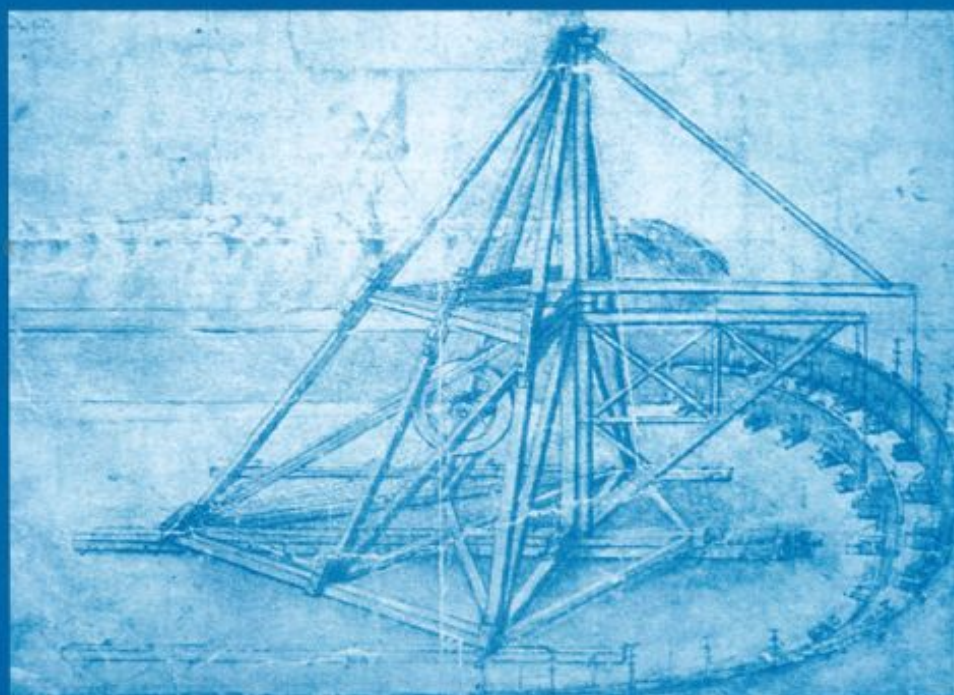


SOLID MECHANICS AND ITS APPLICATIONS

J.-P. Merlet

Parallel Robots

Second Edition



 Springer

Parallel Robots
Second Edition



Free Robotic Resource : <http://www.olayno.com>

SOLID MECHANICS AND ITS APPLICATIONS

Volume 128

Series Editor: G.M.L. GLADWELL
Department of Civil Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?*

The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids.

The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

The median level of presentation is the first year graduate student. Some texts are monographs defining the current state of the field; others are accessible to final year undergraduates; but essentially the emphasis is on readability and clarity.

For a list of related mechanics titles, see final pages.

Parallel Robots

(Second Edition)

by

J.-P. MERLET

INRIA, Sophia-Antipolis, France



Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 1-4020-4132-2 (HB)
ISBN-13 978-1-4020-4132-7 (HB)
ISBN-10 1-4020-4133-0 (e-book)
ISBN-13 978-1-4020-4133-4 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

All Rights Reserved

© 2006 Springer

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Printed in the Netherlands.

Table of Contents

Preface	xv
Acknowledgements	xvii
Notation	xviii
1 Introduction	1
1.1 Characteristics of classical robots	1
1.2 Other types of architecture	4
1.3 Needs for robotics	11
1.4 Parallel robots: definition	12
1.4.1 Generalized parallel manipulators: definition	12
1.4.2 Parallel manipulators	12
1.4.3 Fully parallel manipulators	13
1.4.4 Fully parallel manipulators: analysis	13
1.4.4.1 Planar robots	14
1.4.4.2 General case	15
1.5 Contents	17
1.6 Exercises	18
2 Structural synthesis and architectures	19
2.1 Introduction	19
2.2 Structural synthesis methods	20
2.2.1 Graph theory	20
2.2.2 Group theory approach	21
2.2.2.1 The Lie group and subgroups of displacement	21
2.2.2.2 Subgroup motion generators	22
2.2.2.3 Type synthesis based on group theory	23
2.2.3 The screw approach	23
2.2.3.1 Basics of screw theory	24
2.2.3.2 Type synthesis based on screw theory	24
2.2.4 Structural synthesis and other kinematic performances	25
2.2.5 Structural synthesis and uncertainties	25
2.2.6 Notation for parallel robots	26
2.3 Planar robots	27
2.3.1 3 d.o.f. manipulators	27
2.4 Spatial motion robots	29

2.4.1	Joints and actuators	29
2.4.2	Classification of parallel robots.	30
2.4.3	3 d.o.f. manipulators	31
2.4.3.1	Translation manipulators	31
2.4.3.2	Orientation manipulators	35
2.4.3.3	Mixed degrees of freedom manipulators	39
2.4.4	4 d.o.f. manipulators	43
2.4.5	5 d.o.f. manipulators	44
2.4.6	6 d.o.f. manipulators	47
2.4.6.1	<i>UPS</i> chain robot	48
2.4.6.2	<i>PUS</i> chain robots	51
2.4.6.3	<i>RUS</i> chain robots	52
2.4.6.4	Robots with miscellaneous chains	54
2.4.6.5	Three-legged robots	56
2.4.6.6	Decoupled robots	59
2.5	Redundant robots	62
2.6	Articulated truss and binary actuation	62
2.7	MEMS and micro-positioning robots	66
2.8	Wire robots	66
2.9	Examples of applications	69
2.9.1	Spatial applications	70
2.9.2	Vibration	73
2.9.3	Medical applications	75
2.9.4	Simulators	77
2.9.5	Industrial applications	79
2.9.5.1	Machine-tool	80
2.9.5.2	Positioning devices	86
2.9.5.3	Other industrial applications	88
2.9.6	Miscellaneous applications	91
2.10	Robots studied in this book	93
2.11	Exercises	93
3	Inverse kinematics	95
3.1	Inverse kinematics	95
3.1.1	General methods	95
3.1.1.1	Analytic method	95
3.1.1.2	Geometrical method	96
3.1.2	Examples	97
3.1.2.1	Planar manipulators	97
3.1.2.2	3- <i>UPU</i> manipulator	98
3.1.2.3	6- <i>UPS</i> manipulator	99
3.1.2.4	6- <i>PUS</i> manipulator	100

3.1.2.5	6- <u>RUS</u> manipulator	101
3.1.2.6	General conclusion	102
3.1.3	Extrema of the joint coordinates	102
3.2	Exercises	103
4	Direct kinematics	105
4.1	Planar robots	105
4.1.1	The 4-bar mechanism	106
4.1.2	Coupler curve and circularity	106
4.1.3	Direct kinematics of the 3- <u>RPR</u> robot	107
4.1.3.1	Assembly modes	108
4.1.3.2	Polynomial direct kinematics	108
4.1.3.3	Particular cases	111
4.1.4	Other planar robots	111
4.2	Robots with 3 translational d.o.f.	112
4.3	Robots with 6 d.o.f.	113
4.3.1	Example of analysis: the TSSM	113
4.3.1.1	Upper bound on the number of assembly modes	113
4.3.1.2	Polynomial formulation	114
4.3.1.3	Example of TSSM with 16 assembly modes	117
4.3.2	Analysis of other space mechanisms	119
4.3.2.1	3 degrees of freedom wrist	120
4.3.2.2	MSSM	121
4.3.2.3	6- <u>PUS</u> robot and Stewart platform	121
4.3.2.4	Manipulators <i>PPP-3S, PRR-3S, PPR-3S</i>	122
4.3.3	Special cases of the 6- <u>UPS</u> robot	123
4.3.3.1	6-5 manipulators	123
4.3.3.2	6-4 manipulators	123
4.3.3.3	6-3 manipulators	124
4.3.3.4	5-5 manipulators	124
4.3.3.5	5-4 manipulators	124
4.3.3.6	4-4 manipulators	126
4.3.3.7	Manipulators with 5 aligned points	127
4.3.4	The SSM	127
4.3.5	General case of the 6- <u>UPS</u> robot	128
4.3.5.1	Maximum number of assembly modes	128
4.3.5.2	Determination of the solutions	128
4.3.5.3	Example with 40 real solutions	129
4.3.6	Summary of results	130
4.4	Systematic method for <i>UPS</i> robots	130
4.4.1	Manipulators with 9 legs	130

4.4.2	Manipulators with 7 and 8 legs	133
4.5	Conclusion	134
4.6	Fast numerical methods	135
4.6.1	Newton schemes	136
4.6.1.1	Principle	136
4.6.1.2	Implementation for the direct kinematics .	137
4.6.1.3	Drawbacks of the Newton schemes and real-time issues	139
4.6.1.4	Convergence of the Newton schemes	140
4.6.1.5	Extending the unicity domain: the inflation	142
4.6.2	Interval analysis scheme	142
4.6.3	Methods efficiency and computation time	143
4.6.4	Path tracking	144
4.7	Direct kinematics with extra sensors	145
4.7.1	Type and location of the extra sensors	146
4.7.2	Maximal number of sensors	146
4.7.2.1	Addition of angular sensors	146
4.7.2.2	Addition of linear sensors	147
4.7.2.3	Combination of angular and linear sensors	148
4.7.3	Relationship between sensors accuracy and pose accuracy	148
4.8	Exercises	149
5	Velocity, accuracy and acceleration analysis	153
5.1	Kinematics relations	153
5.2	Inverse jacobian matrix	153
5.2.1	Euler angles inverse jacobian	155
5.2.1.1	Example: 6- \underline{UPS} manipulator	155
5.2.2	Inverse kinematic jacobian	156
5.2.2.1	Example: planar 3- \underline{RPR} manipulator	157
5.2.2.2	Example: 3 - \underline{UPU} manipulator	158
5.2.2.3	Example: 3 - \underline{PUS} rotational wrist	159
5.2.2.4	Example: 6- \underline{UPS} manipulator	160
5.2.2.5	Example: 6- \underline{PUS} manipulator	161
5.2.3	Inverse jacobian and Plücker line coordinates	161
5.3	Jacobian matrix	162
5.4	Kinetostatic performance indices	163
5.4.1	Manipulability and the kinematics polyhedron . . .	163
5.4.2	Condition number and other indices	165
5.4.2.1	Manipulability index and condition number	165
5.4.2.2	Validity of the condition number	167
5.4.2.3	Isotropy	169

5.4.2.4	Global conditioning indices	169
5.4.2.5	Other accuracy indices	170
5.5	Determination of the joint velocities and twist	171
5.5.1	Determination of the joint velocities	171
5.5.2	Determination of the twist	171
5.6	Extrema of the velocities in a workspace	172
5.6.1	Extrema of the twist	172
5.6.2	Extrema of the joint velocities	173
5.7	Accelerations analysis	173
5.7.1	6- <u>UPS</u> robot	173
5.7.2	6- <u>PUS</u> robot	174
5.8	Accuracy analysis	175
5.8.1	Geometrical errors	175
5.8.2	Thermal errors	176
5.8.3	Gravity induced errors	176
5.8.4	Dynamics errors	176
5.8.5	Worst poses for accuracy	176
5.9	Exercises	177
6	Singular configurations	179
6.1	Introduction	179
6.2	Singularity influence and classification	179
6.2.1	Singularities and velocities	179
6.2.2	Singularities and statics	181
6.2.3	Singularities and kinematics	182
6.2.4	Serial singularity	182
6.3	Parallel singularities	183
6.3.1	Motivations for the study of singularity	183
6.3.2	Singularity analysis	184
6.4	Grassmann geometry	185
6.4.1	Variety and geometry	186
6.4.2	Examples of geometrical analysis	189
6.4.2.1	Planar 3- <u>RPR</u> manipulator	189
6.4.2.2	3- <u>UPU</u> manipulator	190
6.4.2.3	MSSM	191
6.5	Motion associated with singularities	200
6.5.1	Determination of the singularity motion	201
6.5.2	Determination of the instantaneous rotation axis	201
6.5.3	Example: the MSSM	202
6.5.3.1	Type 3d configuration	202
6.5.3.2	Type 5a and 5b configuration	203
6.6	Singularity indices	204

6.7	Singularity test	206
6.8	Mechanisms in permanent singularity	208
6.9	Singularity-free path-planning and workspace enlargement	209
6.10	Singularity and design	210
6.11	Exercises	211
7	Workspace	213
7.1	Workspace limits, representation and type	213
7.1.1	The different types of workspaces	213
7.1.2	Orientation representation	214
7.2	Workspace calculation methods	215
7.2.1	Geometrical approach	215
7.2.2	Discretisation method	216
7.2.3	Numerical methods	217
7.3	Planar manipulators	219
7.3.1	Constant orientation workspace	219
7.3.1.1	Joint coordinates limits	219
7.3.1.2	Mechanical limits on the passive joints	220
7.3.1.3	Leg interference	220
7.3.2	Orientation workspace	221
7.3.3	Dextrous workspace	222
7.3.4	Maximal workspace	223
7.3.5	Inclusive orientation workspace	225
7.3.6	Total orientation workspace	227
7.4	3- <i>UPU</i> manipulator	228
7.5	6- <i>UPS</i> manipulator	228
7.5.1	Cross-sections of the constant orientation workspace	229
7.5.2	3D constant orientation workspace	230
7.5.2.1	Workspace area and volume	231
7.5.2.2	Mechanical limits on the joints	233
7.5.2.3	Interference between links	237
7.5.3	Orientation workspace	239
7.5.4	Dextrous workspace	240
7.5.5	Maximal workspace	240
7.5.6	Workspace for machine-tool	242
7.5.7	Comparison between architectures	244
7.6	Workspace performance indices	245
7.7	Trajectory verification	246
7.7.1	Line segment verification	246
7.7.1.1	Constraints on the link lengths	246
7.7.1.2	Mechanical limits on the joints	248
7.7.1.3	Example	248

7.7.2	Parametric trajectory verification	248
7.8	Motion-planning	249
7.8.1	Global motion planning	249
7.8.1.1	Cell decomposition for planar robots	251
7.8.1.2	Cell decomposition for spatial robots	252
7.8.1.3	Roadmaps	254
7.8.2	Machine-tool motion planning and part positioning	255
7.8.3	Prospective for global motion planning	256
7.9	Exercises	257
8	Static analysis	259
8.1	Relations between wrench and joint forces	259
8.1.1	Fundamental relations	259
8.1.2	Determination of the wrench	259
8.1.3	Determination of the joint forces	259
8.2	Maximal joint forces and maximal wrench	260
8.2.1	Maximal joint forces in a pose	260
8.2.2	Maximal joint forces in a translation workspace	261
8.2.3	Maximal joint forces in a general workspace	263
8.2.4	Maximal wrench in a pose	263
8.2.5	Maximal wrench in a workspace	264
8.3	Force performances indices	265
8.4	Parallel robots as force sensors	266
8.5	Stiffness and compliance	266
8.5.1	Stiffness matrix of a parallel robot	267
8.5.1.1	Elastic model	267
8.5.1.2	Beam model	268
8.5.2	Passive compliance and force-feedback control	269
8.5.3	Stiffness maps	270
8.5.3.1	Iso-stiffness maps	270
8.5.3.2	Iso-stiffness of 6- <i>UPS</i> robot	270
8.5.4	Extrema of the stiffness in a workspace	273
8.5.5	Stiffness and design	274
8.6	Static balancing	274
8.7	Exercises	275
9	Dynamics	277
9.1	Introduction	277
9.2	MSSM inverse dynamics	279
9.3	6- <i>UPS</i> manipulator dynamics	281
9.3.1	Hypothesis and notation	281
9.3.2	Algorithm principle	281

9.4	6- <i>PUS</i> robot dynamics	285
9.5	Computation time	286
9.6	Examples	286
	9.6.1 Inverse dynamics	286
	9.6.2 Direct dynamics	287
9.7	Exercises	288
10	Calibration	289
10.1	Introduction	289
10.2	Types and principle of calibration methods	290
	10.2.1 Calibration principle	290
	10.2.2 General comparison of calibration methods	291
	10.2.3 Issues in calibration methods	291
10.3	External calibration	292
	10.3.1 Type of external measurements	292
	10.3.2 Calibration with direct kinematics	293
	10.3.3 Calibration with inverse kinematics	294
	10.3.4 Calibration with constant leg lengths	295
	10.3.5 Calibration with other geometrical elements	296
	10.3.6 Unicity of the solution	296
	10.3.7 Observability	297
10.4	Auto-calibration	298
10.5	Calibration with mechanical constraints	298
10.6	Determination of the calibration poses	299
10.7	Exercises	300
11	Design	301
11.1	Introduction	301
11.2	Reducing the number of design parameters	303
11.3	The atlas approach	305
11.4	The cost function approach	305
11.5	The exact synthesis approach	309
	11.5.1 Workspace synthesis	309
	11.5.2 Velocity synthesis	311
11.6	The parameter space approach	311
	11.6.1 The parameter space	312
	11.6.2 Principle of the method	313
	11.6.3 Calculation of the allowed regions	313
	11.6.4 Search for appropriate robots	317
	11.6.5 Design examples	318
	11.6.6 Advantages and drawback	318
11.7	Other design issues	319

11.8 Exercises	320
12 Appendix: system solution	321
12.1 Homotopy	321
12.2 Elimination	322
12.3 Gröbner basis	323
13 Appendix: interval analysis	325
13.1 Introduction	325
13.2 Function properties and interval evaluation	326
13.3 Generic interval-based algorithm	326
13.4 General purpose applications	328
13.4.1 System solving	328
13.4.2 Global optimization	328
13.4.3 Linear system solving	329
13.5 Robotics applications	329
13.5.1 Workspace calculation	329
13.5.2 Singularity detection	330
13.6 Conclusion	330
14 Conclusion	331
References	333
Index	383

Preface

Parallel robots, also sometimes called *hexapods* or Parallel Kinematic Machines (PKM), are closed-loop mechanisms presenting very good performances in terms of accuracy, rigidity and ability to manipulate large loads. They are been used in a large number of applications ranging from astronomy to flight simulators, and are becoming increasingly popular in the machine-tool industry. This book intends to present a comprehensive synthesis of the latest results on the possible mechanical architectures, on their analysis and design and on possible uses of this type of mechanism. It is a completely updated version of the first edition which was published in 2000.

In a quickly moving domain a book presents the fundamentals of a domain but cannot pretend to remain up to date in term of references and possible applications. Two Web sites will allow a follow-up in term of evolution:

- www-sop.inria.fr/coprin/equipe/merlet/merlet_eng.html: this site provides an extensive, updated bibliographic data base together with open problems and possible mechanical architectures of parallel robot. It will be called the *references Web page* in this book.
- www.parallemic.org: this site maintained by my friend Ilian Bonev presents interesting reviews, web links and up-to-date information on parallel robots

This book is intended to be used by students, researchers and engineers:

- for students there are over 140 exercises and problems¹
- for engineers there are many practical results and applications. Most of the experimental considerations presented in this book are the result of the development of our own prototypes (or for which we have got a design contract) and of numerous discussions with others researchers and industrial partners who have developed parallel robots.
- for researchers a comprehensive list of research topics is addressed together with an important list of references (about 45 % of the references are posterior to the first edition)

As far as references are concerned, I have been confronted by difficult choices: there are numerous references in this field (close to 2000 references may be found in the references Web page at the date of publication

¹Solutions of the exercises are available at www-sop.inria.fr/coprin/equipe/merlet/Solutions/exo.html

of this book) and clearly not all of them can be mentioned. I have favored journal papers over conference papers as they are more easily available. For authors having made a large number of contributions in a given topic I will often only reference their latest related work and indicate by a * that their additional references on the same topic will be found in the references Web page. Then I have tried to include references providing all the possible viewpoints for the problem at hand. Patents will, in general, not be referenced as they are available by conventional means. Even after these steps, the number of references far exceeded what was reasonable for a textbook, so drastic choices had to be made. In the first edition most of the authors were referenced in the index: for lack of space this is no longer the case, but their works can be found in the references Web page.

Web addresses of companies and laboratories are rapidly changing but many of them are however indicated in this book in the following manner[□] *HS* which indicates that a label *HS* followed by the corresponding web link will be found at www-sop.inria.fr/coprin/equipe/merlet/Web

The codes of some algorithms presented in this book are available by anonymous ftp access². I hope that the unsolved problems presented at the end of each chapter will be a source of inspiration for research.

This book is organized so that the abstract at the beginning of each chapter will be, in general, sufficient to understand the problems addressed in the chapter.

The fundamentals of a robotics book involve many scientific domains such as kinematics, dynamics, control theory Elegant and powerful theorems may be established by using only a theoretical approach, which is absolutely necessary. But at some point we have to get numerical results and, given the complexity of the robotics calculation, computers will be involved. They can be used at an early stage with *symbolic computation* to facilitate the manipulation of the complex expressions we will have to deal with, or at the last stage to get numerical results which justify an appendix on system solving. But in many case current computers are not perfect (and sometime they are completely wrong ! see the interval appendix), a point that is often ignored and that may have severe consequences (for example in medical applications). A constant preoccupation we will have in this book is to present algorithms whose results may be *certified*, i.e. that are presented with error bounds that will indicate how much confidence we may have in them. This justifies the appendix devoted to interval analysis, a not so well-known method, that allows the development of such certified algorithms.

²ftp address: [ftp-sop.inria.fr](ftp://ftp-sop.inria.fr), directory `coprin`

Acknowledgments

For the technical support received during the writing of this book, I am indebted to many people. As it involves knowledge in mechanism theory, geometry, symbolic computation, computer science, control theory ... numerous people at INRIA and at other laboratories have contributed to this work. A list of their names could not be presented here but they are all deeply acknowledged, with very special thoughts for my INRIA colleagues and friends Manuel Bronstein, which die recently, and Isabelle Attali, who disappeared tragically with her two sons in Sri-Lanka during the tragedy of December 2004.

I want also to dedicate this book to the late Claude Reboulet, a pioneer in the study of parallel robots. Claude was a model for me, as for all of my colleagues, and we are all missing him.

We must also remember the late J. Duffy, K. Hunt and L-W. Tsai: beside being prominent kinematicians, they were enjoyable people and they played a major role in the Computational Kinematics community.

C. Gosselin, professor at University Laval, Québec, my co-author of many papers, is here gratefully acknowledged. I am also indebted to Professor Jorge Angeles, who encouraged me very early to pursue this work on parallel robots. Many results presented in this book are the result of the works of C. Reboulet, C. Gosselin and J. Angeles; any error in the presentation is mine. I am also indebted to E. Rivière, who produces the initial translation from the french version of the first version of this book and to Pr. G. Gladwell who made numerous suggestions for improving and clarifying this translation.

All individuals and institutions who contributed graphical material are sincerely acknowledged.

To conclude I could not forget my wife Sylvie and my daughters Laurane and Solène for their unlimited patience and understanding of my late arrival at home.

Sophia-Antipolis, October 7, 2005

Notation

In this book, vectors will be denoted in bold font, matrices in capital slanted bold font and other mathematical quantities in italic font. For example $\mathbf{A}\mathbf{X} = \lambda\mathbf{b}$ indicates that the vector \mathbf{X} multiplied by the matrix \mathbf{A} should be equal to the vector \mathbf{b} multiplied by the scalar λ . For the sake of readability, we also try to avoid the use of the transpose in the equations when there is no ambiguity. For example the dot product of two vectors \mathbf{U}, \mathbf{V} will be denoted $\mathbf{U}\cdot\mathbf{V}$ instead of the classical $\mathbf{U}^T\cdot\mathbf{V}$.

The following notation and definitions will be used:

- \times : cross-product of 2 vectors
- \cdot : scalar product of 2 vectors
- \dot{a} : time-derivative of a
- \mathbf{X} : generalized coordinates of the robot in vector form
- Θ : joint coordinates of the robot in vector form
- $\Theta_{\mathbf{a}}$: actuated joint coordinates of the robot in vector form
- $\Theta_{\mathbf{p}}$: passive joint coordinates of the robot in vector form
- ρ_i, Θ_i : usually actuated joint coordinate of link i
- ρ_{min}, ρ_{max} : minimal and maximal value of a joint coordinate
- A_i : center of the joint of link i , attached to the base
- $x_{a_i}, y_{a_i}, z_{a_i}$: coordinates of A_i in the reference frame
- B_i : center of the joint of link i , attached to the end-effector
- $x_{b_i}, y_{b_i}, z_{b_i}$: coordinates of B_i in the moving frame
- O : origin of the reference frame
- (O, x, y, z) : reference frame
- C : origin of the moving frame. It will be used as a pose parameter for the end-effector
- x_c, y_c, z_c : coordinates of C in the reference frame
- (C, x_r, y_r, z_r) : moving frame
- ψ, θ, ϕ : Euler angles defining the orientation of the end-effector. These angles are defined in the following manner: starting from the reference frame, we obtain the moving frame by first rotating around the z axis through an angle ψ , then by rotating around the new x axis through an angle θ and finally by rotating around the new z axis through an angle ϕ .
- \mathbf{R} : rotation matrix from the moving frame to the reference frame
- \mathbf{J} : jacobian matrix of the robot
- \mathbf{J}^T : transpose of the jacobian matrix
- $\mathbf{\Omega}$: angular velocity vector of the end-effector
- \mathbf{V} : cartesian velocity vector of the end-effector
- \mathbf{W} : velocity vector of the end-effector, constituted of \mathbf{V} and $\mathbf{\Omega}$, also called the *velocity twist*

- $\boldsymbol{\tau}$: actuated joints force/torque in vector form
- $\boldsymbol{\mathcal{F}}$: force/torque vector applied to the end-effector. Unless otherwise mentioned the torque will be calculated with respect to C .
- annular region: a region in the plane delimited by two concentric circles defined as the set of points that lie in the larger circle and not in the smaller one
- C : cylindrical joint
- H : helical joint
- P : prismatic joint
- P_a : parallelogram
- R : revolute joint
- S : ball-and-socket joint
- U : universal joint
- d.o.f.: degrees of freedom

Introduction

Mechanical systems that allow a rigid body (here called *an end-effector*) to move with respect to a fixed base, play a very important role in numerous applications. A rigid body in space can move in various ways, in translation or rotary motion. These are called its *degrees of freedom*. The total number of degrees of freedom of a rigid body in space cannot exceed 6 (for example three translatory motions along mutually orthogonal axes, and three rotary motions around these axis). The position and the orientation of the end-effector (here called its *pose*) can be described by its generalized coordinates; these are usually the coordinates of a specific point of the end-effector and the angles that define its orientation, but may be any other set of parameters that allows one to define uniquely the pose of the end-effector. As soon as it is possible to control several degrees of freedom of the end-effector via a mechanical system, this system can be called a *robot*.

The last few years have witnessed an important development in the use of robots in the industrial world, mainly due to their flexibility. However, the mechanical architecture of the most common robots does not seem adapted to certain tasks. Other types of architecture have therefore recently been studied, and are being more and more regularly used within the industrial world. This is so for the parallel robots that we will study in this book.

1.1. Characteristics of classical robots

Currently, most existing manipulators present a decidedly anthropomorphic character, usually strongly resembling a human arm. They are constituted of a succession of rigid bodies, each of them being linked to its predecessor and its successor by a one-degree-of-freedom joint, for example allowing the rotation of a rigid body around an axis, or the translatory motion of a rigid body. This architecture will be called a *serial robot* with analogy to electrical systems. An example of a serial mechanism is the *spherical robot*, where a succession of segments goes from the base to the end-effector, each segment being linked to its successor by a revolute joint. If each of the n joints is actuated, it will usually be possible to control n degrees of freedom

of the end-effector. The serial robot **Scara** represents a good architectural example. It allows the control of 4 degrees of freedom from the end-effector (figure 1.1). Tables 1.1 and 1.2 present the general characteristics of robots

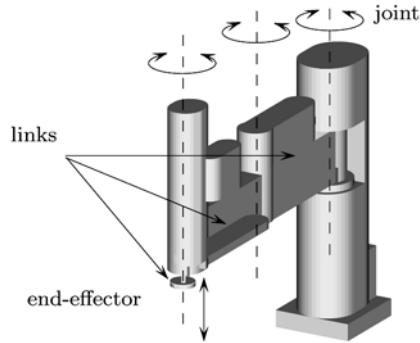


Figure 1.1. The "Scara" robot

of the Scara type and of industrial spherical robots with 6 degrees of freedom.

Robot	d.o.f.	mass	load	Repeatability	$\frac{\text{load}}{\text{mass}}$
Adept I800	4	34	5.5	± 0.02	0.1617
Adept 1XL	4	265	12	± 0.025	0.0452
Adept 3XL	4	266	25	± 0.038	0.0939
Epson E2C251	4	14	3	± 0.01	0.21442
Epson E2S45x	4	20	5	± 0.015	0.25
Epson E2H853	4	37	2	± 0.025	0.054
Seiko EC250	4	14	3	± 0.01	0.21438
Seiko EH850	4	43	10	± 0.025	0.2325
Toshiba SR-504HSP	4	38	2	± 0.02	0.0526

TABLE 1.1. Characteristics of industrial manipulators (Scara type, mass of the robot and load capacity in kg, repeatability in mm, according to the manufacturers notice).

These two tables emphasize several interesting points, of which the first is the value of the ratio of the load capacity/robot mass. For a spherical manipulator with 6 degrees of freedom, this ratio is less than 0.15. Therefore, for a transported mass of about 500 kg, the manipulator mass would be about 3330 kg. Note that compared to the first edition of this book

Robot	mass	load	Repeatability	$\frac{\text{load}}{\text{mass}}$
ABB IRB 140T	98	5	± 0.03	0.051
ABB IRB 2400L	380	7	± 0.06	0.01842
ABB IRB 4400/45	980	45	± 0.1	0.04591
ABB IRB 6400R/3.0-100	1600	100	± 0.15	0.0625
Fanuc Arc Mate 100i	138	6	± 0.08	0.04347
Fanuc Arc Mate 120i	370	16	± 0.1	0.04324
Fanuc M420iA	620	40	± 0.5	0.064516
Fanuc R-2000iA 165F	1210	165	± 0.3	0.13636
Fanuc S-900iB/200	1970	200	± 0.5	0.101523
Kuka KR 6	235	6	± 0.1	0.02553
Kuka KR 60-3	665	60	± 0.2	0.09022
Kuka KR 100	1155	100	± 0.15	0.08658

TABLE 1.2. Characteristics of industrial manipulators (spherical type, mass of the robot and load capacity in kg, repeatability in mm, according to the manufacturers notice).

there has been a relatively large improvement in the ratio load/mass with an average value going from 0.035 to 0.064.

For robots of the Scara type, this ratio is in general better, in particular for the so-called direct-drive robots, without a reduction gear between the motors and the joints. However, it is always less than 0.25 for heavy loads. For a load capacity of 500 kg the robot mass will thus be at least 2000 kg. Note that compared to the first 2000 edition of this book there has been an improvement in the ratio load/mass with an average value going from 0.06846 to 0.08547.

The second noteworthy point concerns the positioning accuracy, for which there are two distinct concepts: *absolute accuracy*, defined as the distance between the desired and the actual position of the end-effector, and *repeatability*, which is the maximum distance between two positions of the end-effector reached for the same desired pose from different starting positions. The accuracy values given by the manufacturers generally indicate repeatability, which is far better than absolute accuracy, even though users are interested mostly in absolute accuracy. Tables 1.1, 1.2 show us that even repeatability may even be insufficient for certain tasks. As for absolute accuracy, it is conditioned by several factors: accuracy of the manipulator internal sensors (the sensors that are used to measure the joint coordinates and to control the robot motions), clearance in the drives, flexure of the

links, quality of the geometric realization (for example, perpendicularity or parallelism between successive rotation axes). It is generally accepted that, in most cases, the absolute accuracy of a serial robot is poor.

The low transportable load and poor accuracy are both inherent in the mechanical architecture of existing manipulators, and in particular of the serial disposition of the links. Each of them has to support the weight of the segments following it in addition to the load: they are therefore all subject to large flexure torques, which means they must be stiffened, and thus become heavier. Positioning accuracy obviously depends on the flexural deformations that are not measured by the robot internal sensors. Moreover the links magnify errors: a small measurement error in the internal sensors of the first one or two links will quickly lead to a large error in the position of the end-effector. For example, for a one meter long arm made up of just one revolute joint, a measurement error of 0.06 degrees leads to an error of 1 mm in the position of the end-effector. The presence of a drive with a reduction gear also induces a backlash which leads to inaccuracy.

The violation of the assumed geometric constraints between the axes of the links also constitutes an important source of positioning errors. A slight perpendicularity error between the first two axes of a spherical manipulator will lead to errors in all vertical motions that, given the amplitude of the motions, must be taken into account. Note that the successive positions of the links, together with the necessity of stiffening them, imply that the moving parts of the robot will have a significant mass. As a consequence, during high velocity motions, the manipulator experiences inertia, centrifugal and Coriolis forces that make the control of the robot complex.

Serial robots operate under the action of two kind of forces: inertia and friction. These forces have different scales: inertia forces essentially vary with the square of the lengths of the links; friction forces are relatively unaffected by such dimensions. This means that one cannot design a micro serial robot simply by scaling down a larger version; under such scaling, the inertia forces are reduced while the friction forces remain relatively unchanged. We conclude that serial robots are inappropriate for tasks requiring either the manipulation of heavy loads, or a good positioning accuracy, or to work at different scales.

1.2. Other types of architecture

In order to introduce other types of mechanical architectures for robots, we present a few formal notions that will allow us to make a clear distinction between the key elements which characterize robots. These elements are taken from C. Gosselin (186). For each link of a manipulator, the *connection degree* is the number of rigid bodies attached to this link by a joint.

Simple kinematic chains are then defined as being those in which each member possesses a connection degree that is less than or equal to 2. Serial manipulators may then be defined as simple kinematic chains for which all the connection degrees are 2, except for two of them, the base and the end-effector, with connection degree 1. Such a chain is also called an *open-loop kinematic chain*.

A *closed-loop kinematic chain* is obtained when one of the links, but not the base, possesses a connection degree greater than or equal to 3. These elements can be shown clearly by representing the kinematic chain by a graph, which can either be a *connection graph* (87) or a *layout graph* (475). A graph will be more easily readable than a pure description of the mechanisms. Mechanism theory also retains another important notion, mobility, represented by the number of independent degrees of freedom of the end-effector. We will return to this concept later.

Some of the problems occurring with serial manipulators can be resolved mechanically by distributing the load on links, i.e. by linking the end-effector to the ground by a set of chains that each support only a fraction of the total load. The use of closed-loop kinematic chains for manipulators thus seems to be quite interesting; actually this option had already been explored even before the term robot had been coined. Some theoretical problems linked to this type of structure were mentioned as early as 1645 by Christopher Wren, then in 1813 by Cauchy (75), in 1867 by Lebesgue (348) and in 1897 by Bricard (56).

One of the main theoretical problems in this field, called the *spherical motion problem*, to which we will return later, was the central point of a competition called *Le Prix Vaillant*, that took place in France in the 1900's and was organized by the Académie des Sciences. The prize was won on equal terms by Borel (52) and Bricard (52).

Later on Bonev (50) mentions a patent filed in 1928 by J.E. Gwinnett (210) for what we will call a *spherical mechanism* to be used as a platform for a movie theater (figure 1.2).

On the practical side of things, in 1947 Gough (203) established the basic principles of a mechanism with a closed-loop kinematic structure (figure 1.3), that allows the positioning and the orientation of a moving platform so as to test tire wear and tear. He built a prototype of this machine in 1955 (204). For this structure, the moving element is a hexagonal platform whose vertices are all connected to a link by a ball-and-socket joint. The other end of the link is attached to the base by a universal joint. A linear actuator allows the modification of the total length of the link; this mechanism is therefore a closed-loop kinematic structure, actuated by 6 linear actuators. This device was still used up to 2000, the year where it was put into retirement (figure 1.4).

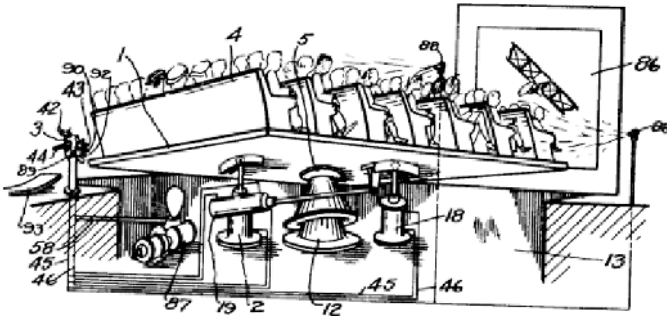


Figure 1.2. The spherical mechanism proposed in 1928 by J.E. Gwinnet

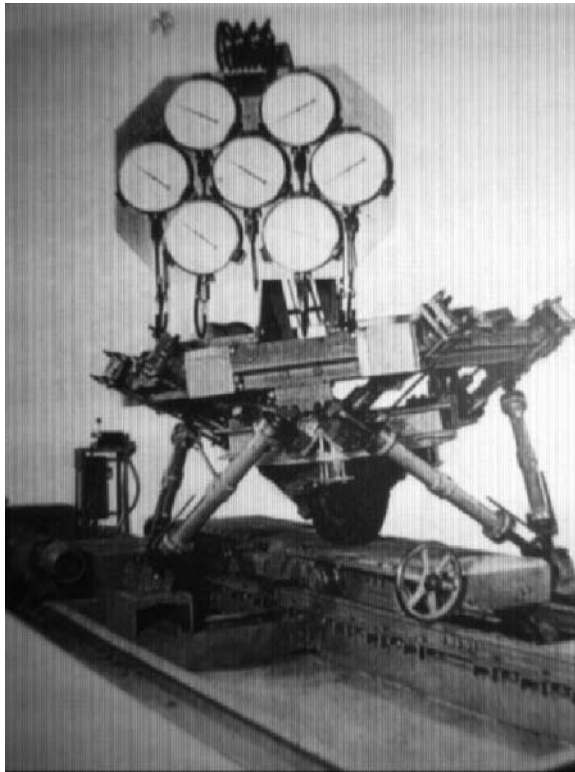


Figure 1.3. Gough platform (1947). The moving platform, to which a tire is attached, is linked to the ground by 6 links with varying lengths. An universal joint is put at one of the ends of each link, a ball-and-socket joint at the other. Changing the length of the links modifies the position and the orientation of the moving platform, and therefore of the wheel. This wheel is driven by a conveyor belt and the mechanism allows the operator to measure the tire wear and tear under various conditions (203).

