a planar eight-bar linkage: application to a platform-type robot, *ASME Proceedings of the 21th Biennial Mechanism Conference*, pp. 37-43, Chicago, September, 1990
- Pierrot, F. & Company, O. (1999). H4: a new family of 4-dof parallel robots, *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 508-513, Atlanta, USA, September, 1999

- Pottmann, H.; Peternell, M. & Ravani, B. (1999). An introduction to line geometry with applications. *Computer Aided Design*, Vol. 31, pp. 3-16
- Reboulet, C. & Robert, A. (1985). Hybrid control of a manipulator with an active compliant wrist, *The 3rd International Symposium of Robotics Research*, pp. 76-80, Gouvieux, France, October, 1985
- Renaud, P.; Andreff, N. & Marquet, F. et al.. (2003). Vision-based kinematic calibration of a H4 parallel mechanism. *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 1191-1196, Taipei, Taiwan, September, 2003
- Robotics Department of LIRMM. H4 with rotary drives: a member of H4 family, dedicated to high speed applications. http://www.lirmm.fr/w3rob/SiteWeb/detail_resultat.php?num_resultat=7&num_topic=1&num_projet=1&num_activite=3
- Rolland, L. The manta and the kanuk: novel 4-dof parallel mechanisms for industrial handling. *ASME Dynamic Systems and Control Division, IMECE'99 Conference*, pp. 831-844, Nashville, USA, November 1999
- Romdhane, L.; Affi, Z. & Fayet, M. (2002). Design and singularity analysis of a 3-translational-DOF in-parallel manipulator. *ASME Journal of Mechanical Design*, Vol. 124, pp. 419-426
- Ryu, J. & Cha, J. (2003). Volumetric error analysis and architecture optimization for accuracy of HexaSlide type parallel manipulators. *Mechanism and Machine Theory*, Vol. 38, pp. 227-240
- Steward, D. (1965). A platform with 6 degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, Vol. 180, Part 1, No. 15, pp. 371-386, British
- St-Onge, B.M. & Gosselin, C.M. (2000). Singularity analysis and representation of the general Gough-Stewart platform. *International Journal of Robot Research*, Vol.19, No. 3, pp. 271-288
- Su, Y.X.; Duan, B.Y. & Peng, B. et al.. (2003). Singularity analysis of fine-tuning stewart platform for large radio telescope using genetic algorithm. *Mechatronics*, Vol. 13, pp. 413-425
- Tsai, L.-W. (1996). Kinematics of a three-DOF platform with three extensible limbs. *Recent Advances in Robot Kinematics*, Kluwer Academic Publishers, pp. 401-410
- Tsai, L.-W. & Joshi, S.A. (2000). Kinematics and optimization of a spatial 3-UPU parallel manipulators. *ASME Journal of Mechanical Design*, Vol. 122, No.4, pp. 439-446
- Wolf, A. & Shoham, M. (2003). Investigation of parallel manipulators using linear complex approximation. *ASME Journal of Mechanical Design*, Vol. 125, No. 3, pp. 564-572
- Wu, J.; Yin, Z. & Xiong, Y. (2006). Singularity analysis of a novel 4-dof parallel manipulator H4. *International Journal of Advanced Manufacturing and Technology*, Vol. 29, pp. 794-802
- Zlatanov, D.; Fenton, R.G. & Benhabib, B. (1994). Singularity analysis of mechanism and robots via a motion-space model of the instantaneous kinematics. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 980-991, San Diego, CA, May, 1994

- Zlatanov, D.; Benov, I.A. & Gosselin, C.M. (2002). Constraint singularities of parallel mechanisms, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 496-502, Washington DC, May, 2002
- Zobel, P.B., Di Stefano, P. & Raparelli, T. (1996). The design of a 3 dof parallel robot with pneumatic drives, *Proceedings of the 27th International Symposium on Industrial Robot*, pp. 707-710, October, 1996, Milan

Human Hand as a Parallel Manipulator

Vladimir M. Zatsiorsky and Mark L. Latash
Department of Kinesiology, The Pennsylvania State University

1. Introduction

When a human hand grasps an object the hand can be viewed as a parallel manipulator. In general, the mathematical analyses of the human hands and multi-fingered robot hands (Murray et al. 1994) are similar. In particular, concepts developed in robotics such as contact models, e.g. soft-finger model, grasp matrix, form and force closure grasps, internal forces, etc. can be applied to analyze the performance of the human hands. Multi-finger prehension is an example of a mechanically redundant task: the same resultant forces on the object can be exerted by different digit forces. People however do not use all the mechanically available options; when different people perform a certain manipulation task they use a limited subset of solutions.

Studies on human prehension deal with four main issues:

1. Description of the behavior: What are the regularities in force patterns applied at the fingertip-object interfaces when people manipulate objects?
2. Are the observed patterns dictated by the task and hand mechanics? The mechanical properties of the hand and fingers are complex, and it is not always evident whether the findings are direct consequences of the mechanical properties of the hand or they are produced by a neural control process.
3. If the observed facts/phenomena are not of mechanical origin are they mechanically necessitated? In other words, can the task be performed successfully in another way?
4. If reproducible phenomena are not mechanical and not mechanically-necessitated, the question arises why the central nervous system (CNS) facilitates these particular phenomena. This is a central question of the problem of motor redundancy in general: Why does the CNS prefer a certain solution over other existing solutions?

The present chapter briefly reviews some specific features of the human hand and the involved control mechanisms. To date, the experimental data are mainly obtained for the so-called *prismatic grasps* in which the thumb opposes the fingers and the contact surfaces are parallel (Figure 1). The contact forces and moments are typically recorded with 6-component force and moment sensors.

Experimental 'inverted-T' handle/beam apparatus commonly used to study the prismatic precision grip. Five six-component force sensors (black rectangles) are used to register individual digit forces. During testing, the suspended load could vary among the trials. The load displacement along the horizontal bar created torques from 0 N·m to 1.5 N·m in both directions. The torques are in the plane of the grasp. While forces in all three directions were recorded the forces in Z direction were very small and, if not mentioned otherwise, were

neglected. When the handle is oriented vertically the force components in the X and Y directions are the *normal* and *shear*, (or *tangential*) forces, respectively. The figure is not drawn to scale.

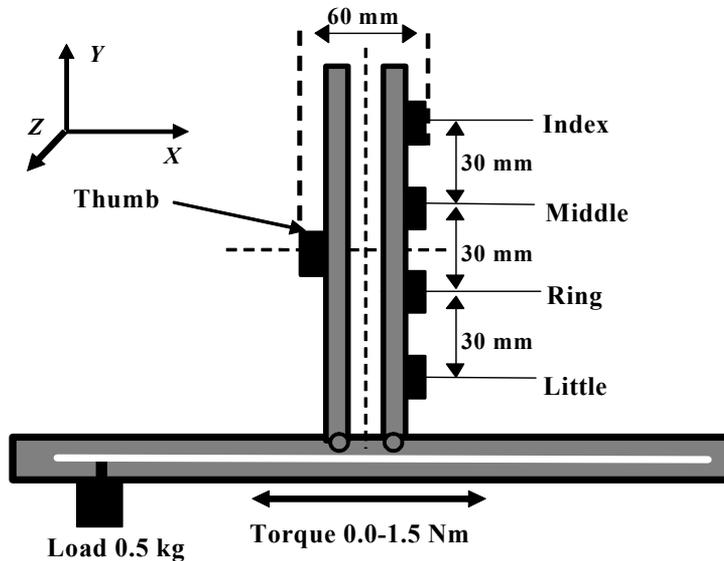


Figure 1. Experimental 'inverted-T' handle/beam apparatus

2. Digit contacts

During an object manipulation the finger tips deform and the contact areas are not constant (Nakazawa et al. 2000; Paré et al. 2002; Serina et al. 1997; Srinivasan, 1989; Srinivasan et al. 1992; Pataky et al. 2005). The fingers can also roll on the sensor surface. As a result, the point of digit force application is not constant: it can displace by up to 5-6 mm for the fingers and up to 11-12 mm for the thumb (Figure 2). Therefore the digit tip contacts should be as a rule treated as the soft-finger contacts (Mason & Salisbury, 1985).

When a soft-finger model of the digit-object contact is employed, the contact is characterized by six variables: three orthogonal force components (the normal force component is unidirectional and the two tangential force components are bi-directional), free moment in the plane of contact, and two coordinates of the point of force application on the sensor. To obtain these data the six-component force and moment sensors are necessary. The coordinates of the point of force application are not recorded directly; they are computed from the values of the normal force and the moment around an axis in the contact plane. Such a computation assumes that the fingers do not stick to the sensor surfaces, in other words the fingers can only push but not pull on the sensors. In such a case the moment of force about the sensor center is due to the application of the resultant force at a certain distance from the center. The displacements of the points of digit force application change the moment arms of the forces that the digits exert on the hand-held object and make the computations more cumbersome.

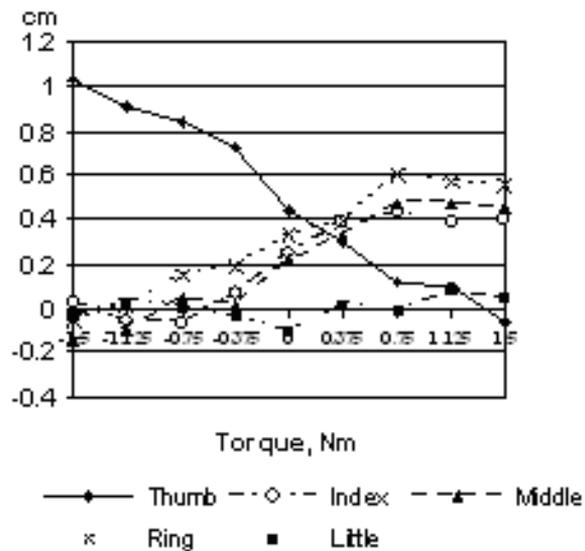


Figure 2. Displacement of the point of application of digit forces in the vertical direction at the various torque levels. The results are for an individual subject (average of ten trials). The positive direction of the torque is counterclockwise (pronation efforts), the negative direction is clockwise (supination efforts). Adapted by permission from V.M. Zatsiorsky, F. Gao, and M.L. Latash. Finger force vectors in multi-finger prehension. *Journal of Biomechanics*, 2003a, 36:1745-1749.

3. Hand asymmetry and hierarchical prehension control

Asymmetry in the hand function is an important feature that differentiates the hand from many parallel manipulators used in engineering as well as from some robotic hands (Fu & Pollard 2006). The functional hand asymmetry is in part due to the hand design (e.g. the thumb opposing other fingers, differences in the capabilities of index and little fingers, etc.) and in part is due to the hand control.

Due to the specific function of the thumb opposing other fingers in grasping, the forces of the four fingers can be reduced to a resultant force and a moment of force. This is equivalent to replacing a set of fingers with a *virtual finger*, VF (Arbib et al. 1985, Iberall 1987; Baud-Bovy & Soechting, 2001). A VF generates the same wrench as a set of actual fingers. There are substantial differences between the forces exerted by individual fingers (IF) and VF forces: (a) The IF force directions are as a rule dissimilar (for a review see Zatsiorsky & Latash 2008) while their resultant (i.e., VF) force is in the desired direction (Gao et al. 2005). (b) VF and IF forces adjust differently to modifications in task conditions (Zatsiorsky et al. 2002a, b). (c) IF forces are much more variable than VF forces (Shim et al. 2005a, b). The desired performance at the VF level is achieved by a synergic co-variation among individual finger forces at the IF level. The above facts support a hypothesis that multi-finger prehension is controlled by a two-level *hierarchical control* scheme (reviewed in Arbib et al. 1985; Mackenzie & Iberall 1994). At the upper level, the required mechanical action on the object is distributed between the thumb and the VF. At the lower level, action of the VF is distributed among individual fingers.

Functional hand asymmetry is also manifested in different responses to perturbations in the supination effort (SE) and pronation effort (PE) tasks. [The anatomical terms supination and pronation refer to the rotation of the forearm and hand along the longitudinal forearm axis in the clockwise and counterclockwise directions, respectively (as seen by the performer).] For instance, when subjects double their initial grasping force whilst maintaining the handle in the air in equilibrium, in the PE tasks the moment of normal forces exerted on the object increases while in the SE tasks it decreases (Figure 3). Such moment changes are not determined by the hand anatomy which is approximately symmetrical about the longitudinal axis of the hand (Li et al. 1998a). The changes in the moments of the normal forces are compensated by equal and opposite moments of the tangential forces such that the total moment exerted on the object does not change.

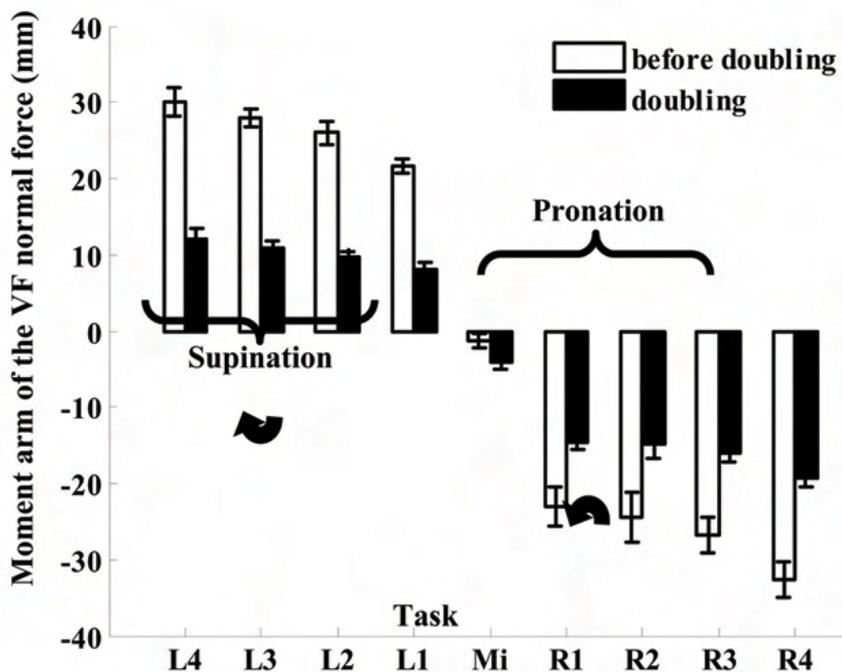


Figure 3. Changes of the moments of the normal forces after the doubling of the grasping force (From an unpublished study by X. Niu, M.L. Latash, V. Zatsiorsky, 2008)

Another example of the functional hand asymmetry in the SE and PE tasks comes from the experiments with transcranial magnetic stimulation (TMS). A single-pulse TMS applied over the hand projection in the left motor cortex (its descending pathways go to the segmental apparatus that controls the right hand) induced different reactions in the SE and PE tasks (Figure 4). Note that the changes in the total moment of force scale with the background moment of force (task moment of force in Figure 4), but supination responses dominate. The reasons behind the asymmetrical hand reactions to the TMS-induced perturbations in the SE and PE tasks are presently unknown.

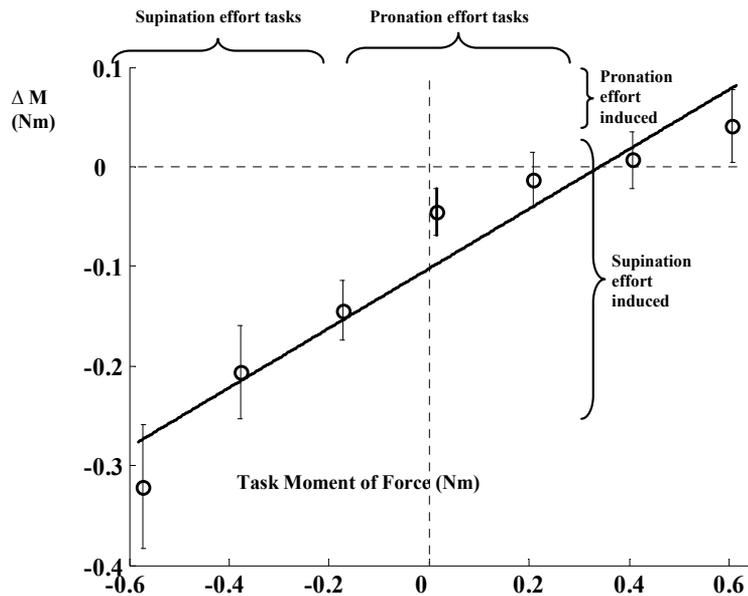


Figure 4. The dependence of the TMS-induced change in the total moment of force on the background moment of force required by the task. The grand average across subjects data ($n=6$) are presented with the linear regression line. (From an unpublished study by X. Niu, M.L. Latash, V. Zatsiorsky, 2008)

4. Finger interdependence and inter-finger connection matrices

In studies on parallel manipulators, the contacts are usually considered independent and identical in their properties. Consequently, all contact points are treated equally. Actions of human fingers are not independent (reviewed in Schieber, Santello, 2004; Zatsiorsky, Latash 2004; 2008). To demonstrate the finger interdependence turn your palm up and wiggle the ring finger. You will see that other fingers also move. This simple demonstration is an example of the so called *finger enslaving*—fingers that are not required to produce any force/motion by instruction are activated (Kilbreath & Gandevia 1994; Li et al. 1998b; Zatsiorsky et al. 2000; Kilbreath et al. 2002). Another type of the finger interdependence is *force deficit*— peak force generated by a finger in a multi-finger maximal voluntary contraction (MVC) task is smaller than its peak force in the single-finger MVC task. The deficit increases with the number of explicitly involved (master) fingers (Li et al. 1998a).

Finger interdependence is commonly described by *inter-finger connection matrices* (IFM) that relate the levels of commands to individual fingers with finger forces via a matrix equation (Zatsiorsky et al. 1998b; Li et al. 2002; Danion et al. 2003; Gao et al. 2003; Latash et al. 2003a):

$$\mathbf{f} = [\mathbf{W}]\mathbf{c} \quad (1)$$

where \mathbf{f} is a (4×1) vector of the normal finger forces, $[\mathbf{W}]$ is a (4×4) inter-finger connection matrix whose elements depend on the number of fingers involved in the task (i.e. they represent both the finger enslaving and force deficit), and \mathbf{c} is a (4×1) vector of *central (neural) commands*, representing by how much the person wants to involve individual

fingers. The elements of vector \mathbf{c} equal 1.0 if the finger is intended to produce maximal force (maximal voluntary activation), they are equal to zero if the finger is not intended to produce force (no voluntary activation). The inter-finger connection matrices can be computed by artificial neural networks based on experimental data involving finger force measurements (Zatsiorsky et al. 1998; Li et al. 2002; Gao et al. 2003, 2004; Latash et al. 2003a) or estimated by simple algebraic equations (Danion et al, 2003). The described approach led to the concept of finger *modes* that are arrays of finger forces caused by a single command to one of the fingers. If matrix $[\mathbf{W}]$ is known and actual finger forces in a prehension task are recorded, the vector of neural commands \mathbf{c} can be reconstructed by inverting equation (1): $\mathbf{c} = [\mathbf{W}]^{-1}\mathbf{f}$ (Zatsiorsky et al. 2002b). Matrix $[\mathbf{W}]$ is 4x4 and is always invertible.

When the vector \mathbf{c} is reconstructed, forces generated by individual fingers can be decomposed into components that are due to (a) direct commands to the targeted fingers, and (b) the enslaving effects, i.e. the commands sent to other fingers (Figure 5).

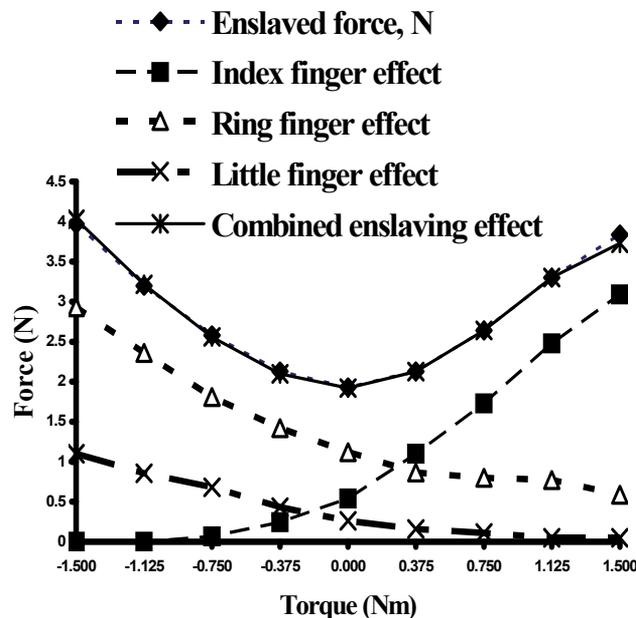


Figure 5. Decomposition of the normal force of the middle finger during holding a 2.0 kg load at different external torques. The data are from a representative subject. (Adapted from V.M. Zatsiorsky, R.W. Gregory, and M.L. Latash. (2002b) Force and torque production in static multifinger prehension. II. Control. *Biological Cybernetics*, 87: 40-49.)

5. Agonist and antagonist fingers

In multi-finger grasps, the finger forces generate the moments of force with respect to the thumb as a pivot. The fingers that are located above and below the thumb, for instance the index and the little fingers, generate moments in opposite directions (Li et al. 1998a, b). Moments in a desired direction – those that resist the external torque – have been termed the *agonist moments* while moments in the opposite direction – assisting the external torque – have been termed *antagonist moments* (Zatsiorsky et al. 2002a, b). The fingers that

generate agonist and antagonist moments with respect to a given task (external torque) are commonly addressed as the *agonist* and *antagonist fingers*, respectively.