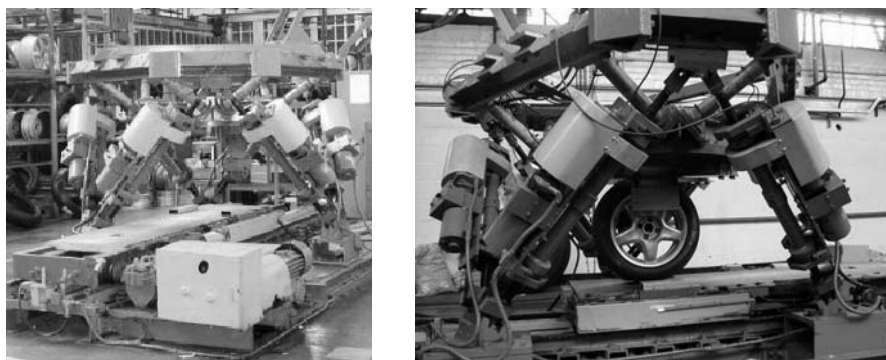


Figure 1.4. The last prototype of a Gough platform to be used in the Dunlop Tyres company (courtesy of Mike Beeson from this company). This machine, called the *Universal Rig*, is on exhibit in the British National Museum of Science and Industry

We will examine the Gough mechanism in detail throughout this book; however, let us suppose, for the moment, that the actuators are able to control the 6 degrees of freedom of the moving platform. The interest that this structure represents for the ratio load capacity/mass ratio is immediately clear. Indeed, while the structure occupies its central position, the actuators support approximately only $1/6$ of the total load. Moreover, the flexure imposed on the links is reduced because the joints may impose only traction-compression constraints. These two factors then allow us to decrease the mass of the moving structure by permitting the use of actuators of lower power and links of smaller size (it has been shown for a 3 d.o.f. parallel robot that the average energy usage was 26% of a serial manipulator of similar size (363)). Employing linear actuators is interesting because this type of element is available with excellent mass, speed, acceleration and motion amplitude characteristics (hydraulic jack for example). Intuitively, one can also imagine that the positioning accuracy is good, and that for two reasons:

- the (unmeasured) deformations of the links due to the flexure are reduced
- the errors in the internal sensors of the robot (measurement of the lengths of the links) only slightly affect errors on the platform position. For example, if all the sensors present the same error, the calculation of the pose of the platform based on the sensor measurements will show an error only for the vertical axis: the amplitude of the error will be about the same as the error in the sensors.

But as mentioned in the historical paper of I. Bonev (50) although Gough was the first to design a functional prototype of parallel robot,

hexapods were known well before. System of this type are known under the acronym *MAST*, which stands for *Multi-Simulation Table* or *Shake Table* with orthogonal disposition of the legs, that are popular in the vibration community because for small variations of the leg lengths the displacement of the platform can be easily interpreted. Figure 1.5 shows such a shake table developed at the department of Civil Engineering of the University of Minnesota (168).

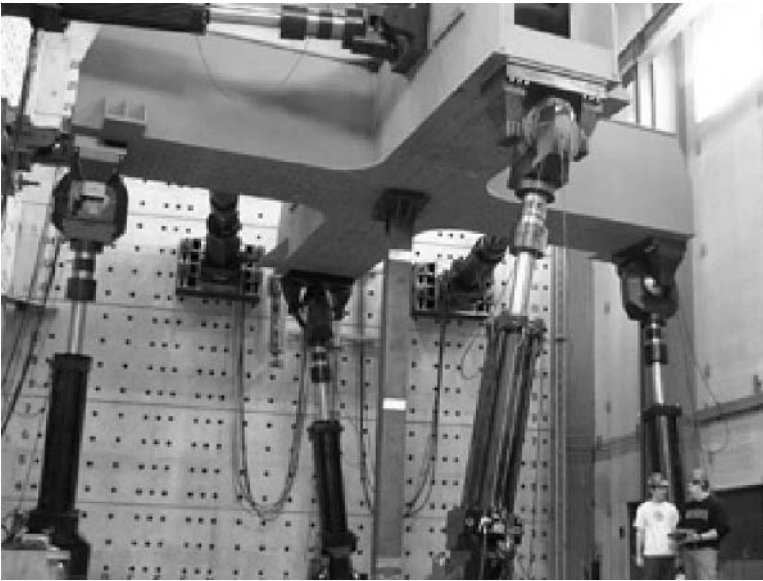


Figure 1.5. The shake table of the University of Minnesota for earthquake simulation

The use of this type of mechanism started, however, only when the first flight simulators were built. During the 1960's, the development of the aeronautics industry, the increase in the cost of pilot's training, together with the need to test new equipment while not flying, brought researchers to look into mechanisms with several degrees of freedom that could simulate a heavily loaded platform with high dynamics (for example the whole cockpit of a plane). Pictures of early simulators are presented in figure 1.6. The manipulator mass is important for dynamics because the disturbing effects (for example the Coriolis force) decrease as the mass of the moving equipment decreases. All those constraints make the use of serial manipulators difficult, because their bandwidth is generally small.

In 1965, Stewart (551) suggested that simulators should be fitted with the mechanism shown in figure 1.7. For this structure, the moving element

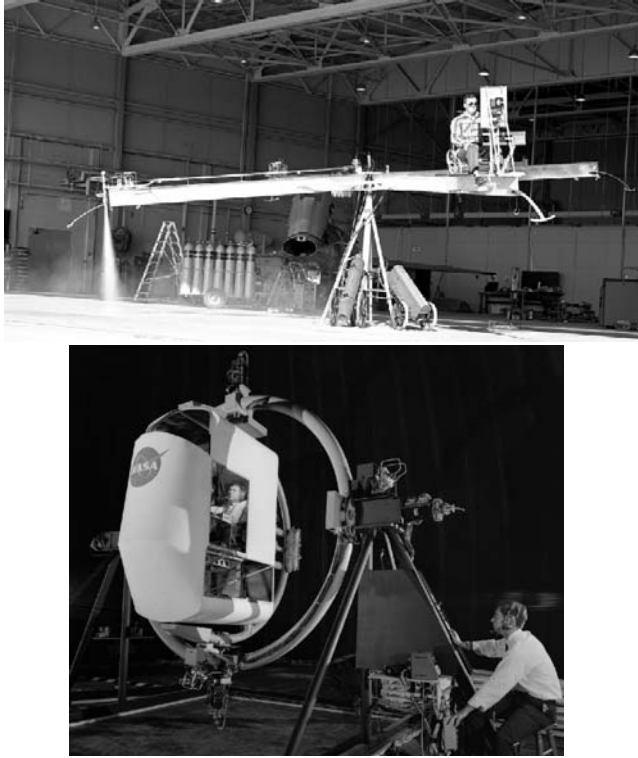


Figure 1.6. Early simulator. On the left the Iron Cross (1956) mounted on an universal joint whose motion was controlled through 6 nitrogen nozzles. On the right the lunar rendez-vous simulator (1962) with 2 d.o.f. (courtesy NASA)

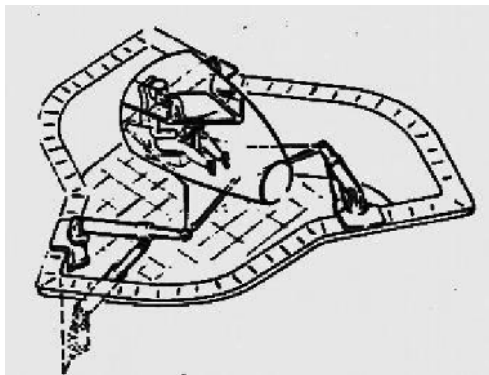
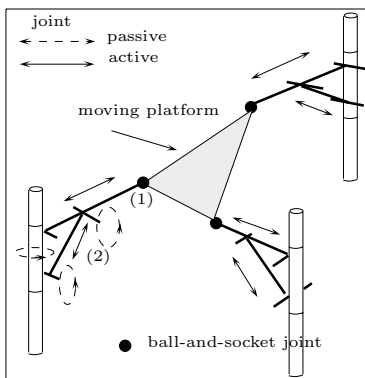


Figure 1.7. Stewart platform (1965). The motions of the moving platform are obtained by modifying the length of the 6 articulated links.

is a triangular platform whose vertices are all connected by a ball-and-socket joint to an under-mechanism constituted of two jacks (1, 2), placed also in a triangular fashion. One of the ends of each of those jacks is linked by a revolute joint to a vertical axis link that can rotate around its axis. The other end of one of the two jacks is attached by a ball-and-socket joint to the moving platform, the other end of the second jack is linked by a revolute joint to the body of the first jack. In the very last section of his paper, Stewart mentions the possibility of joining the ends of the jacks at a point linked to the platform, thus reproducing the idea of the Gough platform.

One of the reviewers of Stewart's paper happened to be Gough, who recalled the existence of his own structure. Stewart's other reviewers even suggested that the Gough platform should be used for off-shore drilling platforms or for milling machines. This revealed itself to be an excellent vision of the future, as will be exposed later on. It seems that the Stewart platform has not yet received any practical application, while the Gough platform has been used extensively. Ironically the Gough platform, which appeared much before Stewart's, is most often known as the Stewart platform!

But although Stewart's paper was instrumental in the development of flight simulator, it must be noted, as mentioned by Ilian Bonev (50), that in 1962 an engineer from the Franklin Institute, Klaus Cappel, was given the task of improving an existing MAST. He came up with the same octahedral arrangement as Gough. This device was patented in 1967 (69) (the patent was filed on December 7, 1964) but as a motion simulator on the request of the Sikorsky Aircraft Division for the design and construction of a 6 d.o.f. helicopter flying simulator (figures 1.8, 1.9).

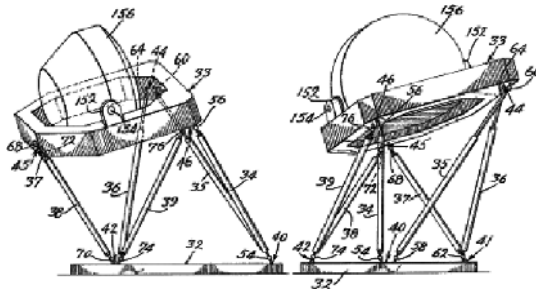


Figure 1.8. A drawing in the patent of K. Cappel (1967).

The patent was infringed by CAE, a leader in flight simulator, but the lawsuit by the Franklin Institute was successful. According to Klaus Cappel,

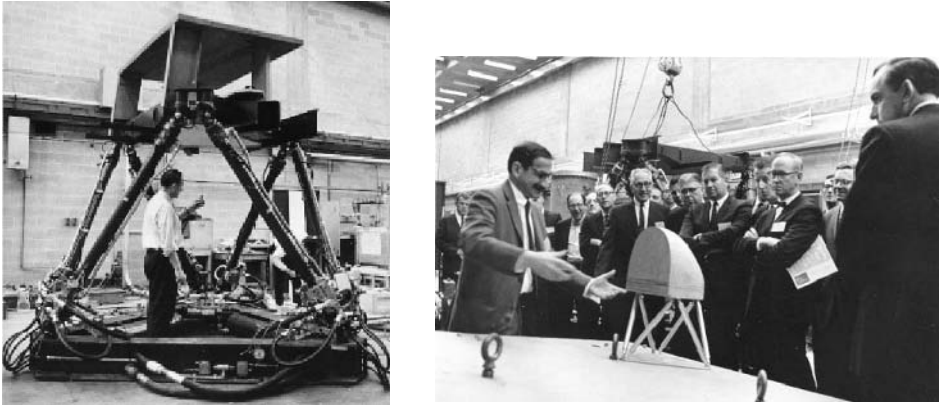


Figure 1.9. On the left the first flight simulator designed in the mid 1960s by Cappel, and on the right a demonstration by K. Cappel to the management of the Franklin Institute (courtesy of Klaus Cappel)

the initial response of his management to his innovative design was very negative (70)... as can be seen on the picture (1.9).

Nowadays, flight simulators of all sorts use the principle of Gough's and Cappel's platform architecture¹. However, it is also used in many other simulators, sometimes surprisingly, as we will see in the next chapter.

But parallel robots were also considered for other applications than flight simulator. According to Ilian Bonev (50) Willard L.V. Pollard establishes a design of a parallel robot for automated spray painting although he never build a prototype. But his son, Willard L.V. Pollard Jr, filed on October 29, 1934 a patent that was issued in June 1942 (481) describing his father's invention.

1.3. Needs for robotics

A positioning device for a simulator imposes constraints which may be quite different from those necessary in a robotic system. In the later case, accuracy may be most important (for example for assembly tasks), while the amplitude of motion is less so. Dynamics is also important for tasks involving a contact between the robot and its surrounding, as in grinding or surface following; or for tasks where execution speed is crucial, like pick-and-place operations, which need a robot with a very light moving part, a so-called *fast robot*.

Another important concept is the *compliance* of the robot. In general

¹see the *Simulator* section in the references Web page

when the end-effector of a serial robot is submitted to external forces/torques there will be slight changes in the pose of the end-effector which are due to backlash in the drive, flexure in the links; these cannot be observed by the internal sensor of the robot, and therefore cannot be corrected using the robot control, hence the name *passive compliance*. In many applications, for example in the machine-tool industry, this compliance has a very negative effect.

Closed-loop kinematic chain stiffness is in general much higher than that of an open-loop structure, and the deformations due to passive compliance will often be measured easily. Elasticities could also be added voluntarily in order to increase the passive compliance (which may be useful in some phases of tasks like assembly) while the controlled actuators could be used in order to obtain a fixed behavior model. For instance, they could be made to be very stiff along a certain direction, and soft in the two orthogonal directions. This type of control is called an *active compliance*, as it makes the manipulator controls intervene actively.

We will now define the types of mechanisms that will be studied in this book.

1.4. Parallel robots: definition

1.4.1. GENERALIZED PARALLEL MANIPULATORS: DEFINITION

General parallel manipulators can be defined as follows:

A generalized parallel manipulator is a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains.

1.4.2. PARALLEL MANIPULATORS

This definition of generalized parallel manipulators is very open: it includes for instance redundant mechanisms with more actuators than the number of controlled degrees of freedom of the end-effector, as well as manipulators working in cooperation .

We will deal mainly with mechanisms with the following characteristics:

- at least two chains support the end-effector. Each of those chains contains at least one simple actuator. There is an appropriate sensor to measure the value of the variables associated with the actuation (rotation angle or linear motion).
- the number of actuators is the same as the number of degrees of freedom of the end-effector.
- the mobility of the manipulator is zero when the actuators are locked.

This type of mechanism is interesting for the following reasons:

- a minimum of two chains allows us to distribute the load on the chains
- the number of actuators is minimal.
- the number of sensors necessary for the closed-loop control of the mechanism is minimal.
- when the actuators are locked, the manipulator remains in its position; this is an important safety aspect for certain applications, such as medical robotics.

Parallel robots can therefore be defined as follows:

*A **parallel robot** is made up of an end-effector with n degrees of freedom, and of a fixed base, linked together by at least two independent kinematic chains. Actuation takes place through n simple actuators.*

1.4.3. FULLY PARALLEL MANIPULATORS

Parallel robots for which the number of chains is strictly equal to the number of d.o.f. of the end-effector are called *fully parallel manipulators* (186; 475).

Gosselin characterizes fully parallel manipulators by the equation

$$p(n - 6) = -6 , \quad (1.1)$$

where p represents the number of chains and n the number of rigid bodies within a chain. Earl (150) also defined a parallelism index with the formula

$$d = \frac{k}{l - 1} , \quad (1.2)$$

where k represents the number of independent loops, i.e. the difference between the number of joints with one degree of freedom and the number of moving bodies; and l is the number of degrees of freedom of the end-effector. This index varies between 0 and 1: 1 for a fully parallel robot, and 0 for a serial robot. Remember that, in certain cases, a manipulator which is not fully parallel may have a parallelism index of 1; see Chapter 2. More recently Rao, has defined various parallelism indices for planar robots (496).

1.4.4. FULLY PARALLEL MANIPULATORS: ANALYSIS

The definition of fully parallel manipulators allows us to characterize chains. There are two main cases: *planar* robots (three degrees of freedom in the plane), and *spatial* robots, which do not move just within a plane.

1.4.4.1 Planar robots

A fully parallel planar manipulator has an end-effector with three degrees of freedom, two translations and one rotation. Planar robots with less than three degrees of freedom will be mentioned only briefly in this book. Three chains support the end-effector; the chains are attached to the end-effector at three points: generically the end-effector is a triangle.

The fact that the mobility is zero when the actuators are locked, and that it becomes 3 when the actuator degrees of freedom are added, can be used to characterize the chains.

It is difficult, however, to define a general mobility criterion for closed-loop kinematic chains, as Hunt (248) and Lerbet (359) already noted (a full section on mobility indices may be found in the references Web page). Classical mobility formulae can indeed lead us to ignore some degrees of freedom. Grübler's formula is nevertheless generally used: it gives a planar mechanism mobility m

$$m = 3(l - n - 1) + \sum_{i=1}^n d_i, \quad (1.3)$$

where l represents the total number of rigid bodies of the mechanism, including the base, n is the total number of joints and d_i the number of degrees of freedom of joint i . Note that if m is negative the mechanism is deemed *overconstrained*.

If we assume that the three chains are identical, and if n_1 represents the number of rigid bodies in a chain, there will be a minimum of $n_1 + 1$ joints with one degree of freedom connecting them, of which one will be actuated. Then

$$l = 2 + 3n_1, \quad n = 3(n_1 + 1), \quad \sum_{i=1}^n d_i = 3(n_1 + 1),$$

thus we get

$$3 = -6 + 3(n_1 + 1),$$

and thus $n_1 = 2$. A chain is therefore made of two rigid bodies linked by a joint. Each of these rigid bodies is linked by a joint either with the base or with the end-effector. There will thus be three independent joints within the chain.

It is worth noting, however, that if the actuated joints are locked we will have

$$l = 5 \quad n = 6 \quad \sum_{i=1}^n d_i = 6,$$

so that the mobility is zero.

1.4.4.2 General case

Fully parallel robots with m degrees of freedom possess m chains supporting the end-effector. If these chains are identical, Grübler's formula for three-dimensional mechanisms may be written as

$$m = 6(l - n - 1) + \sum_{i=1}^n d_i . \quad (1.4)$$

The use of this strictly combinatorial formula can sometimes lead to mistakes because it does not take the geometric relations between the joints into consideration. For example, mechanism with 0 mobility, or even over-constrained may have in fact finite mobility because of the dependency between the constraints.

The most famous counter-examples are Cardan's joint, and Bennet and Goldberg's mechanisms, which are called *paradoxical* mechanisms. In order to take geometry into account, several methods have been considered, for example that of Angeles (10), Gogu (185), Hervé (226), Jin (290) or Martínez (384)*. Their application has however not been generalized yet. Here, Grübler's formula has been deemed sufficient for a preliminary analysis.

If n_1 represents the number of bodies within each chain, and n_2 the number of degree of freedom of the joints of each chain, we have

$$m = 6 + 6 mn_1 - 5 mn_2 .$$

Complete solutions for this equation are sought for different mobility values, and in each case we will look for a solution with a minimal n_2 (the reduction of the number of joints is a way of decreasing the positioning errors of the end-effector). Solutions are obtained only for the following (m, n_1, n_2) triplets: (2, 3, 4), (3, 4, 5), (6, 5, 6).

It is easy to show that in these cases, the mechanism mobility becomes zero when each actuator is locked. We note also that it will not be possible to build fully parallel manipulators with identical chains if the desired mobility is 4 or 5. Different types of chains will have to be introduced. Note that this claim has been discussed in the community as researchers have exhibited structures with identical chains that *in theory* will have mobility 4 or 5. This may be easily explained when looking at the proof of the Grübler formula (see exercise 1.1) that is just based on the difference m between the number of linear equations that relates the generalized velocities of each body in the mechanism to the joint velocities, and the number of generalized and joint velocities, and therefore assumes that the system has full rank. The equations are dependent upon the geometry of the system, and for a mechanism of a given type with arbitrary geometry the full rank assumption

will be true, and the Grübler formula will give the right mobility. But for some particular geometries one (or more) of the equations may just vanish, or equations will become dependent, inducing a loss of rank, and the mechanism will have a higher mobility than the general case, either with an infinitesimal motion (the rank drops only at a given configuration) or with a finite motion. But this drop of rank may usually happen only if very strict geometrical conditions are fulfilled: *in practice* these conditions may not be exactly verified (for example because of manufacturing tolerances) and the mechanism that has in theory n d.o.f. will in fact exhibit m d.o.f (this will be illustrated in the next chapter). Consider for example the planar mechanism described in figure 1.10: with its three revolute joints at A_1, A_2, C , its mobility according to Grübler formula is 0, and this is verified for generic positions of A_1, A_2 and lengths l_1, l_2 . But if $l_1 = l_2$ and $A_1 = A_2$ the mechanism exhibits one d.o.f. (a rotation around A_1). In practice the mechanism may have a different mobility: for example if there is no clearance at $A_1 = A_2$ but the tolerances on the lengths lead to $l_1 \neq l_2$, then the mobility will be 0, as the mechanism cannot even be assembled. On the other hand, clearance at A_1, A_2 may allow us to get a mobility if sufficient forces/torques are applied at C . Determining if a

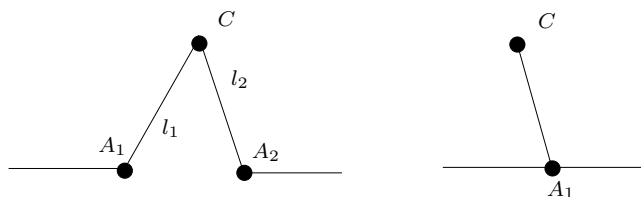


Figure 1.10. A mechanism (on the left) that has generically a mobility 0 may exhibit a larger mobility in specific cases (on the right)

structure that has in theory n d.o.f. (with $n < 6$) will still exhibit the same number of d.o.f. being given bounded manufacturing tolerances, is still an open problem. It must also be noted that the Grübler formula just counts the number of d.o.f. of the end-effector, and neither provides their type nor indicates if the d.o.f. are the same everywhere. Indeed some mechanism may exhibit various d.o.f. For example the surprising *Dymo* robot (667) has 3 d.o.f. which can be of 5 different types: 3 spatial translations, 3 orientations, 3 planar d.o.f., a locked mode (no motion) and a mix of translation and orientation.

Finally note that in the Grübler formula we just count the number of d.o.f. constrained by the joints. The technological means that will be used to realize these constraints do not play a role in the mobility of the mechanism. A direct consequence is that mechanisms that differ by their joint nature

may be equivalent from a mobility view point. For example replacing a prismatic actuated joint by a revolute one (or by more complex components such as cams) in a given mechanism will not change the mobility (but may have an high impact on other kinematic performances). Consequently, mechanisms that differ from another mechanism only by such arrangement will belong to the same kinematic class and can hardly be qualified as "novel".

1.5. Contents

This book is set around different problems that occur for the design, analysis and use of parallel robots.

In Chapter 2, *Structural Synthesis and Architectures*, synthesis methods for designing robot, with given d.o.f. will be presented, and various possible mechanical architectures of parallel robot will be exposed, as well as typical examples of applications.

Chapter 3, *Inverse Kinematics*, will show how to calculate the joint coordinates according to the desired end-effector pose.

The inverse relation allowing us to determine the end-effector pose from the joint coordinates measurements will be studied in Chapter 4, *Direct Kinematics*.

Chapter 5, *Velocity and Acceleration analysis*, will establish the relations between the end-effector velocities and the actuator velocities, together with their limits. Similar relations will be established for accelerations, and an accuracy analysis will be proposed.

Chapter 6, *Singular Configurations*, will discuss special poses for parallel robots, where the mobility of the structure is no longer zero, although the actuators are locked.

Chapter 7, *Workspace*, will deal with the calculation of the boundaries for the possible motions of a parallel robot when it is subjected to limitations on the values of its joint coordinates and to constraints on the motion of its passive joints. Motion planning within a workspace will also be discussed.

Chapter 8, *Static Analysis*, will then give a detailed explanation of the relations between the forces acting on the end-effector and the forces exerted by the actuators, as well as the inverse relation. We also examine parallel robot stiffness in the same chapter.

Chapter 9, *Dynamics*, will then try to show how to calculate the forces that should be exerted by the actuators so that the end-effector reaches a certain speed and acceleration rate. The inverse relation will also be studied.

We will consider the problem of parallel robots *Calibration* in Chapter 10. A book on mechanisms would not be complete without a chapter on design; Chapter 11, *Design*, will look into that.

Finally, as many of the algorithms presented in this book rely on system solving and *interval analysis* (a not so well known domain), we present in two appendices a brief introduction to these topics.

Note that there is no chapter dealing with control per se, although this issue will be addressed in some chapters. We have focused this book on parallel robot modeling, an already very large domain, that it is necessary to master before addressing control problems.

1.6. Exercises

Exercise 1.1: Prove the Grübler formula

Exercise 1.2: Show that the mobility of fully parallel spatial robots with identical chains is null when each actuator is locked.

Exercise 1.3: Show that there are no fully parallel robots with identical chains that possess a mobility of 4 or 5, using Gosselin's formula (equation 1.1).

Problem 1.1: Find structures of parallel robots with identical chains that have exactly 4 or 5 d.o.f. even if manufacturing tolerances are taken into account

Structural synthesis and architectures

This chapter will present general methods to determine the possible structures of parallel robots that have a given number of d.o.f. and then a comprehensive enumeration of parallel robots mechanical architectures as described in current literature on the subject¹. The list will be classified by increasing numbers of degrees of freedom, from 3 to 6, for the end-effector. In the figures, actuated joints will be represented by arrowed vectors, while passive joints will be indicated, if necessary, by dashed vectors.

Examples of the use of parallel robots for very diverse applications will be discussed. Representative types of parallel manipulator will then be chosen to be studied more specifically in the remainder of this book.

2.1. Introduction

Because of serial robot deficiencies, a few researchers have tried to develop new robotic structures. Minsky in 1972 (423), and Hunt in 1978 (248), proposed parallel structures. But such structures were based on the ingenuity of the researchers and not on a systematic approach. *Structure (or type) synthesis* is the domain in which a methodology is used to try to generate all the structures that have a desired kinematic performance. In this chapter we will restrict this kinematic performance to the number of degrees of freedom as we will see that it is quite difficult to consider other kinematics properties. This is one key issue for parallel robots: as opposed to serial robots, for which there is a limited number of possible mechanical structures, there is a very large variety of possible close-loop mechanisms, and it is usually admitted that the topology of the structure will affect the overall performance of the robot. Note that sometime the word *synthesis* is also used for *dimensional synthesis* i.e. to determine what should be the geometry of a given structure to reach some kinematic performance: this very important issue will be addressed in the "Design" chapter.

¹Obviously all possible mechanical architectures cannot be presented in this book, but the Web site: www-sop.inria.fr/coprin/equipe/merlet/Archi/archi_robot.html presents a comprehensive description of more than 150 mechanical architectures. Furthermore it is very often quite difficult to determine who has proposed for the first time a given architecture: we have done our best to reference the earliest journal paper that proposes a full analysis of the robot.

Starting around about the year 2000 there has been a very large increase in the number of papers describing new structures for parallel robots (although we will see that many of them were known well before), especially ones having less than 6 d.o.f.. There is an overriding motivation behind such efforts: for many applications, less than 6 d.o.f may be needed. For example, for milling operation in the machine-tool domain, the rotation of the platform around its normal is not needed, as the spindle will manage this d.o.f.: hence only 5 d.o.f. are needed. In that case the usual claim is that machines having only the necessary d.o.f. will be less costly than the usual 6 d.o.f. parallel robot, as they have less actuators, and that the control will be simpler. In my opinion such a claim is a complex issue, and its veracity is difficult to establish in general terms. First of all, the cost of the machine is only a part of the operating cost, and various factors may increase the cost of less than 6 d.o.f. robot:

- maintenance and fabrication cost may be higher if the chains of the robot involves different actuators and sensors
- the robot will most probably exhibit *parasitic motion* (i.e. motion that is not wanted) that will affect its performances, and will result in poorer quality for the task
- on the other hand the redundancy of a 6 d.o.f. robot may be used to improve the quality of the task (see for example (412) for the use of the redundant mobility of a 6 d.o.f. robot for improving the stiffness of a 5 axis milling machine, and the "Workspace" chapter)

Hence only a careful economic and technical analysis will allow us to determine the best structure for a given task. Therefore, in my opinion, papers presenting new architectures with less than 6 d.o.f. should cover also basic issues (such as kinematics, workspace, singularity) and should provide a critical analysis of the performances of the proposed robot (e.g. the amplitude of the parasitic motion).

Nevertheless it is clear that structural synthesis is an exciting domain with a large number of open problems.

2.2. Structural synthesis methods

Structural synthesis is a old problem in mechanism theory (156). We have seen that mobility formulae may be used for that purpose (see for example (7; 290; 642)) but this approach may have difficulties to deal with the synthesis of robots with less than 6 d.o.f.

We will focus here on the most widely used synthesis approaches (and their variants): *graph theory*, the *group theory* and the *screw theory* approaches.

2.2.1. GRAPH THEORY

The enumeration of all possible structures having a given number of d.o.f. may be conducted by considering that there is only a finite set of possible kinematic pairs, and hence a very large, but finite, set of possible structure combinations. But the manipulation of the combination of kinematic pairs should rely on a formalism that allows one to determine automatically the number of d.o.f. of the structure. Freudenstein was the first to propose the use of graph theory for that purpose. He devised a graph scheme in which the vertices correspond to the links of the mechanism, and the edges correspond to the joints. Initially graph description was used as a simple graphical representation of a mechanism but further works showed that graph theory was a powerful tool to manipulate these graphs, especially with computers. Graph theory was used early by Earl (150) to devise new parallel robots architectures. But graph theory has two drawbacks which are difficult to overcome when dealing with parallel robots:

- *isomorphism*: to avoid the combinatorial explosion of an exhaustive enumeration of all possible mechanisms, contracted graphs are used. But there is no longer a one-to-one correspondence between the graphs and the mechanisms: a given mechanism may be represented by different graphs, and redundant graphs should be eliminated from the enumeration. This complex issue has never been solved completely
- graph theory makes extensive use of the mobility formulae (such as Grübler formula), and for spatial structures there are many mechanisms that do not obey this formula (e.g. the parallelogram) and that play an important role in structural synthesis. Hence such elements cannot be ignored

A somewhat related approach was developed by Assur (84): instead of kinematic chains, basic *families* of mechanisms are considered and a joint simplification method is used to remove joints so that the mechanism has the desired number of d.o.f.: but this method seems to be difficult to use for spatial mechanisms.

2.2.2. GROUP THEORY APPROACH

2.2.2.1 *The Lie group and subgroups of displacement*

The set $\{D\}$ of displacements, which represents the motion of rigid body, have the special structure of a group, the *displacement group*. This group is directly related to the *Special Euclidean matrix group* $SE(3)$ which is defined as the set of matrices of the form:

$$SE(3) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{pmatrix} \right\}$$

where R is a rotation matrix and \mathbf{p} a 3-dimensional vector. $SE(3)$ is a *continuous* group, and any open set of elements of $SE(3)$ has a one-to-one map onto an open set of R^6 . In mathematical terminology $SE(3)$ is a *differentiable manifold* that is called a *Lie group*. There are subgroups of the group of displacements that will play an important rule in structural synthesis. A comprehensive list of these Lie subgroups is given by Hervé (228)*. Let us just mention some of these subgroups:

- $\{T(\mathbf{u})\}$: the translations parallel to the vector \mathbf{u}
- $\{T\}$: all the spatial translations
- $\{X(\mathbf{w})\}$: all the translatory and rotary motions about all axes that are parallel to the axis defined by the vector \mathbf{w} . Such a motion is also called a *Schönflies motion*, as this mathematician extensively studied this type of motion (522).
- $\{Y(\mathbf{w}, p)\}$: all the planar translations perpendicular to the vector \mathbf{w} combined with a screw motion of pitch p along any axis parallel to \mathbf{w} (called also *pseudo-planar motion*)

Serial arrangement of two elements of a subgroup is called *composition*, and may lead to an element of another subgroup. For instance, combining three elements of $\{T(\mathbf{u})\}$ with vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ may lead to an element of the translation motions subgroup $\{T\}$ in space if certain constraints on the vectors \mathbf{u}_i are satisfied.

But for us the most important operation is the *intersection operation* obtained when elements of subgroups act on the same rigid body. Hervé (228)* presents rules that regulate this operation (for instance the intersection of two elements of $\{D\}$ is an element of $\{D\}$).

An interesting intersection case is that of the intersection of two elements of the $\{X(\mathbf{w})\}$ and $\{X(\mathbf{w}')\}$ subgroups with $\mathbf{w} \neq \mathbf{w}'$ is an element of the $\{T\}$ subgroup.

2.2.2.2 Subgroup motion generators

Displacements of Lie subgroups are generated by kinematic chains that are called *motion group generators*.

These generators plays an important part for the architectural possibilities of parallel robots. It therefore seems necessary to list them briefly. We present here the generators of the $\{D\}$ group (that will be important for the design of 6 d.o.f. robots) but Hervé (228)* presents generators for the other subgroups. We assume that only R, P, S types of joints are used.

The $\{D\}$ generators can be classified into 4 main types: $RRPS, RPRS, PRRS, RRRS$ (figure 2.1), with constraints on the positions of the joint axes.

The $RRPS$ type is constituted of an universal joint, followed by a prismatic joint which is itself followed by a ball-and-socket joint. The $RPRS$

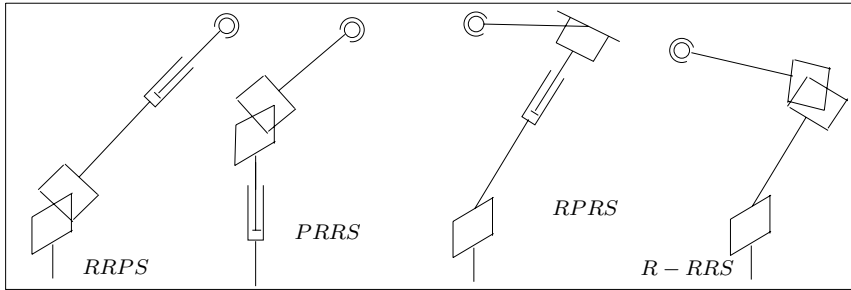


Figure 2.1. $\{D\}$ motions group generators using joints of the R, P, S type.

type is made of a revolute joint on whose axis a prismatic joint is set, followed by a revolute joint, itself followed by a ball-and-socket joint. The $PRRS$ generator is composed of a prismatic joint followed by a universal joint at the end of which is a ball-and-socket joint. Lastly, the $RRRS$ type is made of a revolute joint followed by a universal joint, on which a ball-and-socket joint is set.

2.2.2.3 Type synthesis based on group theory

The basic principles of parallel robot synthesis based on group theory are as follows:

1. determine to which subgroup S the end-effector should belong to have the desired d.o.f.
2. determine all the possible subgroups to which the different kinematic chains that will constitute the legs of the robot may belong so that the intersection of these subgroups belongs to S .
3. determine all the motion generators of these subgroups: they will constitute the kinematic chains of the robot

For example if the robot must have 6 d.o.f, then all its legs should belong to the subgroup $\{D\}$. Various authors have presented examples of this approach, allowing us to obtain an enumerative list of possible structures $(12)^*$, $(230)^*$, $(302)^*$, $(361)^*$.

Synthesis based on group theory is still open for investigation as not all the group structures have been exploited: for example Rico (507) has shown that the concept of *conjugacy classes* of a group may be used to explain the mobility of paradoxical mechanisms.

2.2.3. THE SCREW APPROACH

2.2.3.1 Basics of screw theory

The group theory approach allowed the discovery of numerous new possible structures. Nevertheless the group $\{D\}$ has more special properties which are not reflected by its Lie group structure alone. Consequently the group theory approach has difficulty accounting for very special cases of mobility such as paradoxical mechanisms.

We may extend the Lie group concept by considering the tangent space at the identity element, which is a vector space called the *Lie algebra* of the Lie group. As soon we have chosen an origin for $SE(3)$ the associated Lie algebra $se(3)$ is the vector space of all instantaneous velocities whose elements are 6-dimensional vectors 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 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

$$\boldsymbol{\Omega} \cdot \mathbf{M} + \mathbf{v} \cdot \mathbf{F} = 0$$

When a kinematic chain is connected to a rigid body the key point is that the possible *instantaneous* velocity twists for the rigid body are reciprocal to the wrenches imposed by the kinematic chains (called the *constraint wrenches*). In other words, the d.o.f. of the rigid body are determined by the constraint wrenches. For a parallel robot the following statements hold:

- the velocity twist of the moving platform is the intersection of those of all the legs of the robot
- the wrench of the moving platform is the union of those of all legs of the robot

2.2.3.2 Type synthesis based on screw theory

Using the above statements, we may devise a synthesis methodology:

1. find the wrench system S that is reciprocal to the desired velocity twist of the moving platform
2. determine the wrenches of the kinematic chains of the robot whose union spans the system S
3. determine all the possible structures of the kinematic chains that will generate the corresponding wrenches
4. as all considered twists and wrenches are instantaneous (synthesis based on screw theory is called a *first order* approach) it is necessary to verify that the mobility of the platform is not instantaneous

The disadvantage of this approach is that it is difficult to automate, especially steps 3 and 4. Group theory has the advantage here: by restricting the possibilities at step 2, the motion generators at step 3 are easily identified. On the other hand, group theory may miss structures that will not fall within the Lie subgroup framework.

Nevertheless, several authors have used this method to generate a large number of structures with less than 6 d.o.f (158; 171; 176), (247)*, (327)*, sometime mixing the screw approach with a mobility formula (118; 246).

In conclusion, although the proposed methods of structural synthesis has allowed us to determine a very large number of structures, this problem is not yet solved.

2.2.4. STRUCTURAL SYNTHESIS AND OTHER KINEMATIC PERFORMANCES

Many parallel robots have been proposed, and we will illustrate some of them in the next sections. But an important question should be addressed: *may we design a synthesis approach that deals with any kinematic performance other than the number of d.o.f. of the robot ?*

This is possible to some extent for serial robots. For example, we may compare the workspace ability of a Cartesian 3 d.o.f. translation robot with that of a spherical $3R$ structure. Indeed if we assign a stroke of L to the linear actuator of the Cartesian robot, the workspace volume will be L^3 , while the workspace volume of a $3R$ robot whose links have a length L will be $4\pi(2L)^3/3 \approx 33.5L^3$. Hence from the workspace point of view the $3R$ structure is superior to the Cartesian structure, but the opposite will hold for positioning errors. Consequently, if a task involves both workspace and accuracy constraints we cannot directly state which one of the structures is more appropriate.

For parallel robots, it seems that a simple design rule such as that derived for $3R$ and Cartesian robots, cannot be established (although some attempts have been made for planar robots (496)* or in a more general case (74; 580)), as their performance will be highly dependent on their dimensioning, as will be illustrated in the following chapters. Consequently, structural synthesis cannot be separated from *dimensional synthesis* (a complex issue that will be addressed in the "Design" chapter) and we put forward the following conjecture: *a parallel robot with well-designed dimensions will exhibit overall better performance compared to another parallel robot whose structure seems to be more appropriate but whose dimensions have been poorly chosen.*

This conjecture is also an answer to a question that is frequently asked by end-users: *what is the structure that is most appropriate for my task ?.*

2.2.5. STRUCTURAL SYNTHESIS AND UNCERTAINTIES

Uncertainties are inherent in robotics: we have manufacturing tolerances for the mechanism, sensor measurement noise, tracking errors due to control, and uncertainties in the task. Some of these errors may be corrected (for example through calibration, improvement in the control hardware, . . .) but cannot be eliminated. As mentioned earlier, manipulators with less than 6 d.o.f. will exhibit parasitic motion because their design does not exactly satisfy the stringent geometrical conditions that are usually necessary to obtain the desired d.o.f (see for example (465)*). A preliminary analysis of this problem was performed for the 3-*UPU* robot (figure 2.10), a robot that theoretically has 3 d.o.f. but that may exhibit significant rotational motion, and is very sensitive to manufacturing tolerances (212; 463; 626).