Activation of the antagonist fingers increases the energy expenditure and can be mechanically unnecessary. Patterns of the antagonist moments depend on the load/torque combinations and can be described using a *3-zone model*, Zatsiorsky et al. 2002a): (A) *Large load-small torque* combinations. The antagonist fingers should be activated to prevent object slipping from the hand. In such tasks the antagonist moments are mechanically necessary. (B) *Intermediate load-intermediate torque* combinations. To prevent slipping, a performer has two options: (a) exert larger force by the agonist 'central' (middle or ring) finger while simultaneously decreasing the force of the agonist 'peripheral' (index or little) finger such that the VF normal force is above the slipping threshold, or (b) activate the antagonist fingers. (C) *Small load-large torque* combination. In such tasks, there is no need for the performer to be concerned about the object slipping from the hand because the force exerted by the agonist fingers is sufficient for the slip prevention. In this zone, antagonist moments are not mechanically necessary. They were however observed in all the tasks (Figure 6). One of the mechanisms causing the antagonist moments is enslaving; antagonist fingers are activated because strong commands are sent to the agonist fingers and the antagonist fingers are enslaved by these commands.

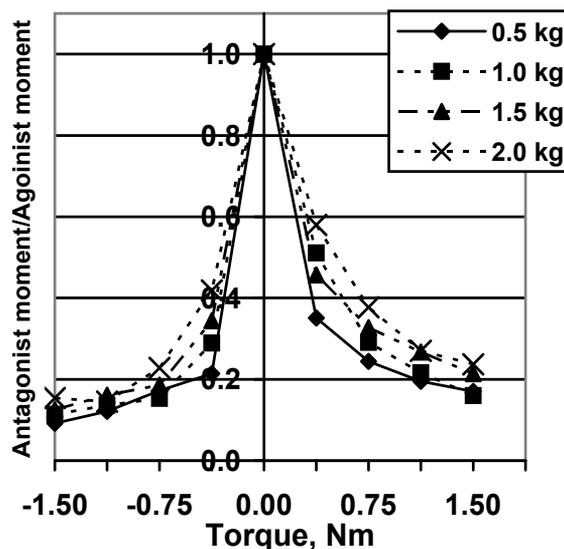


Figure 6. 'Antagonist/agonist moment' ratio in different tasks. Among the tasks, the load varied from 0.5 to 2.0 kg and the torque values were from -1.5 Nm to 1.5 Nm. The ratio for the zero torque conditions was estimated from the equilibrium requirements under the assumption that the normal forces of the two pairs of agonist and antagonist fingers were equal. Antagonist moments were observed over the entire range of load-torque combinations. (Adapted from V.M. Zatsiorsky, R.W.Gregory, and M.L.Latash. Force and torque production in static multifinger prehension: biomechanics and control. I. Biomechanics. *Biological Cybernetics*, 2002, 87:50-57.)

6. Grasp equation

The force-moment transformations from the digit tips to the hand-held object can be described with a *grasp equation*

$$\mathbf{F}=\mathbf{G}\mathbf{f} \quad (2)$$

For a planar task (see Figure 1), \mathbf{F} is a (3×1) vector of the resultant force and moment acting on the object, \mathbf{G} is a (3×10) *grasp matrix* (Mason & Salisbury 1985), and \mathbf{f} is a (10×1) vector of the digit forces. The elements of the first two rows of the matrix are the coefficients at the digit force values. Because in the position shown in Figure 1 the normal and tangential digit forces are along the X and Y axes of the global system of coordinates, the coefficients are either zeroes or ±1. When the normal and tangential digit forces are not parallel to the X and Y axes, e.g. when the handle is not oriented vertically, the coefficients equal the direction cosines, i.e. the projections of the unit vectors along the normal and tangential directions onto the X and Y axes. The elements of the last row in matrix \mathbf{G} are the moment arms of the digit forces about axis Z through the origin of the system of coordinates. \mathbf{G} is also known as the *matrix of moment arms*.

Equation (2) is a linear equation that allows for using the common methods of linear algebra. The equation is based however on a simplifying assumption that the elements of the grasp matrix are constant, i.e. the points of digit force applications do not displace during the period of observation. If they migrate, the elements of \mathbf{G} are not constant anymore and the equations become non-linear: variable values of digit forces are multiplied by the variable values of moment arms. In computations, this obstacle can be avoided if a (10×1) \mathbf{f} vector is expanded to a (15×1) vector where the added elements are the moments exerted by the individual digits with respect to the corresponding sensor centers. Matrix \mathbf{G} in this case is 3×15. For a general 3-D case, matrix \mathbf{G} is 6×30.

7. Internal forces during object manipulation

In multi-digit grasping, a vector of contact forces and moments \mathbf{f} can be broken into two orthogonal vectors: the resultant force vector \mathbf{f}_r (manipulation force) and the vector of the internal force \mathbf{f}_i ($\mathbf{f}=\mathbf{f}_r+\mathbf{f}_i$) (Kerr & Roth 1986; Yoshikawa & Nagai 1990, 1991). An *internal force* is a set of contact forces which can be applied to an object without disturbing its equilibrium (Mason & Salisbury 1985; Murray et al. 1994). The elements of an internal force vector cancel each other and, hence, do not contribute to the manipulation force (a resultant wrench exerted on the object). In human movement studies, the best known example of the internal forces is the *grasp force*, two equal and opposite normal forces exerted by the thumb and VF against each other. The resultant of these forces equals zero. An internal force is not a single force; it is a set of forces and moments that, when act together, generate a zero resultant force and a zero resultant moment.

In five-digit grasps in 3-D space, vector of individual digit forces and moments \mathbf{f} is a 30×1 vector. Its relation with a 6×1 vector \mathbf{F} of the resultant forces and moments acting on the object is described by equation (2) where \mathbf{G} is a 6×30 grasp matrix (Salisbury & Craig 1982; Kerr & Roth 1986). Vector of the internal forces \mathbf{f}_i lies in the null space of \mathbf{G} (the null space of a m by n matrix \mathbf{G} is the set of all vectors \mathbf{f} in R^n such that $\mathbf{G}\mathbf{f}=\mathbf{0}$ $\{\mathbf{f}\in\mathbf{N}(\mathbf{G})|\mathbf{G}\mathbf{f}=\mathbf{0}\}$).

Because the rank of a 6×30 matrix is at most 6, the dimensionality of the null space of the

grasp matrix (its nullity) is at least 24. The dimensionality of each of these vectors equals the total number of the digit forces and moments, i.e. thirty in five-digit grasps (some of the elements of an internal force vector can be zero). Hence there exist many finger force-moment combinations that interact in such a manner that the individual forces and moments cancel each other and do not contribute to the manipulation force. For instance, if individual tangential finger forces are in opposite directions, ulnar and radial, these force components can cancel each other such that the resultant tangential force equals zero. Analysis of all the 24 basic vectors of $\mathbf{N}(\mathbf{G})$ would be a daunting task. The force elements can be of different magnitude (provided that they negate each other's effects) and the 24 independent sets of internal forces (basic vectors) can be combined in different linear combinations, so there can be many internal forces (Gao et al. 2005). A performer can choose innumerable combinations of the internal force elements provided that they cancel each other.

So far, the research was mostly limited to the planar tasks performed with mechanically unconstrained objects and analyzed at the VF level. According to the mathematical analyses (Kerr & Roth 1986; Gao et al. 2005), at this level there exist only three internal forces: the *grasp force*, the *internal moment* (about an axis parallel to axis Z, see Figure 1), and the *twisting moment* - due to the opposite twisting moments exerted by the thumb and VF around the axis normal to the surfaces of the contacts. The latter combination is mechanically possible due to the soft finger contacts but cannot be actually realized in single-hand grasping: people cannot twist the thumb and the finger(s) in opposite directions (in two-hand grasping this option can be realized). Because of that, the twisting moment is neglected in the studies on human prehension.

The manipulation force is prescribed by the task mechanics. The internal forces allow for much freedom. The manipulation force vector and the vector of the internal force are mathematically independent (Kerr & Roth 1986; Yoshikawa & Nagai 1991). Practically this means that the central controller can change manipulation force without changing the internal force and vice versa (Yoshikawa & Nagai 1990, 1991; Gao et al. 2005 b, c). This opportunity is realized in robotics manipulators where the manipulation force and the internal forces are commonly controlled separately (e.g., Zuo & Qian 2000); the control is said to be decoupled. The decoupled control requires less computational resources; the controller does not have to bother about on-line adjustments of the grasp force to object acceleration and/or orientation. This strategy requires, however, exerting unnecessarily large forces and is, in this sense, uneconomical. People do not use this option. Available data suggest that the CNS prefers to face larger computational costs rather than produce excessive forces. In contrast to robots, people adjust the internal forces to the manipulation forces during the object transport (Flanagan & Wing 1993, 1995; Smith & Soechting 2005; Zatsiorsky et al. 2005; Gao et al. 2005 b, c). The pattern of the adjustment depends on the performed movement.

Gao et al (2005 b, c) studied vertical and horizontal object movement at the three handle orientations, vertical, horizontal and diagonal (inclined 45°). In total, six combinations of handle orientation and movement direction were tested: (1) Parallel manipulations. (1a). VV task: Vertical orientation-vertical movement. (1b). HH task: Horizontal orientation-horizontal movement. (2) Orthogonal manipulations. (2a) VH task: Vertical orientation-horizontal movement. (2b) HV task: Horizontal orientation-vertical movement (3) Diagonal manipulations. (3a.) DV task: Diagonal orientation-vertical movement. (3b). DH task:

Diagonal orientation-horizontal movement. In the above description the following terminology is used: When the handle orientation and the direction of manipulation are along the same axis (e. g. a vertically oriented handle is being moved in the vertical direction or a horizontally oriented handle in the horizontal direction) the manipulation is called the *parallel manipulation*. The *orthogonal manipulation* corresponds to the object motion at the right angle to the handle orientation, e.g. a vertically oriented handle is being moved in a horizontal plane or a horizontally oriented handle is moved in a vertical plane.

The summary results on the coordination of internal forces in various manipulation tasks are presented in Table 1. The following terminology is used. *Grasping synergy* (GS) is a conjoint change of the normal digit forces (Zatsiorsky & Latash 2004). The coordination pattern characterized by a simultaneous (in-phase) increase or decrease of the normal forces of the thumb and VF is called *symmetric GS* (in the VV task a symmetric GS was used, see Figure 7 below). The reciprocal thumb force-VF force changes when the normal forces of the thumb and VF change in opposite directions are called *anti-symmetric GS*, see Figure 8 where the internal force demonstrated an 'inverted V' pattern with respect to the handle acceleration.

Manipulation		Coupling of the thumb and VF normal forces (Grasping synergies)	Internal force-manipulation force (acceleration) coupling
<i>Parallel</i>	VV	Symmetric	Positive
	HH	Symmetric	Positive (the internal force increases slightly with the acceleration magnitude)
<i>Orthogonal</i>	VH	Anti-symmetric	Inverted -V pattern (symmetric)
	HV	Depends on the frequency. Approximately anti-symmetric at high frequencies.	Depends on the frequency. At high frequencies an asymmetric inverted-V pattern with the peak at 1 g
<i>Diagonal</i>	DV	Symmetric	Positive
	DH	Anti-symmetric	Positive (for the selected direction of the coordinate axes)

Table 1. Internal forces during different manipulations of the hand-held objects

Consider the cases of symmetric GS and anti-symmetric GS. When people move a vertically oriented object in the vertical direction (the VV tasks) the grip force F_G increases in parallel with the object acceleration and hence the load force F_L ($F_L = W + ma$ where W is the object's weight, m is its mass and a is acceleration), apparently to prevent slip (Johansson & Westling 1984; Flanagan & Wing 1993, 1995; Flanagan et al. 1993; Flanagan & Tresilian 1994; Nakazawa et al. 1996; Kinoshita et al. 1996, Gordon et al. 1999; Gysin et al. 2003; see also Flanagan & Johansson 2002 for a review). The $F_G - F_L$ coupling is so strong that people increase F_G in parallel with F_L even when F_G is already much above the slipping threshold, e.g. when before lifting the object a performer purposefully grasps the object with a high

force (Flanagan & Wing 1995). Figure 7 illustrates the finding. Note that the grasping force is larger for larger object acceleration.

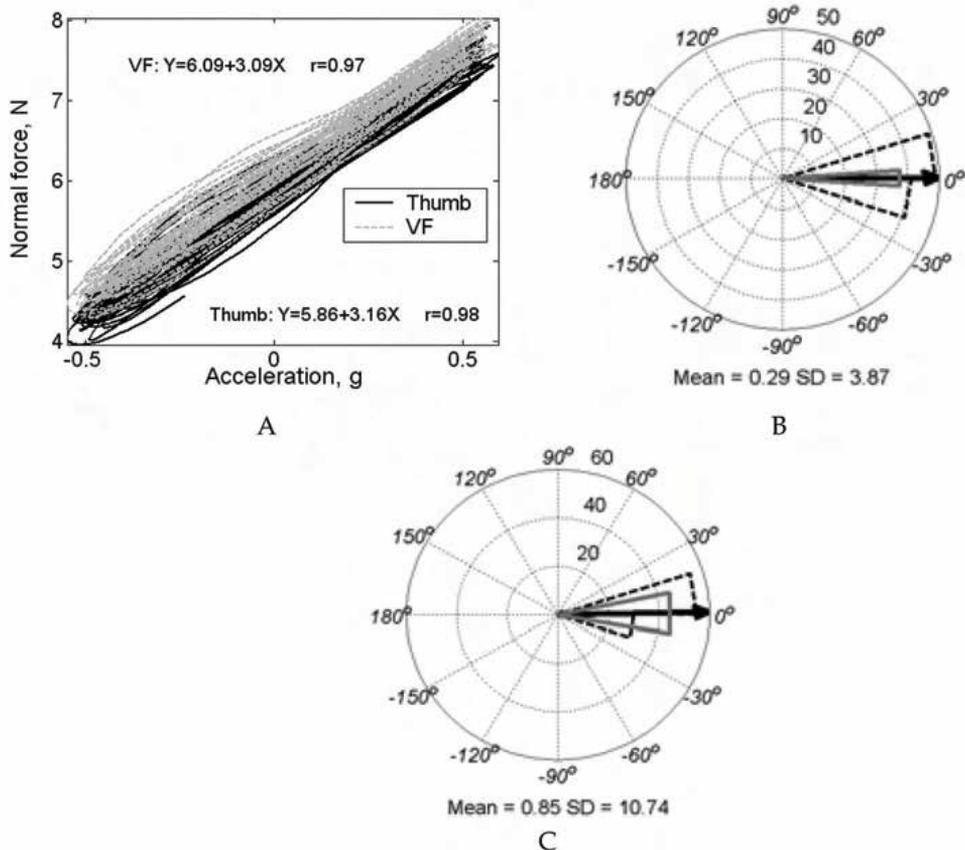


Figure 7. VV manipulation. (A) Normal forces of the thumb and VF (N) versus the handle acceleration in the vertical direction. A representative trial, weight (W) 8.8 N, frequency 1.5 Hz, representative subject. With the acceleration from approximately -0.5 g to 0.5 g the tangential force L (load) varied from approximately 4.4 to 13.2 N ($L = W + ma$). The range of the internal force fluctuations was approximately 4.0 N, from 4.0 to 8.0 N. (B) The VF normal force - acceleration phase angles (circular histogram). The phase angles cluster around zero degree. (C) The VF normal force - thumb normal force phase angles (circular histogram). Thick black arrow represents the mean phase angles and the gray triangle represents the angular standard deviation, The dashed lines illustrate the data for individual bins, the bin size is 20° . $n = 90$ (6 subjects \times 5 loads \times 3 frequencies). Adapted from Gao, F., Latash, M. L., Zatsiorsky, V. M. (2005) Internal forces during object manipulation. *Experimental Brain Research*, 165 (1): 69-83.

Quite a different coordination is observed in the VH tasks (horizontal movement of a vertically oriented object). In these tasks, the maximal grasping force is observed at the instances of maximal speed, and hence zero acceleration (Smith & Soechting 2005; Gao et al. 2005a, b), Figure 8.

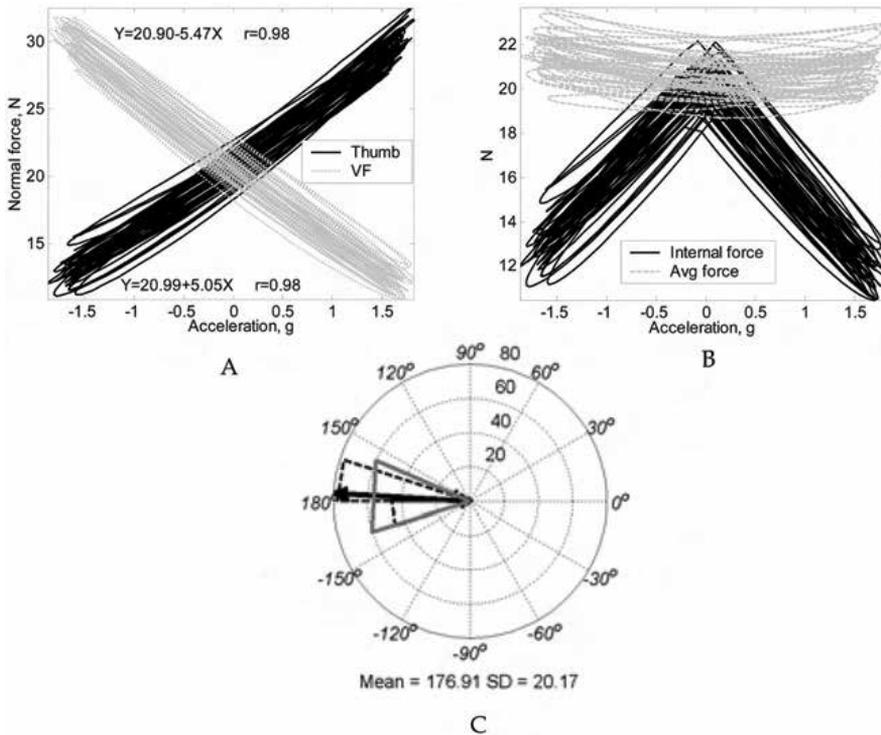


Figure 8. VH manipulation. (A) Normal forces of the thumb and VF versus the handle acceleration in the horizontal direction. A representative trial, the load was 11.3 N, the frequency was 3 Hz, representative subject #1. (B) Internal force and average normal force versus the handle acceleration. The range of the internal force fluctuations was approximately 8.0 N, from 12.0 N to 21.0 N. (C) The thumb normal force-VF normal force phase angle [circular histogram, $n = 150$ (6 subjects \times 5 loads \times 5 frequencies)]. Adapted from Gao F, Latash ML, Zatsiorsky VM (2005) Internal forces during object manipulation. *Experimental Brain Research*, 165 (1): 69-83.

In addition to the grasping force changes, the moments exerted by the normal and tangential digit forces also change during object manipulation while compensating for each other's changes and preserving the object orientation (Figure 9).

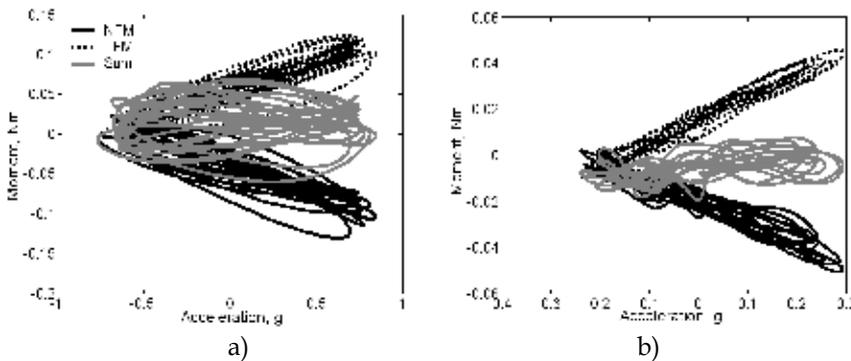


Figure 9. Internal and resultant moments in various tasks. Representative examples. Abbreviations: NFM-Moment of normal force; TFM-Moment of tangential force; Sum =

(NFM+TFM) = resultant moment. (A) VV: 2 Hz, 13.8 N, representative subject; (B) HH: 1 Hz, 6.3 N, representative subject. Adapted from Gao, F., Latash, M. L., Zatsiorsky, V. M. (2005) Internal forces during object manipulation. *Experimental Brain Research*, 165 (1): 69-83.

8. Local and synergic reactions to perturbations

When grasping different objects performers adjust digit forces to the object features. Two types of the adjustments are distinguished, local and synergic. The term *local* describes the responses that start and end at the same digit, i.e. an effect of friction at a given digit on the force exerted by this digit. The term *synergic* refers to changes in a finger's force in response to changes in friction under other finger(s) (Aoki et al. 2006, 2007; Zatsiorsky et al. 2006).

An example of the local and synergic reactions could be seen in experiments in which the friction under each digit was different, either high or low, resulting in eight friction conditions (for the three-digit grasps, Niu et al. 2007) or 32 friction conditions (Aoki et al. 2007). The difference between the high and low friction was three-fold. When friction under a digit was low, its tangential force decreased and the normal force increased (local effects). Digit forces were also adjusted to friction at other digits (synergic effects). The synergic effects were directed to maintain the handle equilibrium. For instance, to keep the total tangential force constant, the tangential forces of the thumb and fingers changed in opposite directions (Figure 10).

Not only the VF tangential force but also the tangential forces of the individual fingers are affected by the local friction conditions (Aoki et al. 2007; Niu et al. 2007). The tangential force adjustments to the local friction support the notion that the VF tangential force sharing is under neural control; the sharing percentage is not determined solely by the passive mechanical properties of the individual fingers and joints. In other words, the metacarpal joints cannot be modeled as simple hinges.

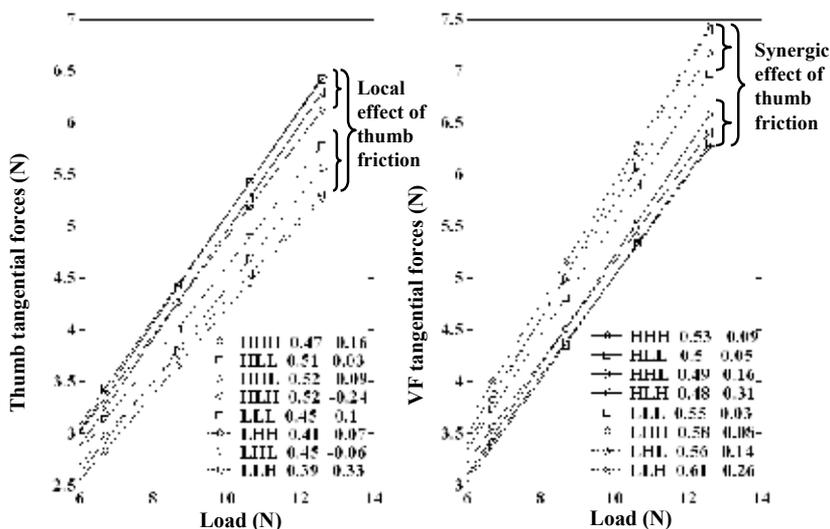


Figure 10. Tangential forces of the thumb and VF in the three-digit grasps as a function of the load and friction, high (H) or low (L). The eight friction conditions were HHH, HLL, HHL, HLH, LLL, LHH, LHL, and LLH, where the letters correspond to the friction condition

for the thumb, index and middle fingers, respectively. The friction sets with the thumb at a low friction contact (LLL, LHH, LHL and LLH) are printed with dotted lines. The solid lines represent the tasks with the high friction contact at the thumb. In the left panel, LFE is the *local friction effect*, i.e. the difference induced by the high or low friction contact at the thumb. Two other smaller figure brackets show the synergic effects, i.e. the effect of friction at other digits on the thumb force. The numbers in the bottom right insets are the regression coefficients and intercepts (the regression model $f_i^t = a_i + k_i^{(1)}L$ was used for computations). Note the small values of the intercepts. Cf. the right and the left panels: the thumb friction, H or L, induced opposite changes of the thumb and VF forces. (The figure is from X. Niu, M. Latash, V.M. Zatsiorsky (2007) Prehension synergies in minimally redundant grasps & the triple-product model of digit force control, *Experimental Brain Research*, 98 (1): 16-28.)

Local and synergic reactions are not limited to adjustments to local friction. For instance, similar reactions were observed when performers were holding a motorized handle while the handle width was forcibly either increased or decreased (Zatsiorsky et al. 2006). Handle expansion/contraction did not perturb the handle equilibrium; both the resultant force and moment acting on the handle remained the same. However, when the handle width increased each digit was perturbed (the length of the flexor muscle increased), and a restoring force tending to return the digit to its previous position arose (the local digit force adjustment). The local mechanisms, e.g. stretch reflexes, were directed to resist the imposed digit displacement. These mechanisms violated the object equilibrium, whilst the synergic force adjustments restored the equilibrium.

9. Principle of superposition in human prehension

The *principle of superposition* refers to decomposition of complex skilled actions into several elemental actions, which can be controlled independently by several controllers. The principle was first suggested in robotics (Arimoto et al. 2001; Arimoto & Nguyen 2001) and was verified for the dexterous manipulation of an object by two soft-tip robot fingers. Such a control can be realized via a linear superposition of two commands, one command for the stable grasping and the second one for regulating the orientation of the object. In robotics, such a decoupled control decreases the computation time.

When applied to multi-finger grasps in humans, the principle claims that at the VF level the forces and moments during prehension are defined by two independent commands: "Grasp the object stronger/weaker to prevent slipping" and "Maintain the rotational equilibrium of the object". The commands correspond to the two internal forces discussed previously, the grip force and the internal moment. The effects of the two commands are summed up. The validity of the principle was confirmed in a set of diverse experiments (for the review see Zatsiorsky et al. 2004; see also Shim et al. 2005 and Shim & Park, 2007)). The principle allows explaining the digit force adjustments to different factors such as (a) the load force and its modulation associated with the handle acceleration; (b) the external torque and its modulation; (c) the object orientation in the gravity field; (c) friction at the digit tips; and some other variables. The variations of the above factors may require similar or opposite adjustments. For instance, an increase of the object weight and a decrease in friction both require a larger gripping force while, a decrease of the load and a decrease in friction require opposite grasp force changes, a force decrease and increase, respectively. It has been

suggested that the CNS responds to a mixture of similar or opposite requirements follows a rule: *Adjustment to the sum equals the sum of the adjustments* (reviewed in Zatsiorsky & Latash 2008).

10. References

- Aoki, T.; Latash, M. L., and Zatsiorsky, V. M. (2007). Adjustments to local friction in multifinger prehension. *J Mot Behav.* Jul; 2007. 39(4):276-90.
- Aoki, T.; Niu, X.; Latash, M. L., and Zatsiorsky, V. M. (2006) Effects of friction at the digit-object interface on the digit forces in multi-finger prehension. *Exp Brain Res.* 2006 Jul; 172(4):425-38.
- Arbib MA; Iberall T, and Lyons D. (1985) Coordinated control programs for movements of the hand. Goodwin AW and Darian-Smith I, eds. *Hand Function and the Neocortex.* Berlin: Springer Verlag; 1985; pp. 111-129.
- Arimoto, S. and Nguyen, P. T. A. (2001) Principle of superposition for realising dexterous pinching motions of a pair of robot fingers with soft-tips. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences.* 2001 Jan; E84A(1):39-47.
- Arimoto, S.; Tahara, K.; Yamaguchi, M.; Nguyen, P. T. A., and Han, H. Y. (2001) Principles of superposition for controlling pinch motions by means of robot fingers with soft tips. *Robotica.* 2001 Jan-2001 Feb 28; 19:21-28.
- Baud-Bovy, G. and Soechting, J. F. (2001) Two virtual fingers in the control of the tripod grasp. *J Neurophysiol.* 2001 Aug; 86(2):604-15.
- Danion, F.; Schoner, G.; Latash, M. L.; Li, S.; Scholz, J. P., and Zatsiorsky, V. M. (2003) A mode hypothesis for finger interaction during multi-finger force-production tasks. *Biol Cybern.* 2003 Feb; 88(2):91-8.
- Flanagan, J. R.; Tresilian, J., and Wing, A. M. (1993) Coupling of grip force and load force during arm movements with grasped objects. *Neurosci Lett.* 1993 Apr; 152(1-2):53-6.
- Flanagan, J. R. and Tresilian, J. R. (1994) Grip-load force coupling: a general control strategy for transporting objects. *J Exp Psychol Hum Percept Perform.* 1994 Oct; 20(5):944-57.
- Flanagan, J. R. and Wing, A. M. (1993) Modulation of grip force with load force during point-to-point arm movements. *Exp Brain Res.* 1993; 95(1):131-43.
- Flanagan, J. R. and Wing, A. M. (1995) The stability of precision grip forces during cyclic arm movements with a hand-held load. *Exp Brain Res.* 1995; 105(3):455-64.
- Flanagan JR and Johansson RS. (2002) Hand movements. Ramshandran VS, ed. *Encyclopaedia of the human brain.* San Diego: Academic Press; 2002; pp. 399-414.
- Fu JL, Pollard NS (2006) On the importance of asymmetries in grasp quality metrics for tendon driven hands. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS);* Beijing, China. Beijing; 2006: 1068-1075.
- Gao F, Latash ML, and Zatsiorsky V M (2004) Neural network modeling supports a theory on the hierarchical control of prehension. *Neural Computing & Applications.* 2004; 13(4): 352-359.
- Gao F, Latash ML, and Zatsiorsky VM (2005a) Control of finger force direction in the flexion-extension plane. *Exp Brain Res.* 2005 Mar; 161(3):307-15.
- Gao F, Latash ML, and Zatsiorsky VM (2005b) Internal forces during object manipulation. *Exp Brain Res.* 2005 Aug; 165(1):69-83.

- Gao F, Latash ML Zatsiorsky VM. (2005c) In contrast to robots, in humans internal and manipulation forces are coupled (ThP01-18). *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*; Chicago, IL, USA. 2005: 404-407.
- Gao F, Li S, Li ZM, Latash ML, and Zatsiorsky VM. (2003) Matrix analyses of interaction among fingers in static force production tasks. *Biol Cybern.* 2003 Dec; 89(6):407-14.
- Gordon AM, Charles J, and Duff SV (1999) Fingertip forces during object manipulation in children with hemiplegic cerebral palsy. II: bilateral coordination. *Dev Med Child Neurol.* 1999 Mar; 41(3):176-85.
- Gysin P, Kaminski TR, and Gordon, A. M. (2003) Coordination of fingertip forces in object transport during locomotion. *Exp Brain Res.* 2003 Apr; 149(3):371-9.
- Iberall T (1987) The nature of human prehension: Three dexterous hands in one. *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*; Raleigh, NC. 1987; 396-401.
- Johansson, R. S. and Westling, G. (1984) Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Exp Brain Res.* 1984; 56(3):550-64.
- Kerr JR, Roth B. (1986) Analysis of multifingered hands. *Journal of Robotics Research.* 1986; 4(4):3-17.
- Kilbreath, S. L.; Gorman, R. B.; Raymond, J., and Gandevia, S. C. (2002) Distribution of the forces produced by motor unit activity in the human flexor digitorum profundus. *J Physiol.* 2002 Aug 15; 543(Pt 1):289-96.
- Kilbreath SL and Gandevia SC. (1994) Limited independent flexion of the thumb and fingers in human subjects. *J Physiol.* 1994; 479 (Pt 3):487-97.
- Kinoshita H; Kawai S; Ikuta K , and Teraoka T. (1996) Individual finger forces acting on a grasped object during shaking actions. *Ergonomics.* 1996; 39(2):243-56; ISSN: 0014-0139.
- Latash MI; Gao F, and Zatsiorsky VM. (2003) Similarities and differences in finger interaction across typical and atypical populations. *Journal of Applied Biomechanics.* 2003; 19(3):264-270.
- Li ZM; Latash ML , and Zatsiorsky VM. Force sharing among fingers as a model of the redundancy problem. *Exp Brain Res.* 1998a; 119(3):276-86.
- Li ZM; Latash ML; Newell KM, and Zatsiorsky VM. (1998b) Motor redundancy during maximal voluntary contraction in four-finger tasks. *Exp Brain Res.* 1998; 122(1):71-7.
- Li ZM; Zatsiorsky VM; Latash ML; Bose NK. (2002) Anatomically and experimentally based neural networks modeling force coordination in static multi-finger tasks. *Neurocomputing .* 2002; 47:259-275.
- MacKenzie CL and Iberall T. (1994) *The grasping hand.* Amsterdam; New York: North-Holland.; 1994.
- Mason MT and Salisbury JK. (1985) *Robot Hands and the Mechanics of Manipulation.* Cambridge, MS: The MIT Press; 1985.
- Murray RM; Li Z, and Sastry SS. (1994) *A Mathematical Introduction to Robotic Manipulation.* Boca Raton, FL: CRC Press; 1994.
- Nakazawa, N.; Ikeura, R., and Inooka, H. (2000) Characteristics of human fingertips in the shearing direction. *Biol Cybern.* 2000 Mar; 8(3):207-14.
- Nakazawa N; Uekita Y; Inooka H, and Ikeura R. (1996) Experimental study on human grasping force. *IEEE International Workshop on Robot and Human Communication.* 1996; pp. 280-285.

- Niu, X.; Latash, M. L., and Zatsiorsky, V. M. (2007) Prehension synergies in the grasps with complex friction patterns: local versus synergic effects and the template control. *J Neurophysiol.* 2007 Jul; 98(1):16-28.
- Pare, M. and Dugas, C. (1999) Developmental changes in prehension during childhood. *Exp Brain Res.* 1999 Apr; 125. 125(3. 3):239-47.
- Pataky, T. C.; Latash, M. L., and Zatsiorsky, V. M. (2005) Viscoelastic response of the finger pad to incremental tangential displacements. *J Biomech.* 2005 Jul; 38(7):1441-9.
- Schieber, M. H. and Santello, M. (2004) Hand function: peripheral and central constraints on performance. *J Appl Physiol.* 2004 Jun; 96(6):2293-300.
- Serina, E. R. and Rempel, D. M. (1996) Fingertip pulp response during keystrokes. *Twentieth Annual Meeting of the American Society of Biomechanics, Conference Proceedings; Georgia Tech, Atlanta, Georgia.* 1996: 237-238.
- Shim, J. K.; Latash, M. L., and Zatsiorsky, V. M. (2005a) Prehension synergies in three dimensions. *J Neurophysiol.* 2005 Feb; 93(2): 766-76.
- Shim, J. K.; Latash, M. L., and Zatsiorsky, V. M. (2005b) Prehension synergies: trial-to-trial variability and principle of superposition during static prehension in three dimensions. *J Neurophysiol.* 2005 Jun; 93(6):3649-58.
- Shim, J. K. and Park, J. (2007) Prehension synergies: principle of superposition and hierarchical organization in circular object prehension. *Exp Brain Res.* 2007 Jul; 180(3):541-56.
- Smith, M. A. and Soechting, J. F. (2005) Modulation of grasping forces during object transport. *J Neurophysiol.* 2005 Jan; 93(1):137-45.
- Srinivasan, M. A. (1989) Surface deflection of primate fingertip under line load. *J Biomech.* 1989; 22(4):343-9.
- Srinivasan, M. A; Gulati, R. J., and Dandekar, K. (1992) In vivo compressibility of the human fingertip. *Advances in Bioengineering, Proceedings of the ASME* (Ed. M.W. Bidez). 1992; 22:573-576.
- Yoshikawa T and Nagai K. (1980) Analysis of multi-fingered grasping and manipulation. Venkarataman ST and Iberall T, eds. *Dextrous Robot Hands*. New York: Springer-Verlag; 1990; pp. 187-208.
- Yoshikawa T and Nagai K. (1991) Manipulating and grasping forces in manipulation by multifingered robot hands. *IEEE Trans Robot. Autom.* 1991; 7:67-77.
- Zatsiorsky, V. M.; Gao, F., and Latash, M. L. (2003) Finger force vectors in multi-finger prehension. *J Biomech.* 2003 Nov; 36(11):1745-9.
- Zatsiorsky, V. M.; Gao, F., and Latash, M. (2006) Prehension stability: experiments with expanding and contracting handle. *J Neurophysiol.* 2006 Apr; 95(4):2513-29.
- Zatsiorsky, V. M.; Gregory, R. W., and Latash, M. L. (2002a) Force and torque production in static multifinger prehension: biomechanics and control. I. Biomechanics. *Biol Cybern.* 2002a; 87(1):50-7.
- Zatsiorsky, V. M.; Gregory, R. W., and Latash, M. L. (2002b) Force and torque production in static multifinger prehension: biomechanics and control. II. Control. *Biol Cybern.* 2002b; 87(1):40-9.
- Zatsiorsky, V. M. and Latash, M. L. (2004) Prehension synergies. *Exerc Sport Sci Rev.* 2004 Apr; 32(2):75-80.
- Zatsiorsky, V. M.; Li, Z. M., and Latash, M. L. (2000) Enslaving effects in multi-finger force production. *Exp Brain Res.* 2000 Mar; 131(2):187-95.

- Zatsiorsky VM and Latash ML. (2008) Multi-finger prehension: An Overview. *Journal of Motor Behavior*. 2008; (In press.).
- Zatsiorsky VM; Latash ML; Danion F; Gao F; Li ZM; Gregory RW, and Li S. (2004a) Inter-finger connection matrices Bubak M; van Albada GD; Sloop PMA, and Dongarra JJ, Eds. *Computational Science - ICCS 2004, 4th International Conference*; Krakow, Poland. Berlin: Springer; 2004: 1056-1064.
- Zatsiorsky VM; Latash ML; Gao F, and Shim JK. (2004b) The principle of superposition in human prehension. *Robotica*. 2004; 22:231-234.
- Zatsiorsky VM; Li ZM, and Latash ML. (1998) Coordinated force production in multi-finger tasks: finger interaction and neural network modeling. *Biol Cybern*. 1998 Aug; 79(2):139-50; ISSN: 0340-1200.
- Zuo, B. R. and Qian, W. H. (2000) A general dynamic force distribution algorithm for multifingered grasping. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*. 2000 Feb; 30(1):185-192.

Mobility of Spatial Parallel Manipulators

Jing-Shan Zhao, Fulei Chu and Zhi-Jing Feng
*Department of Precision Instruments and Mechanology,
 Tsinghua University, Beijing 100084,
 P. R. China.*

1. Introduction

This chapter focuses on the mobility analysis of spatial parallel manipulators. It first develops an analytical methodology to investigate the instantaneous degree of freedom (DOF) of the end-effector of a parallel manipulator. And then, the instantaneous controllability of the end-effector is discussed from the viewpoint of the possible actuation schemes which will be especially useful for the designers of the parallel manipulators. Via comparing the differences and essential mobility of a set of underactuated, over actuated and equally actuated manipulators, this chapter demonstrates that the underactuated, over actuated and equally actuated manipulators are all substantially fully actuated mechanisms. This work is significantly important for a designer to contrive his or her manipulators with underactuated or over actuated structures.

Based on the analytical model of the DOF of a spatia $(\mathcal{S})^T E \mathcal{S}^r = 0$ (16)

where \mathcal{S} and \mathcal{S}^r are column vectors, $E = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix}$, and \mathbf{I}_3 and $\mathbf{0}_3$ are 3×3 identity and zero matrices, respectively.

Similarly, if one gets a set of terminal constraints exerted to a rigid body, its free motion(s) can also be solved through equation (16). Next, one can investigate the instantaneous mobility of the end-effector of a parallel manipulator with equation (16).

2.1. The degree of freedom of the end-effector of a parallel manipulator

The free motions of the end-effector can be instantaneously expressed in a set of Plücker homogeneous coordinates in one Cartesian coordinate system. The main steps are:

1. Investigate the Terminal Constraints of the Kinematic Chains

In general, any parallel manipulator can be decomposed into $n(n \geq 1)$ kinematic chains connecting the end effector with the base. In order to instantaneously analyze the mobility properties of the end-effector, this section only establishes one absolute coordinate system. After establishing the coordinate system, the Plücker homogeneous coordinates of all kinematic pairs in a chain can be obtained. Group all of the kinematic screws of the same chain to be $\mathcal{S}_i (i = 1, 2, \dots, n)$ and solve the terminal constraint(s) \mathcal{S}_i^r with equation (16).

In fact, if all of the terminal constraints of the kinematic chains are gained, the constraints exerted to the end-effector, denoted by \mathcal{S}_E^r , should also be obtained. The dimension of constraint spaces spanned by the terminal constraints of kinematic chains can be simplified as $d = Rank(\mathcal{S}_E^r)$.

2. Solve the Free Motion(s), \mathcal{S}_E^F , of the End-Effector with Equation (16)

Naturally, the mobility properties of the end-effector is fully expressed by \mathcal{S}_E^F . Its number of DOF can be expressed as:

$$M = Rank(\mathcal{S}_E^F) = 6 - d \tag{17}$$

Now, the DOF of the end-effector of the parallel manipulator shown in Fig. 1 can be instantaneously investigated with the above two steps. In this manipulator, the end-effector $C_1C_2C_3$ has three identical *PPRR* kinematic chains connected with the fixed base. For the sake of modelling, one can establish any Cartesian coordinate system for the manipulator. Assume that the direction vector of the prismatic joint $A_i (i=1,2,3)$ is denoted by $e_{A_i} = (a_{i1} \ b_{i1} \ c_{i1})^T$, the direction vector of the prismatic joint $B_i (i=1,2,3)$ is denoted by $e_{B_i} = (a_{i2} \ b_{i2} \ c_{i2})^T$, the rotational vector of the revolute joint B_i is denoted by $e_{B_i^r} = e_{A_i} \times e_{B_i} = (b_{i1}c_{i2} - b_{i2}c_{i1} \ a_{i2}c_{i1} - a_{i1}c_{i2} \ a_{i1}b_{i2} - a_{i2}b_{i1})^r$, the rotational vector of the revolute joint $C_i (i=1,2,3)$ is denoted by $e_{C_i^r} = e_{B_i^r}$. Also suppose that $e_{A_1} \neq e_{A_2} \neq e_{A_3}$ and $e_{B_i} \cdot e_{A_i} = 0$.