An open problem for such a structure is to determine the influence of the manufacturing tolerances on the amplitude of the parasitic motion. Furthermore, this should be done over the whole workspace of the robot (i.e. we must determine what is the maximum of the amplitude of the parasitic motion over the workspace): this will be called the *direct tolerance problem*.

An even more complex issue, the *inverse tolerance problem*, is to determine what should be the manufacturing tolerances for a given structure so that the maximum of the amplitude of the parasitic motion is lower than a given threshold.

Both problems are very difficult to solve (and more so as they cannot be decoupled from the dimensional synthesis). In my opinion, this is one of the biggest kinematic challenges as solutions of these problems are needed to determine if a structure may be used for a given task.

The next sections will present various architectures for robots with 3 to 6 d.o.f. Most of these structures have been found through the synthesis methods that have been presented in the previous sections (although some of them were introduced well before). Clearly not all architectures can be presented: we have chosen a representative subset of robots which have either been extensively studied both theoretically and experimentally, or present theoretical interest. Drawings of other architectures can be found at the references Web page. Still, determining if a given structure has been previously published (or is only a variant of a known architecture) is a difficult task.

2.2.6. NOTATION FOR PARALLEL ROBOTS

There is no uniform notation in the literature for describing the mechanical architecture of a parallel robots. For example a Gough platform may be denoted either as a 6-6 robot (i.e. a robot having 6 attachment points for the legs on the base and on the platform) or as a 6-*UPS* robot (i.e. having 6

legs having a *UPS* mechanical structure) or as a *3T-3R* robot (i.e. having 3 translational and 3 rotational d.o.f., the translations (rotations) being along (about) axes that are *fixed* in the reference frame). None of these notations is satisfactory: the 6-6 notation does not describe the structure of the legs; the 6-*UPS* notation does not indicate that some legs may share the same attachment point, or that the geometry of the legs satisfies some specific constraints; and the *3T-3R* does not indicate the mechanical architecture of the robot.

Such lack of uniform notation is unfortunate because a complete textual description of the mechanical structure of robot is a pre-requisite for the design of software that will provide an automated analysis of performance. The development of such a tool may seem a long term objective, but we will see in the following sections that many problems related to parallel robots may be solved for any robot structure as soon as the problems have been solved for a few classes of robots. Thus a process using a library of known solutions may be the right way to develop this design library. Nevertheless we will use available notations as their meaning will be clear according to the context.

2.3. Planar robots

Conventionally, revolute joints are labeled *R*, while prismatic joints become *P*. Actuated joints are underlined.

2.3.1. 3 D.O.F. MANIPULATORS

Consider a moving platform, within the plane, whose three degrees of freedom, the two translations along the *x* and *y* axes, and rotation through an angle θ around the axis *z*, perpendicular to the plane, are to be controlled. A fully parallel robotic structure is sought: according to the definition given earlier, it should possess three independent kinematic chains, actuated by three actuators. As each of these chains must be linked to the ground, and to the moving platform at the same time, there will then be three attachment points on the ground and three on the moving platform. One can therefore consider, still in a general manner, a triangular moving platform. We showed in the previous chapter that under these conditions, each of the chains is constituted of two rigid bodies linked together by a joint, and that they have a total of three joints.

A chain can be described by the sequence of these three joints, from the base upwards. The chains can present the following sequences: *RRR*, *RPR*, *RRP*, *RPP*, *PRR*, *PPR*, *PRP*, *PPP* (figure 2.2). The sequence *PPP* is excluded as the joints motion must remain independent.

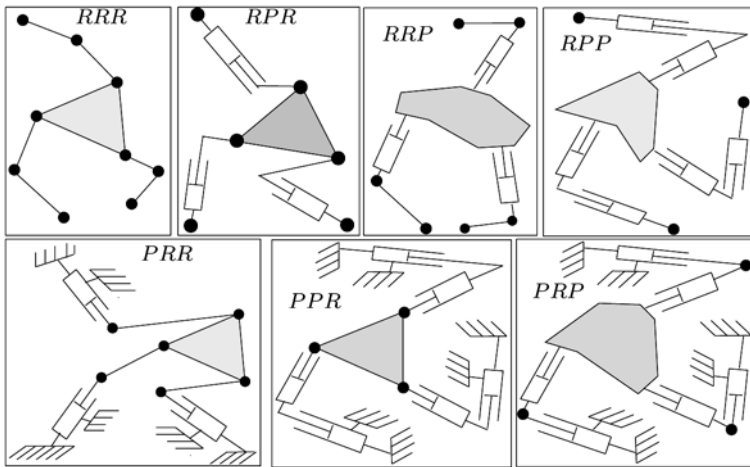


Figure 2.2. The different fully parallel planar robots with three degrees of freedom and identical chains.

It is noteworthy that with a simple exchange of the base and the moving platform, the robots of the RRP type become equivalent to PRR , and the RPP equivalent to PPR . We have not specified the actuated joint, as it can be any of the three. We should generally avoid placing the actuator on the end effector in order to lighten the weight of the moving equipment. It is also possible to build robots with different types for each chain.

A reference book on kinematics, static analysis and stiffness of these robots has been published by Duffy (144), and there have been various studies of these robots: $3\text{-}RRR$ robots were mentioned by Rooney (512), Hunt (250), and were extensively studied by Kassner (304) and Gosselin (186). Additional works are related to the dimensional synthesis problem (16; 536), singularity analysis (51; 199), workspace analysis (175; 336; 373) stiffness (319), balancing (282).

The company Googol Technology is marketing such robots for educational purposes with the feature that the end-effector is reduced to a point, so that the robot is redundant[□] G_{oo} (figure 2.3).

$3\text{-}RPR$ type robots have been extensively studied: synthesis (222; 435), kinematic and singularity analysis (108)*, workspace (255)* and joint force optimization (148).

Robots of the $3\text{-}PRR$ type have been mentioned by Hunt (249), and some of their characteristics were studied in (197) and have been proposed for a high speed robot (297)*. The company Hephaist Seiko is using this architecture for its triaxial stage NAF3[□] HS .

Some papers have presented studies that are generic i.e. they may deal

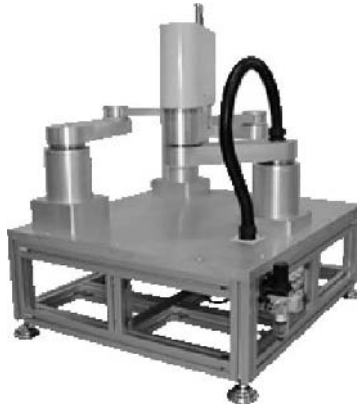


Figure 2.3. The redundant planar 3-RRR robot GPM of Googol Technology (courtesy of Googol technology)

with any robot with R, P joints: synthesis (54)*, workspace (255; 408; 617), singularity (352; 526), forward kinematics (404), architecture selection (224)

Planar parallel robots with 3 d.o.f. and R, P joints have thus been extensively studied. We mention other works related to planar robots:

- the replacement of rigid links by wires. This topic will be studied in another section
- variant of the 3-PRP structure with a triangular arrangement of the linear actuators (58; 127)
- the use of holonomic pairs as actuators (223)*
- redundant robot (for singularity avoidance) (166; 378)
- micro-robot with flexible joints (98; 297; 330; 471; 636)

2.4. Spatial motion robots

2.4.1. JOINTS AND ACTUATORS

For parallel robots, the most commonly used joints are, in increasing order of degrees of freedom: revolute, prismatic, universal and ball-and-socket joints. Such joints, sometimes with reduced motion ability, are available commercially at a low cost. A problem still remains, however, with the ball-and-socket joints, as usually models have a reduced range of motion (typically ± 15 degrees) or do not allow the rotation of several bodies around the same point, as some architectures would need.

Enlarging the amplitude of motion has been addressed mainly by companies, and figure 2.4 shows the spherical joints of INA and of Hephaist Seiko. The joint of Hephaist Seiko allows for a misalignment of 30 or 45

degrees with a size from 7.6mm to 7.6cm and a weight from 15g to 6.73kg.

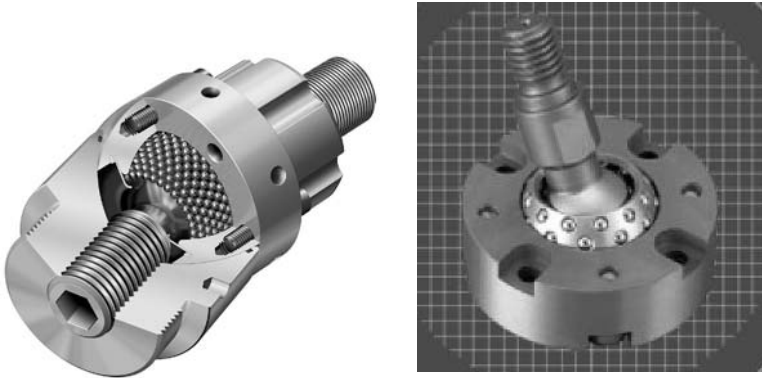


Figure 2.4. Spherical joints of INA and Hephaist Seiko (pictures courtesy of INA-Schaeffler KG and Hephaist Seiko)

As for connecting multiple bodies to the same joint, Bosscher summarizes the solutions that have been proposed (53). For all these solutions, a compromise has to be found between stiffness and range of motion. To improve the stiffness, U joints with skew axes are sometime used (e.g. in the M-850 of Physik Instrumente) as it is claimed that their stiffness is better. Another important point is that we will see that it may interesting for the direct kinematics or for the calibration to have instrumented passive joints (i.e. joints whose motions may be measured at least along one direction).

For small robots in which the motions of the passive joints are of low amplitude, the use of flexible joints (called also *flexure hinge*) has also been proposed, either elastic (605) or using specially designed deformable structures (298; 393; 449; 471; 636) some of which may allow relatively large motion (429).

As for the actuators, all sort have been used: pneumatic, hydraulic, electrical, magnetic, piezo-electric, shape memory alloy, magnetostrictive, ... (consult the "Actuators" section in the references Web page). Specific actuators will be described in the *micro-positioning* section but we can mention the use of two stage struts for parallel robots whose legs are submitted to torsion, a piezo actuator enabling a higher torsional stiffness (444)*. For the sake of curiosity we should mention the use of spreadbands that act at the same time as actuators and joints (521) (figure 2.5). Wire-driven parallel robot have also been intensively studied; as this actuation scheme has a large impact on the analysis, we will devote a full section to that topic. We must also note the increasing use of electrical linear motors, especially for machine-tools.

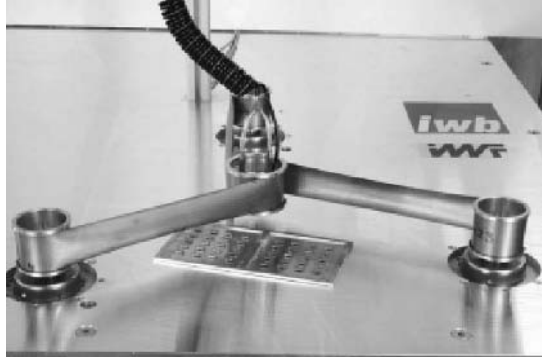


Figure 2.5. A planar parallel robot using spreadband actuators (courtesy of Braunschweig Technical University)

2.4.2. CLASSIFICATION OF PARALLEL ROBOTS.

Parallel robots will be presented by increasing number of d.o.f. of the platform. Clearly, as seen in the synthesis section, not all possible architectures can be presented, and we have selected a few representative robots. Other architectures are presented in the references Web page. Remember also that for robots with less than 6 d.o.f., the presented architecture has only *in theory* the claimed number of d.o.f., as seen in the synthesis section.

2.4.3. 3 D.O.F. MANIPULATORS

2.4.3.1 Translation manipulators

Manipulators with 3 degrees of freedom in translation prove extremely interesting for pick-and-place and machining operations. Results of the synthesis methods for this type of robot may be found in (74)*, (171; 176) (230)*, (247)*, (290; 328).

The most famous robot with three translation degrees of freedom is the *Delta* (figure 2.6), that was initially developed at École Polytechnique from Lausanne (EPFL) by Clavel (99) (see (47) for an history of the development of the Delta and its application). All the kinematic chains of this robot are of the RRP_aR 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 P_a is fixed to this joint, and allows translation in the directions 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.

This robot is marketed by the Demareux company^{□ DD} and by ABB under the name IRB 340^{□ ABB} *FlexPicker*, while CSEM^{□ CS} offers a micro

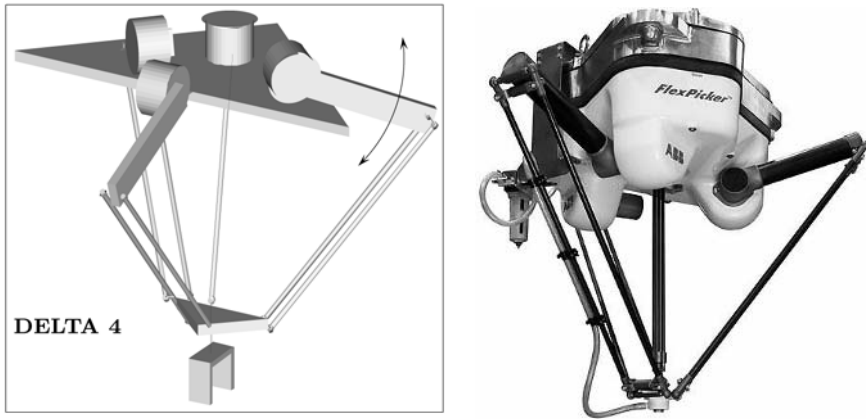


Figure 2.6. The Delta robot (from Clavel (99)) and one of its industrial version, the FlexPicker IRB 340 (courtesy of ABB)

version. We will see that the *Delta* has been used in a large number of applications.

In term of group theory, the chain of the *Delta* robot constitutes a $\{X(\mathbf{w})\}$ generator, and therefore the intersection of 3 of those generators, with axes that are not parallel, will allow us to obtain a generator of the sub-group $\{T\}$ of the translations in space. Another member of this family is the Star robot (227) (figure 2.7), with the notable difference that the Star is over-constrained (each leg restricts two rotational degree of freedom of the platform) while the *Delta* is not.

It is important to remember that the *Delta* ancestor is a mechanism described in 1942 by Pollard (481), intended to be used for car painting (figure 2.7). This mechanism presents three revolute actuators that orientate three arms, the ends of which are linked to the pod by three articulated links. A problem with this design is that to get only translation of the end effector the three distal links must connect at ball-and-socket joints that share the same center. We have already seen the difficulty of designing such joint, a problem that is elegantly solved by the *Delta* architecture.

It is noteworthy that the rotary actuator and lever part of the *Delta* could be replaced by a linear actuator, as suggested by Clavel itself and Zobel (668) (this type of *Delta* is sometime called a *Linapod* or a *linear Delta*). Such a variant is currently being used for machine-tool (figure 2.86). Note also another variant, the *Triax* proposed by Sheldon (figure 2.87).

Work is also going on at EPFL for the development of the *Delta*³, a micro-robot based on this architecture and using flexure hinges (102),(449)*.

Reboulet (501) suggests a mechanism, the *Speed-R-Man*, the structure of

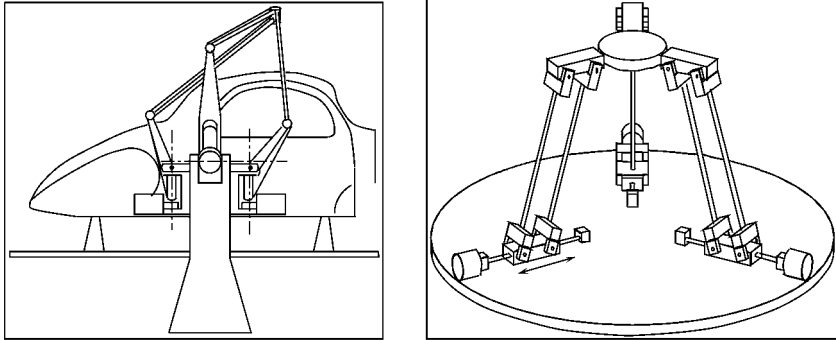


Figure 2.7. Robots with three degrees of freedom in translation. On the left, the Pollard mechanism (from Pollard (481)). On the right, the *Star*: the rotary actuators lead the nut of a ball screw on which a parallelogram is articulated. The parallelogram at the other end is linked by a revolute joint to the end-effector (from Hervé (227)).

which is very similar to the *Delta*, although the rotary actuators are replaced by two linear actuators acting on the same point. His mechanism therefore is redundant, and presents interesting speed characteristics. The interest of the redundancy lies in the fact that the maneuverability is improved, and that the joint forces can be reduced for a given external force.

Another interesting member of the same family is the *Orthoglide* robot (figure 2.8) developed for machine-tool application (618). The main interest of this robot is that it presents relatively homogeneous kinematic performances in its useful workspace.

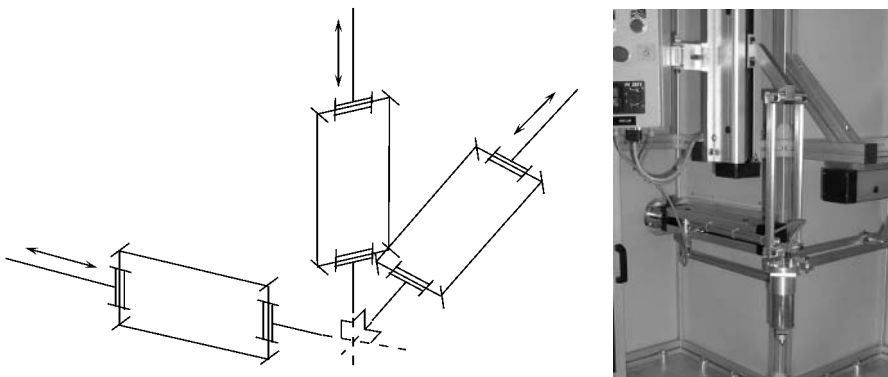


Figure 2.8. The *Orthoglide* robot (from Wenger (618)).

It is not necessary to have revolute joints to be members of the same family. For example Kong proposes the $3\text{-}\underline{C}RR$, a robot with cylindrical joints (figure 2.10) and some of its variants (326).

Another representative 3 d.o.f. robot is the *Tricept* that issued from a patent by Neumann (446). This mechanism presents an end-effector that possesses a stem which is free to translate along its axis. The stem is linked at its base by a universal joint, forbidding the stem to rotate around its axis; three chains of the *RRPS* type act on the end-effector (figure 2.9).

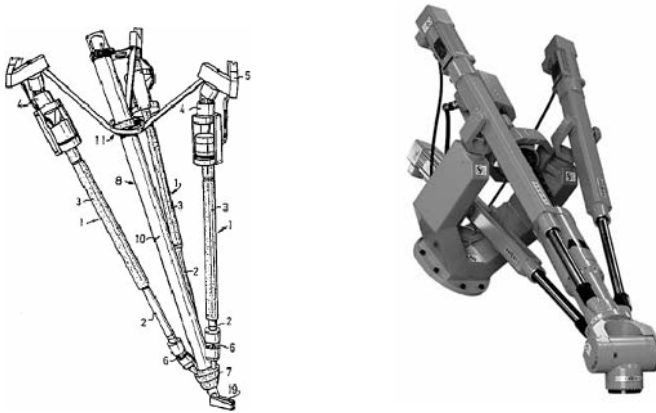


Figure 2.9. On the left, the Neumann patent (courtesy United States Patent and Trademark Office). A stem of varying length, set on a universal joint, connects the end-effector to the base. Three linear actuators allow the modification of the end-effector position (from Neumann (446)). On the right, the Tricept IRB 940 from ABB (courtesy of ABB)

GEC-Marconi (now BAE Systems) uses this architecture to build a bulky parallel-serial hybrid robot designed for assembly, the *Tetrobot*: orientation of the end-effector was obtained through a classical serial wrist (578). To the best of our knowledge this robot has never been marketed.

In 1992 a small company directed by Neumann, Neos Robotics (now SMT Tricept), started marketing the Tricept *TR600* with some success, and is now producing a machine-tool five axis version, the *TM 805* presented in 1998. Later on, Comau and ABB added a Tricept to their catalog under the names *HP1* and *IRB 940*. A similar machine, the *MultiCraft 560*, is also sold by MultiCraft. Note that instead of using links with varying lengths in the Neumann mechanism, it is possible to use fixed length links with ends close to the base moving on a curve. For example, in the *Georg V* robot, these extremities move along a tilted line (581). The Tricept is a representative of a family of 3-d.o.f. translation robots in which a passive mechanism constrains the end-effector d.o.f.: other members of this family may be found on the references Web page.

In academic studies the most studied 3-d.o.f. translational robot is the 3-*UPU* robot proposed by Tsai (586) (figure 2.10), which is a special case of the family of 3-*RRPRR* mechanisms (note that 3-*RPRRR* and 3-*PRRRR* may be used as well (74)*,(171)). The academic interest in this robot is

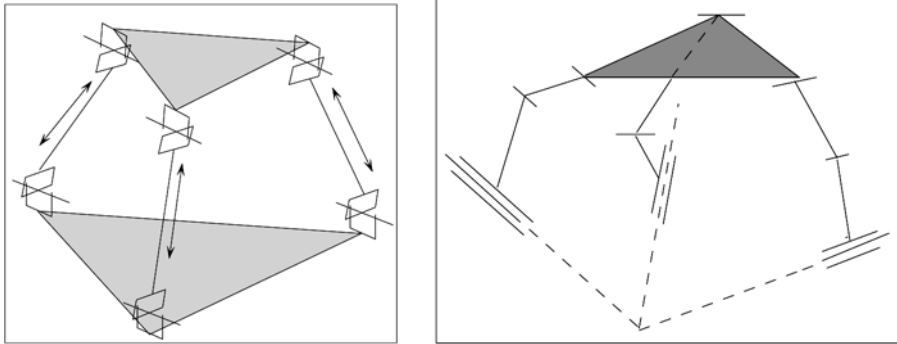


Figure 2.10. On the left, the 3-*UPU* manipulator mentioned by Tsai (586): a convenient choice of the axes of the universal joints enable the end-effector to have only translations. On the right, the 3-*CRR* robot of Kong (326).

justified for various reasons:

- although Tsai has completed an extensive study of his robot (589), strange behavior was observed in the manufactured prototypes: large rotational motions were observed (48) (see an impressive picture of the rotation motion in (212)). The explanation is that this robot is very sensitive to manufacturing tolerances (212; 463) and that it is in a singular configuration in its nominal position, and will remain close to this singularity over large translation motion (48; 626), see section 6.4.2.2
- another interest of this robot is that by modifying the revolute joint axis we will get a rotational robot (229)

According to Carricato, the 3-*PRRRR* robot seems to be more interesting as it does not exhibit the rotational singularities of the 3-*UPU* (74) (a claim that is not exactly true (67)) while the Tricept structure seems to have better workspace and stiffness properties than the 3-*UPU*, according to the designers of the 3-*UPU* (294), who have also compared the workspace and stiffness of the 3-*RUU*, 3-*PUU* (with inclined or parallel rails) and 3-*UPU* robots and favor the 3-*RUU* (590).

2.4.3.2 Orientation manipulators

Manipulators allowing three rotations about one point represent an interesting alternative to the wrist with three revolute joints having convergent

axes classically used for serial robots.

In terms of group theory, the rotary motion of a rigid body around a point N , $\{S(N)\}$ is a subgroup. The Lie subgroups of this subgroup are the subgroups of rotations $\{R(N, \mathbf{u})\}$ about the axis defined by the vector \mathbf{u} and which contain the given point N . The intersection of three elements $\{R(N, \mathbf{u})\}$, $\{R(N, \mathbf{v})\}$, $\{R(N, \mathbf{w})\}$ spans $\{S(N)\}$, as long as \mathbf{u} , \mathbf{v} , \mathbf{w} are linearly independent (a spherical pair is a special example of such intersection, the vectors being mutually orthogonal).

A first possible use of this group theory result in a passive constraint mechanism that allows only rotation of the platform. Three additional kinematics chains which are generator of the motion group $\{D\}$ will be used to actuate the platform. Figure 2.11 presents a wrist using the central mast principle with $\{D\}$ generators of the $RRPS$ types ($PRRS$ may also be used) and an application with the Vertical Motion Simulator (VMS) of NASA.

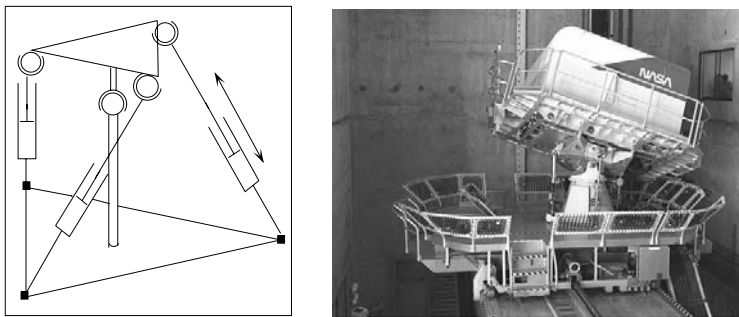


Figure 2.11. On the left, a parallel wrist with the moving plate articulated on a ball-and-socket joint placed on a central mast. The universal joints are represented by a black square. On the right, an example of application with the Vertical Motion Simulator (VMS) (courtesy NASA)

The passive constraint mechanism is not necessary: for example the central mast may rotate while it supports the platform on a revolute joint, and two prismatic actuators allow the other rotations to take place.

Another type of generator of the subgroups $\{R(N, \mathbf{u})\}$ are spherical chains, that have been mentioned by Asada (19). Here three spherical chains that share the N point will lead to a spherical mechanism. Gosselin and his team have intensively studied the realization of a wrist based on this principle (186; 200; 524; 526) in order to make a pointing system (*agile eye*^{AE}). This manipulator uses three actuated spherical chains with rotary actuators with axes converging to a point that is the center of rotation (figure 2.12). The robot mentioned by Gosselin was also studied by Alizade (6)

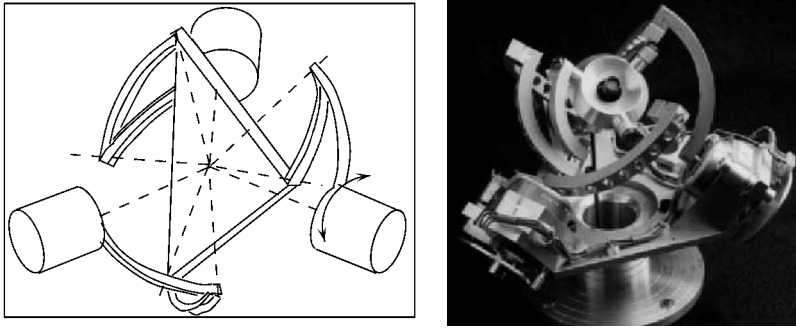


Figure 2.12. Gosselin spherical wrist and its prototype: three spherical chains are used with rotary actuators with axes converging to the center of the moving plate (from Gosselin (186))

and Leguay and Reboulet (356) have suggested redundant spherical robots in order to optimize the manipulator dexterity.

Vischer (596) suggested the *Argos* robot that possess the same structure as the *Star* robot, the parallelogram translation being replaced by a rotation about the motor axis (figure 2.13). The platform then turns about a point that is common to the three axes; therefore the star layout of the axis of the *Argos* joint is essential, while for the *Star* robot it is only a choice in the design. A mechanism of the same family as the *Argos* was suggested by Baumann (28) in his *Pantoscope* manipulator. Interestingly the 3-*UPU*, initially proposed as a translation robot, may also be designed as a wrist, provided there is a specific disposition of the passive joint axis (301) (figure 2.13). But the *U* joint in the 3-*UPU* may be substituted by *RR* pairs

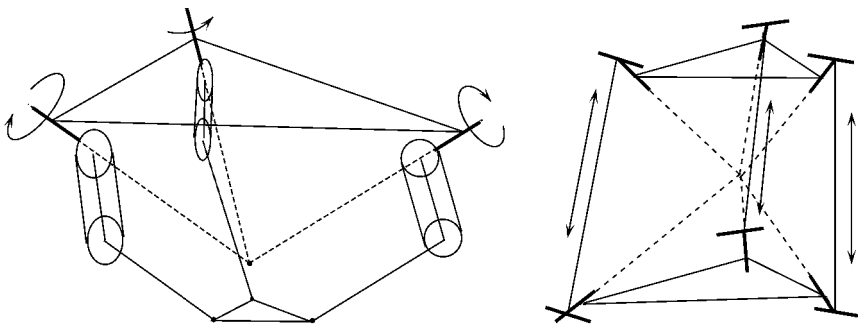


Figure 2.13. On the left, *Argos*, a spherical variant of the Delta robot: the steel-band joints are used as parallelograms so that the distal link is always parallel to the motor axis (from Vischer (596)). On the right, the 3-*UPU*, with the passive joint axis in a configuration that leads to only rotational degrees of freedom for the platform.

(leading to 3-RRPRR), while the prismatic actuator may also be replaced by an R joint, leading to a $3\text{-}(5R)$ structure, (137) (figure 2.14) or to a 3-URU , also called the *Dymo* (665) (figure 2.15). The interest of the *Dymo*

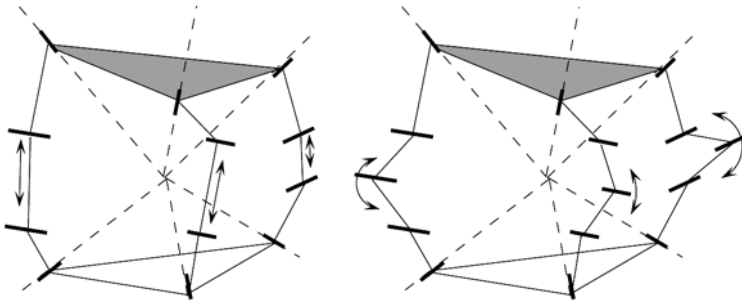


Figure 2.14. Variations of the 3-UPU structure in which the U joint are substituted by RR pairs and the P joint by \underline{RR}

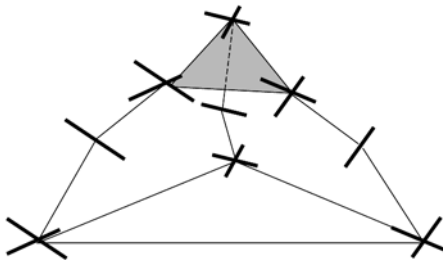


Figure 2.15. On the left, the 3-URU *Dymo* mechanism. On the right the 2 d.o.f. *Omni-Wrist*: a clever arrangement of the passive joints and the use of two linear actuators allows for a ± 90 degrees rotation ability without singularity (courtesy Ross-Hime Designs)

is that, according to its configuration, it may have different d.o.f. modes; this has been finely analysed by Zlatanov. We present here a summary of this analysis. If the axis of the first R joints of the U joints on the base and on the platform are in general position, then this robot has 3 translational d.o.f. But if the axis of the first revolute joint of the base (platform) U joint intersects at the same point, M_b (M_p) and M_b , M_p coincide, then the robot has 3 rotational d.o.f. Now assume that the base, platform and legs all lie in the same plane: we get a planar 3-RRR whose d.o.f. are 2 translations

in the base plane and one orientation around any axis perpendicular to this plane. Zlatanov mentions also the existence of another spatial mode, called the mixed mode, in which the platform has both translation and rotation d.o.f. Amazingly it is shown that a transition between two modes (e.g. translation and orientation) is possible without disassembly!

The 3-*RUU* structure has been proposed by Di Gregorio (133) but some passive joints will almost always be idle (figure 2.16). Hervé (229) has proposed another configuration of the 3-*RUU* without this drawback (figure 2.16).

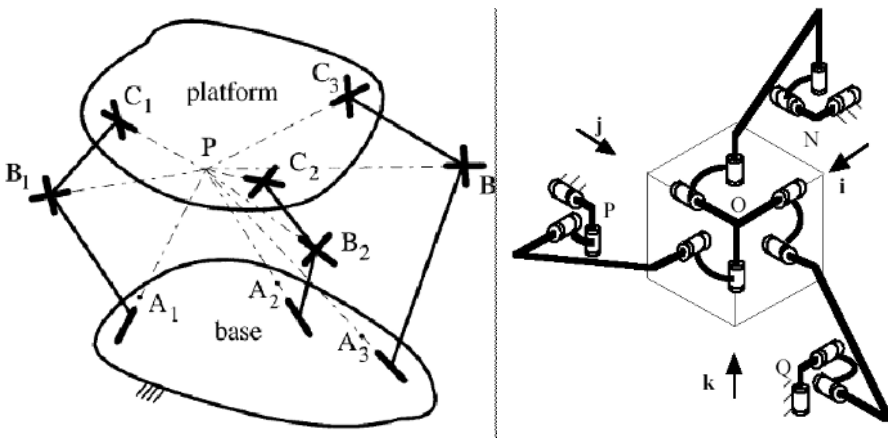


Figure 2.16. Two different designs for the 3-*RUU* wrist structure: the configuration on the right has no idle pair

Many other structures have been proposed as wrist: 3-*RSR* (139), 3-*RRS*, 3-*CRU*, 3-*UPC*, 3-*CRC* (159) and an exhaustive list is proposed by Kong (327)*.

Note that most of the presented structures are overconstrained, and a real rotational behavior will be obtained only with a strict manufacturing of the parts. Recently non-overconstrained mechanisms with asymmetrical, non identical, legs have been proposed (303).

Finally let us mention the *Omni-Wrist*^{OW} of Rosheim with 2 d.o.f. in rotation but with a ± 90 degrees rotation ability without singularity because of a clever arrangement of the passive joints (figure 2.15). It has been used for large fountain displays, antenna pointing and most recently, biomedical applications.

2.4.3.3 Mixed degrees of freedom manipulators

The 3 d.o.f. manipulator of figure 2.17 mentioned by Hunt (250) has been studied by various authors: Gosselin (186), Lee (355)*, Pernette (471) (as

a micro-robot under the name *Orion*), Waldron (599). The link joints with the base are revolute, while the joints on the moving platform are ball-and-socket joints connected to prismatic actuators allowing the variation of the link lengths. The degrees of freedom are a translation along the vertical axis, and rotation along the precession and nutation angles. This robot has been used by Zhang (655) for the balancing of a spaceship arm and as a micro-manipulator (under the name of *Artisan*) by Waldron and Khatib.

A structure equivalent to Hunt's was also suggested by Lambert (340)

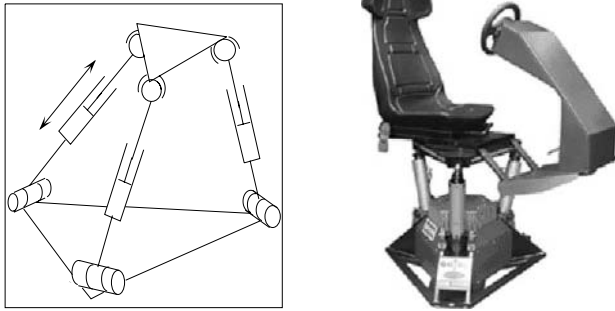


Figure 2.17. A three degrees of freedom manipulator mentioned by Hunt and Lee and its application as an entertainment motion base (courtesy Visible Light[□] ^{V^L}).

and was later studied by Dunlop (146): the prismatic actuators are replaced by a system of articulated links that are actuated at the base by a revolute actuator (figure 2.18). This structure has the advantage of presenting a larger workspace, but requires more passive joints. A variant has been used in the *Dockwelder*[□] ^{DP} European project for ship welding: the rotary actuators were substituted by prismatic ones. In this project, two such tripods were used in series and on top of them was a classical industrial arm.

Using a platform congruent to the base allows the direct kinematics to remain very simple. A similar design was suggested by Canfield (68) under the name *Carpal wrist*. It is also possible to replace the rotary actuators by linear actuators, as we have suggested for our *Mips* robot (416), a micro-robot for endoscopy with a diameter of 7 mm (figure 2.19). Instead of using an actuator having a vertical motion, we may also choose any other axis direction, for example horizontal, as suggested by Carretero (71) and used in the *Sprint Z3* machine tool (figure 2.85).

Lande's mechanism (341) (figure 2.20) has two linear actuators moving vertically articulated stems, to which links are connected by universal joints. The other ends of the links are also connected to the moving platform by universal joints. The third degree of freedom is obtained by the rotation

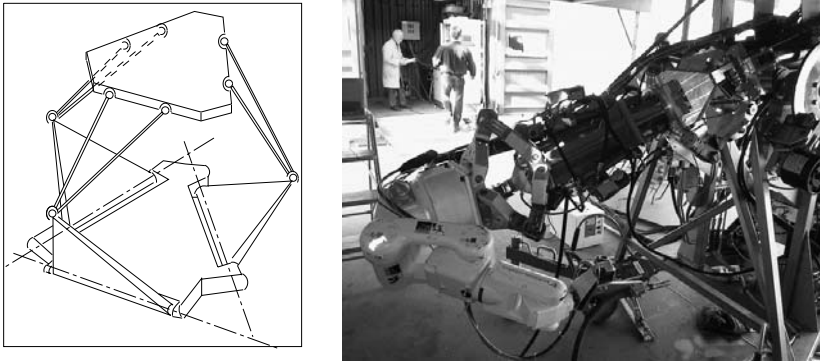


Figure 2.18. A variant of the Hunt's mechanism proposed by Lambert and Dunlop and on the right another variant used in the *Dockwelder* project (courtesy University of Southern Denmark).

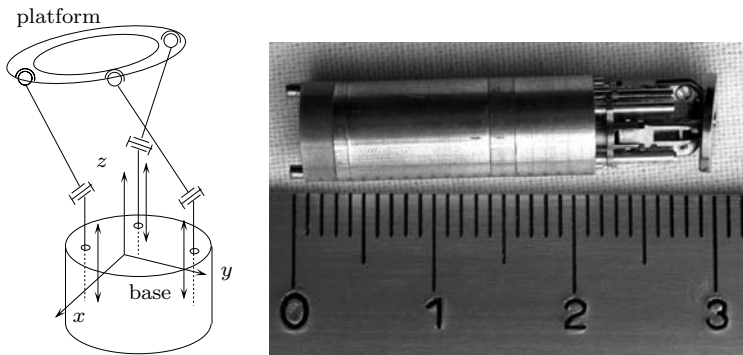


Figure 2.19. The micro-robot *Mips* (416). The linear actuators on the base move along vertical direction revolute joints that are connected to fixed length links.

of a chain that allows control of the rotation of the end-effector around the platform normal. According to this principle, mechanisms with more degrees of freedom could be built. It should however be remembered that such architectures lead to an important number of passive joints (18 for a manipulator with 6 degrees of freedom).