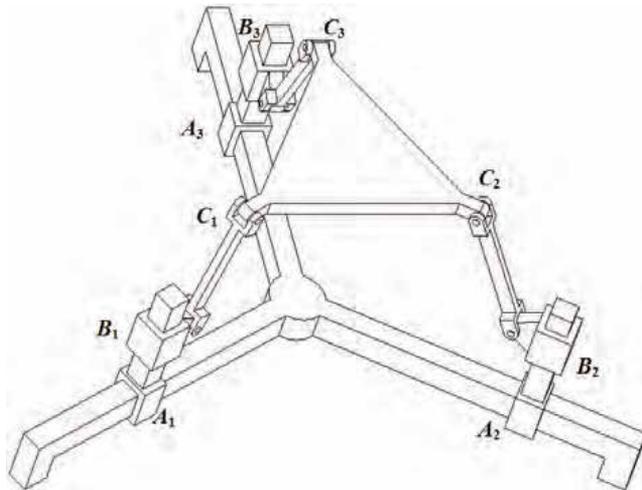


Fig. 1 a 3-PPRR Spatial Mechanism

So, the kinematic screws for each kinematic chain can be expressed as:

$$\mathcal{S}_{A_i B_i C_i} = \begin{bmatrix} \mathcal{S}_{A_i} & \mathcal{S}_{B_i} & \mathcal{S}_{B_i^r} & \mathcal{S}_{C_i^r} \end{bmatrix} \tag{18}$$

where

$$\begin{aligned} \mathcal{S}_{A_i} &= (0 \ 0 \ 0 \ a_{i1} \ b_{i1} \ c_{i1})^T, \ \mathcal{S}_{B_i} = (0 \ 0 \ 0 \ a_{i2} \ b_{i2} \ c_{i2})^T, \\ \mathcal{S}_{B_i^r} &= \begin{pmatrix} b_{i1}c_{i2} - b_{i2}c_{i1} & a_{i2}c_{i1} - a_{i1}c_{i2} & a_{i1}b_{i2} - a_{i2}b_{i1} & y_{B_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & z_{B_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & x_{B_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) \\ -z_{B_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) & -x_{B_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & -y_{B_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & & & \end{pmatrix}^T, \\ \mathcal{S}_{C_i^r} &= \begin{pmatrix} b_{i1}c_{i2} - b_{i2}c_{i1} & a_{i2}c_{i1} - a_{i1}c_{i2} & a_{i1}b_{i2} - a_{i2}b_{i1} & y_{C_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & z_{C_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & x_{C_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) \\ -z_{C_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) & -x_{C_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & -y_{C_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & & & \end{pmatrix}^T. \end{aligned}$$

The terminal constraints of the kinematic chain can be solved with (16):

$$\mathcal{S}_{A_i B_i C_i}^r = \begin{bmatrix} \mathcal{S}_i^{r1} & \mathcal{S}_i^{r2} & \mathcal{S}_i^{r3} \end{bmatrix} \tag{19}$$

where $\mathcal{S}_i^{r1} = (0 \ 0 \ 0 \ a_{i1} \ b_{i1} \ c_{i1})^T$, $\mathcal{S}_i^{r2} = (0 \ 0 \ 0 \ a_{i2} \ b_{i2} \ c_{i2})^T$,

$$\mathcal{S}_i^{r3} = \begin{pmatrix} b_{i1}c_{i2} - b_{i2}c_{i1} & a_{i2}c_{i1} - a_{i1}c_{i2} & a_{i1}b_{i2} - a_{i2}b_{i1} & y_{C_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & z_{C_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & x_{C_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) \\ -z_{C_i}(a_{i2}c_{i1} - a_{i1}c_{i2}) & -x_{C_i}(a_{i1}b_{i2} - a_{i2}b_{i1}) & -y_{C_i}(b_{i1}c_{i2} - b_{i2}c_{i1}) & & & \end{pmatrix}^T.$$

According to the mechanism shown in Fig. 1, $e_{B_1} = e_{B_2} = e_{B_3}$. Therefore, the terminal constraints exerted to the end-effector by these three kinematic chains are:

$$\mathcal{S}_{C_1 C_2 C_3}^r = \begin{bmatrix} \mathcal{S}_1^{r1} & \mathcal{S}_2^{r1} & \mathcal{S}_3^{r1} & \mathcal{S}_1^{r2} & \mathcal{S}_1^{r3} & \mathcal{S}_2^{r3} & \mathcal{S}_3^{r3} \end{bmatrix} \tag{20}$$

It is not difficult to find that the rank of $\mathcal{S}_{C_1 C_2 C_3}^r$ expressed by equation (20) is 5, and the free motions of the end-effector $C_1 C_2 C_3$ can be again solved with equation (16):

$$\mathcal{S}_{C_1 C_2 C_3}^F = (0 \ 0 \ 0 \ a_{12} \ b_{22} \ c_{32})^T \tag{21}$$

Equation (21) indicates that the end-effector has one translational DOF along the direction vector $e = (a_{11} \ b_{11} \ c_{11})^T$. Of course, the number of the DOF of the end-effector is $Rank(\mathcal{S}_{C_1 C_2 C_3}^F) = 1$, the direction is $e = (a_{12} \ b_{22} \ c_{32})^T$ and the type is translation, which is fully represented by the screw expression (21).

2.2. The number of actuators required to control the end-effector of a spatial parallel manipulator

After obtaining the instantaneous mobility of the end-effector, one can directly exert M actuators to the manipulator, and then investigate the CDOF of the end-effector by solving

the free motion(s) of the end-effector within its workspace. If the newly solved motion(s), denoted by $\mathcal{S}_E^{F_{ni}}$, $i=1,2,\dots$, satisfy that $\mathcal{S}_E^{F_{ni}} \neq (0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$, then, additional actuations are needed under this configuration and the actuation scheme. Of course, we can either reselect the actuation scheme or add $\text{Rank}(\mathcal{S}_E^{F_{ni}})$ more actuation(s) under this configuration

until $\mathcal{S}_E^{F_{ni}} = (0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$. The total number of actuations under the configuration with this actuation scheme is the CDOF. However, what must be pointed out is that the actuation(s) should not be exerted to the joint when the newly increased terminal constraint can be transformed by the other actuation(s). Otherwise, the over constraint case will occur. When there are a lot of possible actuation schemes any one of which can be selected to set the actuators, the controllability of the manipulator is also affected by the actuation scheme's selection. For an instance, one can analyze the number of actuation(s) required to control the end-effector of the parallel manipulator shown in Fig. 1. Because the number of DOF of the end-effector is 1, it is reasonable for us to expect that the end-effector can be fully controlled only with one actuation. If one actuation is exerted to any joint of the mechanism, A_1 for example, it is not difficult to find that the end-effector still remains one translational DOF in the direction $e = (a_{12} \ b_{22} \ c_{32})^T$ when one repeats the above two steps in section 2.1. Therefore, one has to add another actuation to the mechanism. Of course, he can add the second actuation to any one of the rest joints. However, it is not difficult to prove that the end-effector will not be controlled unless the second actuation is exerted to the prismatic joint $B_i (i=1,2,3)$ under the condition that the first actuation is exerted to $A_i (i=1,2,3)$.

However, just as mentioned above, the new-added actuation should not be accepted if the newly-increased terminal constraint can be obtained by translating the former actuation(s). For an example, if the second actuation is assigned to the revolute joint B_1 , the newly-increased terminal constraints of the kinematic chain $A_1B_1C_1$ will be:

$$\mathcal{S}_1^n = (a_{11} \ b_{11} \ c_{11} \ y_{C_1}c_{11} - z_{C_1}b_{11} \ z_{C_1}a_{11} - x_{C_1}c_{11} \ x_{C_1}b_{11} - y_{C_1}a_{11})^T \quad (22)$$

Equation (22) is the transformation of the actuation exerted to the prismatic joint A_1 . So, the newly-added actuation is an over actuation for the actuation scheme whose first actuation is assigned to A_1 .

Of course, one can also exert the second actuation to the prismatic joint A_2 after assigning the first actuation to the prismatic joint A_1 . Again, one can find that the end-effector still has the free translation in the direction $e = (a_{12} \ b_{22} \ c_{32})^T$ when one repeats the above two steps in section 2.1. So, one can continue to add the third actuation to the prismatic joint A_3 . However, the end-effector will not be controlled until a fourth actuation is applied to one of the prismatic joints, B_1 , B_2 and B_3 . This forms a second actuation scheme. So, under this actuation scheme, the number of actuations needed to control the end-effector shown in Fig. 1 is 4.

The differences between the second actuation scheme and the first one are that the second one not only completely control the end-effector but also completely control every link in the manipulator. The selections of different actuation schemes can be well accomplished by a computer especially when the possible selections are numerous such as the one shown in Fig. 1. Unfortunately, this properties of a mechanism is ignored by the general mobility formulas.

3. The substantial mobility of underactuated, over actuated and equally actuated manipulators

A manipulator is said to be underactuated when the number of actuators in the manipulator is smaller than the number of degrees of freedom of the mechanism (Laliberté & Gosselin, 1998). When applied to mechanical fingers, the concept of underactuation leads to shape adaptation, i.e. underactuated fingers will envelope the objects to be grasped and adapt to their shape although each of the fingers is controlled by a reduced number of actuators (Laliberté & Gosselin, 1998). The concept of underactuation in robotic fingers—with fewer actuators than the degrees of freedom—allows the hand to adjust itself to an irregularly shaped object without complex control strategy and sensors (Birglen & Gosselin, 2006a). These underactuated manipulators arise in a number of important applications such as space robots, hyper redundant manipulators, manipulators with structural flexibility, etc (Jain & Rodriguez, 1993). The fact that the underactuated robotic fingers allow the hand to adjust itself to an irregularly shaped object makes it possible that no complex control strategy or numerous sensors are necessary in these manipulators (Birglen & Gosselin, 2006b). However, the over actuated mechanical systems often occur in biomechanical systems during the contact with ground and is recently introduced in redundantly actuated parallel robots. Yi and Kim (Yi & Kim, 2002) designed a singularity free load-distribution scheme for a redundantly actuated three-wheeled Omnidirectional mobile robot. The most outstanding advantage of the redundantly actuated mobile robot is that the singularities of the mechanism can be well avoided. Yiu and Li (Yiu & Li, 2003) investigated the trajectory generation for an over actuated parallel manipulator, in which there is one redundant actuator. Of course, the redundant actuator(s) and the required actuator(s) must obey a certain relationship determined by the mechanism, which will be discussed in section 3.2.

This section aims at clarifying the substantial relationships between the underactuated, over actuated and the equally actuated manipulators. The underactuated manipulator, which is also called under-determinate input system, means that the number of actuations provided is less than that is necessary; while the over actuated manipulator, which is also called redundant actuation or redundant input system, means that the number of actuations provided is larger than that is necessary. Equally actuated manipulator, which is also called fully actuated or determinate system, means that the actuations provided is equal to that is needed.

From the viewpoint of mechanisms, this classification of manipulators seems to be reasonable and has been widely used in engineering. However, it is not a properly scientific categorization for mechanisms. Therefore, this section will briefly study the substantial relationships between the underactuated, over actuated and equally actuated manipulators that are easily misunderstood in engineering applications.

3.1 The essence of the underactuated manipulator

To begin with this section, one might first investigate a famous inverted pendulum system shown in Fig. 2, which is also a representative, underactuated mechanical system. This inverted pendulum system is a planar two degrees of freedom catenation mechanical system. The vehicle can only make reciprocal translation along the x -axis and the pendulum can only rotate about the pivot attached to the moving vehicle.

In applications, only one actuation is provided to control the system, which seems to conflict with the definition of a fully actuated mechanism. In order to reveal the essence of this puzzling phenomenon, one might first turn to analyze the dynamics of this two-degree-of-freedom system.

Suppose the mass of the vehicle is denoted by M , the mass of the pendulum is m and the distance from the pivot attached to the vehicle to the mass center of the pendulum is l and the moment of inertia of the pendulum is denoted by J . The dynamics of the system can be immediately established via Lagrange method. The kinetic energy of the vehicle is:

$$T_v = \frac{1}{2} M \dot{x}^2$$

where T_v represents the kinetic energy of the vehicle.

The kinetic energy of the pendulum is:

$$T_p = \frac{1}{2} m \left[\frac{d}{dt}(x + l \sin \theta) \right]^2 + \frac{1}{2} m \left[\frac{d}{dt}(l \cos \theta) \right]^2 + \frac{1}{2} J \dot{\theta}^2$$

where T_p represents the kinetic energy of the pendulum.

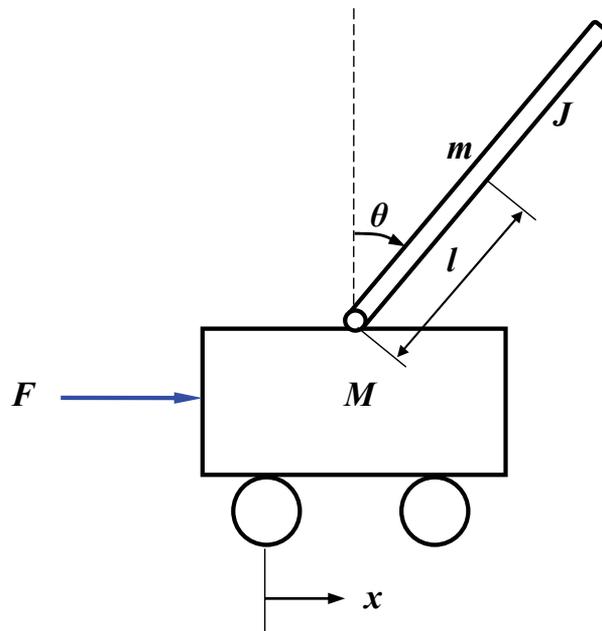


Fig. 2 a single inverted pendulum system

The total kinetic energy of the system is:

$$T = T_v + T_p = \frac{1}{2}(M + m)\dot{x}^2 + ml\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}(J + ml^2)\dot{\theta}^2$$

The potential energy of the system is:

$$V = mgl\cos\theta$$

Therefore, the Lagrange function of the system is:

$$L = \frac{1}{2}(M + m)\dot{x}^2 + ml\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}(J + ml^2)\dot{\theta}^2 - mgl\cos\theta \tag{23}$$

where L indicates the Lagrange function.

The dynamics equations for the two-degrees-of-freedom system shown in Fig. 2 can be expressed as:

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau \end{cases} \tag{24}$$

where τ represents the torque exerted to the revolute joint that connect the inverse pendulum and the vehicle.

In parallel manipulator, this chapter develops a general process to synthesize the manipulators with the specified mobility. The outstanding characteristics of the synthesis method are that the whole process is also analytical and each step can be programmed at a computer. Because of the restrictions of the traditional general mobility formulas for spatial mechanisms, a lot of mechanisms having special manoeuvrability might not be synthesized. However, any mechanism can be synthesized with this analytical theory of degrees of freedom for spatial mechanisms.

2. The valid means to investigate the mobility of a mechanism

The quick calculation approaches based on the algebra summations of the number of the links, joints and the constraints induced by the joints can not be completely perfected by itself. This is true even the analytical methods are applied in seeking the common constraints (Hunt, 1978)(Waldron, 1966)(Huang, 2006). These problems are becoming more and more obvious with the advent of spatial parallel manipulators. The primary considerations of the designers for the parallel manipulators have been focused on nothing but the mobility of the end-effector and its controllability. Therefore, the concept of general mobility of a mechanism should be divided into two basic concepts —the degree of freedom of the end-effector and the number of actuations needed to control the end-effector. With this regard, this chapter first introduces two primary definitions:

Definition 1:

The DOF of an end-effector totally characterizes the motions of the end-effector including the number, type and direction of the independent motions (Zhao et al, 2004a)(Zhao et al, 2006a).

Definition 2:

The configuration degree of freedom (CDOF) of a mechanism with an end-effector indicates the independent number of actuations required to uniquely control the end-effector under a configuration (Zhao et al, 2004b)(Zhao et al, 2006c).

Obviously, the DOF of an end-effector in number is not larger than 6 but the independent number of actuations required to uniquely control the end-effector might be any nonnegative integer. Bearing the above two definitions in mind, one can fall into two steps to investigate the mobility of a mechanism—the DOF of the end-effector and the CDOF of the mechanism with the prescribed end-effector. The former definition indicates the full instantaneous mobility properties of the end-effector through a mathematics concept of free mobility space while the later one presents the instantaneous controllability of the mechanism system. By definition 1, one can find that the DOF of an end-effector is only subjected to the constraint(s) exerted by the kinematic chain(s) connecting the end-effector with the fixed base or ground. Besides, the degree of freedom of the end-effector, instantaneously associated with the spatial configurations of the kinematic chain(s), should clearly depict the number, the direction and the type of the free motion of the end-effector instantaneously. Therefore, only analytical methods can fulfil such a task.

After obtaining the free motions of the end-effector, an engineering question will naturally arise—how many actuations are needed to control the end-effector? By definition 2, one can find that a checking process is given for verifying the controllability of the mechanism with the specified end-effector. Besides, this process can also allow the different selections of the actuation schemes, which is most adapted to the concept design of a manipulator.

Consequently, the valid means to investigate the mobility of mechanisms can be addressed as: (1) investigate the instantaneous DOF of the prescribed end-effector; and (2) investigate the number of actuations required to uniquely control the end-effector of the mechanism. For the instantaneous characteristics of the mobility of a mechanism, only analytical means is acceptable for such a task. Because of the elegance in depicting the relationship between the motions and the constraints, reciprocal screw theory does be a well selection to accomplish the task. Therefore, the following analytical model for the mobility of a parallel manipulator will be built up by applying the reciprocal screw theory.

According to reciprocal screw theory (Hunt, 1978)(Phillips, 1984)(Phillips, 1990)(Phillips et al, 1964)(Waldron, 1966)(Ball, 1900), a screw $\$$ is defined by a straight line with an associated pitch h and is conveniently denoted by six Plücker homogeneous coordinates:

$$\$ = \begin{bmatrix} s \\ s_0 + hs \end{bmatrix} \quad (1)$$

where s denotes direction ratios pointing along the screw axis, $s_0 = r \times s$ defines the moment of the screw axis about the origin of the coordinate system, r is the position vector of any point on the screw axis with respect to the coordinate system. Consequently, the

screw axis can be denoted by the Plücker homogeneous coordinates $\$_{axis} = \begin{bmatrix} s \\ s_0 \end{bmatrix}$.

Assume

$$\begin{cases} \mathbf{s} = (L \ M \ N)^T \\ \mathbf{s}_0 + h\mathbf{s} = (P \ Q \ R)^T \end{cases} \quad (2)$$

Considering $\mathbf{s} \cdot (\mathbf{s}_0 + h\mathbf{s}) = \mathbf{s} \cdot \mathbf{s}_0 + \|\mathbf{s}\|^2 h = \|\mathbf{s}\|^2 h$ and presuming $\mathbf{s} \neq \mathbf{0}$, one obtains the instant pitch of a screw:

$$h = \frac{\mathbf{s} \cdot (\mathbf{s}_0 + h\mathbf{s})}{\|\mathbf{s}\|^2} = \frac{LP + MQ + NR}{L^2 + M^2 + N^2} \quad (3)$$

Therefore, the axis of the screw can also be denoted as:

$$\mathcal{S}_{axis} = (L \ M \ N \ P - Lh \ Q - Mh \ R - Nh)^T \quad (4)$$

Assume that the vector of the projective point of the origin on the screw axis is represented by \mathbf{r}_{O_p} , there will be $\mathbf{s} \perp \mathbf{r}_{O_p}$ and:

$$\mathbf{s} \times (\mathbf{r}_{O_p} \times \mathbf{s}) = (\mathbf{s} \cdot \mathbf{s})\mathbf{r}_{O_p} - (\mathbf{s} \cdot \mathbf{r}_{O_p})\mathbf{s} = \|\mathbf{s}\|^2 \mathbf{r}_{O_p} \quad (5)$$

According to equation (4), for the screw axis, there are:

$$\begin{cases} \mathbf{s} = (L \ M \ N)^T \\ \mathbf{r}_{O_p} \times \mathbf{s} = (P - Lh \ Q - Mh \ R - Nh)^T \end{cases} \quad (6)$$

which yields:

$$\mathbf{r}_{O_p} = \frac{\mathbf{s} \times (\mathbf{r}_{O_p} \times \mathbf{s})}{\|\mathbf{s}\|^2} = \frac{1}{L^2 + M^2 + N^2} \begin{bmatrix} M(R - Nh) - N(Q - Mh) \\ N(P - Lh) - L(R - Nh) \\ L(Q - Mh) - M(P - Lh) \end{bmatrix} \quad (7)$$

Consequently, if the Plücker coordinates of a screw are given, one can easily obtain the unit direction vector, \mathbf{s} , the pitch, h , the screw axis and the vector of the projective point of the origin on the axis, \mathbf{r}_{O_p} , with equations (1) through (7).

If the pitch of a screw equals zero, the screw coordinates reduce to be:

$$\mathcal{S} = \begin{bmatrix} \mathbf{s} \\ \mathbf{s}_0 \end{bmatrix} \quad (8)$$

which is just the Plücker homogeneous coordinates of the screw axis.

In fact, formula (8) uniquely defines a line in a three-dimensional space. Assume that point O_p is the projective point of the origin on a line l and point A is any other point on the line. Then,

$$\mathbf{r}_A = \mathbf{r}_{O_p} + \mathbf{r}_{O_p A} = \mathbf{r}_{O_p} + \frac{a\mathbf{s}}{\|\mathbf{s}\|} \quad (9)$$

where s is a direction vector of line l , a is the length of line segment O_pA .

The moment of line l about the origin at point A will be:

$$s_0 = r_A \times s = \left(r_{O_p} + \frac{a s}{\|s\|} \right) \times s = r_{O_p} \times s \quad (10)$$

From equations (9) and (10), one obtains that the moment of a line about the origin is irrelevant to the point's selection on the line.

If a screw passes through the origin of the coordinate system, the screw coordinates can be denoted as:

$$\mathcal{S} = \begin{bmatrix} s \\ h s \end{bmatrix} \quad (11)$$

On the other hand, if the pitch of a screw is infinite, the screw is defined as:

$$\mathcal{S} = \begin{bmatrix} \mathbf{0} \\ s \end{bmatrix} \quad (12)$$

where $\mathbf{0} = (0 \ 0 \ 0)^T$ is a three dimensional vector.

According to the above definitions, a screw associated with a revolute pair is a twist of zero pitch pointing along the pair axis while a screw associated with a prismatic pair is a twist of infinite pitch pointing in the direction of the translational guide line of the pair.

From equation (11), one has known that the kinematic screw is often denoted in the form of Plücker homogeneous coordinates:

$$\mathcal{S} = (L \ M \ N \ P \ Q \ R)^T \quad (13)$$

where the first three components denote the angular velocity, the last three components denote the linear velocity of a point in the rigid body that is instantaneously coincident with the origin of the coordinate system.

Similarly, \mathcal{S}^r is defined as:

$$\mathcal{S}^r = \left(L^r \ M^r \ N^r \ P^r \ Q^r \ R^r \right)^T \quad (14)$$

where the first three components denote the resultant force and the last three components denote the resultant moment about the origin of the coordinate system.

Two screws, \mathcal{S} and \mathcal{S}^r , are called to be reciprocal if they satisfy the equation:

$$LP^r + MQ^r + NR^r + PL^r + QM^r + RN^r = 0 \quad (15)$$

Obviously, the free motions (general twists) \mathcal{S} and the prescribed constraints (general wrenches) \mathcal{S}^r of an equilibrium rigid body should satisfy equation (15). Equation (15) is often written for short (Kumar, 1992):

From the Lagrange function, one immediately obtains:

$$\frac{\partial L}{\partial \dot{x}} = (M + m)\dot{x} + ml\dot{\theta} \cos\theta, \quad \frac{\partial L}{\partial x} = 0$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml\dot{x} \cos\theta + (J + ml^2)\dot{\theta}, \quad \frac{\partial L}{\partial \theta} = ml(g - x\ddot{\theta}) \sin\theta$$

Substituting the above equations into equation (24), one has:

$$\begin{cases} (M + m)\ddot{x} - ml\ddot{\theta} \sin\theta + ml\dot{\theta}^2 \cos\theta = F \\ ml\ddot{x} \cos\theta + (J + ml^2)\ddot{\theta} - mgl \sin\theta = \tau \end{cases} \quad (25)$$

Of course, in the underactuated condition, there is $\tau = 0$. The first formula in equation set (25) is the apparent actuation formula while the second one in equation set (25) is a hidden relationship of the mechanical system, in which the gravity, the inertia force and moment of the pendulum are associated precisely. As a matter of fact, therefore, this relationship depicted by the second formula in equation set (25) provided another actuation constraint for the two-degrees-of-freedom mechanical system in dynamics but not in statics. Therefore, the mechanical system shown in Fig. 2 is fully actuated in dynamics but not in statics. When

$\ddot{x} = 0$ and $\tau = 0$, equation set (25) can be simplified as:

$$\dot{\theta} = \pm \sqrt{\frac{mgl}{J + ml^2} \cos\theta - \frac{F}{ml \sin\theta}} \quad (26)$$

where “ \pm ” is determined by the initial condition of the system and the sign should be “+” in the case shown in Fig. 2.

From equations (25) and (26), it is not difficult to find that the inverted pendulum system shown in Fig. 2 can only keep a dynamic equilibrium but not a static equilibrium which is a primary requirement for a mechanism.

A much more familiar example is the differential gear train mechanism used in the driving axle of all kinds of automobiles. The basic mechanism structure is shown in Fig. 3. The pinion gear transforms the torque from the engine to the driving axle shafts by a differential gear train mechanism, in which the ring gear shown in Fig. 3 acts as an actuator and the right and the left shafts act as executors.

Obviously, this mechanism also has two degrees of freedom. However, the actuation is just one rotational input from the pinion gear. One might draw a conclusion in haste that this mechanism should be an outstanding representative example for the applications of underactuated mechanical systems because it is so widely used in the modern vehicles. This mechanical system is really quite different from the inverted pendulum system shown in Fig. 2 in that the hidden mechanical constraint or “actuation” is more easily ignored. The *reaction difference* between the right and left wheels from the road surface provides such an

“actuation”, which is apparent when the reactions to the right and left wheels from the road surface are different, and which often occurs when the vehicle makes a right or left turn.

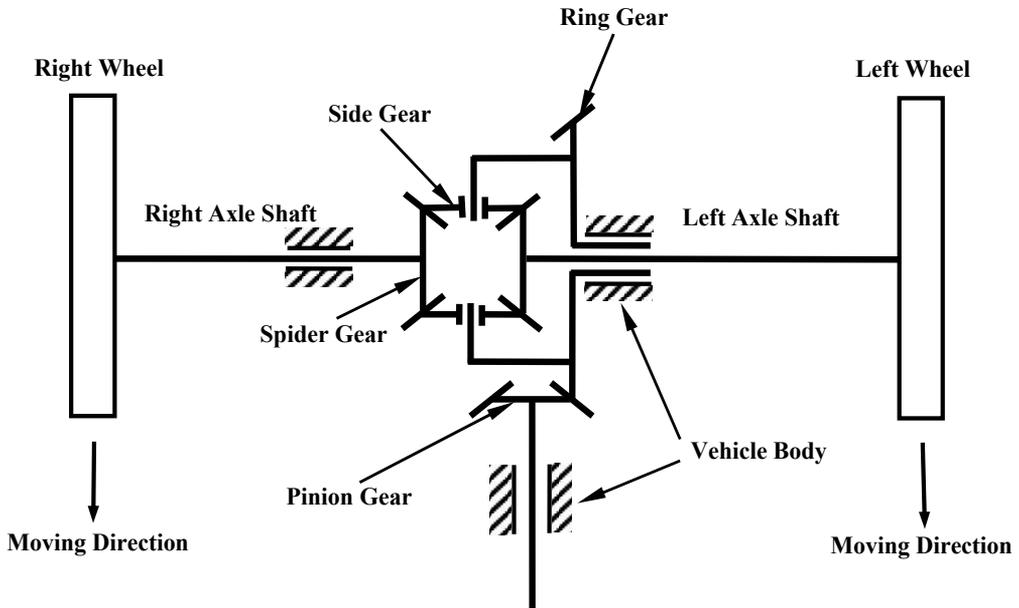


Fig. 3 the differential gear train mechanism

Another facility usually used in civil engineering is the inertial rammer shown in Fig. 4. This can also be modelled with a planar mechanism shown in Fig. 5. The apparent actuation is provided by the eccentric force of the eccentric rotor under the actuation of the electric motor. However, the motion of the rammer’s body is indeterminate if the control of F_h is not exerted to the handle. Therefore, the inertia rammer is not an underactuated mechanical system but a fully actuated system although the apparent actuation seems to be restricted to the eccentric force resulting from the eccentric rotating rotor.

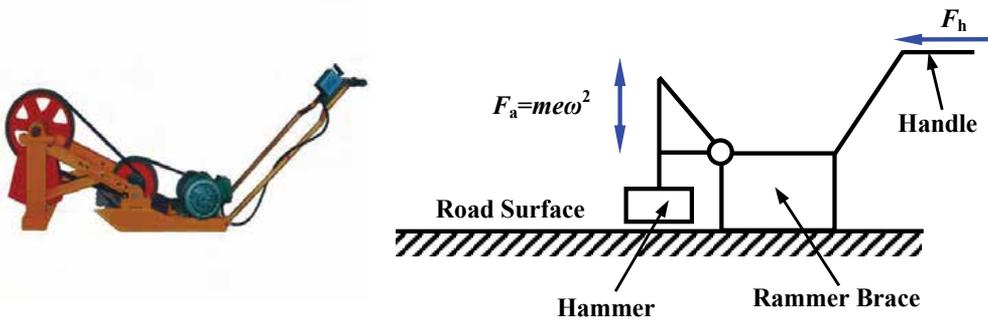


Fig. 4 the inertia rammer

Fig. 5 the mechanism of the inertia rammer

From the above analysis, it is not difficult to find that all the underactuated mechanical systems are substantially actuation determinate from the viewpoint of mechanisms.

3.2 The essence of the over actuated manipulator

Over-determinate actuation manipulators also witnesses wide applications in mechanical engineering, especially in biomechanical engineering. In order to investigate the essence of these manipulators, this section addresses this problem via some mechanism examples.

As a simple example, one might first investigate the motion of a vehicle with one degree of freedom under the actions of two persons shown in Fig. 6. The vehicle can only translate forward and backward along the road direction. However, two different actions are exerted to both sides of the vehicle. So, it is an over actuated mechanical system.

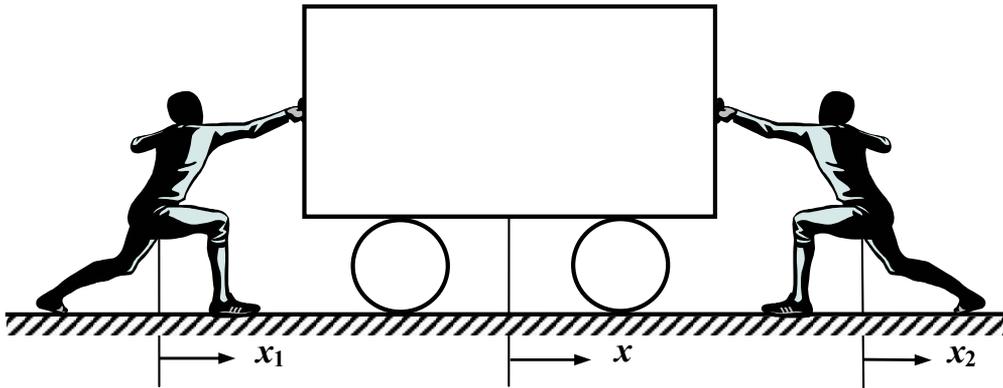


Fig. 6 an over actuated mechanical system

Out of question, the vehicle shown in Fig. 6 will move along the direction of the resultant force of the two persons, in spite of which the two actuations are not independent because these two actuations should satisfy that $x_1 = x_2 = x$. Otherwise, the two actuations might not do continuous work to the vehicle. These additional constraints are also called compliant equations.

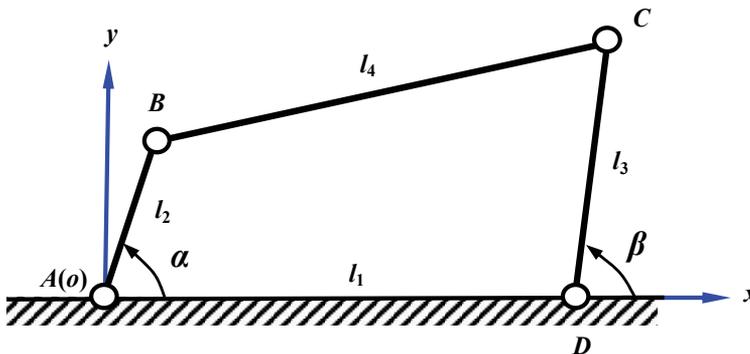


Fig. 7 a planar four-bar mechanism with two actuations

Next, one can consider a planar four-bar mechanism under two actuations shown in Fig. 7. Obviously, only one actuation is needed to control the mechanism. In engineering applications, however, it is also available to exert two actuations to increase the input torque or force to drive the mechanism to output a larger power. Therefore, the mechanism in such a case is a representative of the over actuated mechanical systems.

Following, one can investigate the problems that might be ignored or misunderstood. For the sake of conveniences, a coordinate system is established by setting the origin to superimpose with revolute joint A and x -axis along the link AD and y -axis perpendicular upward to link AD . If the planar four-bar mechanism has a determinate motion, the equation below should hold:

$$(l_3 \cos \beta + l_1 - l_2 \cos \alpha)^2 + (l_3 \sin \beta - l_2 \sin \alpha)^2 = l_4^2 \quad (27)$$

Therefore, differentiating equation (27) with respect to time and rearranging yields:

$$\frac{\dot{\beta}}{\dot{\alpha}} = \frac{l_2 [l_1 \sin \alpha + l_3 \sin(\alpha - \beta)]}{l_3 [l_1 \sin \beta + l_2 \sin(\alpha - \beta)]} \quad (28)$$

where $\dot{\alpha}$ and $\dot{\beta}$ represent the angular velocities of the crank AB and the rocker DC shown in Fig. 7, individually.

Therefore, the actuations exerted to the crank AB and the rocker DC should keep in a precise relationship specified by equation (27). Otherwise, the link BC might be cracked due to the increasing internal forces. Equation (27) or (28) is the compliant equation for the over actuated manipulator shown in Fig. 7.

Consequently, it is not difficult to find that there always are compliant constraint equations for the over actuated mechanical systems. And therefore, these mechanical systems are also substantially equally actuated.

3.3 The problems to be noted in engineering applications

The dexterity of an underactuated manipulator differs from the dexterity of a fully actuated one, even if their mechanical structures are identical. Therefore, the underactuated mechanical systems are widely used in the cases for fault tolerance and energy saving purposes. From the above analysis, one knows that any mechanical system that has a determinate motion should be an equally actuated system in essence. Next, one investigates an underactuated mechanical finger with return actuation shown in Fig. 8.

This mechanism is used in the finger of the United States patent initially applied by Gosselin et al (Gosselin & Laliberté, 1998) for dexterity hand in 1998. The primary structure of the mechanism shown in Fig. 8 (Birglen & Gosselin, 2006a) is a planar four-bar mechanism. Links AB and AD are simultaneously pivoted with the fixed wrist. Links AB and BC are connected by a passive spring. Next, the mobility of the mechanism will be investigated in several cases.

Firstly, when the finger does not contact any object, the links AB and BC connected by a passive spring might be disposed as one link, and therefore, $ABCDE$ forms one link and rotates about the fixed pivot, A , under the actuation of the force F . When AB contacts a target object, the link AB will degenerate to an unmovable base attached to the wrist, and therefore, the spring will be deformed under the action of the force F and finger $ABCDE$ forms a real four-bar mechanism. This will be holding until the side BE also touches the boundary of the target object, after which the continuous increasing of the force F will only result in the deforming of the target object.

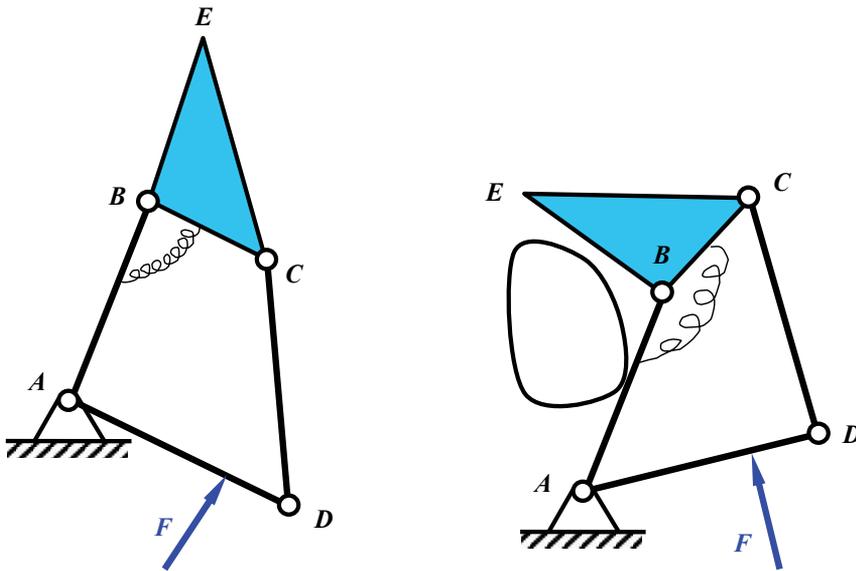


Fig. 8 underactuated mechanical finger with return actuation

The above analysis indicates that the so called underactuated mechanical finger is equally or fully actuated at any instant from the viewpoint of mechanisms. Consequently, no matter what kind does a mechanism belong to, it should have a determinate motion and equal actuation(s) at any instant, which should be particularly noticed in the concept design of underactuated mechanical systems. Theoretical and example analysis indicate that the underactuated, over actuated and fully actuated mechanical systems are all substantially equally actuated mechanisms.

4. Synthesis of a spatial parallel manipulator with a specified mobility

Usually, suspension is a general term of the equipments transforming forces and moments from the wheel to the vehicle body. Its primary function is to determine the geometry of the wheel motion during jounce and rebound, and to withstand forces and moments on the suspension in accelerating motion (Raghavan, 1996). The ride and handling characteristics of a vehicle are heavily dependent on the kinematic and compliance properties of the suspension mechanism (Raghavan, 2005). Compared with dependent suspensions, independent suspensions can eliminate undesirable dynamic phenomena such as shimmy and caster wobble resulting from wheel coupling in solid-axle suspensions (Raghavan, 1996). The most common independent suspension mechanisms utilized in automobiles are short-long arm suspension (Suh, 1998), the MacPherson strut (Raghavan, 2005), the multilink suspensions (Simionescu, 2002), and the short-long arm front suspension with a true kingpin (Murakami, 1989), etc. Most automotive independent suspension mechanisms are single degree-of-freedom mechanisms with the predominant motion being wheel jounce and rebound. In order to allow the wheel to pass the uneven terrain without slipping, Chakraborty and Ghosal (Chakraborty & Ghosal, 2004) investigated the kinematics of a wheeled mobile robot moving on uneven terrain by modeling the wheels as a torus and proposing a lateral passive joint. Applications indicate that the wheel orientation and

position parameters such as kingpin, caster, camber, toe change, axes distance, and the wheel track are primary consideration in the design of suspension mechanism. These parameters, as a matter of fact, are dependent on the wheel jounce and rebound, an independent parameter (Raghavan, 2005).

Therefore, a particular rigid guidance mechanism whose end-effector only has one straight line translation should maintain the orientation and position parameters invariable. Yan and Kuo (Yan & Kuo, 2006) addressed the topological representations and characteristics of variable kinematic joints, which might be utilized in spatial mechanism synthesis. By considering workspace, dexterity, stiffness and singularity avoidance, Arsenault and Boudreau (Arsenault & Boudreau, 2006) discussed the synthesis problems of planar parallel mechanisms. In the history of mechanism synthesis, a significant example is that the creation of linkages to produce exact straight line motion was an important engineering as well as a mathematical problem of the 19th century (Kempe, 1877). While many engineers and mathematicians were searching for a 4- 5- or 6-bar straight line linkage all suffered from the fact that they could not attain such a motion in the middle of 19th century, Peaucellier investigated an eight bar linkage shown in Fig. 9 and discovered he could generate an exact straight line motion from a rotary input.

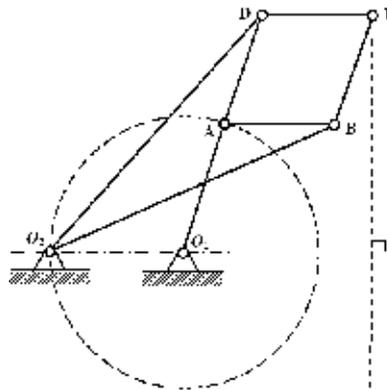


Fig. 9 Structure of Peaucellier-Lipkin Eight-Bar Linkage

This invention was recognized by several mathematicians as being very important to the design of general mathematical calculators (Kempe, 1877). This eight-link linkage was the one of the first to produce exact straight line motion and was independently invented by a French engineer named Peaucellier and by a Russian mathematician named Lipkin (Kempe, 1877), which is therefore often called Peaucellier-Lipkin eight-link linkage.

However, Peaucellier-Lipkin linkage is mostly utilized as a motion generator but not a rigid guidance mechanism. Obviously, because of its complexity, such a mechanism can not be used as a suspension in spite of the fact that it can really make the wheel move in a straight line during jounce and rebound. Therefore, this section first discusses the synthesis processes with the analytical model of the instantaneous mobility of a manipulator for the rigid guidance mechanism with the specified mobility; and then presents a rectilinear motion generating manipulator that can be utilized as a suspension mechanism.

The general synthesis process might be:

Step 1: Express the free motions required for the prescribed end-effector in Plücker coordinates at a Cartesian coordinate system.

The Plücker coordinates of the specified motions should be firstly expressed in a Cartesian coordinate system. This chapter supposes that the twists of the free motion(s) of the end-effector are denoted by \mathcal{S}_{End}^F .

Step 2: Solve the constraint(s) exerted to the end-effector by its kinematic chain(s). According to reciprocities between free motion(s) and constraint(s) of an end-effector, the constraint(s) applied to the end-effector can be solved with the equation (16):

$$\left(\mathcal{S}_{End}^F\right)^T E \mathcal{S}_{End}^C = 0 \tag{29}$$

where \mathcal{S}_{End}^F indicates the specified free motion(s) of the end-effector and \mathcal{S}_{End}^C denotes any constraint applied to the end-effector.

Step 3: Decide the number of kinematic chains, $m(m \geq 1)$, that will be used to connect the end-effector with the fixed base.

If every link in the chain is connected to at least two other links, the chain forms one or more closed loops and is called a closed kinematic chain; if not, the chain is referred to as open (Shigley & Uicker, 1980). For the later open chain case, the synthesis is simply stated as: any kinematic chain is feasible if the twist basis, $\mathcal{S}_{B_i}^F$, of the chain contains \mathcal{S}_{End}^F . However, the following steps should be further discussed if the mechanism is a closed one.

Step 4: Synthesize the terminal constraint(s) of each kinematic chain.

Suppose

$$\mathcal{S}_{End}^C = \left[\mathcal{S}_{End}^{C_1} \quad \mathcal{S}_{End}^{C_2} \quad \cdots \quad \mathcal{S}_{End}^{C_n} \right] \tag{30}$$

where n indicates the dimension of the constraint basis of the end-effector.

Suppose that the terminal constraint(s) of the i th ($i = 1, 2, \dots, m$) kinematic chain is denoted by \mathcal{S}_i^C , the terminal constraint(s) of the chain might be synthesized with:

$$\mathcal{S}_i^C = \mathcal{S}_{End}^C K_i \tag{31}$$

where $K_i = \left[K_{i1} \quad K_{i2} \quad \cdots \quad K_{in_i} \right]$, and $K_{ij} = \left(k_{i1} \quad k_{i2} \quad \cdots \quad k_{in} \right)^T$ and $j = 1, 2, \dots, n_i$.

For a feasible mechanism that makes the end-effector only have the prescribed free motion(s), the necessary and sufficient criterion is that the resultant terminal constraint(s) of all these m kinematic chain(s), $\bigcup_{i=1}^m \mathcal{S}_i^C$, should be equivalent to \mathcal{S}_{End}^C . **This is called the**

construction criterion 1 of the feasible kinematic chains.

The necessity and sufficiency of this criterion can be immediately deduced from equation (29) with linear algebra theory.

Step 5: Solve the twist basis of the i th kinematic chain with the terminal constraints, \mathcal{S}_i^C , synthesized in step 4.

With reciprocal screw theory, a basis of the twist(s) of the i th kinematic chain, denoted by $\mathcal{S}_{B_i}^F$, can be obtained by solving the following equation:

$$\left(\mathcal{S}_i^C \right)^T E \mathcal{S}_{B_i}^F = 0 \quad (32)$$

where \mathcal{S}_i^C represents the terminal constraint(s) of the i th kinematic chain synthesized in step 4.