The addition of a passive link allows us to play quite freely with the nature and the number of resulting d.o.f., as shown in the example in figure 2.20. This robot consists of a moving plate actively controlled by 3 linear actuators. It is also subjected to a passive constraint because it is linked by a ball-and-socket joint to a bar that is itself subjected to constraints. One of the ends of this bar is linked to a shuttle that is itself guided in translation and passes through a hurdle orifice; it can therefore freely twist

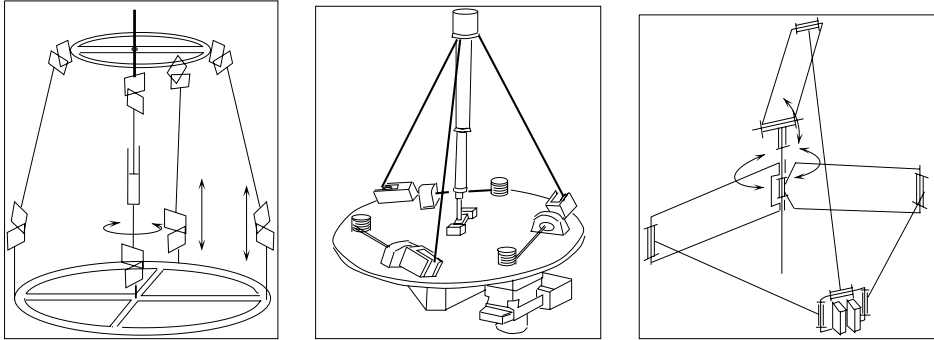


Figure 2.20. On the left, linear actuators ensure a motion of the joint centers: the rotation around the plate normal is obtained by an articulated stem (from Lande (341)). In the middle, a 3 degrees of freedom robot mentioned by Landsberger: the links are wires that can roll up around drums. The tension in the wire is maintained by the central mast that pushes passively towards the top (from Landsberger (342)). On the right, a cylindrical robot suggested by Reboulet (from Reboulet (503)).

about this orifice center, and freely move in translation about its axis. According to the same principle, Reboulet mentioned an architecture where passive constraint is exercised by an extensible bar linked to the base by a universal joint, but unable to twist about its axis (figure 2.21).

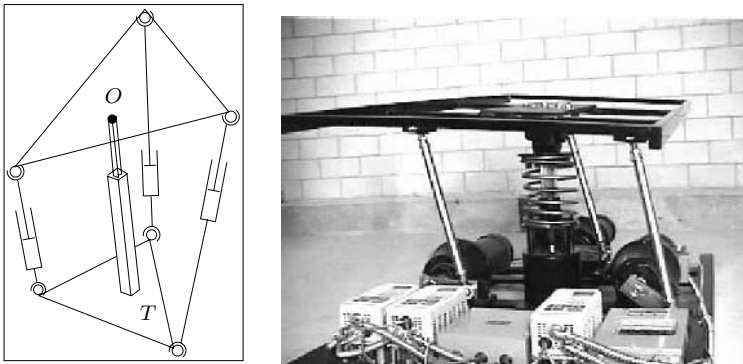


Figure 2.21. On the left, a 3-d.o.f. structure: the central mast allows only vertical translation and is topped by a universal joint which allows only 2 rotations for the platform. On the right, a motion base that uses a variant: the linear actuators are substituted by a combination of a lever and a rotary motor (courtesy InMotion SimulationTM)

Another example of this type of robot is the *CaPaMan* robot of Ceccarelli (76). The initial version uses three actuated 4-bar mechanisms with a

free-sliding ball-and-socket joint, but a later version uses a more classical arrangement (figure 2.22).

Reboulet (503) has also suggested a robot with cylindrical motion (figure 2.20), various variants with similar structures being then proposed by Liu (374)* for its *Half* and *Hana* prototypes.

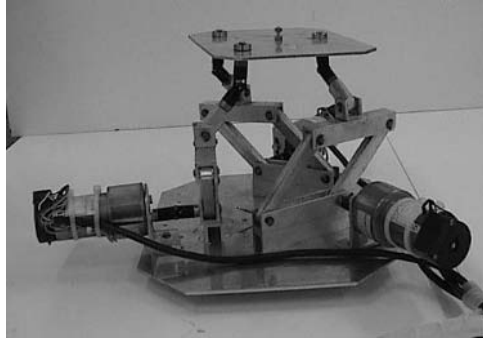
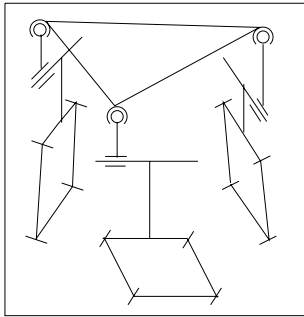


Figure 2.22. The *CaPaMan* robot of Ceccarelli: each chain has an actuated 4-bar mechanism and a free-rolling ball-and-socket joint (from Ceccarelli (76)). In the following versions the free-rolling ball-and-socket joint was substituted by a more classical arrangement

Although it is outside the purpose of this book, the Landsberger mechanism is worth mentioning (342)*. It uses wires as links, a central mast ensuring that they are under tension (figure 2.20). This type of robot is light, but it is difficult to maintain tension in the wires.

Let us also mention a robot presented by Arun (18), made of two piled up octahedra with a common triangular face, and having linear actuators allowing change of the sides lengths (figure 2.57). This robot was studied by NASA as an element of a truss (figure 2.57), and by Hertz (225) and Reinholtz (505).

With these complex manipulators it is often difficult to determine the d.o.f. of the platform. In order to solve this problem, Huang (245)* suggested a screw-based method that he applied to the robot mentioned by Lee, and to spherical ones.

2.4.4. 4 D.O.F. MANIPULATORS

We have seen in the introduction that it is theoretically impossible to design a 4 d.o.f. with identical legs. Hence such a design will have to rely either on a passive constraint mechanism, a specific geometry of the legs, different legs, less than 4 legs, or a specific mechanical design.

Mechanisms with 4 degrees of freedom appeared early in the literature. In 1975, Koevermans (322) presented a flight simulator mechanism based on a passive constraint system (figure 2.23). The degrees of freedom are the three rotations and one translation about the z axis. Another robot with

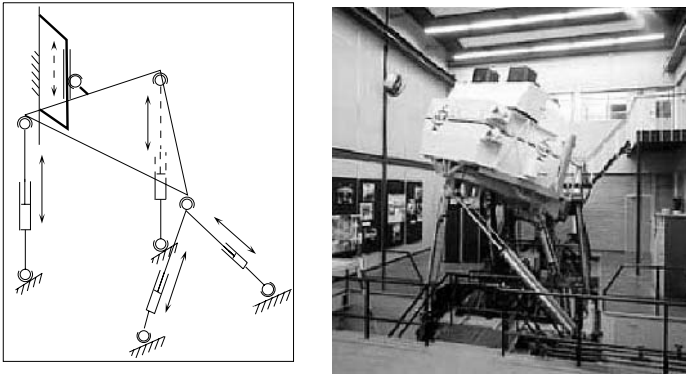


Figure 2.23. Koevermans 4 degrees of freedom manipulator and its use as a flight simulator at NLR. A passive constraint ensures that the only degrees of freedom are the rotations and one translation about the z axis (from Koevermans (322)).

a passive constraint chain is presented in figure 2.27. One way of having the same chains is to use flexible legs, as proposed by Rebman (498) (figure 2.24). Using less than 4 legs may also lead to a manipulator with 4 d.o.f., either with an appropriate actuation scheme as mentioned by Tanev (574), or with a specific arrangement of the joint axis (664). Specific arrangements to get 3T1R motion (i.e. 3 translations and one rotation, also called *Schönflies* motion) have been presented in (361)* (figure 2.24) and exhaustively by Kong (329).

Another possibility is to use 4 non identical chains as in the HITA-SST of Clavel (101) (figure 2.25), or in the robots proposed by Chen (89) or Liu (374) (figure 2.26).

Robots with identical chains are possible as soon as a specific arrangement of the joint axis is satisfied as proposed, for example, in (361)* with *RPRRR* or *RPUR* chains (figure 2.27).

As for a specific design, we may mention the $H4$, $I4$ family of robots proposed by Pierrot and co-workers (110)*,(333)*,(478)* which uses various clever configurations of the platform to get 4 d.o.f., 3 translations and one rotation, with a design that allows for large rotation ability (the *Twice* system, figure 2.28). Hybrid mechanisms mixing serial and parallel mechanisms are another possibility (see the references Web page).

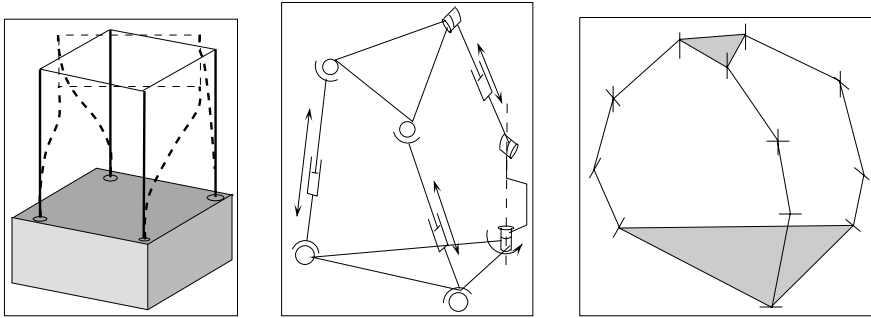


Figure 2.24. 4 d.o.f. manipulator. On the left, a manipulator mentioned by Rebman, using flexible stems (from Rebman (498)). The NAOS mechanism (figure 2.71) is a 3-d.o.f. application of this principle. In the middle, a manipulator with three chains and 4 d.o.f. (from Tanev (574)). On the right, a mechanism with revolute joints whose d.o.f are three translations and a rotation around the normal of the platform (from Li (361)).

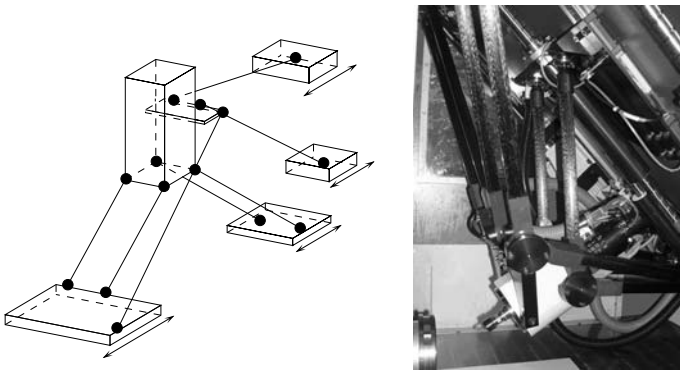


Figure 2.25. 4 d.o.f manipulator, the HITA-STT of Clavel.

2.4.5. 5 D.O.F. MANIPULATORS

Robots with 5 d.o.f will also have to rely on passive constraint mechanisms, specific geometries or design. Such a structure is of particular interest in the machine-tool field for so-called five-axis machining. Indeed 6 d.o.f are not strictly necessary for machining as the rotation of the spindle adds a degree of freedom.

Examples of robots with passive constraint mechanism (627; 652) are presented in figure 2.29. The passive mast is interesting in term of stiffness of the robot (roughly equivalent to a 6 d.o.f robot, except for the d.o.f. that is constrained by the passive mechanism) but decreases the workspace

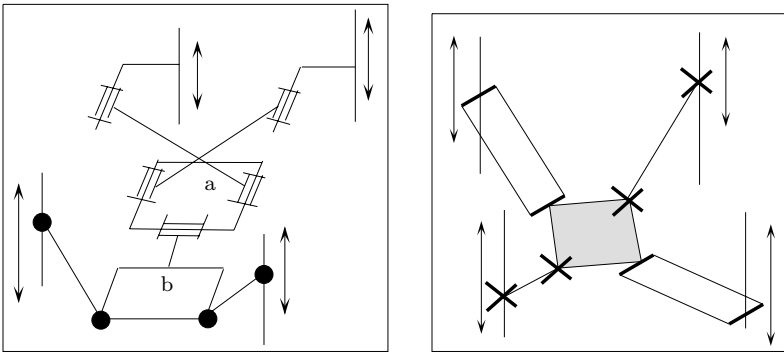


Figure 2.26. 4 d.o.f manipulators with non identical chains (89; 374).

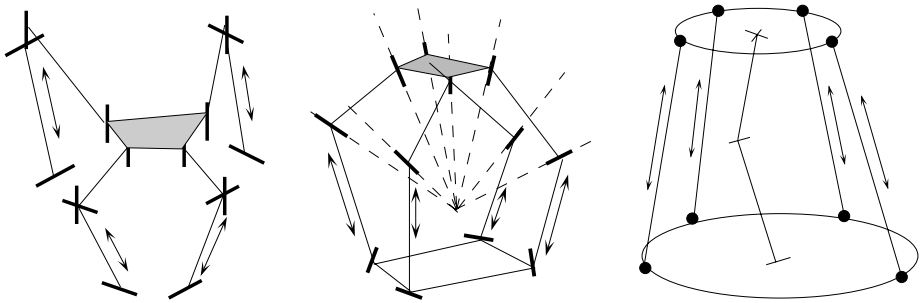


Figure 2.27. 4 d.o.f. robot. On the left and in the middle robots with $RPRRR$ or $RPUR$ chains. On the right, a robot whose passive constraint chain leads to two rotation and two translations d.o.f. (from Zhang (652))

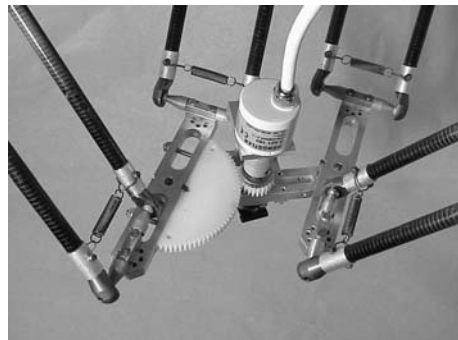
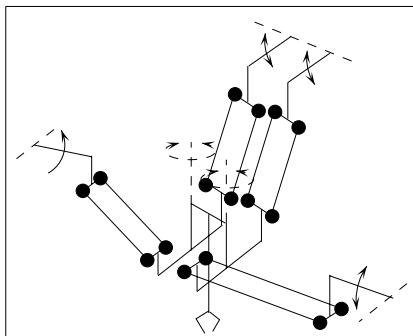


Figure 2.28. 4 d.o.f manipulator. The H4 robot of Pierrot and a detail of its platform with the *Twice* system

because of leg interferences. Another possibility is that one of the active legs constrains one d.o.f. of the platform, as shown in figure 2.29 (176). Five

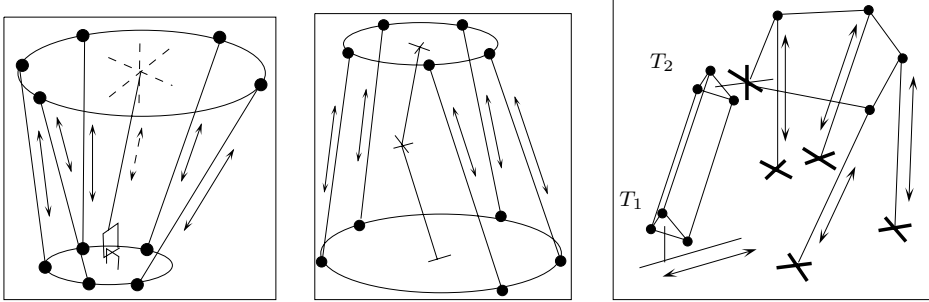


Figure 2.29. 5 d.o.f. robots. The role of the central mast is to prohibit the rotation around the normal to the platform (from (627; 652)). On the right, one of the legs constrains one rotational d.o.f. of the platform (from Gao (176)).

d.o.f. may also be obtained by a specific geometrical arrangement of the legs. Examples of such arrangements are provided in figures 2.30,2.31. A

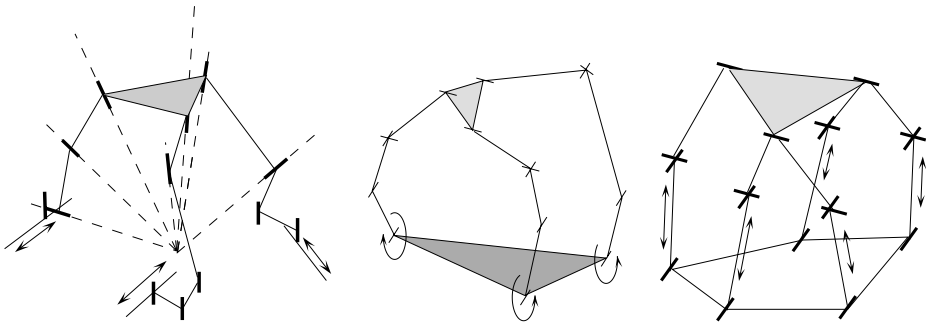


Figure 2.30. 5 d.o.f. robots. Here a specific arrangement of the revolute joint axis allows to prohibit the rotation of the platform around its normal (from (361)*,(158)*)

specific design with interconnected legs may also be used as mentioned by Zoppi (figure 2.31), but in that case both the inverse and direct kinematics are difficult to solve. Alternatively, a hybrid robot mixing parallel and serial structures may be used, as suggested by Austad (20): a parallel positioning device controls the location of a specific point of the end-effector, and a second parallel mechanism ensures two rotations of the moving platform (figure 2.31). A nearly similar structure has been proposed for machining in the ITER vacuum vessel (627).

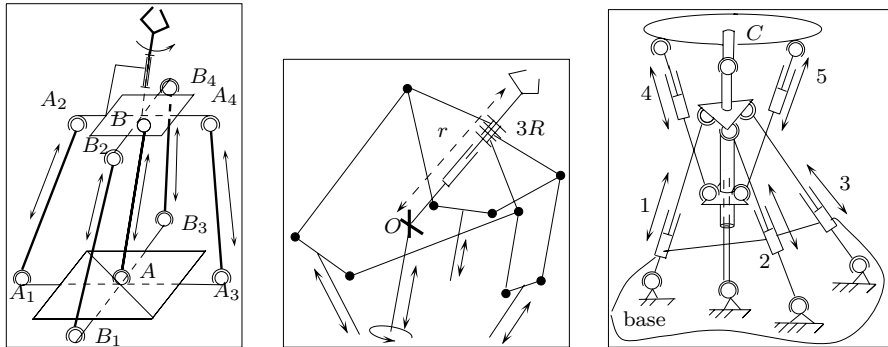


Figure 2.31. 5 d.o.f. robots. On the left is the robot suggested by Zamanov; the sixth degree of freedom (rotation about the normal to the moving platform) may be obtained by yet another mechanism (from Zamanov (647)). In the middle, a 5 d.o.f. robot with interconnected legs; the d.o.f. are the 3 rotations around O , the altitude of O and the distance r between O and the end-effector (from Zoppi (669)). On the right, a hybrid structure (from Austad (20)).

2.4.6. 6 D.O.F. MANIPULATORS

The realizations of 6 degrees of freedom fully parallel manipulators are based on the use of 6 generators of the motion group $\{D\}$, as described in the earlier chapters. They will thus work with chains of the $RRPS$, $RPRS$, $PRRS$, $RRRS$ types. Note that these generators are all basically equivalent with respect to the mobility of the platform, as the order of the joints has no importance and a $P(R)$ may always be replaced by a $R(P)$ joint. Consequently, being given a mechanism whose legs use one of such generators, we may always derive other mechanisms just by changing the order of the joints and by using the replacement of $R(P)$ joints by $P(R)$ joints. In terms of mobility, all the presented mechanisms using these generators will belong to the same class as the Gough platform. But they will not be equivalent when looking at performances other than mobility and it is thus important to present various robots with different leg structures. There are also partially parallel mechanisms offering more complex structures.

2.4.6.1 UPS chain robot

This architecture represented in figure 2.32 is the most frequently used in applications (see section 2.9). This type of manipulator is usually called a Gough platform (see the historical note on this name in the "Introduction" chapter), 6- UPS robot, 6-6 robot or *hexapod* (a word registered as British trade mark 2153930 by Geodetic Technology in the context of machine-tools). It is the most commonly used architecture, and has been used in

numerous laboratories for the realization of prototypes (164; 217; 324; 499).

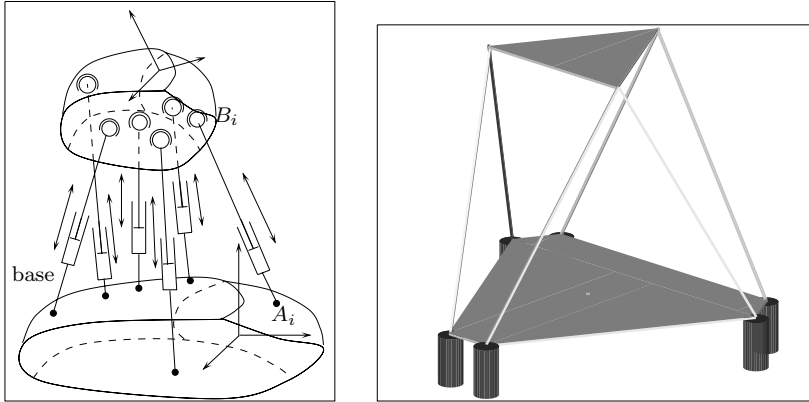


Figure 2.32. The general structure of a 6 d.o.f. parallel robot with chains of the UPS type. The platform is linked to the base by 6 chains. The connection of the chains with the base is usually effected with a U joint, while the chains are attached to the moving platform by a S joint. A prismatic actuator allows the change of the lengths of the links. On the left, the McCallion prototype (1979); the motors lead ball-screws in rotation via a universal joint, enabling change in the lengths of the links (from McCallion (381)).

Historically, it seems that a first realization of this type of parallel manipulator can be attributed to McCallion from Christchurch University (381) for a robotized assembly station. In this prototype, electrical motors that are placed on a fixed base lead ball-screws located in the links via an universal joint. These ball-screws enable changes in the lengths of the links connecting the base to the moving platform (figure 2.32).

As early as the 1980's, Fichter (163)* foresaw that this type of mechanism as well as its possible applications would be posing problems. McCallion's concept was borrowed from Shelef's patent (534), except that its motors are linked to the base via a universal joint and therefore always stayed on the same line as the ball-screws. One of the first people who built this type of manipulator is C. Reboulet from the CERT-DERA, who developed a prototype as early as in 1985 (502). His prototype is an active wrist with pneumatic actuators, and one version of this prototype was marketed, although without success, under the name *Space*.

INRIA developed in 1986 a *left hand* prototype with the help of C. Reboulet (figure 2.33). Its structure is similar to the CERT one, although it uses electric actuators with a stroke of 2cm. This robot was equipped with force sensors in each link, which allows the measurement of the 6 components of an external wrench. The ball-and-socket joints were simply made

of universal joints; the base of the joint can rotate about an axis that is engaged in two miniature ball-bearings set in duplex and pre-constrained. An interesting feature of such parallel structures is the fact that they are

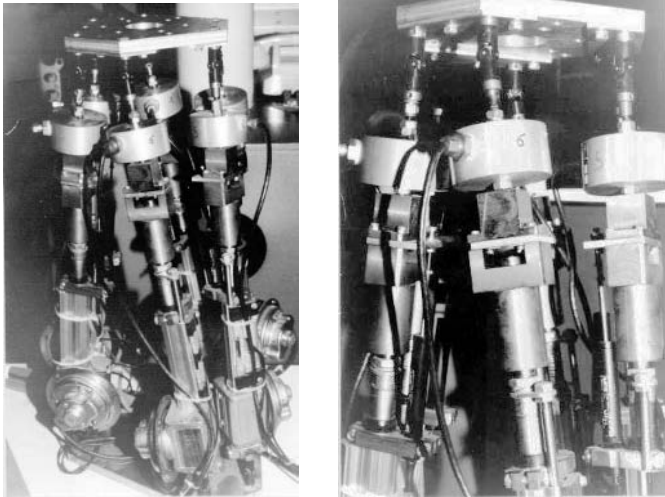


Figure 2.33. INRIA *left hand*, a parallel robot using electrical actuators with links of varying lengths.

much less sensitive than serial robots to scale effects. We will present in section 2.9 both huge 6-*UPS* robots and very small scale robots, for example Arai's 6-*UPS* micro-robot (555) with piezo-electric actuators having a stroke of 8 micrometers, a positioning accuracy of 30 nanometers and flexible beams as passive joints.

It seems that the Marconi company was the first to try to offer this type of manipulator for the market. They created a manipulator, named *Gadfly*, aimed at pick-and-place and electronic component assembly (487). It seems, however, that this prototype has never been marketed.

Special layouts for the joint centers were suggested quite early: for example in the *Toro* robot, designed by Zamanov (645), which possesses double ball-and-socket joints on the moving platform or in a patent by Griffis (207) in which the platform and the base are triangles with three joint centers on each edge, a configuration that leads to a singular design (257). Special layouts may also be obtained by using reconfigurable robots in which the location of the joint centers may be modified at will, in order to optimize the robot geometry for the task at hand, as suggested by Bande (25), Ji (289)* and Yang (632). The modularity concept is interesting but de-

signing algorithms for adapting the geometry to the task is not a trivial task (286).

2.4.6.2 *PUS* chain robots

The first example of such architecture is the INRIA patented *active wrist* (398) (figure 2.34) that shows a vertical actuated prismatic joint that is connected to a fixed length link by a universal joint. The other end of the link is attached to the moving platform by a *S* joint. This structure has been used, for example, by Gerber Coburn for the manufacture of lenses. Such a structure possesses the advantages of having a very low center of mass, a very light moving mass and reduced risk of collision between the links, compared to the 6-*UPS*. Note that the direction of motion of the prismatic

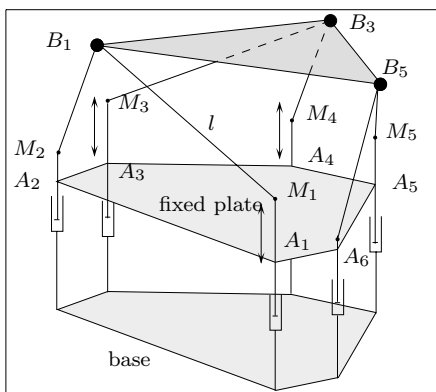


Figure 2.34. The 6 degrees of freedom *active wrist*, the joints of which that are near the base move vertically, using a chain of the *PUS* type (INRIA patent).

actuators does not matter: tilted in the *Hexa-M* machine-tool of Toyoda Machine Works (figure 2.89), horizontal and parallel in the *Hexaglide* robot from École Polytechnique Fédérale of Zürich (figure 2.35) (237) or vertical with only 3 guide-ways in the *Linapod* (488) (figure 2.36). We may also mention the *Nabla 6* (36), with horizontal axis (figure 2.36): there are only three distinct prismatic joint axes, with 2 A_i points sliding on the same axis. Moreover, three of points B_i are articulated on a triple ball-and-socket joint, and therefore are identical: the position of this common point can thus be controlled with the help of three associated actuators, while the other three control the platform orientation; the result is a *decoupled robot*, a topic addressed in section 2.4.6.6.

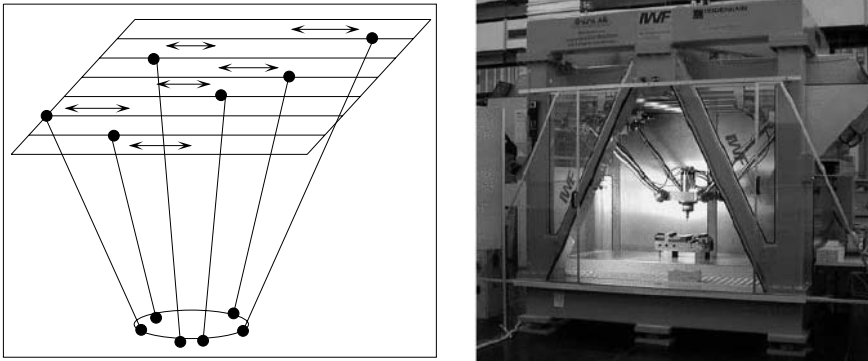


Figure 2.35. The Hexaglide robot from École Polytechnique Fédérale of Zürich and its implementation as a machine-tool

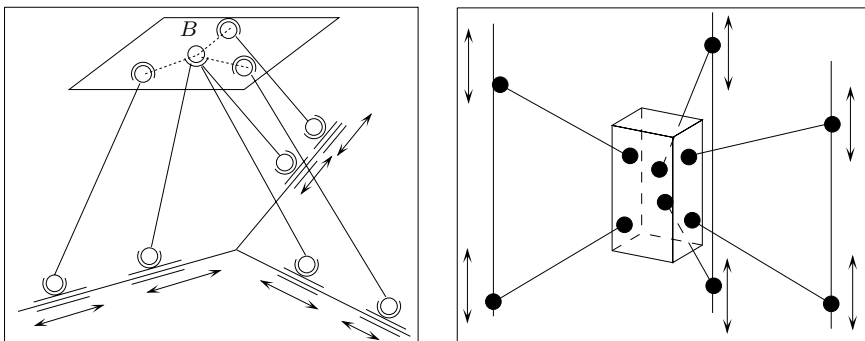


Figure 2.36. On the left, the Nabla 6, a decoupled robot mentioned by Bernier (from Bernier (36)). On the right, the Linapod robot (from Pritschow (488)).

Another interesting device is the *Cobot* mechanism (162): the linear actuation is obtained by a wheel rolling on a rotating cylinder and whose rotation axis may be modified (figure 2.37).

2.4.6.3 *RUS* chain robots

Hunt (250) suggested as early as 1983 a robotic architecture using this type of chain (figure 2.38). The prototype presented in figure 2.39 was built by Zamanov and was based on this principle.

The *Delta* principle was extended by Pierrot from LIRMM[□] LIRMM, and by Uchiyama, who suggested the 6 degrees of freedom fast parallel manipulator *Hexa* (475; 591). This mechanism differs from Hunt's architecture because of the position of the axes of the revolute joints on the base and of the

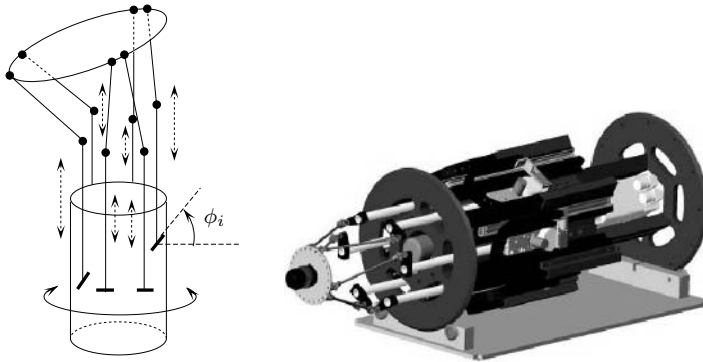


Figure 2.37. The *Cobot* mechanism. Wheels are rolling on a rotating cylinder. By changing the angle ϕ_i of the rotation axis of the wheels, it is possible to modify the vertical velocity of the joint (from Faulring (162)).

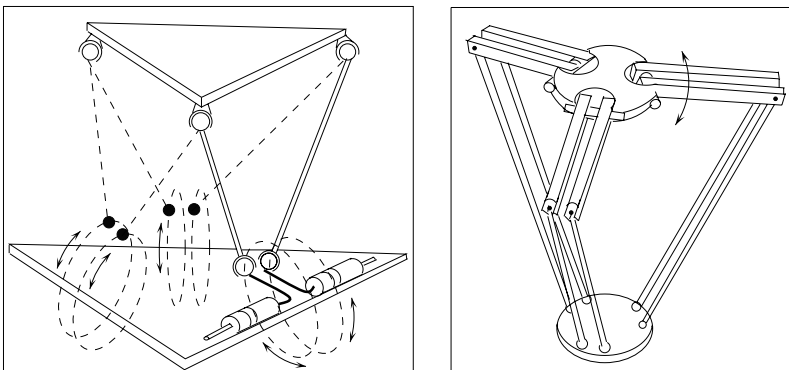


Figure 2.38. On the left, the robot using \underline{RUS} chains as suggested by Hunt in 1983 (from Hunt (250)). On the right, Pierrot's *Hexa* robot, a generalization of the concept of the *Delta* (from Pierrot (475))

location of the joint centers on the moving platform (figures 2.38,2.40).

However, this difference meets the *Delta* operating mode if a pair of links performs an identical motion. It is also noteworthy that the actuated lever motions do not have to be in a vertical plane; a version of it could thus be built, for instance, with the plane horizontal. Several versions of the *Hexa* robot were suggested with various directions for the actuator axis, tilted or vertical (518) as in the the *Rotobot*^{□ RH} device distributed by Hexel with actuated joints moving on a circular rail, figure 2.39.

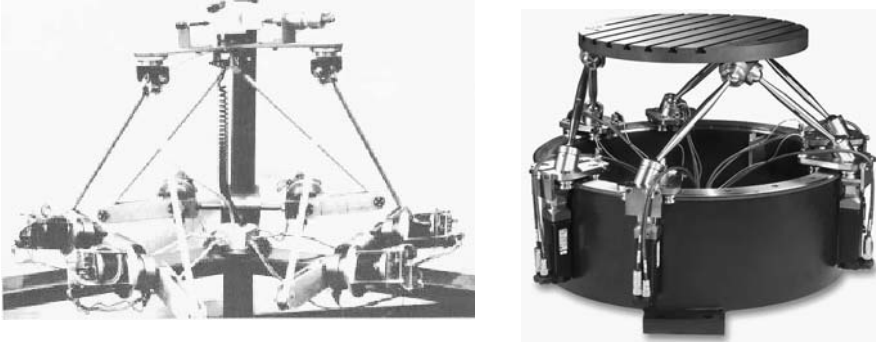


Figure 2.39. On the left, a prototype created by Zamanov and set on an architecture invented by Hunt, where the joint points move on circles (photograph by kind permission of Pr. Zamanov). On the right, the Hexel Rotobot: the actuated joints move on a circular rail (courtesy Hexel)

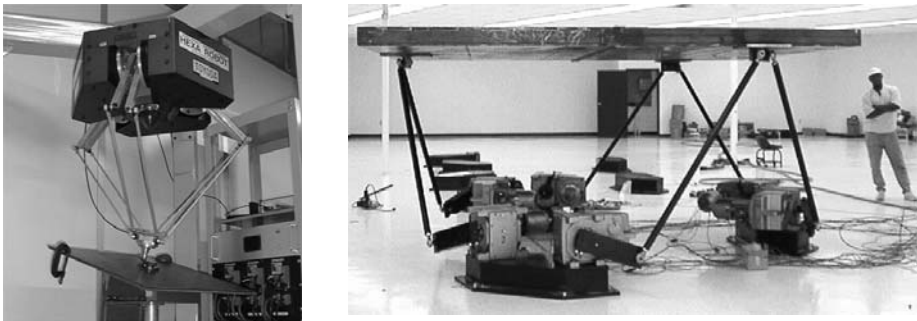


Figure 2.40. On the left, the robot version of the *Hexa*. On the right, an application of the *Hexa* for an entertainment simulator motion base (courtesy Servos Simulation Inc.).

2.4.6.4 Robots with miscellaneous chains

This category brings together manipulators studied within the literature and which have actuation principles which cannot be directly linked to any of the classifications presented earlier.

One example is the complex manipulator suggested by Han, Hudgens and Tesar (213), where 4-bar mechanisms are used in order to move the joint centers (figure 2.41). The same principle is used by Tsai (585) and Tahmasebi (568): two rotary actuators allow the change of position of the joint centers (figure 2.41). The use of flexible joints is also proposed by Wang (605).

For certain tasks, such as assembly, it is important that the mecha-

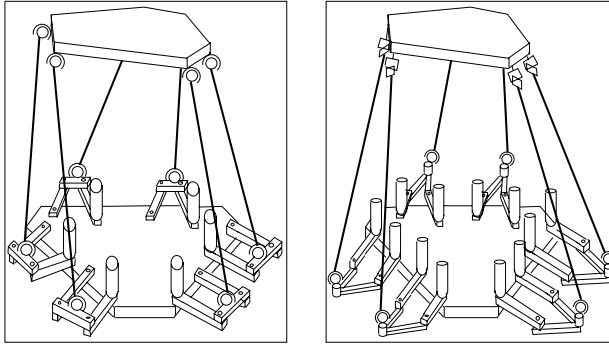


Figure 2.41. On the left, Han's manipulator: 4-bar mechanisms on the base, actuated by electric motors, move the location of the link joint centers (from Han C.S. (213)). On the right, Tahmasebi's robot: two rotary actuators placed in D_i allow the control of the location of the C_i (from Tahmasebi (568)).

nism stiffness (which will be studied later on) be identical in all directions. Figure 2.42 presents an architecture realizing this aim, called *Limbro*, suggested by Artigue (17). Redundancy may be used in order to increase the robot workspace. Thus, Merkle (396) suggests an architecture called the *double tripod*; two linear actuators tripods give five degrees of freedom to the end-effector, and a screw system ensures the rotation about the end-effector axis (figure 2.42).

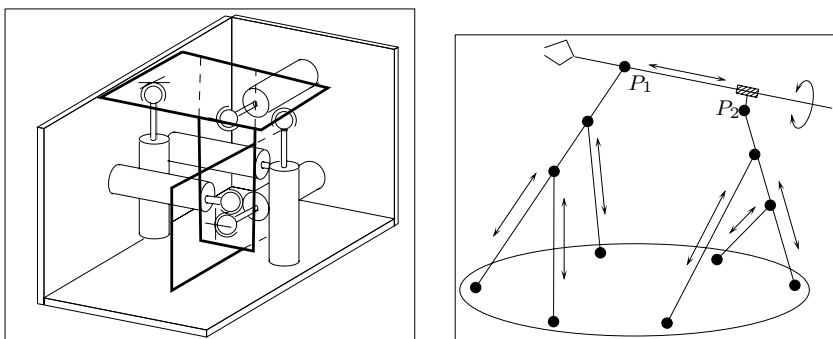


Figure 2.42. On the left, the *Limbro* parallel manipulator architecture. The linear actuators, rigidly attached to the base, possess at their other ends an S joint linked to the moving platform by a P joint (from Artigue (17)). On the right is Merkle's redundant *double tripod*: the groups of 3 linear actuators allow the points P_1, P_2 to move in space. A mechanism controls the rotation of the terminal axis about its axis (from Merkle (396)).

2.4.6.5 Three-legged robots

Many authors have proposed 6 d.o.f. robots with only three legs that will have two actuators per leg (hence they are not fully parallel). This allows one to decrease the risk of interference between the legs (thereby increasing the workspace size), but has the drawback of reducing the stiffness while increasing the positioning errors. Podhorodeski (480) presents a systematic study of possible architecture with three chains.

In this category are the robots with $RRPS$ chains as Alizade's manipulator (5) or $RPSR$ chains (540) (figure 2.43).