Step 6: Synthesize the twist(s) of the i th kinematic chain with the twist basis of the i th chain, $\mathcal{S}_{B_i}^F$, obtained in step 5.

Suppose

$$\mathcal{S}_{B_i}^F = \left[\mathcal{S}_{B_i}^{F_1} \quad \mathcal{S}_{B_i}^{F_2} \quad \dots \quad \mathcal{S}_{B_i}^{F_{n_i}} \right] \quad (33)$$

where n_i indicates the dimension of the twist basis of the i th ($i=1,2,\dots,m$) kinematic chain. According to linear algebra, any twist of the i th kinematic chain can be expressed as the linear combinations of the twist basis of the chain:

$$\mathcal{S}_i^{F_a} = \mathcal{S}_{B_i}^F C \quad (34)$$

where $C = (c_1 \quad c_2 \quad \dots \quad c_{n_i})^T$.

Consequently, the twists of each kinematic chain can be synthesized through equation (34). However, in order to keep the twists of the i th chain to be equivalent to the twist basis of the chain, the rank of the total twists synthesized through equation (34) should equal the dimension of the twist basis of the chain. **This is called the construction criterion 2 of the feasible kinematic chains.**

The necessary and sufficient of this criterion can be immediately obtained from equation (32).

According to the construction criteria 1 and 2, the required synthesis target of a mechanism can be gradually accomplished with the above six steps. Obviously, with these six steps, different person might synthesize different kinematic chains and different mechanisms. However, all the end-effectors of the mechanisms synthesized with the same criteria will surely have the identical specified free motion(s).

The next section will apply these steps to synthesize a rigid guidance mechanism that can be utilized as a suspension of an automobile.

The synthesis target now is to use the least number of links and pure revolute joints to design a mechanism whose end-effector has one pure translation along an exact straight line; therefore, the mechanism must be a closed one. The reason is that it will need at least two actuations to generate a pure straight line translation with an open chain mechanism. And therefore, for the purpose of the suspension required, one at least needs two kinematic chains to generate a pure straight line translation with one actuation input. According to step 1, the specified free motion of the end-effector should be expressed in a Cartesian

coordinate system. Without loss of generality, the precise straight line translation of the end-effector can be assumed to parallel z -axis. Therefore, the free motion can be described in Plücker coordinates as:

$$\mathfrak{S}_{End}^F = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T \tag{35}$$

So, the target now can be depicted as whether one can find two sets of screws whose pitches represented by equation (3) are all zeros provided that they were all reciprocal to \mathfrak{S}_{End}^F of equation (35).

According to step 2, substituting equation (35) into equation (29) yields the constraints exerted to the end-effector, \mathfrak{S}_{End}^C :

$$\mathfrak{S}_{End}^C = \left[\begin{matrix} \mathfrak{S}_{End}^{C_1} & \mathfrak{S}_{End}^{C_2} & \mathfrak{S}_{End}^{C_3} & \mathfrak{S}_{End}^{C_4} & \mathfrak{S}_{End}^{C_5} \end{matrix} \right] \tag{36}$$

where $\mathfrak{S}_{End}^{C_1} = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$ represents a force along x -axis, $\mathfrak{S}_{End}^{C_2} = (0 \ 1 \ 0 \ 0 \ 0 \ 0)^T$ represents a force along y -axis, $\mathfrak{S}_{End}^{C_3} = (0 \ 0 \ 0 \ 1 \ 0 \ 0)^T$ represents a torque about x -axis, $\mathfrak{S}_{End}^{C_4} = (0 \ 0 \ 0 \ 0 \ 1 \ 0)^T$ represents a torque about y -axis, and $\mathfrak{S}_{End}^{C_5} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ represents a torque about z -axis.

From equation (32), it is not difficult to find that the sum of the number of the independent twists and the number of the terminal constraints of a chain is six. In order to reduce the number of revolute joints, one might have to increase the number of the terminal constraints of the chains as many as possible. According to equations (31) and (36), the maximum number of the terminal constraints of a chain is five. However, if such a structure scheme is used, one may find each kinematic chain only consists of one revolute joint, which is unfeasible in reality. Similarly, it is not difficult to find that only when each kinematic chain provides three terminal constraints at most, can the structure scheme is feasible.

With equation (31), one can synthesize the terminal constraints of these two kinematic chains, individually. Selecting different $k_i (i=1,2,\dots,5)$ and substituting them into equation (31), one can synthesize three independent terminal constraints for the first kinematic chain, for example:

Assuming $K_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, one obtains

$$\mathfrak{S}_1^C = \left[\begin{matrix} \mathfrak{S}_1^{C_1} & \mathfrak{S}_1^{C_2} & \mathfrak{S}_1^{C_3} \end{matrix} \right] \tag{37}$$

where $\mathfrak{S}_1^{C_1} = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$ indicates a force along x -axis, $\mathfrak{S}_1^{C_2} = (0 \ 0 \ 0 \ 0 \ 1 \ 0)^T$ indicates a torque about y -axis, and $\mathfrak{S}_1^{C_3} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ indicates a torque about z -axis.

Assuming $K_2 = \begin{bmatrix} a & 0 & 0 \\ b & 0 & 0 \\ 0 & b & 0 \\ 0 & -a & 0 \\ 0 & 0 & 1 \end{bmatrix}$, one can obtain

$$\mathfrak{S}_2^C = \begin{bmatrix} \mathfrak{S}_2^{C_1} & \mathfrak{S}_2^{C_2} & \mathfrak{S}_2^{C_3} \end{bmatrix} \tag{38}$$

where $\mathfrak{S}_1^{C_1} = (a \ b \ 0 \ 0 \ 0 \ 0)^T$ denotes a force along the direction $(a \ b \ 0)^T$, $\mathfrak{S}_1^{C_2} = (0 \ 0 \ 0 \ b \ -a \ 0)^T$ denotes a torque about the direction $(b \ -a \ 0)^T$, $\mathfrak{S}_1^{C_3} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ denotes a torque about z -axis and $ab \neq 0$.

Because $\dimspan\{K_1, K_2\} = 5$, the resultant terminal constraints of these 2 kinematic chains, $\bigcup_{i=1}^2 \mathfrak{S}_i^C$ must be equivalent to \mathfrak{S}_{End}^C . So the construction criterion 1 is satisfied.

According to equation (32), one immediately obtains the twist bases for the two kinematic chains with equations (37) and (38):

$$\mathfrak{S}_{B_1}^F = \begin{bmatrix} \mathfrak{S}_{B_1}^{F_1} & \mathfrak{S}_{B_1}^{F_2} & \mathfrak{S}_{B_1}^{F_3} \end{bmatrix} \tag{39}$$

where $\mathfrak{S}_{B_1}^{F_1} = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$ represents a rotation about x -axis, $\mathfrak{S}_{B_1}^{F_2} = (0 \ 0 \ 0 \ 0 \ 1 \ 0)^T$ represents a translation along y -axis, $\mathfrak{S}_{B_1}^{F_3} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ represents a translation along z -axis, and

$$\mathfrak{S}_{B_2}^F = \begin{bmatrix} \mathfrak{S}_{B_2}^{F_1} & \mathfrak{S}_{B_2}^{F_2} & \mathfrak{S}_{B_2}^{F_3} \end{bmatrix} \tag{40}$$

where $\mathfrak{S}_{B_2}^{F_1} = (\cos\alpha \ \sin\alpha \ 0 \ 0 \ 0 \ 0)^T$ denotes a rotation about the direction $(\cos\alpha \ \sin\alpha \ 0)^T$, $\mathfrak{S}_{B_2}^{F_2} = (0 \ 0 \ 0 \ -\sin\alpha \ \cos\alpha \ 0)^T$ denotes a translation along the direction $(-\sin\alpha \ \cos\alpha \ 0)^T$, $\mathfrak{S}_{B_2}^{F_3} = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ denotes a translation along z -axis, and $\cos\alpha = \frac{a}{\sqrt{a^2 + b^2}}$ and $\sin\alpha = \frac{b}{\sqrt{a^2 + b^2}}$.

According to step 6, one can synthesize the twists of the two kinematic chains with their twist bases (39) and (40), individually. Considering the construction criterion 2, one can find

that the least number of twists in each kinematic chain is three. Therefore, the twist of the first kinematic chain can be synthesized below with equation (34):

$$\mathfrak{S}_1^{Fa} = c_1 \mathfrak{S}_{B_1}^{F_1} + c_2 \mathfrak{S}_{B_1}^{F_2} + c_3 \mathfrak{S}_{B_1}^{F_3} = (c_1 \ 0 \ 0 \ 0 \ c_2 \ c_3)^T \quad (41)$$

Substituting equation (41) into equation (3) yields:

$$h_1^{Fa} = 0 \quad (42)$$

Equation (42) indicates that any twist having the form of equation (41) will naturally satisfy the free motion requirements of the end-effector. The Cartesian coordinates of the joint, r_{A_1} , can be found from equations (7) and (9):

$$r_{A_1} = \frac{\mathbf{s} \times \mathbf{s}_0}{\|\mathbf{s}\|^2} + \frac{a \mathbf{s}}{\|\mathbf{s}\|} = \left(a \quad -\frac{c_3}{c_1} \quad \frac{c_2}{c_1} \right)^T$$

To make the twists of the chain be equivalent to the twist basis, there are at least three twists indicated in the form of equation (41).

Suppose $c_1 = 1$ and the three joints' coordinates are

$$\begin{cases} r_A = (a \ 0 \ 0)^T \\ r_B = (a \ y_B \ z_B)^T \\ r_C = (a \ y_C \ z_C)^T \end{cases}$$

then, the twists of the first kinematic chain will be:

$$\mathfrak{S}_{ABC} = \left[\mathfrak{S}_1^{F_A} \quad \mathfrak{S}_1^{F_B} \quad \mathfrak{S}_1^{F_C} \right] \quad (43)$$

where $\mathfrak{S}_1^{F_A} = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$ represents a rotation about x -axis, $\mathfrak{S}_1^{F_B} = (1 \ 0 \ 0 \ 0 \ z_B \ -y_B)^T$ represents a rotation about a line passing through point $(x_B \ y_B \ z_B)$ and paralleling x -axis, and $\mathfrak{S}_1^{F_C} = (1 \ 0 \ 0 \ 0 \ z_C \ -y_C)^T$ represents a rotation about a line passing through point $(x_C \ y_C \ z_C)$ and paralleling x -axis.

According to equation (34), a twist of the second kinematic chain, denoted by \mathfrak{S}_2^{Fa} , can be expressed as:

$$\mathfrak{S}_{B_2}^{Fa} = \eta_1 \mathfrak{S}_{B_2}^{F_1} + \eta_2 \mathfrak{S}_{B_2}^{F_2} + \eta_3 \mathfrak{S}_{B_2}^{F_3} = (\eta_1 \cos \alpha \quad \eta_1 \sin \alpha \quad 0 \quad -\eta_2 \sin \alpha \quad \eta_2 \cos \alpha \quad \eta_3)^T \quad (44)$$

where η_i denote real numbers and $i = 1, 2, 3$.

Substituting equation (44) into equation (3) yields:

$$h_2^{Fa} = 0 \quad (45)$$

Equation (45) indicates that any twist having the form of equation (44) will naturally satisfy the free motion requirements of the end-effector.

The Cartesian coordinates of the joint, \mathbf{r}_{A_2} , can be found from equations (7) and (9):

$$\mathbf{r}_{A_2} = \frac{\mathbf{s} \times \mathbf{s}_0}{\|\mathbf{s}\|^2} + \frac{b \mathbf{s}}{\|\mathbf{s}\|} = \begin{pmatrix} \frac{\eta_3}{\eta_1} \sin \alpha + b \cos \alpha & -\frac{\eta_3}{\eta_1} \cos \alpha + b \sin \alpha & \frac{\eta_2}{\eta_1} \end{pmatrix}^T$$

To keep the twists of the chain be equivalent to the twist basis, one can only select three independent twists indicated with equation (44) by selecting three sets of $(\eta_1 \ \eta_2 \ \eta_3)$.

If one supposes $\eta_1 = 1$, $\eta_2 = z_F$, $\eta_3 = x_F \sin \alpha - y_F \cos \alpha$ and $b = x_F \cos \alpha + y_F \sin \alpha$, he obtains the coordinates of revolute joint F , $\mathbf{r}_F = (x_F \ y_F \ z_F)^T$; similarly, if one supposes $\eta_1 = 1$, $\eta_2 = z_E$, $\eta_3 = x_E \sin \alpha - y_E \cos \alpha$ and $b = x_E \cos \alpha + y_E \sin \alpha$, he obtains the coordinates of revolute joint E , $\mathbf{r}_E = (x_E \ y_E \ z_E)^T$; and if one supposes $\eta_1 = 1$, $\eta_2 = z_D$, $\eta_3 = x_D \sin \alpha - y_D \cos \alpha$ and $b = x_D \cos \alpha + y_D \sin \alpha$, he can obtain the coordinates of revolute joint D , $\mathbf{r}_D = (x_D \ y_D \ z_D)^T$. Therefore, the three joints' coordinates can be assumed

$$\begin{cases} \mathbf{r}_F = (x_F \ y_F \ z_F)^T \\ \mathbf{r}_E = (x_E \ y_E \ z_E)^T \\ \mathbf{r}_D = (x_D \ y_D \ z_D)^T \end{cases}$$

then, the twists of the second kinematic chain will be:

$$\mathcal{S}_{FED} = \left[\mathcal{S}_2^{F_F} \ \mathcal{S}_2^{F_E} \ \mathcal{S}_2^{F_D} \right] \quad (46)$$

where $\mathcal{S}_2^{F_F} = (\cos \alpha \ \sin \alpha \ 0 \ 0 \ 0 \ 0)^T$ represents a rotation about a line passing through the origin of the coordinate system and in the direction $(\cos \alpha \ \sin \alpha \ 0)^T$, $\mathcal{S}_2^{F_E} = (\cos \alpha \ \sin \alpha \ 0 \ -z_E \sin \alpha \ z_E \cos \alpha \ x_E \sin \alpha - y_E \cos \alpha)^T$ represents a rotation about a line passing through point $(x_E \ y_E \ z_E)$ and in the direction $(\cos \alpha \ \sin \alpha \ 0)^T$, and $\mathcal{S}_2^{F_D} = (\cos \alpha \ \sin \alpha \ 0 \ -z_D \sin \alpha \ z_D \cos \alpha \ x_D \sin \alpha - y_D \cos \alpha)^T$ represents a rotation about a line passing through point $(x_D \ y_D \ z_D)$ and in the direction $(\cos \alpha \ \sin \alpha \ 0)^T$.

With equations (43) and (46), one can synthesize a spatial six link mechanism $ABCDEF$ shown in Fig. 10. It is not difficult to find that α is the angle from x -axis to the y' -axis of the revolute joint F and the revolute joints F , E and D have the same axis direction, which is denoted by $\mathbf{n}_{FED} = (\cos \alpha \ \sin \alpha \ 0)^T$.

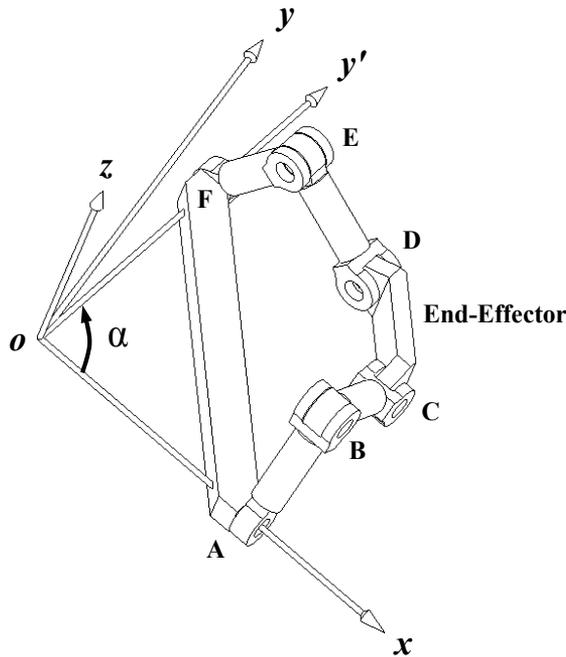


Fig. 10 a Spatial Six-Link Mechanism with a Straight line Translational End-Effector

From the above analysis, it is not difficult to find that the two kinematic chains ABC and FED can surely guarantee the pure straight line translation of the end-effector CD so long as \mathcal{S}_{ABC} and \mathcal{S}_{FED} do not descend in ranks. To analyze the sensitivity of the structure stability to the angle α , one should turn to the equations (37) and (38) and investigate the resultant terminal constraints, which can be expressed with:

$$\mathcal{S}_{CD}^C(\alpha) = \begin{bmatrix} 1 & \cos\alpha & 0 & 0 & 0 \\ 0 & \sin\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin\alpha & 0 & 0 \\ 0 & 0 & -\cos\alpha & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

where $\cos\alpha = \frac{a}{\sqrt{a^2 + b^2}}$ and $\sin\alpha = \frac{b}{\sqrt{a^2 + b^2}}$.

If the terminal constraints denoted by $\mathcal{S}_{CD}^C(\alpha)$ are well conditioned, the mechanism will have fine structure stability. From equations (29) and (47), one can find that the end-effector will have one straight line translation along z -axis so long as $rank(\mathcal{S}_{CD}^C(\alpha)) = 5$, which can be immediately transformed to investigate the following sub matrix of $\mathcal{S}_{CD}^C(\alpha)$:

$$A(\alpha) = \begin{bmatrix} 1 & \cos\alpha & 0 & 0 & 0 \\ 0 & \sin\alpha & 0 & 0 & 0 \\ 0 & 0 & \sin\alpha & 0 & 0 \\ 0 & 0 & -\cos\alpha & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Letting $\det(A(\alpha))=0$, one immediately obtains $\alpha=0$ or $\alpha=\pi$. Therefore, in order to keep the end-effector CD have one straight line translation along z -axis, there will be $\alpha \neq 0$ and $\alpha \neq \pi$. So, the rigid guidance mechanism synthesized in this chapter has a wider adaptation of angle between the planes of its two kinematic chains. Now, the sensitivity of the structure stability to the angle α of the mechanism can be judged by the condition number of matrix $A(\alpha)$ (Kelley, 1995). Let

$$cond(A)_2 = \|A\|_2 \|A^{-1}\|_2 = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}} = 1 \tag{48}$$

where $cond(A)_2$ indicates the condition number of matrix A , $\|A\|_2$ indicates the 2-norm of matrix A , and $\lambda(A^T A)$ indicates the eigenvalues of matrix $A^T A$.

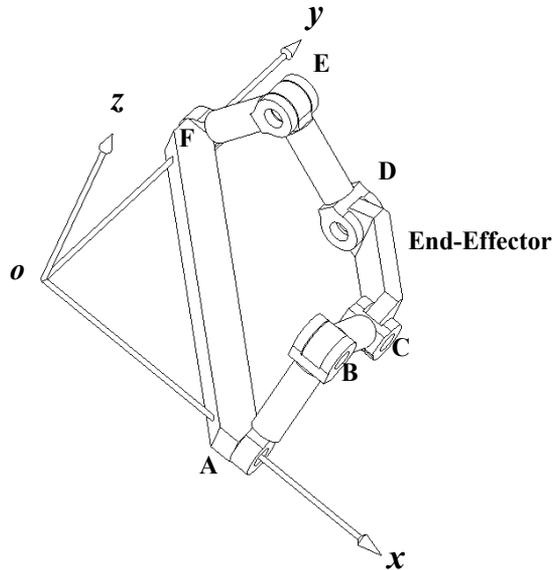


Fig. 11 a Spatial Six-Link Mechanism with the Best Structure Stability

The solution of equation (48) is:

$$\alpha = \frac{\pi}{2} \tag{49}$$

Equation (49) indicates that the mechanism will have the best structure stability when $\alpha = \frac{\pi}{2}$, which is shown in Fig. 11. Compared with Peaucellier-Lipkin eight-link linkage, the spatial six-link mechanism synthesized in this chapter has the least links and revolute joints, and the whole end-effector CD can make an exact straight line translation while Peaucellier-Lipkin eight-link linkage can only allow one specified point to make such a motion. As a matter of fact, the mechanism shown in Fig. 11 is a Sarrus linkage. However, the mechanism proposed here does not necessarily require that the two kinematic chains must within two orthogonal planes which are needed for Sarrus linkage. The so-called Sarrus linkage, which is shown in Fig. 12, is a linkage that converts circular motion to linear motion by using hinged squares. The square end-effector C_1C_2 can make an exact straight line translation along z -axis which shows better properties both in mechanical structure and in kinematics than those of Peaucellier-Lipkin eight-link linkage. However, because of the limited workspace and the uneconomic mechanism architecture, the restrictions of Sarrus linkage shown in Fig. 12 compared with the one shown in Fig. 11 are obvious.