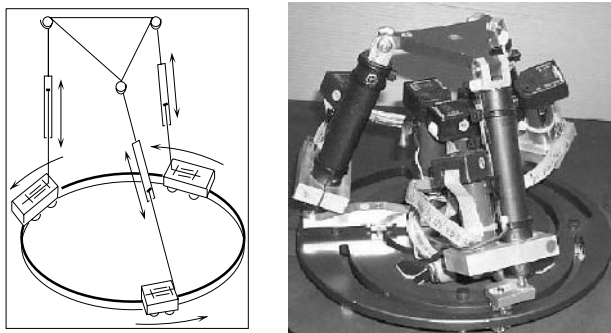


Figure 2.43. Robots with $RRPS$ or $RPSR$ chains, from (5), (540)

Instead of using a linear actuator in the links, the carriages may have 2 d.o.f. in the plane, as in the robot mentioned by Ben-Horin (31) and Tsai (587) (figure 2.44) or in the *SpaceFab* robot of Micos (figure 2.94).

The mechanisms suggested by Kohli (323) (figure 2.45), or Behi (29)

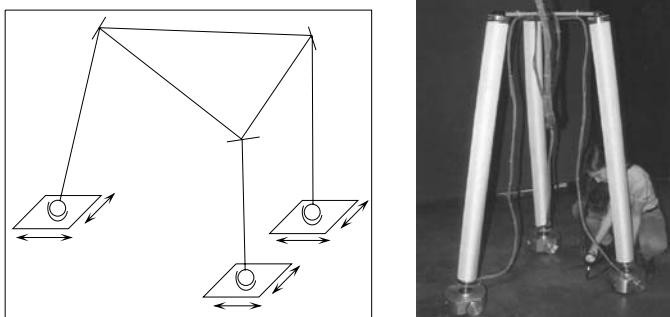


Figure 2.44. On the left, a manipulator mentioned by Ben-Horin: three 2 degrees of freedom planar carriages support fixed length links (from Ben-Horin (31)). On the right, a prototype of the Technion Haifa in which the legs are inflatable.

should also be remembered. They use double actuators which are either linear and rotary, or linear and linear (figure 2.46). Double linear actuators

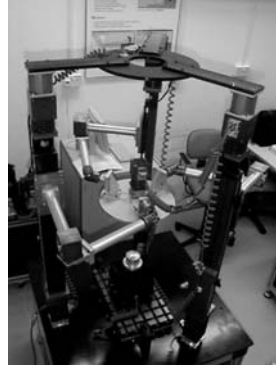
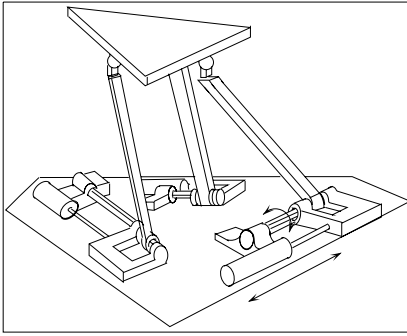


Figure 2.45. On the left, Kohli's prototype, with linear and rotary double actuators (from Kohli (323)); on the right, a variant with $RPRS$ chains (Nanyang Tech. University)

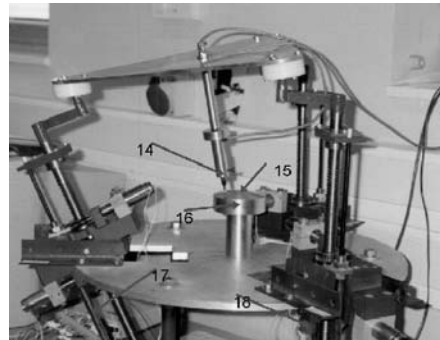
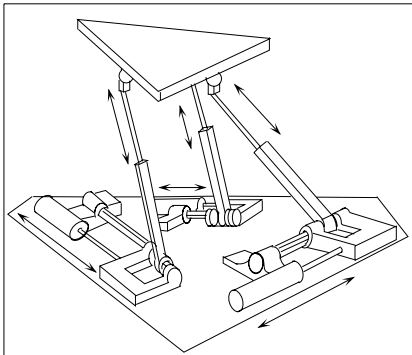


Figure 2.46. On the left, Behi's prototype, with linear-linear double actuators (from Behi (29)) and on the right a robot with RPS chains (Ben-Gurion University of the Negev).

are used in a robot mentioned by Byun (66) while a manipulator using pantographs has been proposed by Ebert-Uphoff (152) with a balancing interest as we will mention later (figure 2.47). Parallelograms were present very early, for example in the prototype of Inoue (277), later studied by Collins (107). They have the interest that they allow us to obtain virtual links with lengths which have a larger range than classical mechanisms, although accuracy suffers because of the larger number of passive joints.

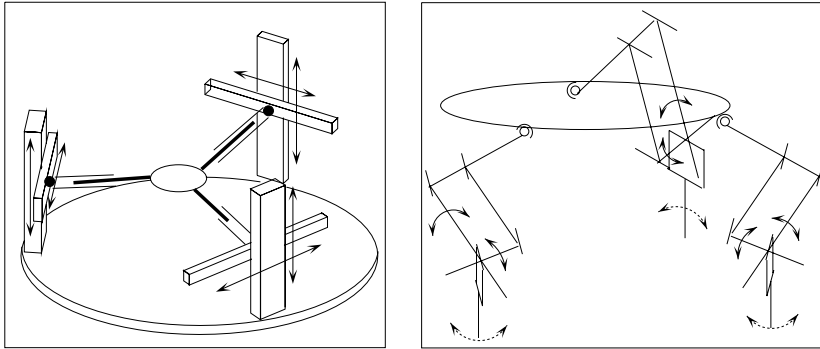


Figure 2.47. On the left, a manipulator mentioned by Byun with *PPSP* chains (from Byun (66)). On the right, a manipulator mentioned by Ebert-Uphoff: in each chain two motors control two sides of a parallelogram while the revolute joint on the base is passive (from Ebert-Uphoff (152)).

The use of 4-bars has also been mentioned in (24) for a possible application in MEMS as the manufacturing of such hinge joints is well mastered (figure 2.48). But 4-bars may also be used for fast robots, as the *Ninja* robot (437) (figure 2.48). Let us also mention the 6 d.o.f. robot called

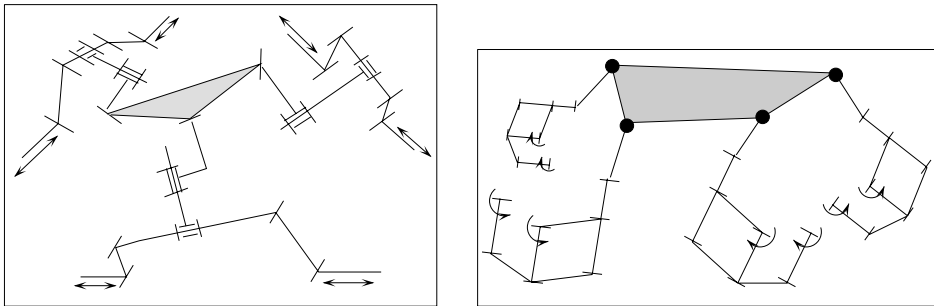


Figure 2.48. On the left, a 3-legged robot with only revolute joints: the robot is designed for a possible use as a MEMS and in this domain hinge joint are mastered (from Bamberger (24)). On the right, the *Ninja* robot, a fast redundant robot (from Nagai (437)).

Smartee, suggested by Cleary (103) marketed without success by Hughes Stx. The end-effector of this robot is connected to the base by three kinematic chains made of two links. The link that is attached to the end-effector is connected to the preceding link by a passive revolute joint, and a differential mechanism controls two d.o.f. of the link attached to the base (figure 2.49).

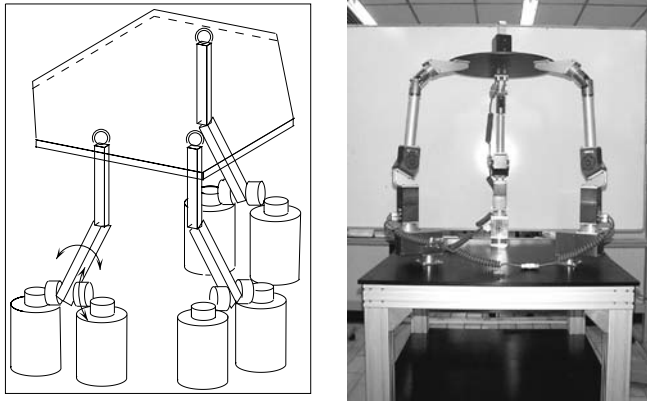


Figure 2.49. On the left, the *Smartee* robot: the differential mechanism controls the two degrees of freedom of the first link of the three chains (from Cleary (103)). On the right, a variant with \underline{RRRS} chains (Nanyang Technological University[□] PMR)

Let us also cite the *Turin robot*, developed by Romiti and Sorli (547). In each leg, a stack of two actuated parallelograms ensure the motion within a plane of a ball-and-socket joint. This ball-and-socket joint is set on a sliding device with axis perpendicular to the parallelogram plane. The robot *CaPaMan* (figure 2.22) is the 3 d.o.f. version of this robot.

Another interesting mechanism addresses the workspace problem of parallel robots. We will indeed see that one of the main drawback of such mechanisms is their relatively limited workspace, especially in term of orientation abilities. To overcome this drawback, Ryu (515) proposed the redundant *Eclipse* robot with seven actuators (figure 2.50): three carriages supporting stems move on a circular rail, and on the stems are 3 linear actuators supporting three revolute joints connected to fixed length links, one of which is actuated. The other ends of the links are connected to the moving platform through ball-and-socket joints. Such a configuration allows for a full rotation around the z axis and over 90 degrees for the other rotations. A further development was proposed the *Eclipse II* (317)*. In this version, figure 2.51, the mechanism is no longer redundant but the use of circular railways allows for a 360 degrees rotation of the platform.

2.4.6.6 Decoupled robots

It is noteworthy that almost all the manipulators presented so far have actuators which influence both the position and orientation of the moving platform. From the control view point, it may be interesting to design a manipulator where three actuators control the translation while the remaining three control the orientations. A *decoupled robot* would thus be obtained.

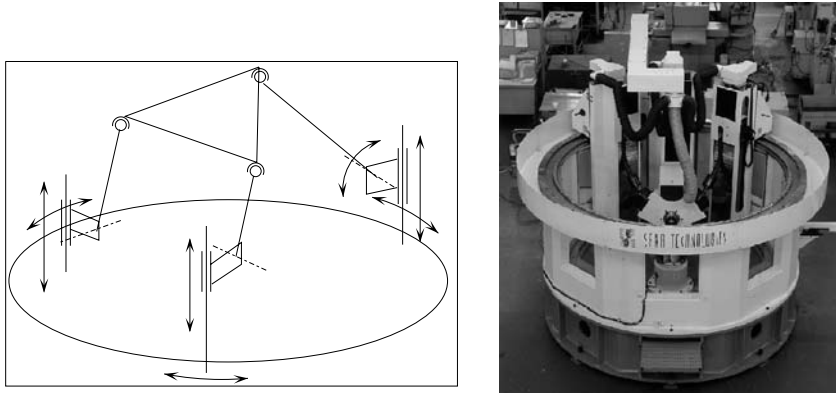


Figure 2.50. The *Eclipse* robot: three carriages supporting stems move on a circular rail and on the stems are 3 linear actuators supporting three revolute joints connected to fixed length links, one of which is actuated. The other ends of the links are connected to the moving platform through ball-and-socket joints (from Ryu (515)). On the right, a first prototype of milling machine based on the *Eclipse* concept.

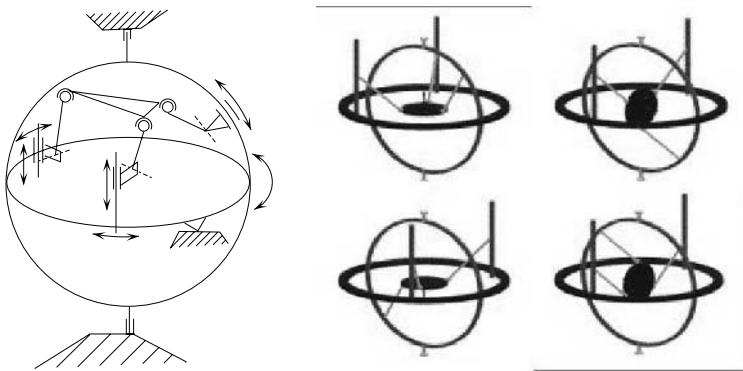


Figure 2.51. The *Eclipse II* 6 d.o.f. robot and some of its possible motions (from Kim (317))

Three different types of decoupling may be distinguished (291):

- *strong coupling*: each pose parameter is a function of all actuated joint variables (e.g. a 6-UPS robot of general geometry)
- *complete decoupling*: each pose parameter is a function of only one actuated joint variable
- *partial decoupling*: neither of these types

Note also that decoupling may occur *locally* (i.e. for a given pose only, see for example the *C5* robot (116)) or globally, but we will mention here only

globally decoupled robots.

Among the partially decoupled robots a special type has attracted interest: those decoupling orientation and translation. We have already presented such a robot with the *Nabla 6* mentioned by Bernier (36) (figure 2.38). Another architecture presenting this characteristic was suggested by Innocenti (264) (figure 2.52); three links share a common ball-and-socket joint; the modification of these link lengths allows the control of the position of the ball-and-socket joint center. The control of the three remaining link lengths thus allows us to modify the moving platform orientations around this center. The delicate point in the practical realization of this manipulator is the design of the triple ball-and-socket joint. Another interesting

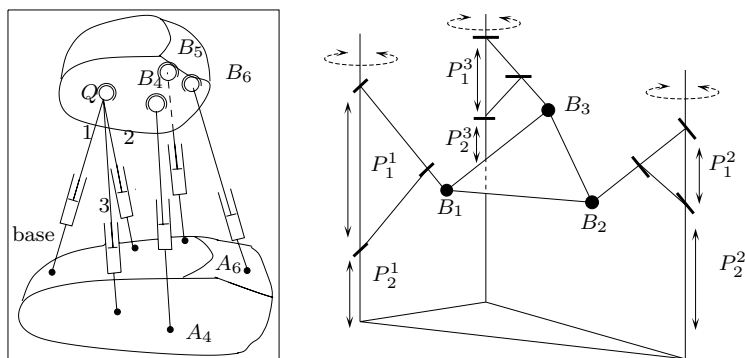


Figure 2.52. Decoupled parallel robots. On the left, Innocenti's robot; three actuators allow the motion of point Q , while the other three allow the control of the rotations around Q (from Innocenti (264)). On the right, the *Tri-Scott*, a variant of the Stewart platform; the upper linear actuator controls the planar motion of the platform while the lower actuator controls the altitude and the remaining orientation (from Zabalza (643)).

example of a partially decoupled robot is a variant of the Stewart platform, the *Tri-Scott* robot (643), figure 2.52. In this robot the upper linear actuators control the planar motion of the platform while the lower actuators control the altitude and the remaining orientation d.o.f.

Another possibility for the creation of a partially decoupled robot is the combination of two robots allowing translations; this is what Lallemand offers with the *2-Delta* (339) made of two robots of the *Delta* type that fit into each other (figure 2.52). Along the same line, we may mention the robot suggested by Ben-Horin (32) which uses two planar parallel robots, as described by Brodsky (58) (figure 2.53).

Completely decoupled parallel robots also exist (for example the *Orthoglide*, figure 2.8) and many other alternatives for decoupled robots have been suggested (see the references Web page). Note however a drawback

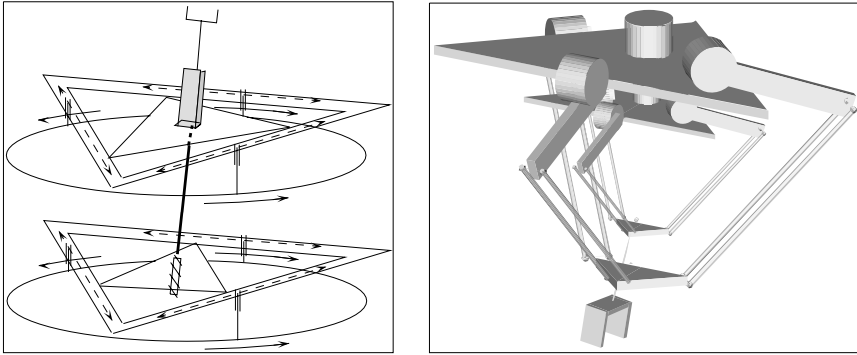


Figure 2.53. On the left, a 6 d.o.f robot using two planar parallel robots (from Ben-Horin (32)). On the right, Lallemand's 2-Delta robot (from Lallemand (339)).

of decoupled robots: the wrench applied on the platform is no longer distributed on all legs, and consequently they will usually have a lower nominal load than non decoupled robots.

2.5. Redundant robots

Although this study is not meant to deal primarily with redundant robots, parallel robot characteristics nevertheless make them very attractive for this particular field. We will see for example that redundancy offers a good opportunity to deal with important and complex issues such as singularity avoidance or solving direct kinematics, problems that will be mentioned in the next chapters. Redundancy may also be used to design fault-tolerant parallel robots (see for example (451) for a failure analysis).

Pierrot (478) distinguishes 3 different types of redundancy, that are illustrated in figure 2.54 on a planar 3-RPR robot:

1. *kinematic redundancy*: at least one of the legs is a motion generator with a larger number of d.o.f than necessary. This may be used for enlarging the workspace (370), see section 6.9. Such redundancy is found in the double tripod of Merkle (figure 2.42),
2. *actuation redundancy*: the end-effector is over-constrained by the actuator, as in the *Archi* robot of LIRMM (478) (figure 2.55). Such redundancy is mostly used for singularity avoidance (370; 612),
3. *measurement redundancy*: the number of sensors is larger than the number of actuated joints. This redundancy plays a role in solving the forward kinematic problem, to reduce the positioning errors (383) and for robot calibration.

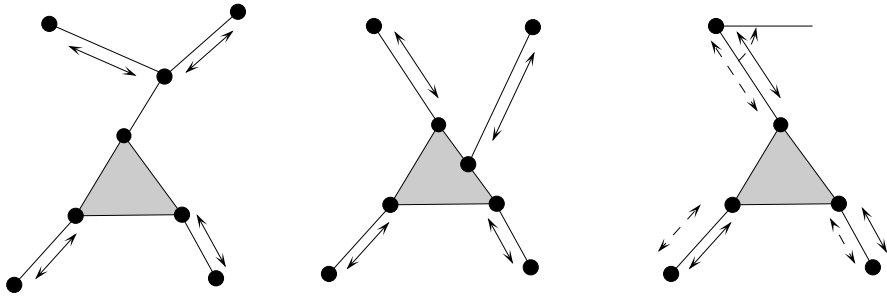


Figure 2.54. From left to right, kinematic redundancy, actuation redundancy and measurement redundancy (measurements are indicated by dashed arrow)

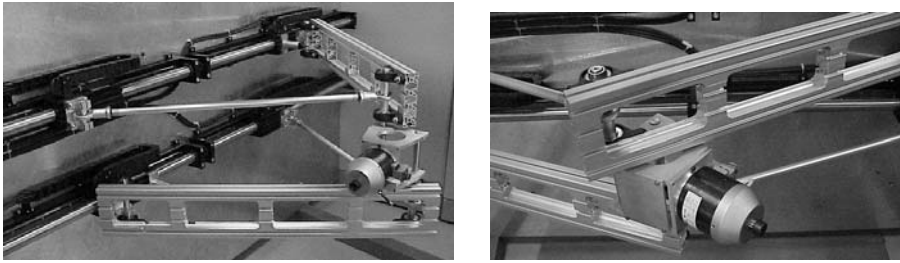


Figure 2.55. The Archi robot of LIRMM, a 3 d.o.f. redundant planar robot with 4 actuators

2.6. Articulated truss and binary actuation

Articulated trusses are examples of highly redundant parallel manipulators. One possible truss architecture simply consists in piling up parallel robots. The now defunct Logabex company put this idea into practice with its *LX4* robot presented in figure 2.56. This type of manipulator is interesting because it is extremely redundant, its workspace is large and the ratio (load capacity)/mass is good: the *LX4* mass is 120 kg for a transportable load of 75 kg (87). However, such a robot is difficult to control.

The truss and variable geometry manipulators (called *VGT*, for Variable Geometry Truss) appear in the works of Miura (424), Sincarsin (543), Reinholtz (505), and Seguchi (527)* to name a few (additional references will be found on the references Web page). The truss they suggest is made of piled up articulated octahedra with three links of varying lengths (figure 2.57). This type of structure allows us to obtain manipulators with a large workspace, light mass and high redundancy, but their kinematics, synthesis and control pose interesting problems. The main envisioned use of the VGT was a spatial truss, but the truss developed by NASA was a

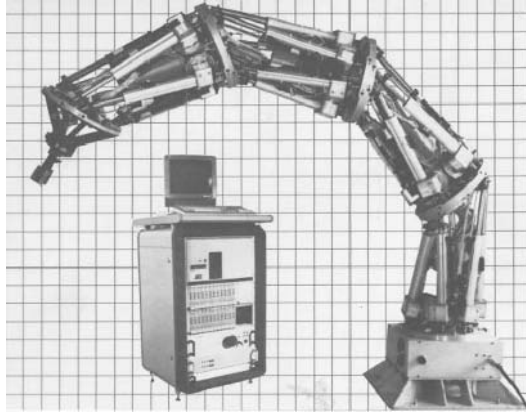


Figure 2.56. The Logabex robot *LX4*, made by piling up *left hands* (photograph by kind permission of the Logabex company).

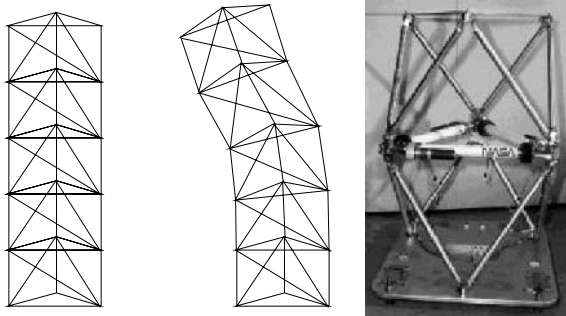


Figure 2.57. On the left, one of Seguchi's truss structures. Each module possesses a octahedral structure with three variable length links (from Seguchi (528)). On the right, an element of the truss developed by NASA, initially proposed by Reinholtz (505).

key element of a long-reach arm developed for the DOE for waste removal (figure 2.58). We may also mention the possible use of active trusses for modifying the shape of an aircraft wing (494).

Koliskor's truss is also worth mentioning (324) as it introduces a new mode of actuation by using linear *binary* actuators that have only 2 possible states, either fully extended or fully retracted. This truss is made of piled up parallel manipulators of the *6-UPS* type, a structure that was studied in detail by Chirikjian (94). The interesting feature of this manipulator is the very large number of poses it can reach (figure 2.59). For instance, with five modules the number of positions that the end-effector can reach rise to 2^{30} (if we assume that the robot pose is always the same for a given set of leg lengths, otherwise the number of reachable poses is

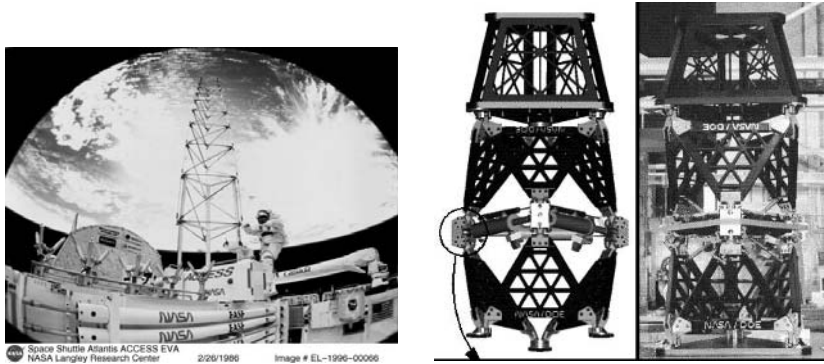


Figure 2.58. A truss deployed in space and the long-reach arm developed for the DOE for waste removal.

even higher), while their control is quite simple. A similar mechanism, but based on a different module structure and with binary polymer actuators, has been recently designed by the MIT under the name Binary Robotic Articulated Integrated Devices (*BRAID*), figure 2.59,(211). Binary robots may also be an interesting solution for micro-positioning. For that purpose Culpepper proposes using Discrete Nano-Actuation Technology (*DNAT*) actuators (113). The moving part of the actuator is connected to its base by 2 springs with different stiffnesses. Opposed constraint elements, such as electrical magnets, may attract the moving part. According to the actuation scheme, the actuator has 4 states. Although the kinematics and

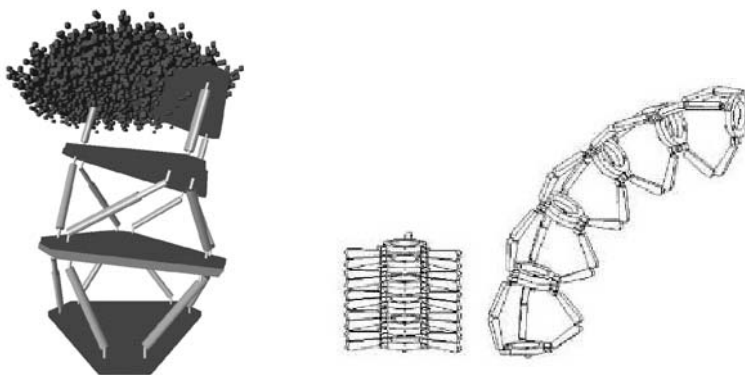


Figure 2.59. On the left the poses that can reach a binary robot with 3 elements (from Ebert-Uphoff (151)). On the right, the *BRAID* mechanism of MIT.

workspace analysis of binary manipulators are complex, as mentioned by Ebert-Uphoff (151) and Lee (351)*, it may be believed that they are ap-

appropriate for many applications (433). Indeed they offer a low-cost, rugged solution for fast pick-and-place or spatial application.

2.7. MEMS and micro-positioning robots

Parallel robot were considered early for application as Micro Electro Mechanical Systems (MEMS). Probably one of the first such systems was proposed by Behi (30), who uses polysilicon micromachining for designing a passive planar 3-*RRR* mechanism. The joints that were used were the same as those used at the macro scale. But an interesting point is that the small motion of the joints allows us to use flexure hinges, as shown in figure 2.60 with a planar 3-*PRR* robot. Merkle also proposed very early a micro hexapod (396) (figure 2.60). Parallel robots are also often used

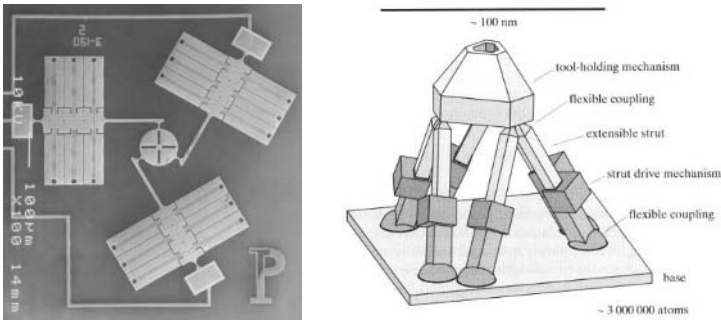


Figure 2.60. On the left, a 3-*PRR* planar mechanism and on the right a proposal of a micro hexapod (from Merkle (396)).

for micro-positioning as in the Physik Instrumente *F206* based on the active wrist concept (figure 2.61). Fine positioning devices may be designed to have a high stiffness, for example Portman (482) proposed a "rigid" Gough platform in which motions are obtained by small variations of the leg lengths through the deformation of a hollow cylinder under hydraulic pressure and elastic deformation of the attachment points of the leg. Another interesting device is the 6 d.o.f. *HexFlex* which uses a planar compliant mechanism (114)*. Three magnetic actuators in the mechanism plane act as linear actuators in a 3-*PRR* planar robot, while three magnetic actuators are perpendicular to the mechanism (figure 2.62) (this robot has therefore some similarity with the Stewart platform). Another interesting device has been proposed by Culpepper (115), figure 2.63, with two possible actuation schemes. In the first scheme, balls are attached to the platform and slide on grooves whose sides are connected to linear actuators. In the second scheme, an eccentric ball-shaft is connected to a rotary-linear actuator.



Figure 2.61. The F206 micro-positioner of Physik Instrumente based on the active wrist concept (courtesy Physik Instrumente)



Figure 2.62. The HexFlex 6 d.o.f. robot for micro-positioning. The planar compliant mechanism on the left is deformed by 6 magnetic actuators, 3 in the mechanism plane and 3 perpendicular to the plane

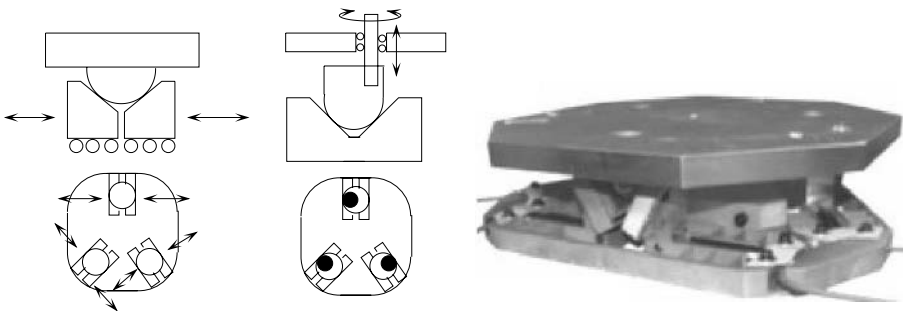


Figure 2.63. Two actuation schemes for a 6 d.o.f. robot based on the motion of balls in a groove. On the right, the sides of the groove are linearly actuated (A), and on the left, an eccentric ball-shaft connected to a rotary-linear actuator allows control of the motion of the platform (from Culpepper (115)). The picture on the right shows a prototype of concept (A) using piezo-electric actuators.

2.8. Wire robots

The replacement of the rigid links by wires that can be coiled or uncoiled (or any other wire shortening method, see (538) for a twisting wire actuation scheme) may have interesting advantages:

- a larger workspace, as wires allow for large changes in the leg lengths and simplify the management of leg interference
- a smaller mobile mass that favor high dynamics
- low interaction with the user, that may be useful for haptic devices.

However, wires impose a constraint: they can pull but cannot push. This constraint will play an important role in the calculation of many of the performances of the robot: for example workspace computation cannot be based only on the allowed change in the leg lengths, but must take also into account the statics of the robot. This issue will not be addressed in this book, but useful references on the subject can be found in the references Web page.

Apart from the robot proposed by Landsberger (342) (figure 2.20), the use of wire robots was suggested by Albus (4)* and his team at the National Institute of Standard and Technology (NIST) in order to realize a crane in the *Robocrane* project. For this system, two platforms shaped like equilateral triangles are connected by 6 wires that can coil and uncoil. The whole system is placed at the end of a classical crane: a semi-rigid mechanism is thus obtained. It allows control of both translation and rotation, which is an unusual feature for cranes in general. Target applications are manipulation of ship components, ship repairs ("Flying Carpet" project), beam assembly (application of wire robots in the building industry has already been mentioned by Higuchi (235) and Ming (421)*) and waste management. In the shipping trade, August Design Inc.[□] ^{WC} uses a parallel crane for the loading of ships, and particularly for automated cargo transfer (*AACTS* system: *Automated All-weather Cargo Transfer System*) and intermodal transfert (Direct Acquisition Rail To Ship, *DARTS* system).

Other applications of wire robot are tasks involving high speed: ultra-fast assembly (with the *Falcon* suggested by Kawamura (305), figure 2.65) or vibration testing (with the *Segesta* robot of Duisburg University (236)*, figure 2.65).

Another possible use of wire robots is as a measuring device for the pose of objects: this was suggested by Geng (182), Zhuang (660), Sheldon (533), and more recently by Thomas (576) and Williams (622). A related application is the *EasyTeach* device proposed by Fanuc Robotics for trajectory teaching: the 3-wires system allows one to determine the position of the painting tool, while a gyroscopic head measure its orientation (figure 2.66).



Figure 2.64. On the left, the NIST Robocrane (courtesy NIST). On the right, the DARTS system (courtesy August Design)

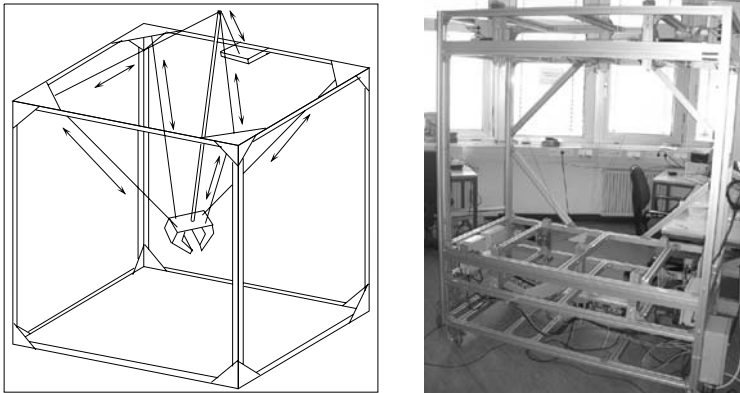


Figure 2.65. On the left, the wire robot *Falcon* (from Kawamura (305)). On the right, the Segesta robot of Duisburg-Essen University: this robot has been used for vibration testing (from Hiller (236)).

The high speeds of wire robots offer possibilities for simulators. This is illustrated by the SACSO robot that will be used to test a plane model in a wind tunnel (338). Wire robots may also be used to design a virtual reality motion base with the advantage of a larger workspace (567).

Finally a very promising application of wire robots is the *SkyCam*^{SC}, a 3 d.o.f. wire robot with a wrist head mounting a camera that allows a spectacular overview of sports events (figure 2.66).

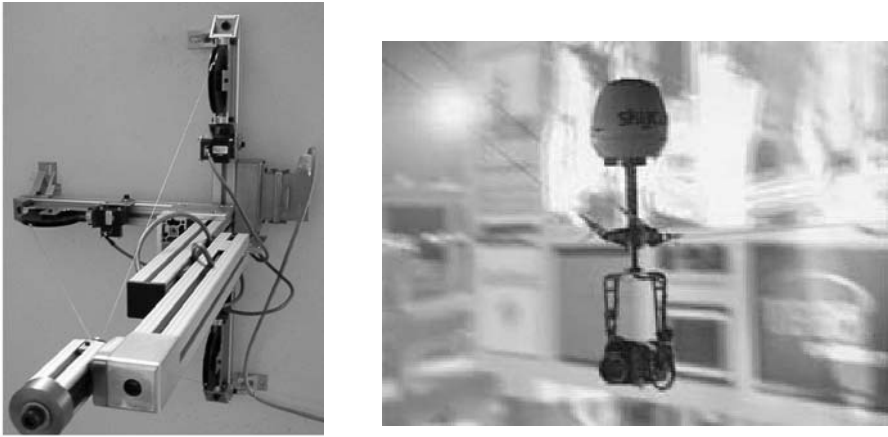


Figure 2.66. On the left, the Easyteach device, here seen from above, used for trajectory teaching. The wire system allows the determination of the location of a painting tool (courtesy Fanuc Robotics). On the right, the SkyCam wire robot, a 3 d.o.f. robot allowing spectacular overviews of sports events (courtesy CFInFlight Ltd)



Figure 2.67. The SACSO wire robot for testing a plane model in a wind tunnel (copyright Onera).

2.9. Examples of applications

The number of applications in which parallel structures have played a role is so large that we cannot mention them all. We will present a few representative examples.

2.9.1. SPATIAL APPLICATIONS

Parallel mechanisms have been proposed for spatial devices for a long time: a very early application had actually been studied for the lunar module landing gear (510).

The truss structures mentioned earlier are particularly adapted to the building of re-configurable spatial structures (562)*, and inflatable hexapods have been considered for the deployment of large lightweight structures for space use (35).

A simulator for the study of robotized assembly in space has been developed, for example the 6-UPS robot *CKCM* studied for the NASA Goddard Space Flight Center by Nguyen and his collaborators (448)*. Simulation of micro-gravity to test a vibration isolation system was proposed by Idle (262), while Dubowsky (143)* designed the *VES* simulator, where a parallel robot was used for the simulation of the behavior of a serial robot, and for the study of the impacts between a free object in space and a structure.

A tendon suspended platform robot, *Charlotte*, was designed by McDonnell Douglas (now Boeing) to automate crew tasks, and flew in the space shuttle mission STS-63 of February 1995 (577) (figure 2.68).

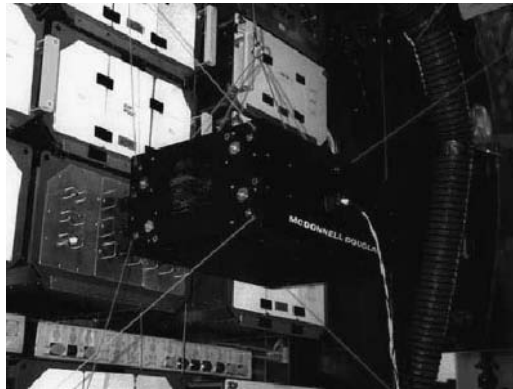


Figure 2.68. The *Charlotte* tendon suspended platform robot designed by McDonnell Douglas that has been tested in the space shuttle mission STS-63

Another application of parallel robots, which does not exactly enter the field of spatial activities, could be that of aerial pointing devices, as suggested early by a group at Canterbury University^{□ NZ} (147); these are now commercially available (figure 2.69).

Along the same lines very successful utilisation of parallel structures is as a pointing device for telescopes. Almost all recent land-based telescopes

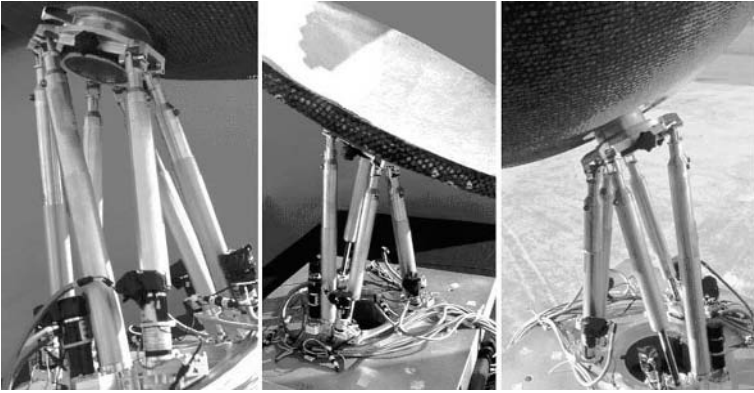


Figure 2.69. An example of use of a parallel robot for the control of antenna orientation. Note the small dimension of the actuators compared to the dish (courtesy IN-SNEC^{□ IS})

use hexapods, either as a secondary mirror alignment system (Telescopio Nazionale Galileo, University of Arizona MMT, both designed by ADS International, UKIRT, the ESO VISTA^{□ TH} or the GRANTECAN on the Canary Islands, figure 2.70), as a primary mirror pointing device, or as a scientific instrument (figure 2.71). Let us also mention the possible use



Figure 2.70. Hexapod used for secondary mirror alignment: on the left at the United Kingdom Infra-Red Telescope (UKIRT) and on the right at the Telescopio Nazionale Galileo (courtesy ADS International^{□ ADS})

of wires robots and 6-UPS structures for a very large radio telescope in China (558)*.