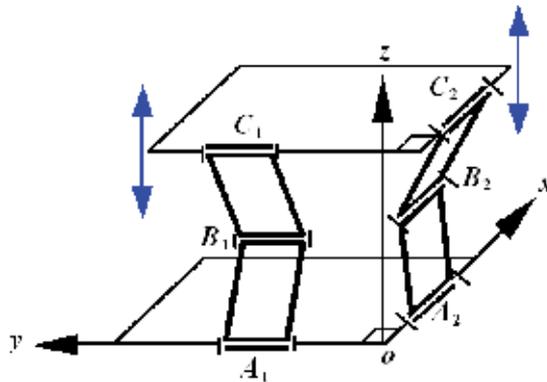


Fig. 12 the Structure of Sarrus Linkage

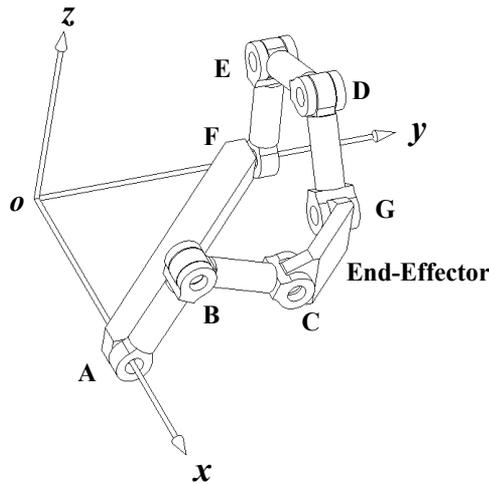


Fig. 13 Configuration of the Spatial Seven-Link Mechanism

As mentioned in step 6, in order to keep the twists of the chain to be equivalent to the twist basis, the rank of the twists of each kinematic chain synthesized through equation (34) should equal the dimension of its twist basis. Therefore, if one or more such twists are added to each kinematic chain, the free motions of the end-effector will not be changed. As an example, the mechanism shown in Fig. 13 is the derivative form of that in Fig. 10 by adding one twist \mathcal{S}_{G_1} to the kinematic chain FED . Where

$$\mathcal{S}_G = (0 \quad 1 \quad 0 \quad -z_G \quad 0 \quad x_G)^T$$

The two kinematic chains of the end-effector CD are now changed to be ABC and FEG_1D . The twists of them two are:

$$\begin{cases} \mathcal{S}_{ABC} = [\mathcal{S}_1^{F_A} & \mathcal{S}_1^{F_B} & \mathcal{S}_1^{F_C}] \\ \mathcal{S}_{FEGD} = [\mathcal{S}_2^{F_F} & \mathcal{S}_2^{F_E} & \mathcal{S}_2^{F_G} & \mathcal{S}_2^{F_D}] \end{cases}$$

It is not difficult to find that the terminal constraints of kinematic chains ABC and FEG_1D are still expressed by equation (36). And therefore, the free motion of the end-effector CD is still a straight line translation along z -axis shown in Fig. 13. As a result, the free motions of the end-effector will not be changed if one or more revolute joints whose Plücker coordinates have the form of equation (44) are added in the second kinematic chain. Similarly, the free motions of the end-effector will not be changed either if one or more revolute joints whose Plücker coordinates have the form of equation (41) are added to the first kinematic chain.

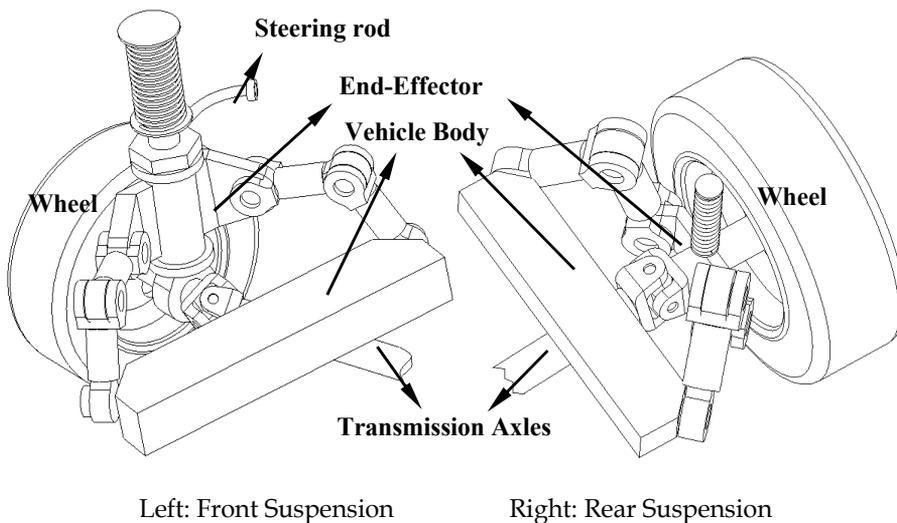


Fig. 14 a Front Suspension and a Rear Suspension

For engineering applications, the end-effector CD in Fig. 11 or Fig. 13 can be utilized as the guiding equipment of a mechanism that requires a precise linear translation, such as the independent suspension of automobile. Because the end-effector of the rigid guidance mechanism can make an exact straight line translation, the front and rear suspensions made up of such a mechanism shown in Fig. 14 allow the orientation and position parameters of the wheels such as kingpin, caster, camber, and axes distance and wheel track to be constant. These merits not only enhance the ride and handling of the vehicles, but also reduce the wearing of the tires during jounce and rebound.

5. Conclusion

This chapter focuses on the mobility analysis and synthesis of spatial parallel manipulators. It focuses on developing an analytical methodology to investigate the instantaneous DOF of the end-effector of a parallel manipulator and the instantaneous controllability of the end-effector from the viewpoint of the possible actuation schemes for the parallel manipulator. Via comparing the differences and essential mobility of a set of underactuated, over actuated and equally actuated manipulators, this chapter demonstrates that the underactuated, over actuated and fully actuated manipulators are all substantially equally actuated mechanisms. This work is significantly important for a designer to contrive his or her manipulators with underactuated or over actuated structures. Based on the analytical model of the DOF of a spatial parallel manipulator, this chapter also investigates a general process to synthesize the manipulators with specified mobility. The outstanding characteristics of the synthesis method are that the whole process is also analytical and each step can be programmed at a computer. Because of the restrictions of the traditional general mobility formulas for spatial mechanisms, a lot of mechanisms that might not be synthesized directly with the general mobility formulas could be synthesized with this analytical theory of degrees of freedom for spatial mechanisms.

6. Acknowledgements

This research was supported by FANEDD under Grant 200741 and the National Natural Science Foundation of China under Grant 50425516. The authors gratefully acknowledge these support agencies.

7. References

- Arsenault, M. & Boudreau, R. (2006). Synthesis of Planar Parallel Mechanisms While Considering Workspace, Dexterity, Stiffness and Singularity Avoidance, *ASME Journal of Mechanical Design*, Vol. 128, No. 1, (January, 2006)69-78 ISSN: 1050-0472
- Ball, R. S. (First Published 1900, Reprinted 1998). *a Treatise on the Theory of Screws*, Cambridge University Press, ISBN 0-521-63650-7, Cambridge
- Birglen, L. & Gosselin, C. M. (2006a). Geometric Design of Three-Phalanx Underactuated Fingers, *ASME Journal of Mechanical Design*, Vol. 128, No. 2, (March 2006)356-364, ISSN: 1050-0472

- Birglen, L. & Gosselin, C. M. (2006b). Grasp-state plane analysis of two-phalanx underactuated fingers, *Mechanism and Machine Theory*, Vol. 41, No. 7, (July 2006)807-822, ISSN: 0094-114X
- Chakraborty, N. & Ghosal, A. (2004). Kinematics of Wheeled Mobile Robots on Uneven Terrain, *Mechanism and Machine Theory*, Vol. 39, No. 12, (December 2004)1273-1287, ISSN: 0094-114X
- Dai, J. S. & Jones, J. Rees. (1999). Mobility in Metamorphic Mechanisms of Foldable/Erectable Kinds, *ASME Journal of Mechanical Design*, Vol. 121, No. 3, (1999)375-382, ISSN: 1050-0472
- Gogu, G. (2005a). Mobility of Mechanisms: a Critical Review. *Mechanism and Machine Theory*, Vol. 40, No. 9, (September 2005) 1068-1097, ISSN: 0094-114X
- Gogu, G. (2005b). Mobility and Spatiality of Parallel Robots Revisited via Theory of Linear Transformations, *European Journal of Mechanics A/Solids*, Vol. 24, No. 4, (July-August 2005) 690-711, ISSN: 0997-7538
- Gosselin, C. M. & Laliberté, C. T. (1998). Underactuated Mechanical Finger with Return Actuation , Patent Number: 5762390, Date of Patent: June 9, 1998, <http://robot.gmc.ulaval.ca/en/publi/patents.html>.
- Huang, Z.; Kong, L. F. & Fang, Y. F. (1997). *Theory of Mechanism of Parallel Robotics and Control*, Machinery Industry Press, ISBN: 7-111-05812-7, Beijing
- Huang, Z.; Zhao, Y. S. & Zhao, T. S. (2006). *Advanced Spatial Mechanism*, Advanced Educational Press of China, ISBN:7-04-019265-9, Beijing
- Hunt, K. H. (1978). *Kinematic Geometry of Mechanisms*, Oxford University Press, ISBN 0-19-856124-5, Oxford
- Jain, A. & Rodriguez, G. (1993) An Analysis of the Kinematics and Dynamics of Underactuated Manipulators, *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, (August 1993)411-422, ISSN: 1042-296X
- Kelley, C. T. (1995). *Iterative Methods for Linear and Nonlinear Equations*, North Carolina State University, Society for Industrial and Applied Mathematics, ISBN: 0898713528(pbk.), Philadelphia
- Kempe, A. B. (1877). *How to Draw a Straight Line*, London: Macmillan. Cited by Henderson, D. W. & Taimina, D., <http://kmoddl.library.cornell.edu/tutorials/11/>.
- Kumar, V. (1992). Instantaneous Kinematics of Parallel-Chain Robotic Mechanisms, *ASME Journal of Mechanical Design*, Vol. 114, (September 1992)349-358, ISSN: 0738-0666
- Laliberté, T. & Gosselin, C. M. (1998). Simulation and Design of Underactuated Mechanical Hands, *Mechanism and Machine Theory*, Vol. 33, No. 1-2, (January-February 1998)39-57, ISSN: 0094-114X
- Murakami, T. et al. (1989). Development of a new multi-link front suspension, SAE 890179.
- Phillips, J. (1984). *Freedom in Machinery*, Vol. 1: Introducing Screw Theory, Cambridge University Press, ISBN 0-521-23696-7, Cambridge
- Phillips, J. (1990). *Freedom in Machinery*, Vol. 2: Screw Theory Exemplified, Cambridge University Press, ISBN 0-521-25442-6, Cambridge

- Phillips, J. R. & Hunt, K. H. (1964). On the Theorem of Three Axes in the Spatial Motion of Three Bodies, *Australian Journal of Applied Science*, Vol. 15, (1964) 267-287, ISSN: 0572-1156
- Raghavan, M. (1996). Number and Dimensional Synthesis of Independent Suspension Mechanisms, *Mechanism and Machine Theory*, Vol. 31, No. 8, (November, 1996)1141-1153, ISSN: 0094-114X
- Raghavan, M. (2005). Suspension Synthesis for N:1 Roll Center Motion, *ASME Journal of Mechanical Design*, Vol. 127, No. 4, (July 2005) 673-678, ISSN: 1050-0472
- Rotman, Joseph J. (2002). *Advanced Modern Algebra*, Prentice Hall, (May 2002) ISBN : 0130878685
- Shigley, J. E. & Uicker, J. J. (1980). *Theory of Machines and Mechanisms*, New York: McGraw-Hill Companies, Inc. , ISBN: 0070568847, New York
- Simionescu, P. A. & Beale, D. (2002). Synthesis and Analysis of the Five-Link Rear Suspension System Used in Automobiles, *Mechanism and Machine Theory*, Vol. 37, No. 9, (September, 2002)815-832, ISSN: 0094-114X
- Suh, C. H. (1989). Synthesis and analysis of suspension mechanisms with use of displacement matrices, SAE 890098.
- Tsai, L.-W. (2001). *Mechanism Design: Enumeration of Kinematic Structures According to Function*, CRC Press LLC, ISBN 0-8493-09018, Florida
- Waldron, K. J. (1966). The Constraint Analysis of Mechanisms, *Journal of Mechanisms*, Vol. 1, (1966)101-114, ISSN: 0094-114X
- Yan, H.-S. & Kuo, C.-H. (2006). Topological Representations and Characteristics of Variable Kinematic Joints, *ASME Journal of Mechanical Design*, Vol. 128, No. 2, (March, 2006)384-391, ISSN: 1050-0472
- Yi, B.-J. & Kim, W. K. (2002). The Kinematics for Redundantly Actuated Omnidirectional Mobile Robots, *Journal of Robotic Systems*, Vol. 19, No. 6, (June 2002)255-267, ISSN: 0741-2223
- Yiu, Y. K. & Li, Z. (2003). Trajectory Generation for a 2-dof Over-actuated Parallel Manipulator with Actuator Speed and Torque Limits Consideration, *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.1058-1063, ISBN: 0-7803-7866, July 16-20, 2003, Kobe, Japan.
- Zhao, J.-S. et al. (2004a). A New Method to Study the Degree of Freedom of Spatial Parallel Mechanisms. *The International Journal of Advanced Manufacturing Technology*, Vol. 23, No. 3-4, (February 2004) 288-294, ISSN: 0268-3768
- Zhao, J.-S.; Zhou, K. & Feng, Z.-J. (2004b). A Theory of Degrees of Freedom For Mechanisms, *Mechanism and Machine Theory*, Vol. 39, No. 6, (June 2004)621-643, ISSN: 0094-114X
- Zhao, J.-S. et al. (2006a). Re-analysis of the Degree-of-Freedom Configuration of the Platforms in Spatial Parallel Mechanisms with Constraints Spaces, *The International Journal of Advanced Manufacturing Technology*, Vol. 28, No. 1-2, (February 2006)190-196, ISSN: 0268-3768

- Zhao, J.-S. et al. (2006b). The Free Mobility of a Parallel Manipulator, *Robotica*, Vol.24, No.5, (September 2006)635-641, ISSN: 0263-5747
- Zhao, J.-S.; Feng, Z.-J. Dong, J.- X. (2006c). Computation of the Configuration Degree of Freedom of a Spatial Parallel Mechanism by Using Reciprocal Screw Theory, *Mechanism and Machine Theory*, Vol. 41, No. 12, (December 2006)1486-1504, ISSN: 0094-114X

Feasible Human-Spine Motion Simulators Based on Parallel Manipulators

Si-Jun Zhu^{1,2}, Zhen Huang² and Ming-Yang Zhao¹

¹ *Shenyang Institute of Automation (CMS)*

² *Yanshan University
P.R. China*

1. Introduction

In biological kinematics, motion of a spine is realized by a number of functional spinal units (FSU), as shown in Fig.1 (spineuniverse.com). Each FSU consists of two adjacent vertebrae and physiological organization joining FSUs end-to-end (Hou, 2005). Owing to the spine, vertebrates have more flexible torsos than others. To understand, simulate and utilize the motion of vertebrate's torso, international researchers have made many bio-vertebrate robots such as robotic dog (bostondynamics.com), fish (robotic-fish.net), snake (Hirose; nasa.gov; ri.cmu.edu; snakerobots.com), rabbit (jsk.t.u-tokyo.ac.jp), lizard (birg.epfl.ch) and humanoid robot (kawada.co.jp; sony.net; world.honda.com; Giuseppe et al., 2003).

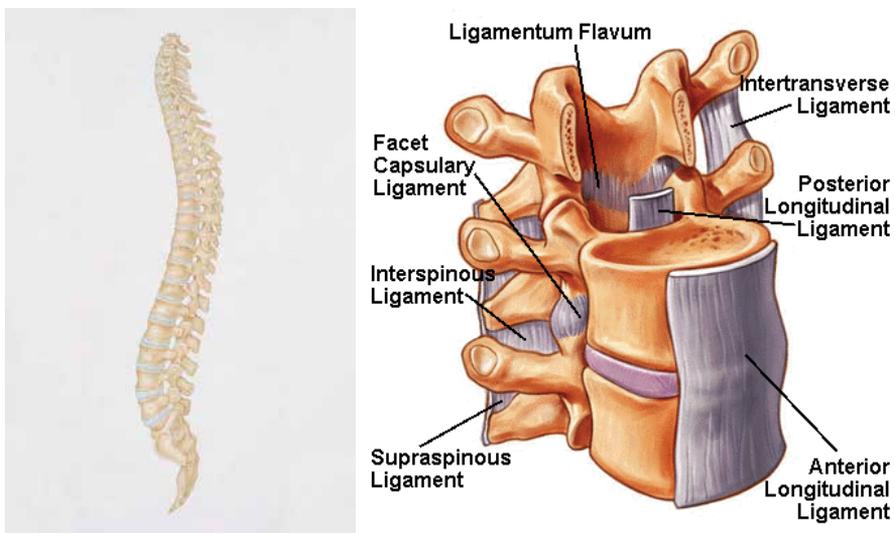


Fig. 1. A human-spine and FSU physiological organization

Torsos of some bio-vertebrate robots adopt torsos without bio-spine structure. As the most advanced quadruped robot on Earth, BigDog (bostondynamics.com) adopted rigid torso which result in the failure to realize the motion between shoulder and waist.

To improve flexibility, some bio-vertebrate robots adopt bio-spine torso, which means a number of functional kinematic units (FKU, working as FSU) joined end-to-end through joining structure. Similar with the spine for vertebrates, the bio-spine torso enable maneuverability of bio-vertebrate robots. High maneuverability qualifies robotic fishes for the oceanographical observation, the leak detection on pipelines, the search for mines and the underwater archaeological exploration (robotic-fish.net). It also makes robotic snake be competent for search-and-rescue mission (ri.cmu.edu), exploring and building in space (nasa.gov) and moving both in water and on ground (Hirose).

Humanoid robots, as an special class of bio-vertebrate robots, have much wider applications than others. Researchers believe humanoid robots may be adopted in entertainment, cooperative works, maintenance and security task (Kazuo, 2003). Torsos of these robots are carried out in different ways (kawada.co.jp; sony.net; world.honda.com; Giuseppe, Hun-ok et al., 2003). ASIMO will realize 3-DoF for head and 1 for torso in next generation (world.honda.com); SDR-4X II realized 4-DoF for neck and 2 for torso (body) (sony.net); HRP-2 realized 2-DoF for head and 1 for torso (body) (kawada.co.jp); WABIAN-RV realized 4-DoF for neck and 3 for torso (Trunk) (Giuseppe, Hun-ok et al., 2003). From a medical point of view, vertebrae of human being are joined by intervertebral disc, ligaments and etc. Translation for one end of a spine to another along the axis of spine is mainly realized by bending, stretching and compressing ligaments and intervertebral discs. The range of such a translation is so small that can be ignored. Therefore, one end of spine for human has 3 rotational and 2 independent translational DoFs (3R2T) relative to another instead of theoretical 6 DoFs.

To the best of our knowledge, most existing humanoid robots accomplish motion of torso through a serial mechanism. For a serial manipulator, the motor closed to the base has to bear the mass of the motors closed to the manipulating end. Consequently, the link closed to the base is much stronger than that closed to the manipulating end, which exhausts extra energy and slows down the reaction. Different with a serial manipulator, a parallel manipulator may assembled all motors on the base. Such a base-actuator structure will lighten links and consequently improve working speed. For example, **DELTA**, a famous successful parallel manipulator, "*the use of base-mounted actuators and low-mass links allows the mobile platform to achieve accelerations of up to 50 G in experimental environments and 12 G in industrial applications*" (parallemic.org). Moreover, kinematic performance of a spine for vertebrate is closed to isotropy. However, a parallel manipulator does not necessarily have isotropic kinematical performance as a serial manipulator. To achieve the kinematical performance closed to isotropy, a parallel manipulator should be fully-symmetrical (FSPM) (Mohamed, 1984), which means identical limbs, symmetrical assembly condition and actuating mode.

To simulating the motion of a human spine, the manipulator should satisfy not only on mobility property, reachable workspace and isotropic kinematical performance, but static and dynamic performance. A spine of human usually can work under two different modes: active mode and passive mode. Under active mode, muscles and ligaments control the motion of the spine following the person's will. Under passive mode, the spine passively moves under the outside load. To satisfy both modes, the manipulators had better to have less prismatic or cylindrical pairs considering that passive translation may result in extra resistance caused by non-coaxial and low working speed especially for ball screw, which will lead poor mechanics performance.

Considering the requirement mentioned above, a 3R2T 5-DoF base-actuator FSPM without prismatic or cylindrical pairs is necessary for a simple and efficient spine motion simulator. In the currently study, three such manipulators, $5\text{-}\underline{RRR}(RR)$, $5\text{-}(RRR)\underline{RR}$ and $5\text{-}(RRR)(RR)$ had been selected as the spine motion simulators. Respective kinematics properties are also illustrated and compared with each other. On this basis, a $5\text{-}\underline{RRR}(RR)$ prototype is designed and manufactured as the human spine motion simulator prototype. Motion capacity between the prototype and different parts of human spine are compared. In the rear part of the literature, future work for further improve the simulation capacity of the prototype is planned.

2. Manipulator enumeration

Type synthesis of 3R2T FSPM had been a difficulty (Merlet, 2000) and hot topic until dozens of them are proposed (Jin et al., 2001; Fang & Tsai, 2002; Huang & Li, 2002; Kong & Gosselin, 2002; Giuseppe, Lim et al., 2003; Li et al., 2004). Comparing with the degree of concern for them in type synthesis, their application seems to be an inactive area.

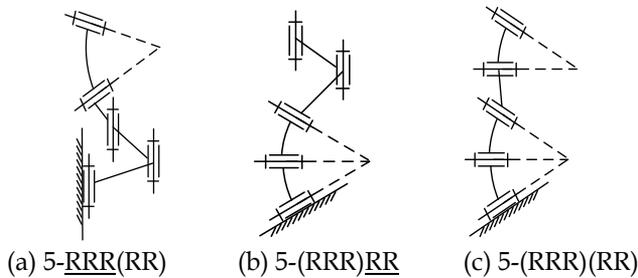


Fig. 2. Three FSPMs for spine-motion simulator

Among near twenty theoretical types of 3R2T FSPMs currently, there are only three manipulators without passive prismatic and cylindrical pairs, including $5\text{-}\underline{RRR}(RR)$, $5\text{-}(RRR)\underline{RR}$, $5\text{-}(RRR)(RR)$ as shown in Fig.2, where “ \underline{AB} ” denotes axes of pairs A and B are parallel, “(AB)” denotes axes of pairs A and B intersect at a common point.

2.1 $5\text{-}\underline{RRR}(RR)$

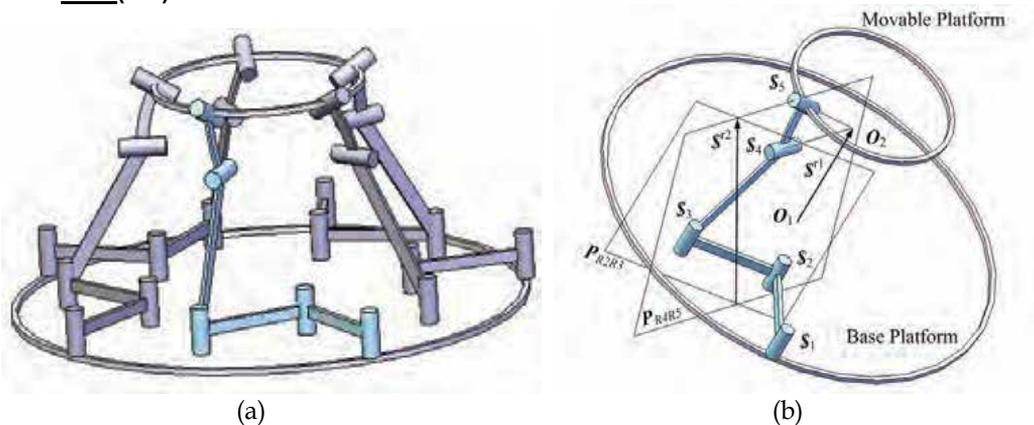


Fig. 3. $5\text{-}\underline{RRR}(RR)$

For a 5-RRR(RR) parallel manipulator shown in Fig. 3, the movable and base platforms are connected by five identical limbs each with five revolute joints. Axes of three joints adjacent to the base platform are perpendicular to the base; the other two intersect at point O_2 . The line passing through the origin O_1 and O_2 is perpendicular to the base platform. Let the x-axis be normal to axis of the first joint and z-axis point from O_1 to O_2 . In such a coordinate frame, the reciprocal screw (Ball, 1900; Hunt, 1978) of limb screw system is $\$r^1 = [0,0,1; 0,0,0]$, whose axis is normal to the base, as shown in Fig. 3(b). The reciprocal screws of five limbs are the same. So the five constraints exerting on the movable platform form a common wrench with zero-pitch which constrains the translational freedom along the z-axis. So the movable platform has three rotational and two translational freedoms in a plane parallel to the base platform. As all actuators are locked, the screw system changes to be

$$\begin{aligned}\$2 &= [S_2; S_{02}] = [0,0,1; p_2, q_2, 0] \\ \$3 &= [S_3; S_{03}] = [0,0,1; p_3, q_3, 0] \\ \$4 &= [S_4; S_{04}] = [l_4, m_4, n_4; p_4, q_4, 0] \\ \$5 &= [S_5; S_{05}] = [l_5, m_5, n_5; p_5, q_5, 0]\end{aligned}\quad (1)$$

According to the screw theory (Ball, 1900; Hunt, 1978), it can be found that $\$r^1$ and $\$r^2$ are the reciprocal screws for the screw system expressed in Eq.(1) by inspection, shown in Fig. 3(b). The axis of $\$r^2$ is the intersecting line of two planes $P_{R_2R_3}$ and $P_{R_4R_5}$. $P_{R_iR_j}$ denotes the plane determined by the axes of kinematic pairs R_i and R_j . The equation system for the axis of $\$r^2$ is

$$\begin{aligned}N_1 \cdot [x-x_{R_2}, y-y_{R_2}, z-z_{R_2}] &= 0 \\ N_2 \cdot [x-x_{o_2}, y-y_{o_2}, z-z_{o_2}] &= 0\end{aligned}\quad (2)$$

where N_1 and N_2 denote the normal vectors of the plane $P_{R_2R_3}$ ($N_1 \cdot S_2 = N_1 \cdot S_3 = 0$) and plane $P_{R_4R_5}$ ($N_2 \cdot S_4 = N_2 \cdot S_5 = 0$), respectively. $[x_{o_2}, y_{o_2}, z_{o_2}]$ and $[x_{R_2}, y_{R_2}, z_{R_2}]$ are the coordinates of O_2 and the center point of the 2nd kinematic pair adjacent to the base, respectively.

In the general configuration, five $\$r^2$ of five limbs are linear independent and every $\$r^2$ is linear independent with the common constraint $\$r^1$. So there are six linear independent constraints exerting on the movable platform when the five actuators are locked. Therefore the selection of the base actuators is feasible and the manipulator is fully-symmetrical.

However, the manipulator will be singular if the configurations of all five limbs are identical as shown in Fig. 3(a). Under such a configuration, the five reciprocal screw $\$r^2$ with zero-pitch of five limbs are distributed on a single hyperboloid of one sheet. Since the rank of screws on a hyperboloid of one sheet with one current is three, so the rank of constraints exerting on the movable platform is not six but four when all actuators are locked. That means the movable platform can still move after locking the actuators, and it is the second class of singularity in ref. (Gosselin & Angeles, 1990). To avoid such a singularity, the kinematic pairs which are not mounted on the base or movable platforms should be assembled with a little difference. For example, two limbs can be assembled in another current as shown in Fig. 4. This method is valid for currently existing 5-DoF FSPMs.

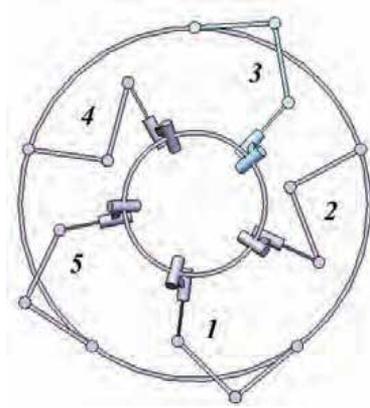


Fig. 4. Assembly condition for singularity avoidance

2.2 5-(RRR)RR

Although 5-(RRR)RR has a different structure with 5-RRR(RR), they have the similar constraint relationship, that is constraint screw of five limbs are the same. Hence, the mobility of 5-RRR(RR) is analyzed here in a simple way. In the similar coordinate frame in Fig.3(b), the reciprocal screw of five limb screw systems for 5-(RRR)RR are the same which form a common wrench $S^{r1} = [0,0,1; 0,0,0]$ constraining the translation along z-axis. The manipulator is the same with 5-RRR(RR) after locking all actuators, which means movable platform is fixed when all five actuators are locked at general configuration. Hence, base-actuator mode is valid for the manipulator 5-(RRR)RR.

2.3 5-(RRR)(RR)

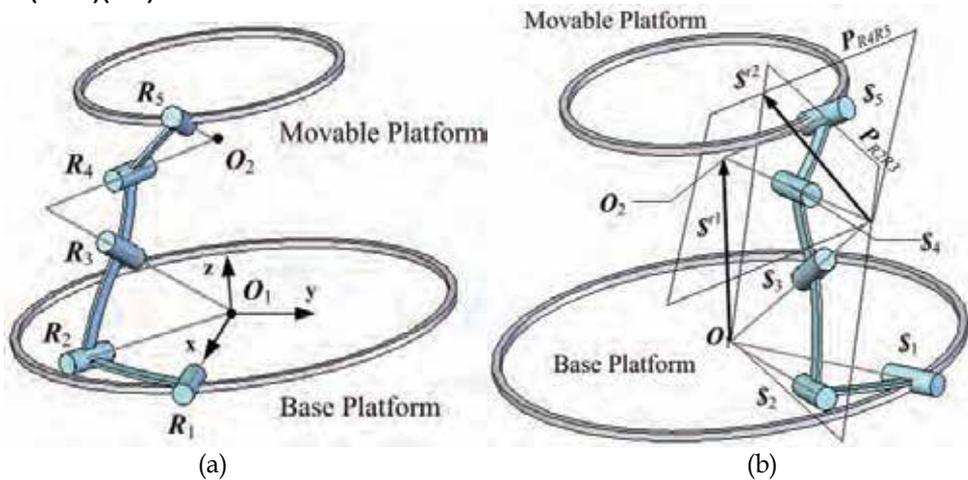


Fig. 5. 5-(RRR)(RR)

For a 5-(RRR)(RR) manipulator, the movable and base platforms are connected by five identical limbs each also with five revolute joints. Axes of three revolute joints adjacent to the base platform (R_1, R_2, R_3) intersect at a point O_1 . The other two (R_4, R_5) intersect at another point O_2 . Let O_1 be the origin, the x-axis be along the axis of R_1 and z-axis be

perpendicular to the base platform. In such a coordinate frame, the reciprocal screw of the screw system is a wrench with zero-pitch

$$\mathcal{S}^{r1} = [\mathbf{O}_2; 0,0,0] \quad (3)$$

whose axis passes through both \mathbf{O}_1 and \mathbf{O}_2 . The five reciprocal screws of five limbs are the same. So these constraints exerting on the movable platform form a common wrench with zero-pitch which constrains the translational freedom along a line passing through \mathbf{O}_1 and \mathbf{O}_2 . So the movable platform has three rotational and two translational freedoms. As all actuators are locked, the screw system changes to be

$$\begin{aligned} \mathcal{S}_2 &= [\mathbf{S}_2; \mathbf{S}_{02}] = [l_2, m_2, n_2; 0,0,0] \\ \mathcal{S}_3 &= [\mathbf{S}_3; \mathbf{S}_{03}] = [l_3, m_3, n_3; 0,0,0] \\ \mathcal{S}_4 &= [\mathbf{S}_4; \mathbf{S}_{04}] = [l_4, m_4, n_4; \mathbf{O}_2 \times \mathbf{S}_4] \\ \mathcal{S}_5 &= [\mathbf{S}_5; \mathbf{S}_{05}] = [l_5, m_5, n_5; \mathbf{O}_2 \times \mathbf{S}_5] \end{aligned} \quad (4)$$

The reciprocal screws for the screw system in Eq.(4) are \mathcal{S}^{r1} and \mathcal{S}^{r2} shown in Fig. 5(b). The axis of \mathcal{S}^{r2} is the intersection of two planes P_{R2R3} and P_{R4R5} . The parameter equation for the axis of \mathcal{S}^{r2} is

$$\begin{aligned} N_1 \cdot [x-x_{o1}, y-y_{o1}, z-z_{o1}] &= 0 \\ N_2 \cdot [x-x_{o2}, y-y_{o2}, z-z_{o2}] &= 0 \end{aligned} \quad (5)$$

where N_1 and N_2 denote the normal vector of the plane P_{R2R3} ($N_1 \cdot \mathbf{S}_2 = N_1 \cdot \mathbf{S}_3 = 0$) and plane P_{R4R5} ($N_2 \cdot \mathbf{S}_4 = N_2 \cdot \mathbf{S}_5 = 0$), respectively. Similar to the 5-RRR(RR), five \mathcal{S}^{r2} and \mathcal{S}^{r1} are six linear independent screws. So the selection of base actuators is feasible.

Note that, there is a obviously difference between the 5-(RRR)(RR) with the other two manipulators. For 5-RRR(RR) and 5-(RRR)RR, the constrained translation is always along the z-axis under different configurations. However, for the manipulator 5-(RRR)(RR), the direction of constrained translation is different under different configurations. As shown in Eq.(3), it always passes through \mathbf{O}_1 and \mathbf{O}_2 . The characteristic makes it be unique in currently existing 5-DoF FSPMs.

3. Prototype

A prototype is manufactured to verify the kinematic analysis and comparing the motion capacity of a human-spine and the prototype (Zhu, 2007). Considering that it is easy to guarantee parallelism than intersection at a common point in machining, 5-RRR(RR) is adopted as the manipulator prototype.

3.1 Structure

As mentioned in section 2.1, the axes for three joints adjacent to the base in one limb are parallel and the other two intersect at a common point. For the convenient to guarantee the machining accuracy, the prototype structure parameters are designed with special values. The axes of three joints adjacent to the base are designed to be perpendicular to the base

platform to guarantee the parallelism. The axis of R_4 is perpendicular to that of R_3 . Five arc links are manufactured through cutting a cylindrical ring averagely after drilling ten holes with indexing plate. One big and 15 small hole are drilled for lightening the movable platform. To avoid actuator singularity mentioned in section 2.1, limb are assembled as shown in the Fig. 4



Fig. 6. 3-D model of the prototype

Diameters for movable and base platform are 109mm and 200mm. The length of both links connecting joints R_1 and R_2 , R_2 and R_3 are 44mm. To allow each arc-link rotate around axis of R_5 freely, the radian of the arc-link is 24 degrees. Five stepper motor controlled by a motion control card actuate five R_1 , respectively. The minimize step of the stepper motor is 0.018 degree under the cooperation with motion control card.

3.2 Reachable workspace

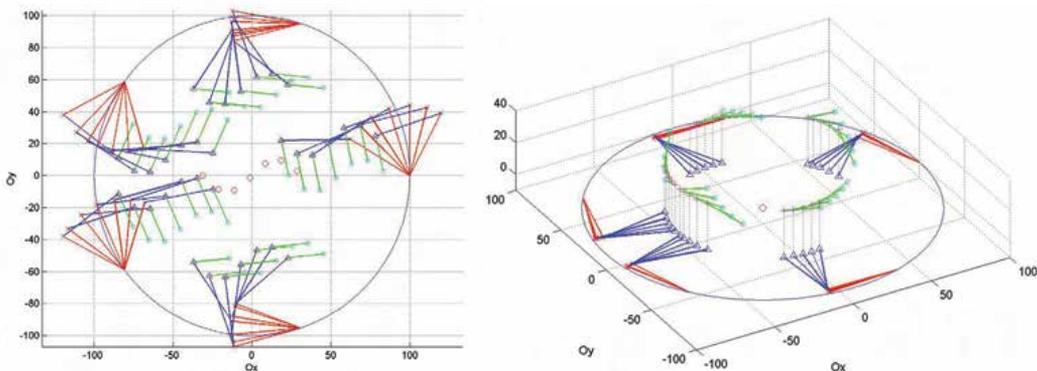


Fig. 7. Translation and rotation simulation of the prototype



Fig. 8. Translation of prototype

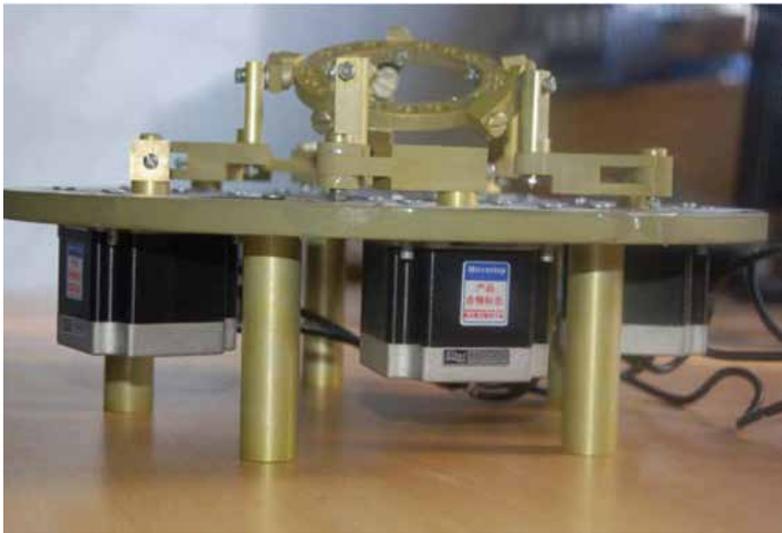


Fig.9. Rotation of prototype

According to the simulation, the reachable positions form a circle similar with a pentagon. The max translational distance is 89 mm, 44.5% of the diameter of the base. However, the max translation of the prototype is about 75mm because of interference. The rotation angles of the prototype around x-axis, y-axis and z-axis are 48, 48 and 66 degrees which is similar with the simulation. The motion of a spine is mainly realized by cervical spine, thoracic spine and lumbar spine. Considering rotation ability, cervical spine is the strongest (123, 61 and 77 degrees); the lumbar spine is the weakest (74, 29 and 9 degrees). Comparing with the three parts of human spine, the rotation ability of the prototype is similar to the thoracic spine, whose rotation angles are 76, 76 and 71 degrees, respectively.

5. Future work

Although mobility and kinematical performance closed to isotropy are realized through 5-DoF FSPM with base-actuator, there are still several aspects to be improved for further simulation capacity for the spine motion.

(a) Enlarge the reachable workspace. The reachable workspace of the prototype is smaller than that of human spine except the rotation around z-axis (yaw). Such a problem may be solved by rearranging the five R_4 . Immature hypotheses include, 1. arranging them in both sides of the movable ring platform, such as two inside and three outside; 2. Control link connecting R_3 and R_4 rotating within 180 degree instead of 360 degree through better trajectory plan to prevent link interference, which may enlarge the rotation angles around x-axis and y-axis to about 96 degree.

(b) Reaction time. The manipulator structure should be redesigned to ensure and improve the reaction time of the manipulator.

(c) Mechanics analysis. As mentioned in the literature, spine for human being may work under passive mode, in which passive force and torque should be calculated and evaluated under outside load. Hence, to simulate the bio-mechanics, static and dynamic behavior should be researched.

(d) Simulate with 5-(RRR)(RR) to make use of its unique characteristic.

6. Conclusion

Considering the characteristics of a human spine including nearly isotropic kinematical performance, fast speed, available under both active and passive modes and reachable workspace, three 3R2T 5-DoF fully-symmetrical parallel manipulators with base-actuator, including 5-RRR(RR), 5-(RRR)RR, 5-(RRR)(RR) are adopted as feasible human spine motion simulators. To decrease machining difficulty and guarantee the machining precision, 5-RRR(RR) is designed and manufactured as the prototype of spine motion simulator. After comparing reachable workspace of the prototype and that of human spine, the future work are planned for further improving simulation capacity of the prototype.

7. References

- Ball, R. (1900). *A Treatise on the Theory of Screws*. Cambridge, Cambridge University Press birg.epfl.ch/page65446.html.
- Fang, Y. F. and Tsai, L. W. (2002). "Structure Synthesis of a Class of 4-DoF and 5-DoF Parallel Manipulators with Identical Limb Structures." *The International Journal of Robotics Research* 21(9), 799-810.
- Giuseppe, C. L.; Lim, H.-O., et al. (2003). Numerical and Experimental Estimation of Stiffness Performances for the Humanoid Robot Wabian-RV. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*.
- Gosselin, C. M. and Angeles, J. (1990). "Singularity Analysis of Closed-Loop Kinematic Chains." *IEEE Transactions on Robotics and Automation* 6(3), 281-290.
- Hou, S. X. (2005). *Spine Surgery*. Beijing, People's Military Medical Press. (in chinese)
- Huang, Z. and Li, Q. C. (2002). "General Methodology for Type Synthesis of Lower-Mobility Symmetrical Parallel Manipulators and Several Novel Manipulators." *International Journal of Robotics Research* 21(2) 131-145.

- Hunt, K. H. (1978). *Kinematic Geometry of Mechanisms*. Oxford, Oxford:Claredon Press
- Jin, Q.; Yang, T. L. et al. (2001). Structure Synthesis of A Class of Five-DoF (Three Translation and Two Rotation) Parallel Robot Mechanisms Based on Single-Opened-Chain Units. *ASME Design Engineering Technical Conferences*, Pittsburgh.
- Kazuo, T. (2003). Humanoid Robot and its Application Possibility. *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, Tokyo, Japan.
- Kong, X. W. and Gosselin, C. M. (2002). Type Synthesis of 3-DOF Spherical Parallel Manipulators Based on Screw Theory. *ASME Design Engineering Technical Conferences*, Montreal, Canada.
- Li, Q. C.; Huang, Z., et al. (2004). "Type Synthesis of 3R2T 5-DoF Parallel Manipulators Using the Lie group of Displacements." *IEEE Transactions on Robotics and Automation* 20(2) 173-180.
- Merlet, J.-P. (2000). *Parallel Robots*. Dordrecht, Kluwer Academic Publishers.
- Mohamed, M. G. and Duffy, J. (1984). A Direct Determination of Instantaneous Kinematics of Fully Parallel Robot manipulators. *ASME Design Engineering Technology Conference*, Cambridge.
- www.bostondynamics.com.
- www.nasa.gov/centers/ames/news/releases/2000/00_66AR.html.
- www.jsk.t.u-tokyo.ac.jp/~ikuo/msm/200001/index-e.html.
- www.kawada.co.jp/global/ams/hrp_2.html.
- www.parallemic.org/Reviews/Review002.html.
- www.ri.cmu.edu/projects/project_407.html.
- www.robot.mes.titech.ac.jp/robot/snake_e.html.
- www.robotic-fish.net.
- www.snakerobots.com.
- www.sony.net/SonyInfo/News/Press_Archive/200303/03-0324E/.
- www.spineuniverse.com/displayarticle.php/article1394.html.
- world.honda.com/ASIMO/next-generation/.
- Zhu, S.-J. (2007). Kinematics of Lower-Mobility Parallel Manipulator and Theory on 5-DoF Parallel Manipulator. Qinhuangdao, Yanshan. Ph.D Thesis.(in Chinese)

Edited by Huapeng Wu

In recent years, parallel kinematics mechanisms have attracted a lot of attention from the academic and industrial communities due to potential applications not only as robot manipulators but also as machine tools. Generally, the criteria used to compare the performance of traditional serial robots and parallel robots are the workspace, the ratio between the payload and the robot mass, accuracy, and dynamic behaviour. In addition to the reduced coupling effect between joints, parallel robots bring the benefits of much higher payload-robot mass ratios, superior accuracy and greater stiffness; qualities which lead to better dynamic performance. The main drawback with parallel robots is the relatively small workspace.

Photo by Artem_Egorov / iStock

IntechOpen