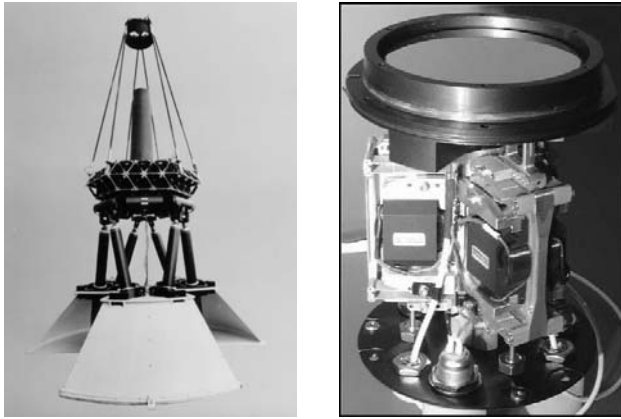


Figure 2.71. On the left, a telescope pointing system (courtesy Vertex Antennentechnik[□] ^{V^A}) that has been installed at the Cerro Armazones observatory in Chile. On the right the 3-d.o.f. tip-tilt-piston mirror NAOS mechanism for the VLT–Naos field selector: three linear actuators on the base each translate a steel wire rigidly connected to the mirror mount; the flexibility of the wires allows one to control the orientation/vertical translation of the mirror (courtesy of CSEM[□] ^{C^{SN}}, Switzerland)

Parallel structures may also be used for satellite instrumentation. A parallel robot has been considered for the SAGE III experimentation, and a version that should be mounted on the ISS has been developed for ESA by ADS International (figure 2.72).

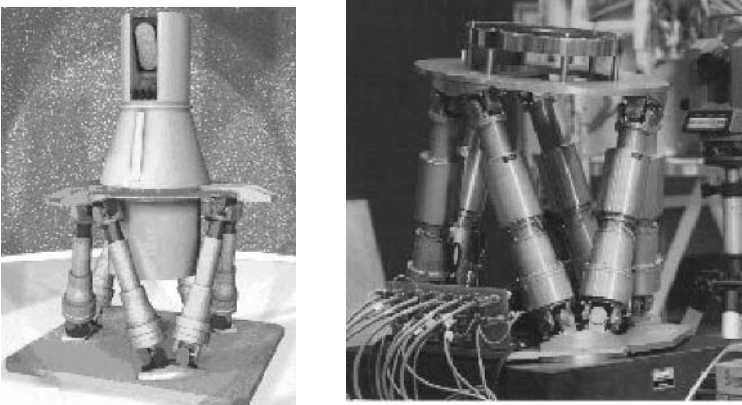


Figure 2.72. A hexapod for the pointing of the SAGE III instrumentation (courtesy ADS International)

2.9.2. VIBRATION

The possible high bandwidth of parallel structures make them good candidates for vibration damping. Although such applications have been considered for many years (180)* theoretical studies are still going on (332), (394)*, (529); active vibration suppressors are now commercially available (figures 2.73,2.74).

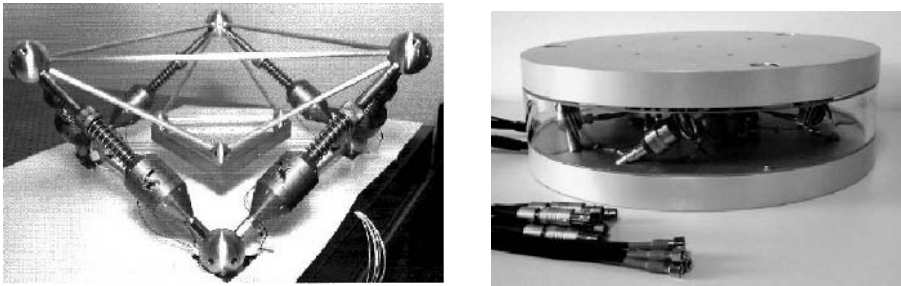


Figure 2.73. On the left, the vibration damper of the University of Wyoming uses voice coil actuators (from McInroy (394)). On the right, an active hexapod of MicroMega (courtesy Micromega Dynamics^{□ MD})

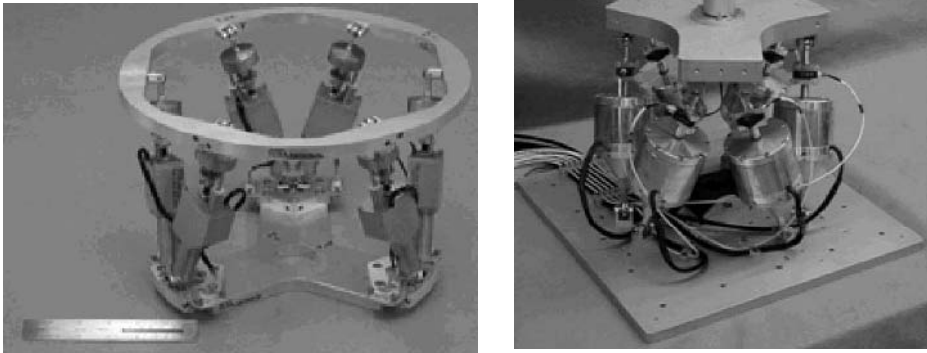


Figure 2.74. The vibration isolator PH1 and PHEX1 of CSA Engineering (courtesy CSA Engineering^{□ CSA})

An interesting example is that of the VISS (*Vibration, Isolation, Suppression and Steering System*)^{□ VISS}, developed jointly by the American Air Force, Honeywell, Trisys and JPL in order to isolate on-board measurement systems (whether optical, laser, etc.) from the body of a satellite.

It has been used very successfully for various satellites, and research is still going on in this domain (520): the sequel of the VISS, the HXA

(*Hexapod Assembly*) has been used on board the PICOSat launched in 2001^{□ HXA}. We may also mention the hexapod of Energen, using magnetostrictive actuators^{□ EN}. Not all vibration isolator have 6 legs: figure 2.75 shows an octopod that was used in 1997 and 2002 during the Space Shuttle missions STS 82 and STS 109 to isolate the shuttle payload. Parallel structures may also be used to submit components to vibration testing. For example OHE Hagenbuch^{□ OHE} proposes the hydraulic *Hexamove* (figure 2.75).

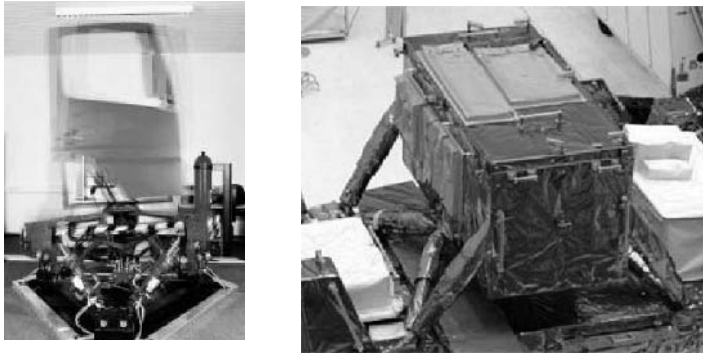


Figure 2.75. On the left, the Hagenbuch *Hexamove* in a vibration test. On the right, an octopod used during a space shuttle launch (courtesy OHE Hagenbuch, CSA).

2.9.3. MEDICAL APPLICATIONS

Robots are slowly entering the medical field with systems such as the Da Vinci (Intuitive Surgical) or Zeus (Computer Motion) robots. Parallel structures play also a role in this evolution and their potential use was mentioned early on. For example in the *Crigos* system of Brandt (55), a parallel robot was used for orthopedic surgery operations, while the INRIA active wrist has been successfully employed for ophthalmological surgery operations on dogs (205). But as for their serial counterpart not many of the laboratory prototypes have found their way into the field (although research is still going on (360; 362; 613)) but we may mention two examples of such use.

In the *SurgiScope* system provided by ISIS Robotics^{□ ISIS}, a *Delta* type robot is used as a microscope stand (figure 2.76).

Another example addresses one difficulty for surgical robot which is to follow the patient's motion. This has motivated the development of the *MARS* robot which has a 6-UPS structure (537); the robot is directly mounted on the patient's bony structure near the surgical site. This robot has been used as a surgical tool guiding spinal pedicle screws placement (figure 2.77), and is sold by Mazor^{□ MAZ} as the *Spine Assist* robot. A similar robot for knee arthroplasty, the *MBARS* (Mini Bone-Attached Robotic

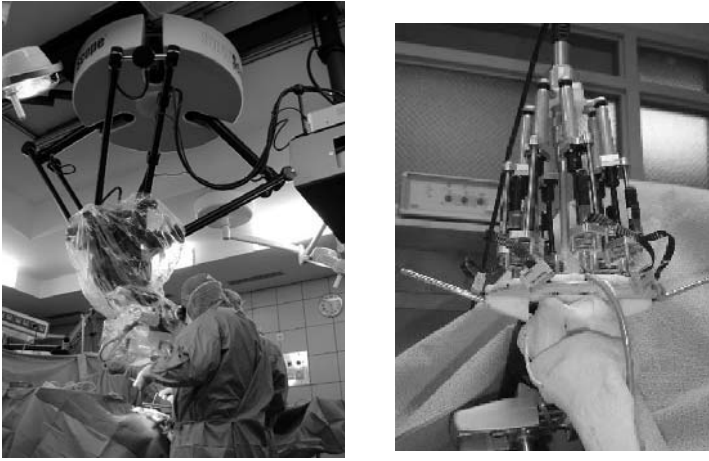


Figure 2.76. On the left, is the ISIS/SurgiScope system using a Delta robot as microscope stand (here at the Val de Grâce hospital in Paris, courtesy of ISIS). On the right, is the CMU MBARS (Mini Bone-Attached Robotic System).

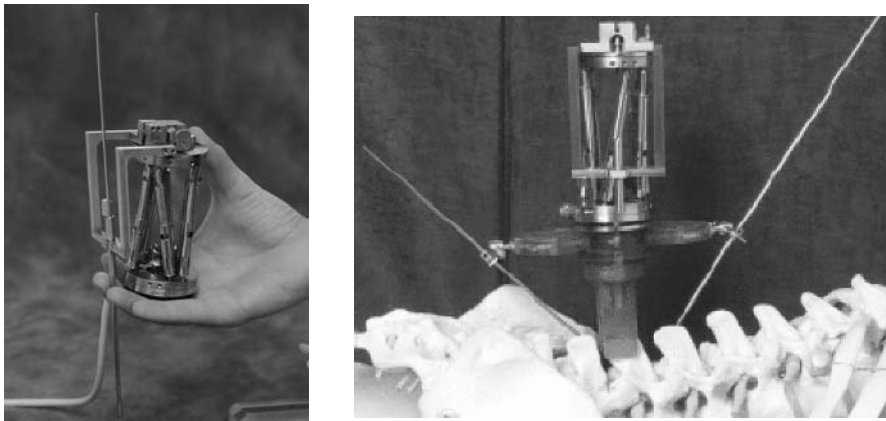


Figure 2.77. The *MARS/Spine Assist* robot for spinal surgery (courtesy of Mazor Surgical Technologies)

System) is currently being developed at CMU[□] CMU (figure 2.76). The Spine Assist and MBARS may indicate a trend toward another approach to surgical robotics, based on small, adaptive and relatively low-cost robots, compared to expensive, large scale structures (see for example the surgical robot *Romed* proposed by the Fraunhofer IPA, figure 2.78, for the same purpose).

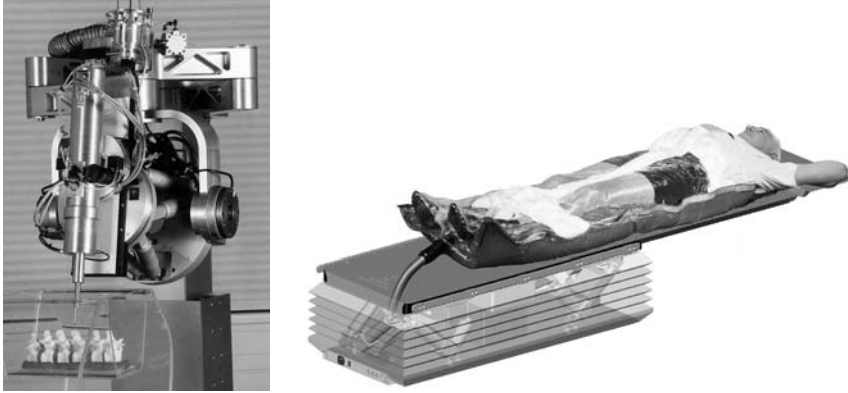


Figure 2.78. On the left, the *Romed* surgical robot proposed by the Fraunhofer IPA. On the right, the HexaPOD 6 d.o.f. robotic treatment couch for patient positioning during radiation therapy (courtesy Medical Intelligence^{□ MI}) with a ± 3 degree orientation ability, ± 30 mm translation motion, a 0.1 mm resolution and a load of 185 kg

It is possible to use another advantage of parallel structures: compared to their serial counterpart they are much less sensitive to scaling effect, and are therefore appropriate for micro-robots. For medical applications this is appropriate for minimal invasive surgery, especially endoscopy. Wendlandt (616) built an endoscope active head made of a 3 degrees of freedom robot that is actuated by wires^{□ BU}. We have suggested another solution for an endoscopic head with the *Mips* 3-d.o.f. robot illustrated in figure 2.19, which uses electrical motors (416).

A chapter of the references Web page is devoted to medical applications; apart from the use of parallel structures in various medical fields we will just mention that kinematic of parallel structures are also used for a better understanding of the kinematics of complex human joints such as the knee (466)* or the shoulder complex (358)*, for correction of bone-deformities (134; 184; 279) (replacement of the Ilizarov apparatus by an hexapod), for rehabilitation and sports training with the *Caren* platform of Motek^{□ MO} (figure 2.82), and for mouth opening and closing training (570)*.

2.9.4. SIMULATORS

After the initial proposal of Stewart, there have been numerous developments of parallel robots for flight simulators (322). Nowadays, many companies are building virtual reality motion simulators, not only for aircrafts but also for ships, train, truck driving^{□ INRT}; this sector is probably the one in which parallel structures are the most successful. We mention CAE

(Canada)^{□ CAE}, Thalès (France)^{□ TH}, Frasca (USA)^{□ FRA}. An example of such a simulator is presented in the photograph 2.79. The largest driv-



Figure 2.79. The Airbus A340 simulator (realization THOMSON-CSF, photograph by P. Palomba).

ing simulator is the National Advanced Driving Simulator (NADS) at the University of Iowa (figure 2.80)^{□ NADS}.



Figure 2.80. The NADS driving simulator at the University of Iowa

One impressive parallel manipulator, the *Turret Motion Based Simulator* (TMBS) was built by the US Army Center for Tanks Research (TACOM)[□] *TMBS*. The actuators are hydraulic, the carrying capacity is 27 tons, with vertical accelerations of about 4-6 g. The aim is to test the ergonomics of the interior of the tank, and to study arm stabilization systems (figure 2.81). A surprising simulator is *Persival*, initially developed by the french École Nationale d'Équitation (figure 2.81): it aims at giving novice horse riders a preliminary training without putting a good horse education in jeopardy; it is now commercially available[□] *PHS*.

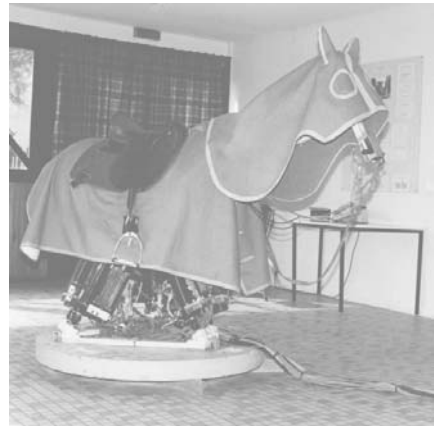
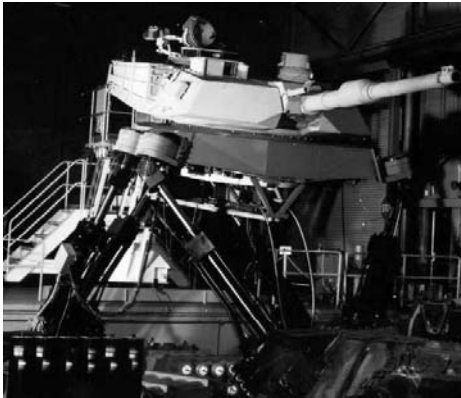


Figure 2.81. On the left, the Turret Motion Based Simulator of the US Army TACOM. On the right, the horse simulator *Persival* from the École Nationale d'Équitation, built in collaboration with ENAC (photograph E.N.E.).

Other surprising simulators are the bicycle simulator developed by KAIST in Korea (figure 2.82), and the *Caren* system of Motek which is used for sports training and medical rehabilitation.

2.9.5. INDUSTRIAL APPLICATIONS

Historically, robotic parallel structures were first designed for assembly tasks (381). Numerous feasibility demonstrations of assembly (with or without force-feedback) have been made by Pierrot (476)*, Reboulet (499)* and ourselves, to name a few. It must be recognized that assembly is by no means the largest field of industrial application for parallel structures (although it may be thought that assembly may become important for the development of MEMS or for the assembly of large aerospace components (59)). We will present here industrial domains in which parallel structures play, or will play, an important role.

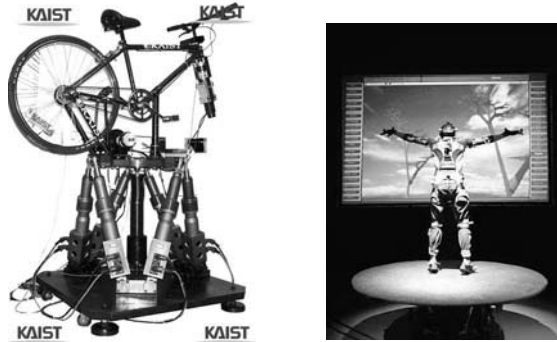


Figure 2.82. On the left, a bicycle simulator (courtesy KAIST). On the right, the *Caren* motion base used for sports training and medical rehabilitation (courtesy Motek).

2.9.5.1 Machine-tool

Among the applications of parallel robots the one that may have the largest economic impact is in machine-tools. The first milling machine was presented by the Giddings & Lewis company (now part of Thyssen Krupp) at the IMTS machine-tool exhibition in Chicago in 1994; it was the main attraction, under the name *Variax* (figure 2.83). It was based on the principle of the Gough platform, thus realizing the vision of the reviewers of Stewart's paper. According to its manufacturer, despite the fact that the machine possesses 6 degrees of freedom, it was 5 times stiffer than a classical machine and had much superior advance speed.

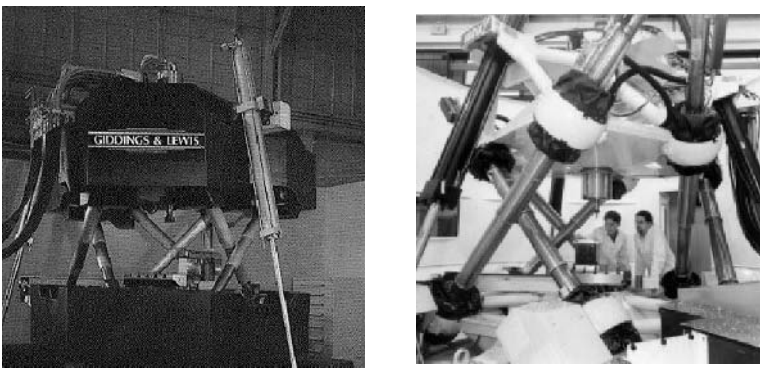


Figure 2.83. The *Variax* of Gidding & Lewis, the first industrial parallel milling machine presented in 1994 at IMTS

Although we will present promising machine-tools based on parallel structures, only a few are already in use in industry. Various factors ex-

plain this slow progression:

- after only 10 years of development such structures have not reached the level of sophistication of classical machines with centuries of experience
- the properties of parallel structures are fundamentally different from their classical serial counterparts. This has often been neglected by manufacturers that have focused mainly on designing the mechanical components while discarding a more global design approach. This has led to the development of prototypes with disastrous design errors (on that topic let us quote Paul Sheldon, the designer of the Variax: "there have been so many poor designs—designs that ignore fundamental engineering truths—that the machines now have a reputation for not being stiff and not being accurate")
- parallel robots are intrinsically *non-linear*, while the CNC controllers that are used to control parallel structures have been designed for linear machines. It is not necessary to be an expert in control theory to understand that using a classical CNC controller for a parallel robot will not allow one to get the best out of the machine. We have conducted an extensive simulation of the whole manufacturing system (CAD, controller and the parallel machine) including all possible sources of errors, and we have studied typical manufacturing trajectories. For the positioning error (i.e. the maximum distance between the desired trajectory and the real one) the result is that, provided the machine is well designed, (a complex issue, see the "Design" chapter), 10 to 20% of the errors come from the CAD system, 70% from the control, while only 10% are induced by the machine itself. Hence it is clear that ad-hoc controllers for parallel structures must be developed, and that the output of the CAD system must be adapted for a parallel machine.

The potential use of parallel structures in the field of machine-tools has motivated a large research effort, especially in mechanical components, vibration control, and for the synthesis of structures with fewer than 6 d.o.f. Indeed most machining tasks require only from 3 to 5 d.o.f. (although we have already seen that redundancy may have advantages). We will mention here only industrial products, although a large number of academic prototypes have been developed (for example we have already mentioned the Orthoglide, figure 2.8 and the Eclipse, figure 2.50) and can be found in the corresponding chapter in the references Web page. We must also mention that some industrial products that were proposed in 2000 are no longer available, either because the focus of the companies has changed or because in the fast-moving machine-tool market several companies have disappeared.

Machine-tools with 2 d.o.f. planar parallel structure are commercially

available. We may mention for example the *Genius 500* of Cross Hüller (part of Thyssen Krupp) or the *Trijoint 900H* of Kocosvit Mas (figure 2.84).



Figure 2.84. On the left, the *Genius 500^{G500}* (courtesy Cross Hüller) and on the right, the *Trijoint 900H^{900H}* (courtesy Kocosvit Mas)

Both machines use PRRP chain allowing to get 2 planar translation d.o.f. for the spindle.

Various machines with parallel 3 d.o.f. have been proposed: the *Tricept* that we have already presented (figure 2.9), the *Sprint Z3^{Z3}* of DS Technologie with 2 d.o.f. in orientation (± 45 degrees) and one translation (up to 370 mm), figure 2.85, both being compared in (629); the *SKM 400^{SKM}* of Starrage-Heckert with a traverse rate of 100 mm/mn, the linear *Delta* type *Quickstep HS500^{HS}* of Krauseco & Mauser, *Index V100*, figure 2.86,^{□ V100} and *Urane SX* of Comau.

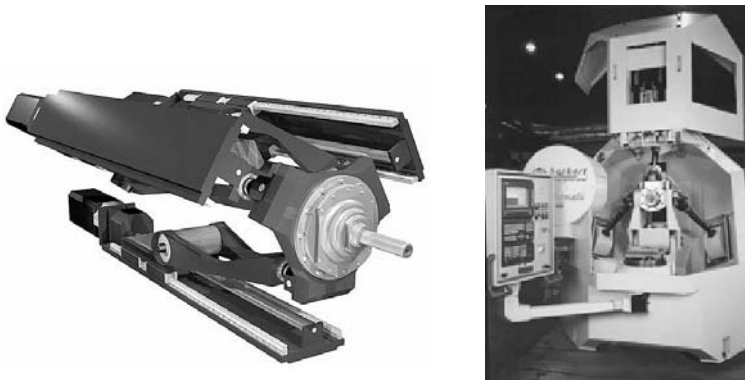


Figure 2.85. On the left, the *Sprint Z3*, 2 d.o.f. in orientation, 1 in translation (courtesy DS Technologie). On the right, the *SKM 400* (courtesy Starrag-Heckert).

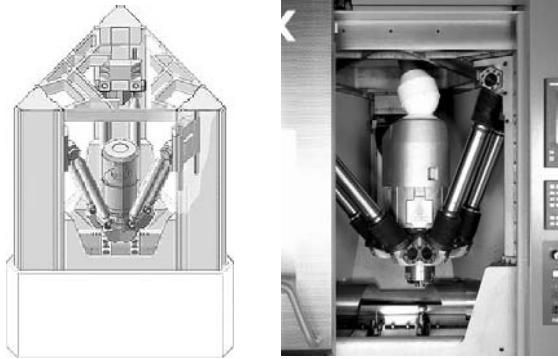


Figure 2.86. The linear Delta type V100 (courtesy Index).

In a recent interview Paul Sheldon, one of the prime movers behind the Variax machine, as former vice-president of Research of Giddings & Lewis, mentions that the future may lie in 3-axis parallel structures, because for metal-cutting machines the number of 3-axis machines far exceeds 5-axis ones. A direct consequence is that he has patented the *Triax* machine that apparently may be related to the *Delta*, although the actuation scheme is different (figure 2.87).

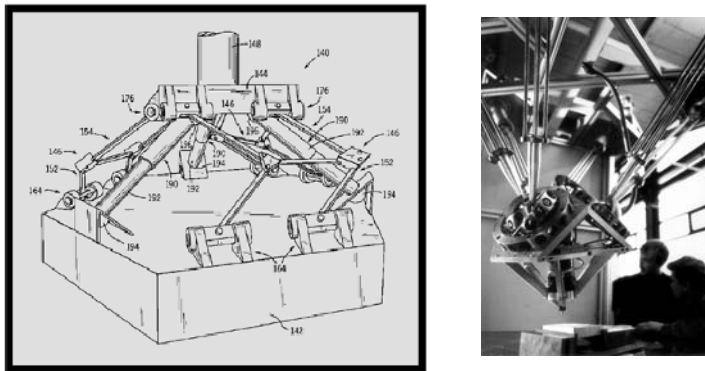


Figure 2.87. On the left, the 3-axis machine *Triax* patented by Sheldon. On the right, the *DiGihex* hexapod of FachHochSchule Bielefeld^{□ FHB}.

An interesting 5-axis machine is the *P800/P2000* of Metrom; this has a clever head mechanism that allows it to use only 5 legs (figure 2.88). More classical 5-axis machines are the *Octahedral Hexapod* of Ingersoll (now part of the Camozzi Group), the *Mikromat 6X^{□ MK}* with seven legs (figure 2.90), the Okuma Cosmo Center *PM-600^{□ OK}* (figure 2.91), the *HexaM* of Toyoda (figure 2.89), the *CMW 380^{□ CMW}* which has been developed

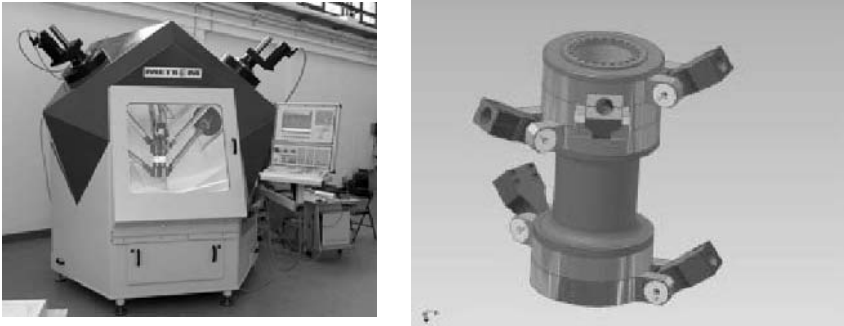


Figure 2.88. The 5-axis *P800* machine-tool. The head mechanism with 3 legs attached to revolute joints sharing the same axis allows it to use only 5 legs (courtesy Metrom)

with the help of various laboratories including INRIA, the *Hexamech-1* of the Savelovo Machine Building Company^{□ HS1}, a hexapod proposed by the Russian company Lopic, the *Pegasus*^{□ PR} of Reichenbacher (figure 2.89) and the DR Mader *Hexapod*^{□ DR}, a company that also produces simulators, (figure 2.92).

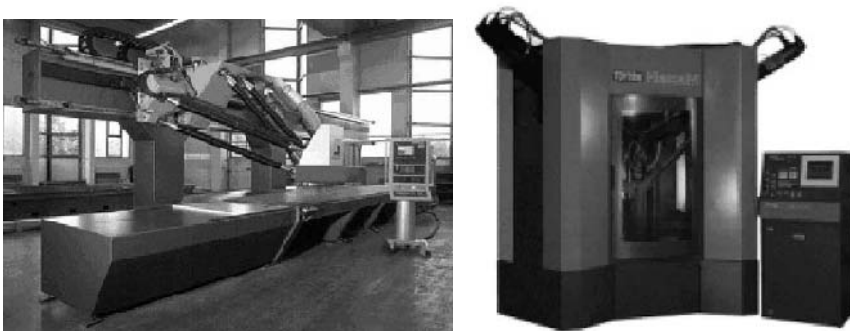


Figure 2.89. On the left, the *Pegasus* (courtesy Reichenbacher). On the right, the *HexaM* of Toyota: the axis of the linear actuator is tilted.

We may also mention a Tricept machine sold by the Norwegian company MultiCraft under the name *MultiCraft 560*^{□ M560} whose main application is the grinding of propellers, although it is unclear if this machine is still sold by this company, Greif Robot Schleifsystem is using it in its robotic grind station.

Research is still going on to improve these parallel machines. European funded research projects have been devoted to this subject: *ROBOTOOL*^{□ RT} and *MACH21*^{□ M21}. National initiatives have also taken place (for example



Figure 2.90. On the left, the *Octahedral Hexapod* of Ingersoll (photo by Kathie Koenig Simon/NIST). On the right, the *6X* (courtesy Mikromat)

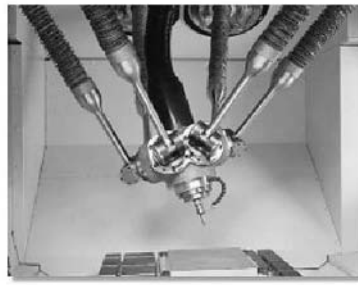


Figure 2.91. The Okuma 5-axis Cosmo Center *PM-600* and a detail of its head (courtesy Okuma)

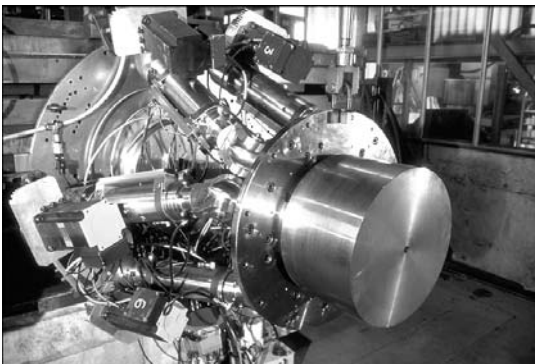


Figure 2.92. On the left, the *CMW 380* (courtesy CMW). On the right, the *DR-Mader hexapod* (courtesy DR-Mader)

in Germany with the *Dynamil* project^{□ DY} funded by the BMBF which led to the design of the *Linapod*, the *Hexact* and the *GeorgV* machine-tool, or in France with the ROBEA project funded by the CNRS) but not all of them have been completed (for example the consortium led by the Sandia National Laboratories was disbanded due to funding problems after only one year). The main research axes are vibration control (with the use of of micro piezo actuator (141)), calibration (we will devote a chapter to this subject), stiffness (243), trajectory determination (532), accuracy analysis (468) and design (a chapter will deal with this subject). Beside the machines designed for relatively heavy-duty, we must also mention that smaller machines for soft material (such as wood) may be designed, see for example the *LME* project^{□ LME}. It must be noted that it is in the field of machine-tools that the largest number of configurations for the attachment point on the base and on the platform have been investigated, see for example the Metrom or Mikromat machines or the *Digihex* hexapod of FH Bielefeld (figure 2.87) with its 6-3 arrangement. We will see that these arrangements have an important influence on the performance of machines.

In the chapter "Singular Configurations" we mention some very special Gough platforms which may be of interest in manufacturing processes.

2.9.5.2 Positioning devices

Fine positioning devices are favorite parallel robots applications. Several companies offers fine positioning devices and we will mention some of them:

- Physik Instrumente^{□ PI} offer a large panel of devices: *M-840*, *M-850* and *M-824* hexapods, Gough platforms showing a resolution better than 1 micrometer, *F206* manipulator, based on the principle of the INRIA active wrist (figure 2.61) and the *P-857* nanopositioning device with flexure hinges
- Micos^{□ MI} proposes the *Paros* positioning device with a repeatability of $\pm 5\mu\text{m}$ for a $350 \times 350 \times 80$ mm workspace (figure 2.93) and the *SpaceFab* for fiber optic alignment with a repeatability of $\pm 0.5\mu\text{m}$ for a $25.4 \times 25.4 \times 12.7$ mm workspace (figure 2.94).
- Alio^{□ AL} exhibits the *HR4* hexapod with a repeatability of $\pm 1\mu\text{m}$ for a $100 \times 100 \times 20$ mm workspace (figure 2.93)
- Hephaist Seiko proposes various 6 d.o.f. positioning robots (*F6*, *SWF6*) with a travel range of ± 15 mm and ± 15 degrees and 3 or 4 d.o.f. robots *NAF3*^{□ HS}

Many other companies produce on-demand positioning hexapods (see for example the piezo-actuated hexapod of Marco^{□ MA}).

Let us also mention the European Synchrotron Radiation Facility (ESRF) positioning devices studied in collaboration with INRIA. The ESRF synchrotron is used for the production of an X-ray beam that is very finely

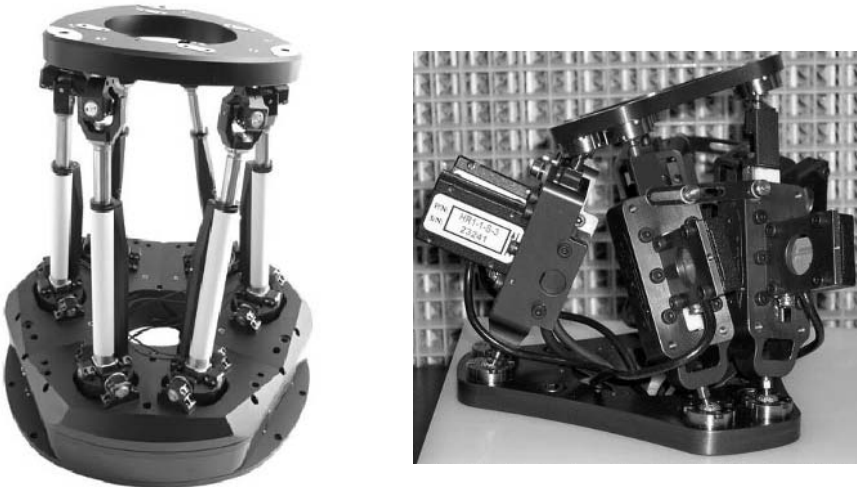


Figure 2.93. On the left, the *Paros* robot (courtesy Micos). On the right, the *HR1* robot (courtesy Alio Industries)

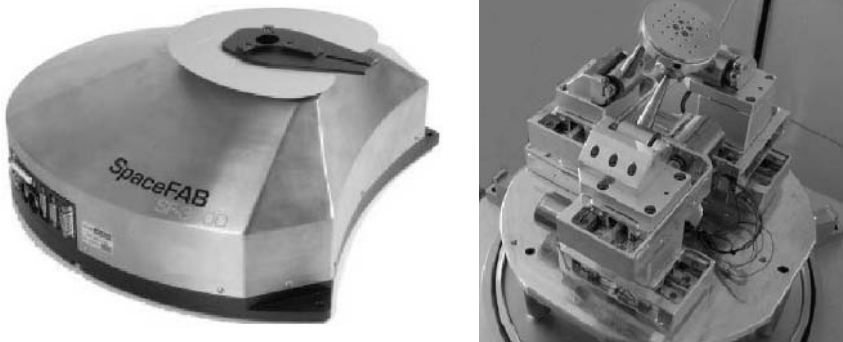


Figure 2.94. The *SpaceFab* robot, a 3-PPS robot for fiber optic alignment (courtesy Micos).

adjusted in frequency and on which experimentation chambers are interposed. This beam is focused with the help of a specific optical device which must be positioned extremely precisely. It should therefore be placed on a 6 d.o.f. positioning device. The weight of the optical device may vary between 500 kg and 2500 kg. Its motions need to be controlled with an error less than a micrometer within a restricted space (typically, a cube whose sides are a few centimeters long). Our study showed that these conditions could be satisfied, and the ESRF realized several prototypes with repeatability,

measured with a load of 230 kg, better than 0.1 micrometer and $1\mu\text{rad}$, and a first resonance peak at 61 Hz (109). Over 40 hexapods of this type are now in use at ESRF. A similar parallel robot has been studied for the

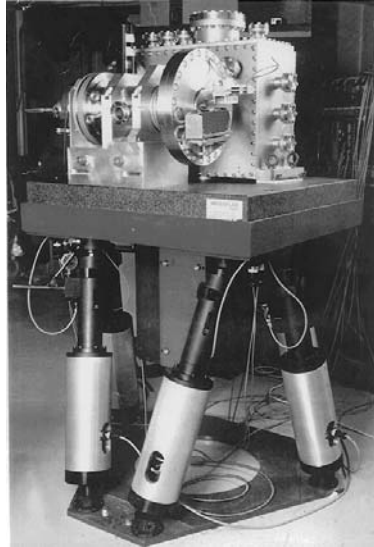


Figure 2.95. The European Synchrotron Radiation Facility positioning device. This robot moves loads that weigh up to 500 kg with an accuracy better than one micrometer.

Laue Langevin Institute for the SALSA project (Strain Analyser for Large & Small scale engineering Applications)^{□ ILL}.

2.9.5.3 Other industrial applications

The high positioning accuracy and large stiffness of parallel robots should mean they will become useful instruments in various industrial fields.

Part handling and spot welding are domains in which parallel robot may play a role. Two companies already provide robots for such application: Fanuc Robotics with the *F-200i*,^{□ F200} and Hexel with the *Hexabot* (figure 2.96), although the later robot is more oriented toward machining. An example of the use of the F-200i in the automotive industry is shown in figure 2.97, and some features of both robots are presented in table 2.1. Note that the load of the F-200i is 100 kg for a 190 kg weight (load/mass ratio of 0.526) while the Hexabot has a load of 91 kg for a mass of 450 kg. Hence the load/mass ratio is 0.526 for the F-200i and 0.202 for the Hexabot; this ratios are much higher than their serial counterparts (see table 1.1).

One of the most successful applications for parallel robots is in packaging, especially with the *Delta* robot, whose very high transfer rate and

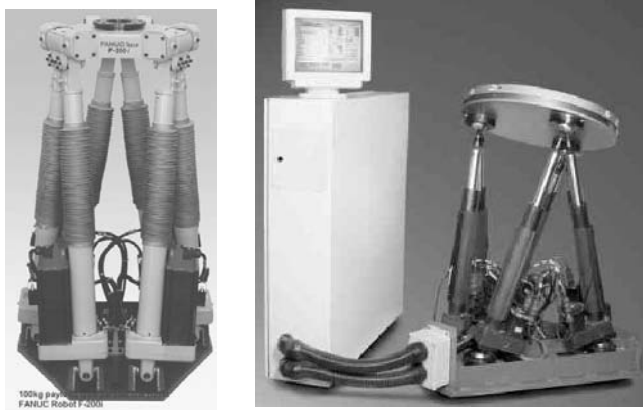


Figure 2.96. On the left, the Fanuc *F-200i* (courtesy Fanuc Robotics,) and on the right, the Hexel *Hexabot* (courtesy Hexel)

Robot	translation (mm)	orientation	load	Repeatability
F-200i	$\pm 750(X-Y)$, 1500(Z)	NA	100 kg	$\pm 0.1\text{mm}$
Hexabot	$\pm 152.5(X-Y)$, 178(Z)	± 25 degree	91 kg	$10 \mu\text{m}$

TABLE 2.1. Some performance indices for the Fanuc F-200i and Hexel Hexabot

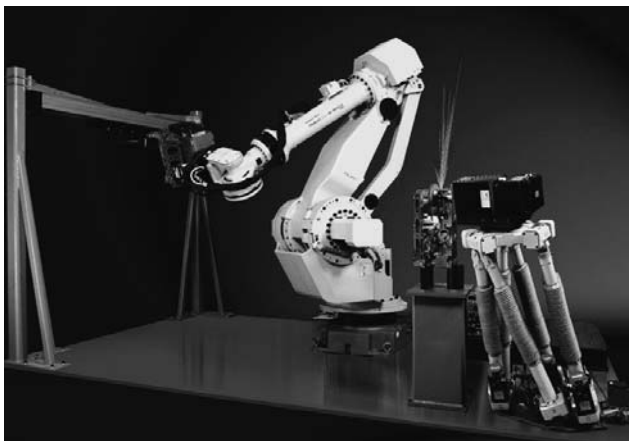


Figure 2.97. An example of engine block handling and deflashing with Fanuc M-900iA/350 serial robot and F-200i parallel robot (courtesy Fanuc Robotics)

high accuracy allows the manipulation of fragile objects, such as those that are met in the food industry. Figure 2.98 presents examples of such applications.

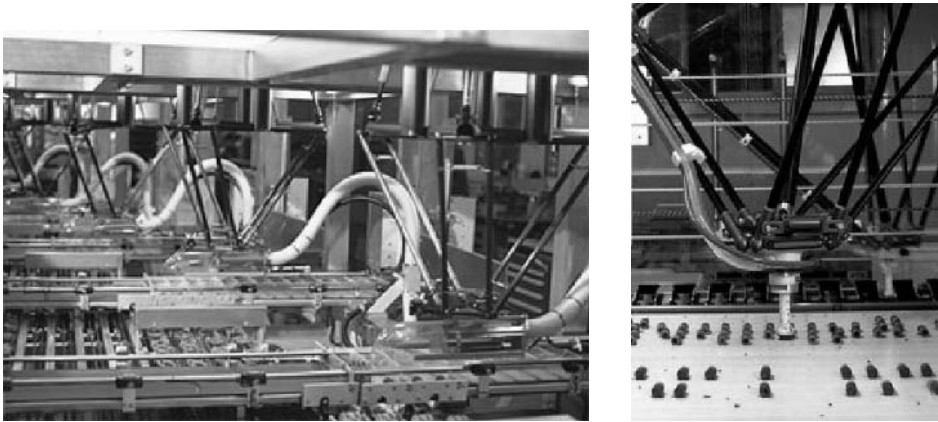


Figure 2.98. The Delta robot being used for packaging in the food industry (courtesy Demarex and SIG Robotics)

The high accuracy of parallel structures makes them good candidates for measuring machines. This application has been mentioned for wire robots but rigid legs may be used as well (figure 2.99). The Lopic Company of



Figure 2.99. On the left, the measuring machine *HCCM* developed by the University of Florida and Perry Automation^{□ PA} with 5 actuated legs and a passive one. A laser interferometry system allows to measure the leg lengths (courtesy Perry Automation). On the right, the *KIM 750* measuring machine of Lopic company, Russia^{□ LA}

Russia is also proposing the *KIM 750* measuring machine (figure 2.99).

We will see also that parallel structures are also very appropriate for force and vibration measurements. This was used very early (1990) by Schönherr (523) to test small flight aircraft (figure 2.100). The FCS^{□ FCS} company has developed a large number of test rigs using force measurements (figure 2.100), and also numerous simulators.

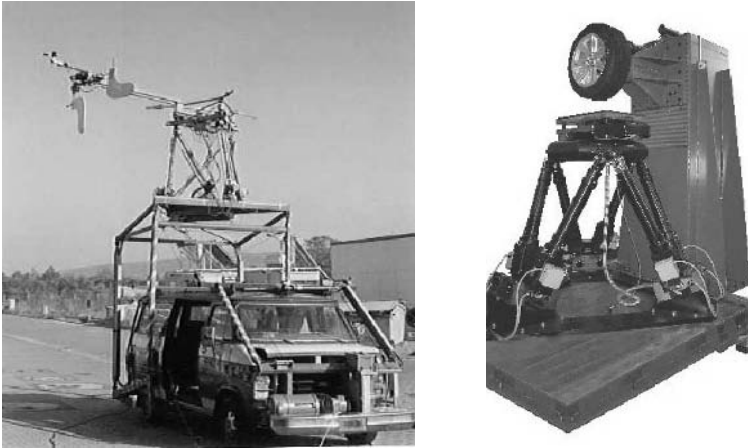


Figure 2.100. On the left, a test device for flight by Schönherr (1990). On the right, a tire test rig (courtesy FCS)

Parallel robots have also been used in miscellaneous applications: footwear testing (426), porcelain decoration with the Copra hexapod^{□ PO}, lens manufacturing with the Hexapod of Gerber Coburn^{□ GC} (a license of the INRIA active wrist). The very high speed of the *Delta* robot has also been used for pick up of microtubes for biological screening tests (100 000 pick ups per day according to RTS Life Science^{□ BS}) or for high-speed packaging (Delta C1000/C2000 of Phoenix Packaging Systems^{□ PD}).

2.9.6. MISCELLANEOUS APPLICATIONS

Parallel robots are in use in many different application domains, and we will now present some examples, without pretending to be exhaustive.

Much research work has been devoted to the use of parallel structures as haptic devices or joysticks (see the corresponding chapter in the references Web page). The high accuracy and force sensitivity is an advantage in such applications while the smaller workspace compared to serial haptic devices is a disadvantage. An example of such a device, the *Omega*^{□ OM} of Forced-dimension, based on a *Delta* robot, is presented in figure 2.101. An original application is the elevator of Hydro, which is used for the installation of the main landing gear of the Airbus A380^{□ ELG}.

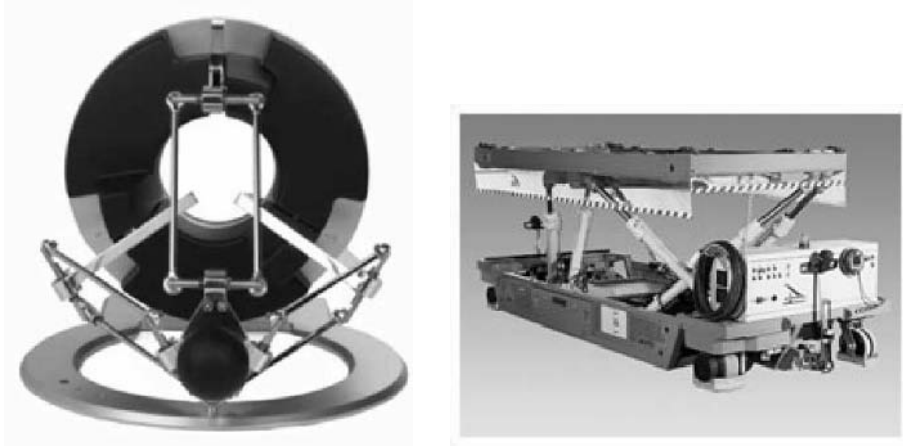


Figure 2.101. On the left, the *Omega*, an haptic 3 d.o.f device based on the Delta structure (courtesy Forcimension). On the right, an elevator for the landing gear of the Airbus A-380 (courtesy Hydro-Gerätbau)

In the leisure industry, numerous parallel structures are used for motion ride simulators. Many companies are offering such motion bases in various dimensions, and this domain is one of the most successful for parallel structures. We mention *AI Group* that has designed the Disney-MGM Star Tour simulator (figure 2.102), *Flight-Avionics*^{FA}, *Servos Simulation*^{SMB} (figure 2.40), *Virtogo*^{VMP} with a pneumatic motion base. Many of them are based on Moog^{MO} or Rexroth^{REX} Hydrauline motion bases, electric or hydraulic, some of them with nominal loads well over 10 tons.

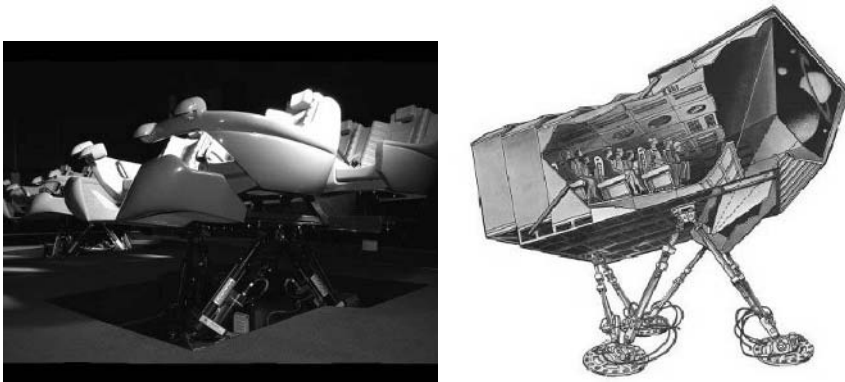


Figure 2.102. On the left, the seven passenger *StarGazer* motion base. On the right, the Disney-MGM *Star Tours* motion base installed in 1987 (courtesy AI Group^{AI})

The *Cinaxe* cinema in La Villette is a good example: 60 spectator seats move as the film is projected on a hemispherical screen.

Other original applications are the munition loader of Bryfogle (63), earthquake simulation (168) and a palm-tree climber (8).

Lastly, we conclude this section with the mention of the existence of a learning set, the *EX 800*, distributed by DeltaLab[□] *DEX*, and made of a Gough platform with electric linear actuators that is controlled by a PC, and of a model allowing easy changing of the location of the passive joints.

2.10. Robots studied in this book

In order to illustrate parallel robot problems, our study will focus on

- planar robots: the 3-*RPR* robot, figure 2.2
- less than 6 d.o.f.: the 3-*UPU* robot, figure 2.10
- 6 d.o.f. robots: the 6-*UPS* with 3 variants presented in figure 2.32. They all possess a planar base and moving platform, but differ in their joint center layouts. The first robot, called MSSM, has a triangular base and end-effector, like the Bricard articulated octahedron (56). The second robot, called TSSM, has a triangular end-effector and a hexagonal base. The third is the SSM which has two hexagonal platforms.
- 6 d.o.f. robots: the 6-*PUS* presented in figure 2.34 sometime mentioned as the *active wrist*.

2.11. Exercises

Exercise 2.1: Show that the 3 degrees of freedom manipulator of figure 2.17 possesses an Earl parallelism index of 1.

Exercise 2.2: Show how taking geometry into account influences calculations on Lambert robot mobility (figure 2.17).

Exercise 2.3: Determine how the joint axes in Lambert's structure may be defined so that the platform d.o.f. are a rotation around a y axis and translation along y, z axis

Exercise 2.4: Calculate the mobility of Gosselin's spherical robot (presented in figure 2.12).

Exercise 2.5: Show that the robot mentioned by Lee (figure 2.17) is a fully parallel manipulator, as defined by Earl.

Exercise 2.6: Show that the Stewart platform has an Earl parallelism index of 1 although it is not a fully parallel robot.

Exercise 2.7: Show that the *Hexa* robot is fully parallel, as understood by Gosselin.

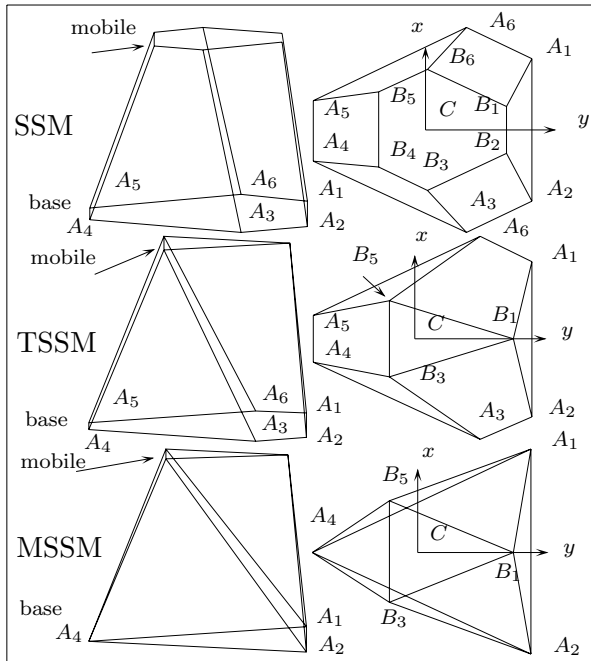


Figure 2.103. The most studied manipulators in this book: SSM, TSSM, MSSM, in perspective, and seen from above.

Problem 2.1: Is there a parallel robot synthesis method allowing one to obtain automatically all possible architectures with a specified number of degrees of freedom?

Problem 2.2: Being given the manufacturing tolerances of a parallel robot with less than 6 d.o.f., and its workspace, determine the maximal amplitude of the parasitic motion (i.e. the motion along unwanted d.o.f.) over the workspace of the robot

Problem 2.3: For a parallel robot with less than 6 d.o.f. and a given workspace, determine the maximal value of the manufacturing tolerances such that the maximal amplitudes of the parasitic motion (i.e. the motions along unwanted d.o.f.) over the workspace of the robot are lower than given thresholds

Problem 2.4: Establish a list of n d.o.f. robots with decoupled rotation and translation

Problem 2.5: Establish a list of n d.o.f. completely decoupled robots

Inverse kinematics

This chapter will examine the relations between the actuated joint coordinates of a parallel robot and the end-effector pose. The relation giving the actuated joint coordinates for a given pose of the end-effector is called the *inverse kinematics*, and we will show that usually this relation is simple for parallel robots

3.1. Inverse kinematics

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 (we have already seen the Euler angles that we will use throughout this book, but other angles may be used as, see the "Workspace" chapter). 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.

3.1.1. GENERAL METHODS

3.1.1.1 Analytic method

If we consider each of the chains linking the base to the moving platform, A will represent the end of the chain that is linked to the base, and B the end of the chain that is linked to the moving platform. By construction the coordinates of A are known in a fixed reference frame, while the coordinates of 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.

If \mathbf{X} represents the generalized coordinates of the moving platform we have

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OB} = \mathbf{H}_1(\mathbf{X}) . \quad (3.1)$$

This gives the positions of the extreme points of all the chains for which we want to calculate the joint coordinates (which often simply means the actuated joint coordinate). We therefore need to calculate the direct kinematics of each single chain which, in general, involves only the moving platform generalized coordinates and the given chain joint coordinates, but not those of the other chains: the solution can therefore be done in parallel for each of those chains. Such a parallel solution will be possible only if chains do not share an actuated joint variable, as in the robot of Zoppi (669), figure 2.31.

The chain joint coordinates Θ allow us to determine the vector \mathbf{AB} , with, if needed, the help of \mathbf{X} :

$$\mathbf{AB} = \mathbf{H}_2(\mathbf{X}, \Theta) \quad (3.2)$$

The joint coordinate calculation can thus be done by solving the following system of equations:

$$\mathbf{H}_1(\mathbf{X}) = \mathbf{H}_2(\mathbf{X}, \Theta) \quad (3.3)$$

If we have p chains connecting the base and the end-effector, the number of unknowns in (3.3) will be $3p$ ($2p$ for planar robots). Alternatively, let us assume that we have N joints, n of which are actuated, which implies that we have also n unknowns in \mathbf{X} . When the actuators are locked (i.e. n unknowns in (3.3) have a fixed value) there remain N unknowns. As in that case the mobility of the end-effector should be 0, there should be N equations in (3.3).

In the most general case (for example for a 6- R serial chain), this solution could be complex. However, the chains used for parallel robots are in general very simple and the solution does not cause any problems (only a very strong impact on some performances may justify having a chain with complicated direct kinematics, thus leading to a complex inverse kinematics for the robot). The interest of this approach is that it can be fully automated: the equations may be derived from a kinematic description of the chains and there are solution methods that are sufficiently general to deal with most of the systems.

Note also that solving equation (3.3) will enable us to determine not only the joint coordinates, but also the coordinates of the passive joints.

3.1.1.2 Geometrical method

A more geometrical approach to the inverse kinematics problem is to consider that the extremities A, B of each leg have a known position in 3D

space. Then we may cut the leg at a point M and get two different mechanisms M_A, M_B constituted of the chain between A, M and the chain between B, M . 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 we assume that the mechanisms 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 to determine the 3 coordinates of the points that are common to V_A, V_B we need 3 independent equations that are obtained from the $3 - d_A, 3 - d_B$ equations describing V_A, V_B . Hence we must have $3 - d_A + 3 - d_B = 3$, or $d_A + d_B = 3$.

Although we end up with a system solution problem, as in the previous section this approach has some advantages:

- as the cutting point is free we have some freedom in the final system of equations, and this may help the solution
- the varieties we consider describe geometrical objects whose intersection may already have been studied in geometry
- the intersection of algebraic varieties is a well studied topic, and various methods allow us to determine bounds on the number of intersection points without actually computing the intersection points (for example, Bezout bounds).

Compared to the previous approach, the drawback of such an approach is that the complexity of the system is heavily dependent upon the choice of the cutting point; an automated treatment is difficult.

3.1.2. EXAMPLES

The following inverse kinematics calculations for a few robot architectures having different types of chains will help explain the basic principles.

3.1.2.1 Planar manipulators

Figure 3.1 shows a planar manipulator with 3 degrees of freedom of the 3-RRR type. Being given the position of C and the rotation angle Φ , we may calculate the rotation matrix R of the platform and the position of the points B_i . We have

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + \mathbf{RCB}_r = \mathbf{H}_1(\mathbf{X}) , \quad (3.4)$$

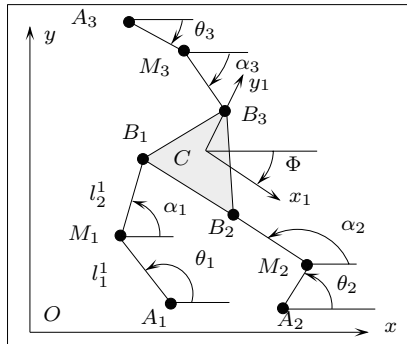


Figure 3.1. The 3-RRR type planar robot

where \mathbf{CB}_r is the known coordinate vector of B_i in the moving frame. We then write that B_i belongs to the serial chain $A_iM_iB_i$ with joint variables θ_i, α_i :

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OM}_i(\theta_i) + \mathbf{M}_i\mathbf{B}_i(\alpha_i) = \mathbf{H}_2(\mathbf{X}, \Theta) \quad (3.5)$$

The inverse kinematics is obtained by solving the system $\mathbf{H}_1(\mathbf{X}) = \mathbf{H}_2(\mathbf{X}, \Theta)$ i.e. 2 equations in the unknowns θ_i, α_i .

In the geometrical approach, we choose M_i as a cutting point. The mechanism A_iM_i imposes the constraint that M_i lies on a circle centered at A_i with radius l_1^i , while for mechanism B_iM_i , it must be on the circle centered at B_i with radius l_2^i . Hence the point M_i lies at the intersection of two circles: there are therefore 2 solutions (or none). As this is valid for each chain we get a total of at most $2^3 = 8$ solutions for the inverse kinematics.

3.1.2.2 3-UPU manipulator

We consider a 3-UPU robot in its translational version (figure 2.10). The inverse kinematics problem is to determine the length ρ of AB , being given the generalized coordinates of the robot \mathbf{X} which are the coordinates of C . We may write

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + \mathbf{CB} = \mathbf{H}_1(\mathbf{X}) \quad (3.6)$$

For this robot the orientation is assumed to be constant, so that the vector \mathbf{CB} is identical to the vector \mathbf{CB}_r which express the coordinates of the B point in the moving frame. Hence the components of \mathbf{AB} are completely determined.

Now we will consider B as the extreme point of a RRP chain. Let Υ be the 2-vector constituted of the two rotation angles around the R joint, and let ρ be the leg length. The unit vector \mathbf{n} defining the direction of the

leg may be calculated as a function of Υ and we have

$$\mathbf{AB} = \rho \mathbf{n} = \mathbf{H}_2(\Upsilon, \rho) \tag{3.7}$$

Combining equations (3.6, 3.7), we get a system of 3 equations in the 3 unknowns ρ, Υ . To solve this system we may notice that the norm of \mathbf{AB} in equation (3.7) is ρ^2 , while we may calculate the norm value directly from equation (3.6). Hence ρ^2 can be calculated directly, and it is then easy to determine Υ from the remaining equations.

Let us now investigate the geometric approach. We will choose B as the cutting point. Point B has to lie on a sphere centered at A with radius ρ and, seen from the platform, B is a fixed point. Hence to obtain the intersection of these 2 varieties we will just state that the known distance between A, B should be equal to ρ : this equation will give the joint coordinates ρ directly.

3.1.2.3 6-UPS manipulator

For this type of manipulator (figure 3.2) the problem is to determine the length ρ of AB , being given the generalized coordinates of the robot \mathbf{X} , viz. the vector \mathbf{OC} and a set of parameters that define the orientation of the platform. Being given the orientation parameters, we can calculate the rotation matrix R between the moving frame and the reference frame. We may write

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + R\mathbf{CB}_r = \mathbf{H}_1(\mathbf{X}) , \tag{3.8}$$

where the vectors $\mathbf{AO}, \mathbf{CB}_r$ are known respectively in the base reference frame and in the moving platform frame. Hence the components of \mathbf{AB} are completely determined. As for the 3-UPU robot, B can be considered

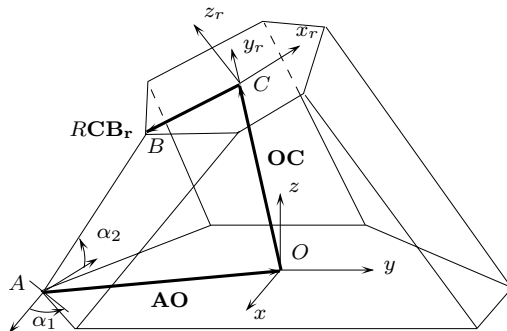


Figure 3.2. Fundamental vectors for establishing the inverse kinematics of a 6-UPS robot.

as the extreme point of an RRP chain, and the unit vector \mathbf{n} defining the

direction of the leg may be calculated as a function of $\Upsilon = (\alpha_1, \alpha_2)$:

$$\mathbf{AB} = \rho \mathbf{n} = \mathbf{H}_2(\Upsilon, \rho) \quad (3.9)$$

Combining equations (3.8, 3.9), we get a system of 3 equations in the 3 unknowns ρ, α_1, α_2 . As for the 3-*UPU*, we notice that the norm of \mathbf{AB} in equation (3.9) is ρ^2 , and we may calculate the norm directly from equation (3.8). Hence the joint coordinates ρ_i are therefore determined by

$$\rho_i = \|\mathbf{A}_i \mathbf{B}_i\| = \|\mathbf{A}_i \mathbf{O} + \mathbf{OC} + \mathbf{RCB}_{i_r}\| . \quad (3.10)$$

The geometric approach is identical to that used for the 3-*UPU*.

Note

From equation (3.10), the squares of the lengths of the links may be written as

$$\rho^2 = \|\mathbf{AO}\|^2 + \|\mathbf{CB}_r\|^2 + 2(\mathbf{AO} + \mathbf{RCB}_r) \cdot \mathbf{OC} + 2\mathbf{AO} \cdot \mathbf{RCB}_r + \|\mathbf{OC}\|^2 . \quad (3.11)$$

It is notable that this expression contains terms linear in the C coordinates and a quadratic part in those same terms ($\|\mathbf{OC}\|^2$). This last term is the same whatever the considered link. If we calculate the difference between two squares of link lengths, this quadratic term will then disappear, and only linear terms will remain. As a consequence, if we calculate this difference for three pairs of links, we will obtain a linear system of three equations in the three coordinates of C . We can then solve this system and obtain the coordinates of C from the rotation matrix and the link lengths.

3.1.2.4 6-PUS manipulator

For this type of manipulator (figure 3.3) the problem is to determine the length λ of AA_0 , being given the generalized coordinates of the robot \mathbf{X} , i.e. the coordinates of C and a set of parameters that define the orientation of the platform. Equation (3.8) giving \mathbf{H}_1 remains valid. Being given the 2 rotation angles of the U joint, $\Upsilon = (\alpha_1, \alpha_2)$ and the length l of the link A_0B , we can calculate the vector $\mathbf{A}_0\mathbf{B}$ as $\mathbf{A}_0\mathbf{B} = l\mathbf{n}$ where \mathbf{n} is the unit vector defining the axis of the link. We suppose that the prismatic joint axis is defined by the unit vector \mathbf{u} , and that the position of the linear actuator at rest is A . Vector \mathbf{AB} may be written as:

$$\mathbf{AB} = \mathbf{AA}_0 + \mathbf{A}_0\mathbf{B} = \lambda\mathbf{u} + l\mathbf{n} = \mathbf{H}_2(\lambda, \Upsilon) \quad (3.12)$$

Equating (3.8) and (3.12) leads to a system of 3 equations in the 3 unknowns $\lambda, \alpha_1, \alpha_2$.

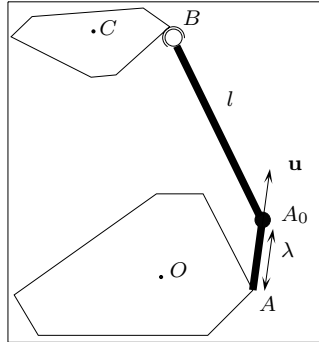


Figure 3.3. Parameters for the inverse kinematics of the 6-PUS manipulators.

To calculate the joint variable λ only we can compute the numerical value of the square of the norm of \mathbf{AB} based on equation (3.8). We can also calculate the norm of \mathbf{AB} based on equation (3.12) with

$$\|\mathbf{AB}\|^2 = \lambda^2 + 2\lambda l \mathbf{u} \cdot \mathbf{n} + l^2, \tag{3.13}$$

We must therefore solve the quadratic equation (3.13) in order to calculate the displacement corresponding to a certain configuration; this will usually lead to two solutions. We will choose one of these two solutions, generally the one compatible with the maximum actuator strokes.

In the geometrical approach we will choose A_0 as cutting point. The mechanism BA_0 imposes that A_0 lie on a sphere centered at B with radius l , while the mechanism AA_0 constrains A_0 to lie on the line going through A with unit vector \mathbf{u} . Intersecting a sphere and a line leads to two possible points. Hence the inverse kinematic admits generally $2^6 = 64$ solutions. Let us also note that the note 3.11 remains valid.

3.1.2.5 6-RUS manipulator

For this type of manipulator the joint center A_0 rotates around a unit vector axis \mathbf{u} , and is located on a circle of radius r with center A (figure 3.4). The inverse kinematic problem is to determine the angle β between the vector \mathbf{AA}_0 and an arbitrary unit vector \mathbf{X}_1 that is perpendicular to \mathbf{u} .

Being given the 2 rotation angles of the U joint, $\mathbf{Y} = (\alpha_1, \alpha_2)$ and the length ρ of the link A_0B , we can calculate the vector $\mathbf{A}_0\mathbf{B}$ as $\mathbf{A}_0\mathbf{B} = \rho\mathbf{n}$, where \mathbf{n} is the unit vector defining the axis of the link. Let us define $\mathbf{Y}_1 = \mathbf{u} \times \mathbf{X}_1$. We can then write

$$\mathbf{AA}_0 = r(\cos \beta \mathbf{X}_1 + \sin \beta \mathbf{Y}_1). \tag{3.14}$$

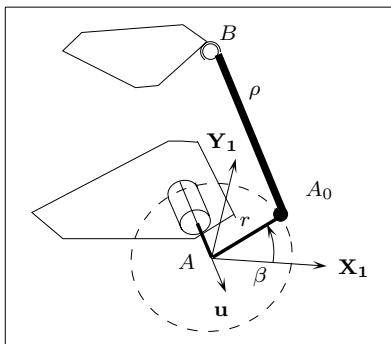


Figure 3.4. The parameters for the inverse kinematics of \underline{RUS} manipulators

and we get

$$\mathbf{AB} = \mathbf{AA}_0 + \mathbf{A}_0\mathbf{B} = r(\cos \beta \mathbf{X}_1 + \sin \beta \mathbf{Y}_1) + \rho \mathbf{n} = \mathbf{H}_2(\beta, \Upsilon) . \quad (3.15)$$

Combining equations (3.8) and (3.15) leads to a system of 3 equations in the 3 unknowns $\beta, \alpha_1, \alpha_2$. To get only the actuated joint β we use the norm of $\mathbf{A}_0\mathbf{B}$ that can be calculated by

$$\|\mathbf{A}_0\mathbf{B}\|^2 = \rho^2 = r^2 + 2\mathbf{AB} \cdot \mathbf{A}_0\mathbf{A} + \|\mathbf{AB}\|^2 . \quad (3.16)$$

where \mathbf{AB} is known through equation (3.8). Defining $\gamma = \rho^2 - r^2 - \|\mathbf{AB}\|^2$ and using the classical Weierstrass substitution, this equation may be written in the form:

$$x^2(\gamma - 2r\mathbf{X}_1 \cdot \mathbf{AB}) + (4r\mathbf{Y}_1 \cdot \mathbf{AB})x + 2r\mathbf{X}_1 \cdot \mathbf{AB} + \gamma = 0 . \quad (3.17)$$

which is a quadratic equation in the variable $x = \tan \frac{\beta}{2}$ that will usually lead to 2 solutions.

For the geometrical method we choose as cutting point A_0 . The mechanism BA_0 imposes that A_0 lies on a sphere centered at B with radius ρ . For the mechanism AA_0 , the point A_0 is constrained to lie on a circle centered at A with radius r lying in the plane perpendicular to \mathbf{u} . The intersection of these 2 varieties is 2 points.

3.1.2.6 General conclusion

As may be seen from the examples, solving the inverse kinematics problem for parallel robots is usually simple, as long as the geometry of the legs is simple enough, and allows for an analytic determination. However it may happen that for complex legs such an analytic formulation cannot be obtained (see exercise 3.11).

3.1.3. EXTREMA OF THE JOINT COORDINATES

While designing a parallel robot, we may have to determine what maximum and minimum values of the joint coordinates will be when the end-effector has to lie within a given workspace. We will summarize known results for 6-UPS robots for a given translation workspace, i.e. for which the orientation of the end-effector is constant (this result may be extended to other manipulators as well):

- the workspace is a *box*: when the point C moves within its box, the joint B moves within a box S of the same size obtained by translating the box described by C by the vector \mathbf{CB} , which is constant since the orientation is fixed. We define A_p^i as the projection of A on the plane of face i of the box S and we consider only such points that belong to face i of the box S . Let d^f be the smallest distance between the points A_p^i and A , and let d be the smallest distance between the point A and the corners of S . Then:
 - the maximum length of a link is attained when the point B occupies a vertex of the box S
 - the minimum length of a link is the smallest value between d and d^f , unless A is included in the workspace, in which case the distance is zero.
- the workspace is a *sphere*: when the point C moves within its sphere, the joint B moves within a sphere S of the same radius, obtained by translating the sphere described by C by the vector \mathbf{CB} , which is constant since the orientation is fixed. The line going through A and the center of the sphere S intersects S at 2 points S_1, S_2 and we define d_i as the distance between A and S_i . The maximum leg length is the maximum of d_1, d_2 and the minimal distance is either the minimum of d_1, d_2 , or zero if A is included in S

Similar results may be obtained for other geometrical shapes of translation workspace. For the most general workspace, we have to solve a constrained optimization problem for which interval analysis methods are appropriate (see the interval appendix).

3.2. Exercises

Exercise 3.1: Show that the inverse kinematics of the RRP planar robots has two solutions.

Exercise 3.2: Show that the inverse kinematics of RPP planar robots has two solutions.

Exercise 3.3: Show that the inverse kinematics of PPR planar robots has one solution.

Exercise 3.4: Show that the inverse kinematics of \underline{PRP} planar robots has one solution.

Exercise 3.5: Determine the angle of rotation of the revolute joints close to the base of a 3- UPU translational robot, being given the location of the platform.

Exercise 3.6: Show that the inverse kinematics of a 6 degrees of freedom, fully parallel robot with identical chains has a maximum of 64 solutions.

Exercise 3.7: Determine what is wrong in the following text (which has been found in a journal paper!): *the inverse problem is to determine the length ρ_i of the actuated joint i.e. the distance between the points A_i, B_i . Being given the pose of the platform, the coordinates of B_i are known, while the coordinates of A_i are known geometrical parameters. Hence $\rho_i^2 = \|\mathbf{A}_i \mathbf{B}_i\|^2 = U$, from which we deduce that the inverse problem has two solutions $\rho_i = \pm\sqrt{U}$. However the negative length cannot be obtained without a reassembling of the mechanism.*

Exercise 3.8: Establish the values of the Euler angles of the universal joint of the links of a 6- UPS robot, given the coordinates of its points A, B in the reference frame.

Exercise 3.9: Establish the quadratic relation characterizing the solutions of the inverse kinematics for spherical robots.

Exercise 3.10: Consider Reinholtz's articulated robot, which consists of two MSSM resting on top of each other, with the lower MSSM moving platform being the base of the upper MSSM. The platform consists of three links of various lengths, while the MSSM links are of fixed lengths (figure 2.57). Determine the number of solutions of its inverse kinematics.

Exercise 3.11: Han's 6 degrees of freedom robot is described in figure 2.41. Find the maximum number of solutions for the inverse kinematics of this robot. You may need to use some results for planar mechanisms presented in Chapter 4, section 4.1.

Problem 3.1: Determine the maximal number of solutions, and the solutions, of the inverse kinematics of the manipulator proposed by Zoppi (669), described in figure 2.31.

Direct kinematics

This chapter will address the problem of determining the pose of the end-effector of a parallel robot 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.

Determining the pose of the end-effector from measurements of the minimal set of joint coordinates that are necessary for control purposes, is equivalent to solving the system of inverse kinematics equations. We will show that, in general, the solution for this problem is not unique i.e. there are several ways of assembling a parallel manipulator with given actuated joint coordinates, and that generally we cannot express in an analytical manner the generalized coordinates as functions of the actuated joint coordinates. We will present methods for finding all the solutions for this problem. We will see that their computation times, although decreasing, are still too large for use 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. We will then present numerical methods using a-priori information on the current pose, that are more compatible with a real-time context, and emphasize that their convergence and robustness is an important issue. Finally we will investigate the influence of additional sensory information on the direct kinematics.

4.1. Planar robots

In this section we will consider 3 d.o.f. planar manipulators, as presented in the "Architecture" chapter. The inverse kinematics equations form a system of three non-linear equations. We will first show that this system may possess several solutions, i.e. that there are several poses of the end-effector for given values of the joint coordinates. It is therefore possible to assemble the manipulator in different ways, and these different configurations will be called the *assembly modes* of the manipulator. In order to obtain the assembly modes, we will manipulate the equations to reduce the problem to finding the roots of a polynomial in one variable, a univariate polynomial. When using this algebraic elimination approach for solving the direct kinematics it is first necessary to determine beforehand an upper bound for the

number of assembly modes that will then be used to guide the elimination to obtain a polynomial with the lowest possible degree. For finding this upper bound we will first recall a few basic notions of mechanism theory which we illustrate on a specific mechanism, the 4-bar mechanism.

4.1.1. THE 4-BAR MECHANISM

The 4-bar mechanism is described on figure 4.1: it uses only revolute joints, fixed length links, and it has one degree of freedom.

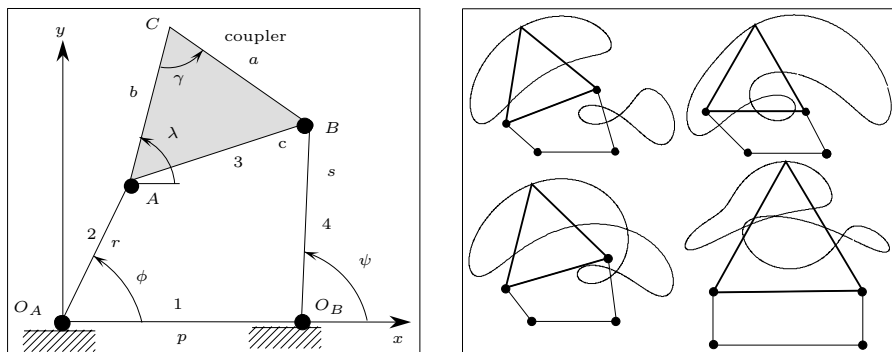


Figure 4.1. The 4-bar mechanism and some possible coupler curves.

This mechanism is made of 4 articulated bars 1, 2, 3, 4. A body, the *coupler*, is rigidly linked to bar number 3. Its geometry is defined by the lengths a, b and the angle γ . The mechanism is defined by the lengths p, r, s of the bars 1, 2, 4 and by two angles: the angle ϕ between the bars 1 and 2, and the angle ψ between 1 and 4. If we change one of these angles, each point of the coupler, C for instance, will describe a curve, called the *coupler curve*. This curve is algebraic, as shown by Freudenstein (169).

4.1.2. COUPLER CURVE AND CIRCULARITY

The 4-bar mechanism has been extensively studied (248). We assume that a motor rotates the segment number 2, and thus modifies the angle ϕ , and we focus on the coupler curve described by point C , with coordinates (X, Y) ; the curve is a sextic¹ \mathcal{S} , some examples of which are illustrated in figure 4.1. One interesting property of this sextic is that it is *tricircular*, i.e. it possesses three double points on the imaginary circle, a concept we will now describe. Let us recall some useful notions relating to the intersection

¹A software allowing the visualization of coupler curves is available by anonymous ftp, in the directory `coprin/4bar`

of algebraic curves, see for example (248). According to Bezout's theorem, an algebraic curve of order n generally meets an algebraic curve of order m in nm points. This result, which is quite well known, may seem like a paradox if we apply it to circles: it would then lead to the existence of 4 intersection points. This paradox can be explained simply in the following manner:

Let us consider a circle of radius r , with its center at coordinates (a, b) :

$$(x - a)^2 + (y - b)^2 - r^2 = 0 ;$$

that is written in projective space by introducing a new unknown w :

$$\left(\frac{x}{w} - a\right)^2 + \left(\frac{y}{w} - b\right)^2 - r^2 = 0 .$$

The unknown w may be considered as a scale factor. The preceding equation is said to be *homogeneous* since it can be written as:

$$(x - aw)^2 + (y - bw)^2 - r^2w^2 = 0 .$$

where all the terms are now of degree 2 with respect to the unknowns x, y, w . The coordinate system x, y, w is called a *homogeneous planar coordinate system*. In this coordinate system (x, y, w) and $(\lambda x, \lambda y, \lambda w)$ represent the same point. It is easy to verify that the two points with coordinates $(1, \pm i, 0)$ lie on the circle: these two imaginary points are called the *imaginary circular points* $\mathcal{S}_1, \mathcal{S}_2$. These points satisfy $x^2 + y^2 = 0$ which defines the *imaginary circle*. Since the imaginary circular points do not depend on the terms a, b, r , they belong to all the circles. The intersection of any two circles will therefore always include the two imaginary circular points and, as a result, two circles cannot intersect in more than two real points. If a planar curve includes the points $\mathcal{S}_1, \mathcal{S}_2$ as double points, triple points, ... we will say that this curve has a *circularity* of 2, 3, ...

The 4-bar sextic \mathcal{S} is tricircular, which simply means that its circularity is 3. Indeed, (see (248)) the sextic equation expressed in homogeneous coordinates for $w = 0$ is

$$(x^2 + y^2)^3(a^2 + b^2 - 2ab \cos \gamma) = 0 . \quad (4.1)$$

Except when one side of the quadrilateral $O_A A B O_B$ has zero length, so that it reduces to a triangle, equation 4.1 is equivalent to $(x^2 + y^2)^3 = 0$. The sextic thus contains the points $\mathcal{S}_1, \mathcal{S}_2$ as triple points. Let us also note that the circularity of the curve is maximal, since a sextic cannot have a circularity greater than 3.

Note that the notion of circularity may be extended to the spatial case, the equation $x^2 + y^2 + z^2 = 0$ being the *absolute conic*.

4.1.3. DIRECT KINEMATICS OF THE 3- $R\underline{P}R$ ROBOT

We consider the 3- $R\underline{P}R$ robot described in figure 4.2. In order to estimate

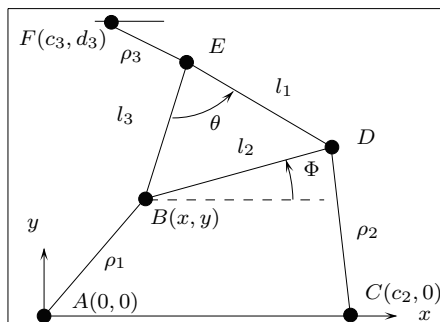


Figure 4.2. The 3- $R\underline{P}R$ robot

an upper bound for the number of assembly modes of this robot, we will examine a sub-mechanism obtained from this robot by detaching one of the joints of the moving platform from its link.

4.1.3.1 *Assembly modes*

If we consider the mechanism defined in figure 4.2 and disconnect the link EF , the remainder of the mechanism becomes an articulated 4-bar mechanism.

The number of intersection points of the coupler curve with the circle with center F and radius $\|\mathbf{EF}\|$ then gives us the number of possible location of E , and hence the number of assembly modes. According to Bezout's theorem, the number of intersection points of a sextic with a circle is at most 12. However, given the tricircularity of the sextic, the points $\mathcal{S}_1, \mathcal{S}_2$ will count in Bezout's number as 6 imaginary intersection points. We can therefore state that there are at most 6 possible assembly modes for this type of parallel robot.

4.1.3.2 *Polynomial direct kinematics*

Among the methods that may be used to solve the inverse kinematics equations (see the solving appendix) we will use an elimination approach that will allow us to reduce the initial problem of solving a system of 3 equations to solving a univariate polynomial equation.

Various methods have been suggested, such as that of Kassner (304), Pennock (469); we will follow C. Gosselin (192) with the notation of figure 4.2. The origin of the reference frame is chosen as the joint center A , and its axis x is defined by the line that joins A to the joint center C . The

axis y is then perpendicular to x . The pose of the moving platform is defined by the position of the joint center B associated with the point A and having coordinates (x, y) . The orientation of the moving platform is determined by the angle Φ between the axis x and the edge BD of the moving platform. The moving platform itself is made of the 3 points B, D, E , and its geometry is uniquely defined by the length of its 3 edges (l_1, l_2, l_3) and by one of the vertex angles, e.g. the angle θ between the sides EB and BD . The 3 link lengths are denoted ρ_1, ρ_2, ρ_3 . Based on these conventions, the coordinates of the three joints that are linked to the reference frame are:

$$A : (0, 0) \quad , \quad C : (c_2, 0) \quad , \quad F : (c_3, d_3) .$$

Under these conditions the inverse kinematics may be written as

$$\rho_1^2 = x^2 + y^2 , \quad (4.2)$$

$$\rho_2^2 = (x + l_2 \cos \Phi - c_2)^2 + (y + l_2 \sin \Phi)^2 , \quad (4.3)$$

$$\rho_3^2 = (x + l_3 \cos(\Phi + \theta) - c_3)^2 + (y + l_3 \sin(\Phi + \theta) - d_3)^2 , \quad (4.4)$$

and can also be written in an algebraic form as

$$\rho_1^2 = x^2 + y^2 , \quad (4.5)$$

$$\rho_2^2 = x^2 + y^2 + Rx + Sy + Q , \quad (4.6)$$

$$\rho_3^2 = x^2 + y^2 + Ux + Vy + W . \quad (4.7)$$

The equation system (4.5-4.7) may be written in a simpler manner:

$$\rho_1^2 = x^2 + y^2 , \quad (4.8)$$

$$\rho_2^2 - \rho_1^2 = Rx + Sy + Q , \quad (4.9)$$

$$\rho_3^2 - \rho_1^2 = Ux + Vy + W . \quad (4.10)$$

The equations (4.9-4.10) are linear in x, y , and this system, with determinant $\Delta = RV - SU$, has the solution

$$x = -(SA_1 - VA_2)/(RV - SU) , \quad y = (RA_1 - UA_2)/(RV - SU) ,$$

where

$$A_1 = \rho_3^2 - \rho_1^2 - W \quad A_2 = \rho_2^2 - \rho_1^2 - Q$$

provided $\Delta \neq 0$. This result can then be transferred into equation (4.8) in order to obtain the equation:

$$(SA_1 - VA_2)^2 + (RA_1 - UA_2)^2 - \rho_1^2(RV - SU)^2 = 0 , \quad (4.11)$$

which only depends on the variable Φ . The Weierstrass substitution is then used:

$$T = \tan\left(\frac{\Phi}{2}\right) \quad , \quad \cos(\Phi) = \frac{1 - T^2}{1 + T^2} \quad , \quad \sin(\Phi) = \frac{2T}{1 + T^2} \quad .$$

This substitution converts (4.11) to a sextic in T :

$$C_0 + C_1T + C_2T^2 + C_3T^3 + C_4T^4 + C_5T^5 + C_6T^6 = 0 \quad (4.12)$$

where the coefficients C_i depend only on the geometry of the manipulator². Each real solution of this equation gives a value of Φ , which in turn gives a pair x, y . However, a sextic may have 0, 2, 4, or 6 real solutions. This means that we do not know whether there exists even one assembly mode for a set of joint coordinates. Since there is no closed form for the solution of a sextic we must find the solution numerically; it is then easy to find an example of a manipulator (figure 4.3) that has 6 assembly modes, i.e. the sextic has 6 real roots. Figure 4.3 also shows the coupler curve; it intersects the circle described by the end of the dissociated link at 6 points. Note a

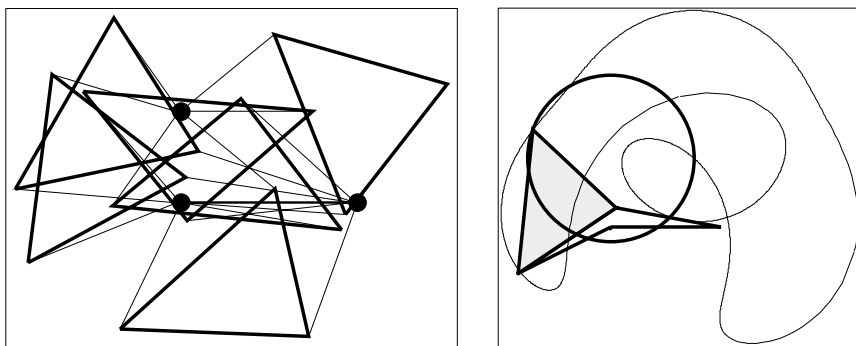


Figure 4.3. On the left, the 6 assembly modes of a planar parallel robot with . The dimensions are $OA_1(0,0)$, $OA_2(15.91, 0)$, $OA_3(0,10)$, the link lengths: 14.98, 15.38, 12 and the sides lengths: B_1B_2 : 17.04, B_1B_3 : 20.84, B_2B_3 : 16.54. On the right is its coupler curve. This robot has 6 assembly modes: the circle described by the end of the dissociated link (in thick line) intersects the coupler curve at 6 different points.

surprising point: it may be thought that by adding a fourth leg there will be only one solution (we have now 4 equations for only 3 unknowns). This is not the case, as shown by Husty in a private communication during the conference ARK 2000: even with 4 legs there still may be up to 6 solutions to the direct kinematics problem.

²A software package designed for solving the direct kinematics of a 3-RPR robot is available by anonymous ftp, in the directory `coprin/FK/3-RPR`.

4.1.3.3 Particular cases

We consider the particular case of the planar parallel robot for which the joints linked to the mobile as well as those linked to the fixed reference, are collinear, as shown in figure 4.4.

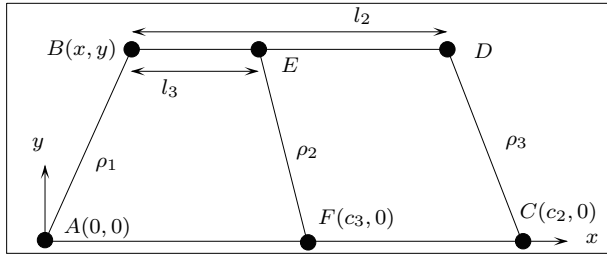


Figure 4.4. A special case of planar parallel robots: it enables the control of the position and the orientation of the bar BED when the lengths ρ_i are modified.

In this case, the same procedure that has been used in the general inverse kinematics equations leads to a cubic equation in $T = \cos \Phi$:

$$f_0(T) = a_3 T^3 + a_2 T^2 + a_1 T + a_0 = 0 . \quad (4.13)$$

There are thus at most three values of T , and hence at most three pairs $\pm\Phi$, six values in all. Here the solutions of this equation can be determined analytically. In fact, it is quite easy to show, for example by using Sturm's method, that this polynomial cannot have more than two real solutions in the interval $[-1, 1]$, and consequently that the manipulator has at most 4 assembly modes; see exercise 4.1. Other geometries, in which the degree of the polynomial is lower than 6, have been studied in (194).

4.1.4. OTHER PLANAR ROBOTS

A complete study of the direct kinematics of planar robots was presented in (404). This study showed, from the direct kinematics viewpoint, that all possible chains are equivalent to the four generic chains presented in figure 4.5 and denoted type 1, 2, 3, 4 (the latter can appear only once among the robot's three chains). The equivalences between the chains and the generic chains are given in table 4.1.

If we have a solution method for all manipulators with these generic chains, we will be able to analyze the direct kinematics for all planar robots³. For these generic chain robots, it is relatively easy to find the

³This is a phenomenon that occurs frequently when studying parallel robots: very often the direct kinematic analysis for two robots with different mechanical architecture will be the same

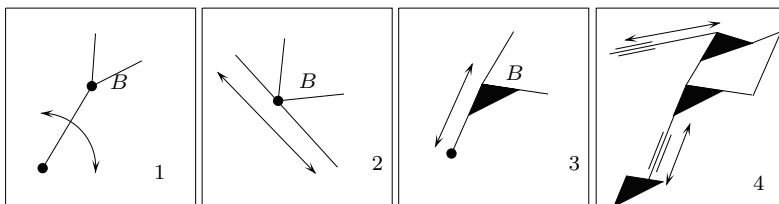


Figure 4.5. The 4 generic chains to which all the possible chains of planar parallel robots are equivalent.

\underline{RRR}	\underline{RRR}	\underline{RRR}	\underline{RPR}	\underline{RPR}	\underline{RPR}	\underline{RPP}	\underline{RPP}	\underline{PRR}
1	1	1	2	1	3	3	3	1
\underline{PRR}	\underline{PRR}	\underline{PRP}	\underline{PRP}	\underline{PPR}	\underline{PPR}	\underline{RRP}	\underline{RRP}	\underline{RRP}
2	2	3	2	2	2	3	3	1

TABLE 4.1. Equivalences between chains.

maximal number of assembly modes, and to find a polynomial form for the direct kinematics, the degree of which corresponds to that number. The results are presented in table 4.2⁴.

chain	1-1-1	2-2-2	3-3-3	1-1-2	1-1-3	2-2-1	2-2-3	3-3-1
solutions	6	2	2	6	6	4	4	4
chain	3-3-2	1-2-3	1-1-4	1-2-4	1-3-4	2-2-4	3-3-4	2-3-4
solutions	4	6	2	2	2	1	1	1

TABLE 4.2. Maximum numbers of solutions for generic chain robots.

4.2. Robots with 3 translational d.o.f.

For parallel robots in translation it is in general easy to solve the direct kinematics. For example for the *Delta* robot (figure 2.6), calculation for the direct kinematics can be done explicitly. Indeed let us denote by M_i the lever end points: these points have a fixed location when the actuators are locked. The structure of the legs allows the point B_i to move on a sphere centered at M_i . As the orientation of the platform is constant when

⁴The introduction of type 4 is due to C. Wampler, who kindly provided us with the results for this chain

B_i moves on a sphere, then C , the center of the platform, also moves on a sphere S_i with same radius, and a center that is obtained by translating M_i by the constant vector $\mathbf{B}_i\mathbf{C}$. For a solution of the direct kinematics, point C should lie on the 3 spheres S_i and is thus obtained as the intersection of these spheres. This leads to two solutions which are symmetrical with respect to the plane defined by the three points corresponding to the lever ends (it is often possible to eliminate one or two solutions for geometric reasons, or because of mechanical limits on the joints). Note that the same method may be applied to solve the direct kinematics of the 3-*UPU*.

4.3. Robots with 6 d.o.f.

The direct kinematics of 6 d.o.f. robots will usually be the most difficult to solve, even though certain mechanisms admit an explicit solution. In this section we will show how a geometrical analysis may lead to an upper bound for the number of real solutions, and we will also present methods that may be used to determine all the solutions.

We will also show that the direct kinematics of large classes of mechanisms can be solved by the study of a few generic mechanisms, here called *equivalent mechanisms*, for which an upper bound of the number of assembly modes can be found and for which the direct kinematics can be reduced to the solution of a univariate polynomial equation.

4.3.1. EXAMPLE OF ANALYSIS: THE TSSM

4.3.1.1 Upper bound on the number of assembly modes

For a TSSM with fixed length links, we consider the triangles formed by the joint B_i and the corresponding joints A_1^i, A_2^i on the base. For these triangular faces, the only possible motion of B_i is a rotation around the line $A_1^i A_2^i$. Hence each point B_i must lie on a circle with center on this line.

Hence we may construct a 3-*RS* mechanism that is equivalent, from the direct kinematics viewpoint, to the TSSM, and made of three links (called the *equivalent links*) that all rotate around a revolute joint, and which all have one end connected to the moving platform by an *S* joint (figure 4.6). Hunt (250) showed that the maximal number of assembly modes for a TSSM was 16. His method was the following: starting from the equivalent mechanism of the TSSM, he disconnected one of the links from the moving platform and considered the remaining part of the mechanism; this was a four-bar spatial mechanism, called a *RSSR*, made of 4 links linked by two revolute joints and by two ball-and-socket joints (figure 4.6). The point B of this mechanism, with coordinates (X, Y, Z) , lies on a surface. In order to establish the degree of this surface, we use Cayley's theorem (248) which

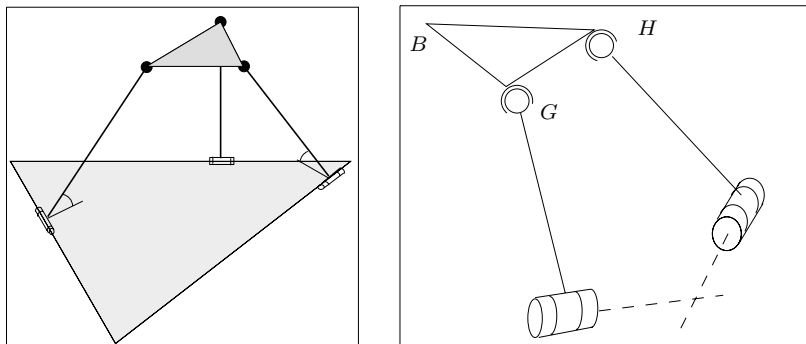


Figure 4.6. On the left, the equivalent mechanism of the TSSM. On the right, the *RSSR* mechanism obtained after cutting one of the links of the equivalent mechanism.

states that a line for which two fixed points C, D are constrained to move on two algebraic curves of order n_c, n_d and of circularity p_c, p_d , will generate a ruled surface of order $2n_c(n_d - p_d) + 2n_d(n_c - p_c)$ if the algebraic curves do *not* lie on parallel planes, and of order $2n_c(n_d - p_d) + 2n_d(n_c - p_c) - 2p_cp_d$ if they do lie on parallel planes.

For an *RSSR*, the hypotheses of Cayley's theorem are satisfied for the line going through the points G, H , which lie on circles. Hence we have $n_c = n_d = 2, p_c = p_d = 1$, which leads to an 8th order surface (if the circles were lying on parallel planes, the order would be 6). If we take into account the rotation of the platform around GH , we find that B lies on a surface of degree 16.

If we now consider the number of intersection points of this surface with the circle described by the end of the equivalent link we have disconnected, we obtain the number of possible assembly modes for the mechanism; the corresponding results when the circles lie within parallel planes are shown in brackets. A surface of order 16 (12) meets a circle in a maximum of 32 (24) points. However, among these points, some will be situated on the imaginary sphere: they must therefore be subtracted from the number of 32 (24). Thus, everything depends on the circularity of a surface of order 16 (12). Hunt suggested that this circularity might be 8 (6), and this was later proved. It is hence possible to assert that 16 (12) points are imaginary, and that there are at most 16 (12) assembly modes.

4.3.1.2 Polynomial formulation

We will reduce the solution of the direct kinematics to the solution of one univariate polynomial equation. It is worth noting that this study was started by Nanua and Waldron (442)*, who determined a 24th order poly-

nomial for the MSSM. We have established in the previous section that there are a maximum of 16 assembly modes for the TSSM. Our goal is therefore to obtain an equation with degree equal to this number.

Numerous researchers have worked on this problem. A 16th order polynomial was found as early as 1988 by Charentus and Renaud (87) from LAAS; using their method we were able numerically to find several configurations with 16 solutions (399). Another solution method, which uses the notion of a spherical mechanism, was also suggested by Griffis and Duffy (206). Finally, Nanua also mentioned a correct solution (442)*.

The method developed by Innocenti (263) will be used to establish this polynomial. We will establish the *RSSR* polynomial for general positions of the revolute joint axes, notwithstanding the fact that for the TSSM these axes are coplanar, using the notation defined in figure 4.7. The mechanism

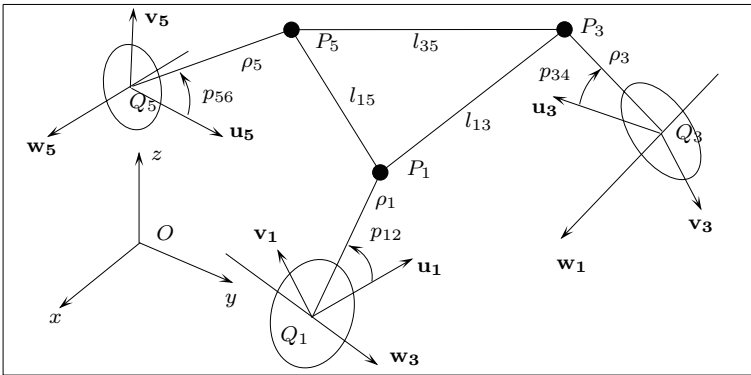


Figure 4.7. Notations used for the polynomial formulation of a *RSSR*

is defined by three links Q_1P_1, Q_3P_3, Q_5P_5 with lengths ρ_1, ρ_3, ρ_5 , that are able to rotate around the points Q_1, Q_3, Q_5 . The angles p_{12}, p_{34}, p_{56} define the rotations. The reference frame is designed so that O lies in the plane $Q_1Q_3Q_5$. The lengths of the edges of the moving triangle are l_{13}, l_{15}, l_{35} .

The initial system of equations is defined by the 3 following equations:

$$\|\mathbf{P}_1\mathbf{P}_3\| - l_{13} = 0, \quad \|\mathbf{P}_1\mathbf{P}_5\| - l_{15} = 0, \quad \|\mathbf{P}_3\mathbf{P}_5\| - l_{35} = 0. \quad (4.14)$$

All these equations can be expressed in terms of the 3 unknowns p_{12}, p_{34}, p_{56} and they may be written as

$$\begin{aligned} q_1 \cos(p_{12}) \cos(p_{34}) + q_2 \cos(p_{12}) \sin(p_{34}) + q_3 \sin(p_{12}) \cos(p_{34}) + q_4 \sin(p_{12}) \sin(p_{34}) + q_5 \cos(p_{12}) \\ + q_6 \sin(p_{12}) + q_7 \cos(p_{34}) + q_8 \sin(p_{34}) + q_9 = 0, \quad (4.15) \\ r_1 \cos(p_{12}) \cos(p_{56}) + r_2 \cos(p_{12}) \sin(p_{56}) + r_3 \sin(p_{12}) \cos(p_{56}) + r_4 \sin(p_{12}) \sin(p_{56}) + r_5 \cos(p_{12}) \end{aligned}$$

$$+r_4 \sin(p_{12}) \sin(p_{56}) + r_6 \sin(p_{12}) + r_7 \cos(p_{56}) + r_8 \sin(p_{56}) + r_9 = 0, \quad (4.16)$$

$$s_1 \cos(p_{34}) \cos(p_{56}) + s_2 \cos(p_{34}) \sin(p_{56}) + s_3 \sin(p_{34}) \cos(p_{56}) + s_5 \cos(p_{34}) \\ + s_4 \sin(p_{34}) \sin(p_{56}) + s_6 \sin(p_{34}) + s_7 \cos(p_{56}) + s_8 \sin(p_{56}) + s_9 = 0. \quad (4.17)$$

where the coefficients q_i, r_i, s_i do not depend on the unknowns (p_{12}, p_{34}, p_{56}) . We introduce the variables t_{12}, t_{34}, t_{56} :

$$t_{12} = \tan\left(\frac{p_{12}}{2}\right), \quad t_{34} = \tan\left(\frac{p_{34}}{2}\right), \quad t_{56} = \tan\left(\frac{p_{56}}{2}\right).$$

Equations (4.16), (4.17) are written as

$$At_{56}^2 + Bt_{56} + C = 0, \quad (4.18)$$

$$Rt_{56}^2 + St_{56} + T = 0, \quad (4.19)$$

in which the coefficients A, B, C are second degree polynomials in t_{12} , and the coefficients R, S, T are second degree polynomials in t_{34} :

$$A = A_2 t_{12}^2 + A_1 t_{12} + A_0, \quad B = B_2 t_{12}^2 + B_1 t_{12} + B_0, \quad (4.20)$$

$$C = C_2 t_{12}^2 + C_1 t_{12} + C_0, \quad R = R_2 t_{34}^2 + R_1 t_{34} + R_0, \quad (4.21)$$

$$S = S_2 t_{34}^2 + S_1 t_{34} + S_0, \quad T = T_2 t_{34}^2 + T_1 t_{34} + T_0. \quad (4.22)$$

We eliminate the unknown t_{56} by calculating the resultant d of the equations (4.18,4.19) (see the solving appendix):

$$d = \begin{vmatrix} 0 & A & B & C \\ A & B & C & 0 \\ 0 & R & S & T \\ R & S & T & 0 \end{vmatrix} = 0.$$

Expanding d we get

$$Gt_{34}^4 + Mt_{34}^3 + Nt_{34}^2 + Ut_{34} + V = 0, \quad (4.23)$$

where G, M, N, U, V are 4th order polynomials in t_{12} . Equation (4.15) also is an equation in t_{34} which may be written as

$$Dt_{34}^2 + Et_{34} + F = 0, \quad (4.24)$$

where D, E, F are 2nd order polynomials in t_{12} . Equations (4.23), (4.24) therefore are polynomial equations in t_{34} . This unknown is eliminated by writing the resultant d_1 of equations (4.23,4.24):

$$d_1 = \begin{vmatrix} 0 & G & M & N & U & V \\ G & M & N & U & V & 0 \\ 0 & 0 & 0 & D & E & F \\ 0 & 0 & D & E & F & 0 \\ 0 & D & E & F & 0 & 0 \\ D & E & F & 0 & 0 & 0 \end{vmatrix} = 0.$$

d_1 then is a polynomial⁵ in t_{12} , of order 16. In the particular case of the TSSM d_1 only contains even powers of t_{12} : it is a polynomial of degree 8 in t_{12}^2 ; this means that if the basic equations admit the triplet solution (p_{12}, p_{34}, p_{56}) , they then also admit the opposite triplet $(-p_{12}, -p_{34}, -p_{56})$ as a solution; each solution configuration has a corresponding *mirror* configuration with respect to the base.

We can see that the fact that the direct kinematics may have up to 16 solutions is confirmed by the degree of the resulting polynomial. However, at this stage we do not know whether there is a manipulator with 16 assembly modes for a given set of joint coordinates. The polynomial may only be solved numerically: we present an example in the next section.

4.3.1.3 Example of TSSM with 16 assembly modes

Consider a manipulator which joint coordinates given in table 4.3.

link	x_a	y_a	z_a	x_b	y_b	z_b
1	-9.7	9.1	0.0	0.0	7.3	0.0
2	9.7	9.1	0.0	0.0	7.3	0.0
3	12.76	3.9	0.0	4.822	-5.480722	0.0
4	3.0	-13.0	0.0	4.822	-5.480722	0.0
5	-3.0	-13.0	0.0	-4.822	-5.480722	0.0
6	-12.76	3.9	0.0	-4.822	-5.480722	0.0

TABLE 4.3. Position of the A, B points for the TSSM with 16 assembly modes.

Choosing an arbitrary set of joint variables will usually lead to 0 solution for the direct kinematics. Hence to ensure that there is at least one solution we choose a nominal configuration from which we calculate the link lengths. The chosen nominal configuration is $x_c = y_c = 0, z_c = 20, \psi = -10^\circ, \theta = -5^\circ, \phi = 10^\circ$. Numerical solution for this set of link lengths shows that the robot admits 16 assembly modes (and the solutions may be certified with interval analysis, see the interval appendix). Table 4.4 presents the pose parameters of the 8 configurations with the moving platform situated above the base; the other 8 configurations are mirror images of these. The first 8 configurations are shown in figure 4.8. Some authors have stated

⁵An implementation of this algorithm is available by anonymous `ftp`, in the directory `coprin/FK/6p-3`.

x_c	y_c	z_c	ψ	θ	ϕ
0.1099	-6.8071	15.1572	178.790092	104.247298	-179.3975
0.0	0.0	20.0	170.0000	4.999992	-170.0000
2.8029	-4.6660	12.7406	55.389531	89.178208	136.1996
1.3617	4.9038	17.3824	-106.331771	149.931849	58.9676
0.1606	5.3765	17.1868	-170.380852	164.013963	7.9545
-0.3524	-3.8663	11.9183	-12.559631	45.110726	-168.3013
-1.4134	4.8262	17.4299	102.640488	147.384474	-61.9768
-2.3355	-4.4679	12.5478	-50.849043	79.039617	-137.3532

TABLE 4.4. 8 assembly modes with the moving platform over the base.

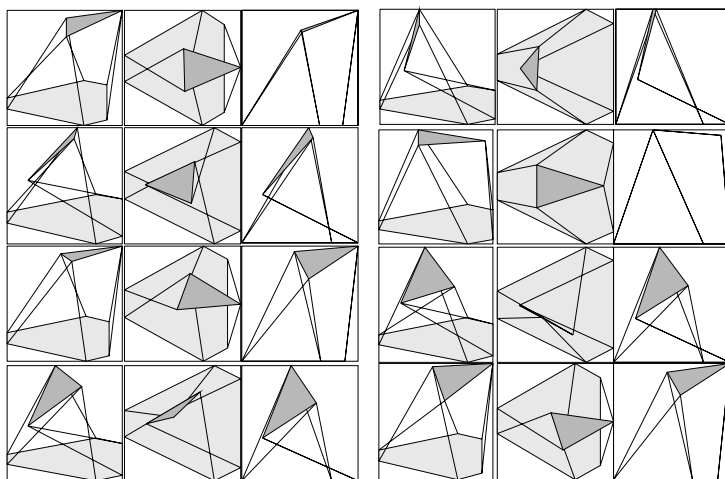


Figure 4.8. 8 assembly modes with the moving platform over the base (for each solution we present a perspective view, a top view and a side view).

that solving the direct kinematics with the polynomial formulation may lead to numerical errors: we have never experienced such a problem, but we agree that the computation must be implemented carefully.

It is interesting to know how the solutions are distributed within the workspace. A systematic study allowed us to establish this distribution, as shown in the table 4.5. This result has been established by solving the direct kinematics for 297381 points within the workspace defined by ± 8 cm

for the coordinates x, y , 19 and 21 cm for the coordinate z , $\pm 15^\circ$ for the rotational angles, using a step size of 1 cm for the coordinates x, y, z and 5° for the angles. The configurations with the minimum, 2, or maximum, 16, numbers of assembly modes constitute only a small of the whole (2.5%); 92.38% of the whole have 4-12 configurations.

solutions	2	4	6	8	10	12	14	16
number	2060	77446	31309	134443	11764	31524	3255	5580
in %	0.69	26.04	10.53	45.21	3.96	10.60	1.09	1.88

TABLE 4.5. Distribution of the number of solutions of the direct kinematics of the TSSM over its workspace

This analysis of the TSSM allows us to note the astonishing duality between parallel and serial robots: the inverse kinematics of a serial 6R robot also leads to a 16th order polynomial. Murthy (436) actually studied a particular 6R robot and showed that a parallel equivalent could be found, and that it is a TSSM. Bruyninckx (62), Collins (106), Duffy (144), Waldron (600)* and Zamanov (646)* all tried to explain this duality. The duality is complete for the velocity and static aspects, but it remains difficult to establish it for the kinematics.

4.3.2. ANALYSIS OF OTHER SPACE MECHANISMS

The TSSM has shown us how the notion of equivalent mechanism allows us to treat the problem of the direct kinematics, by reducing the system of non-linear equations to a univariate polynomial. Although the equivalent mechanism of the TSSM has a particular characteristic (the revolute joint axes are coplanar) the method used to obtain the polynomial did not depend on this characteristic.

The notion of *equivalent mechanism*⁶ allows us to obtain a polynomial formulation of the direct kinematics for many important spatial mechanisms suggested in current literature on the subject. We will see, however, that this notion is not always applicable.

⁶A generic implementation for the resolution of the direct kinematics of robots with equivalent mechanisms of the 3-RS type is available by anonymous ftp, in the directory `coprin/FK/3RS`

4.3.2.1 3 degrees of freedom wrist

The notion of equivalent mechanism may be used for a large number of mechanisms and not only for a 6 d.o.f. robot. We consider in this section a wrist with a 3 degrees of rotary freedom (figure 4.9). The locations of the joints A_i are fixed, and hence the corresponding joints B_i on the moving platform can describe a circle, with center C_i situated on the line going through the ball-and-socket joint C and the point A_i . Using the joint coordinates and the manipulator geometry, we may determine the centers C_i and radii r_i of the circles.

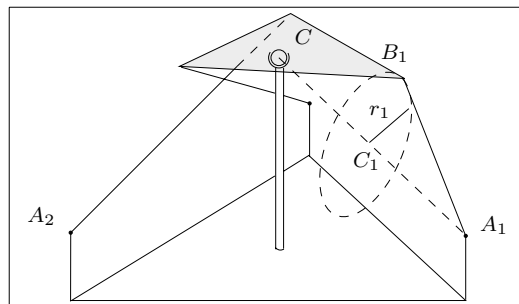


Figure 4.9. The wrist with three degrees of rotary freedom. When the heights of the points A_i are fixed, the points B_i move on circles with centers C_i and radii r_i . The center of the circle is situated on the lines joining A_i to the ball-and-socket C .

The equivalent mechanism is therefore a 3-*RS* with revolute joint axes having a common point. An upper bound of the number of possible assembly modes is 16, and we can find a 16th order polynomial for the direct kinematics. However, it is possible to go even further: we notice that at the level of the 3-*RS* mechanism, for a set of solution angles, the angles corresponding to the symmetrical pose with respect to the ball-and-socket center will also provide a solution. This solution is unacceptable because it does not satisfy the linking constraint with the ball-and-socket joint. Consequently, there can be no more than 8 solutions. This shows that there is a limit to the use of the notion of equivalent mechanism; it may provide a polynomial with a degree which is too high. In fact, in this particular case, as we will see in section 4.3.3.2, the direct application of a result of Innocenti (264) shows that we can obtain an 8th order polynomial directly, as Innocenti (268) has noted, and it is not difficult to find examples with 8 certified real solutions. In our test, we compute the number of solutions, taking as input the joint variables calculated for a regular sampling of the possible end-effector poses and we found out that we get 8 solutions for about 2% of the cases.

4.3.2.2 MSSM

The MSSM is a special case of the TSSM, but we will describe here another solution approach proposed by Dedieu (131). He uses the Lagrange identity that relates the squares of the distances of p points M_i in space. If $L_{ij} = d(M_i, M_j)^2$ represents the square of the distance between the points M_i and M_j , we have:

$$\begin{vmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & L_{12} & \dots & L_{1p} \\ 1 & L_{21} & 0 & \dots & L_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & L_{p1} & L_{p2} & \dots & 0 \end{vmatrix} = 0$$

Dedieu considers five of the vertices of the MSSM and uses this identity to build 6 polynomial equations. A clever manipulation of these polynomials shows that there is a maximum of 16 solutions and that, in all cases, there are either 0 or 2 convex solutions, the other being concave; this is a direct exemplification of one of Cauchy theorems (75).

4.3.2.3 6-PUS robot and Stewart platform

The use of the equivalent mechanism can be illustrated on the 6-PS active wrist (figure 2.34). The vertex on the moving platform of the triangle, made of two links sharing the same double S joint, can rotate only around an axis going through the U joints. We can therefore replace the two links by a single link connected to the base by a revolute joint. We then obtain an equivalent mechanism similar to that obtained for the TSSM, although the revolute joints axes are now in a general position. Using the result of the previous section, we may state that there are at most 16 assembly modes, and that it is possible to find a 16 degree polynomial for the direct kinematics. This calculation was implemented, and we found configurations in which 16 certified assembly modes were indeed possible. A numerical study over the workspace of our prototype has shown that the direct kinematics will have 8, 10 or 12 solutions in 85% of the cases.

The notion of equivalent mechanism may also be applied to the Stewart platform. When the linear actuators have fixed lengths, the only possible motion for the joint center is a rotation around a vertical axis. Given the lengths of the actuators, it is easy to find the position of the centers of the circles on which the joints lie, as well as their radii. An equivalent mechanism is therefore a 3-RS with revolute joint axes vertical, i.e. the circles lie within parallel planes. We apply the particular case of Cayley's theorem: an upper bound of the number of possible assembly modes is 12.

Lazard (347) showed that this result can be refined. He shows that the solutions of the direct kinematics may be obtained directly from the

solutions of the direct kinematics of two planar 3-*RPR* parallel robot⁷ i.e. by solving numerically a polynomial of degree 6. He also exhibits a constructive proof for getting configurations with 12 real solutions.

4.3.2.4 Manipulators *PPP-3S, PRR-3S, PPR-3S*

The analysis of the *RRR-3S* mechanism by Parenti and Innocenti (263) was used later for other mechanisms. Parenti (459) thus suggests an analysis of the mechanism *PPP-3S* (figure 4.10). For this mechanism, three param-

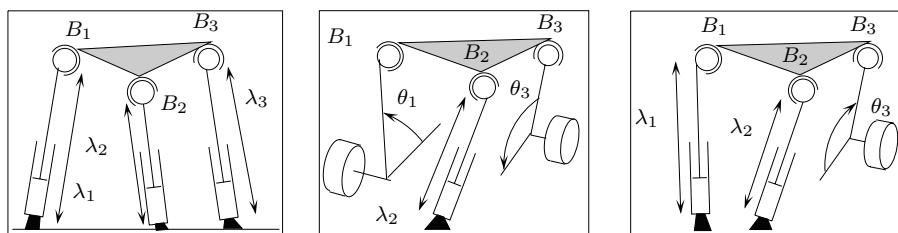


Figure 4.10. The robots *PPP-3S, PRR-3S, PPR-3S*.

eters, $\lambda_1, \lambda_2, \lambda_3$ define the *P* joint displacements. The distance between two joints B_i, B_j may be written as a quadratic form in the two parameters λ_i, λ_j . We therefore have three equations describing the direct kinematics. Considering the resultant of these equations, Parenti obtains an 8th order polynomial. This mechanism therefore has 8 possible assembly modes, and Parenti presents an example with 4 assembly modes. Note that this result is coherent with the upper bound of the number of solution that can be found by using Cayley's theorem. Indeed according to this theorem point, B_1 , which belongs to the coupler of the *PSSP* mechanism $A_2B_2B_3A_3$, lies on a 8th order surface. The number of intersection points of this curve and the line originating from A_1 is at most 8.

Parenti (460) then continues by considering the *PRR-3S* and *PPR-3S* mechanisms (figure 4.10). For these mechanisms, calculation of the distance between the joint centers on the platform brings about chains of the *PSSR, RSSR, PSSP, RSSP* type, for which Parenti establishes the distance as a function of the parameters of the actuated joints. Manipulations on these distance equations then gives a 16th order univariate polynomial for the *PRR-3S*, and a 12th order one for the *PPR-3S*. However, the examples suggested by Parenti for these two mechanisms have only four assembly modes. Both results are consistent with the geometrical approach: they correspond to the maximum number of intersection points of the line

⁷A software for Stewart platform direct kinematics is available by anonymous ftp, in the directory `coprin/FK/Stewart`

originating from A_2 and the surface of the coupler curve of degree 12 for the $PSSR$ and of degree 16 for the $RSSR$.

4.3.3. SPECIAL CASES OF THE 6- \underline{UPS} ROBOT

The notion of TSSM equivalent mechanism is unfortunately unable to deal with the case of the general 6- \underline{UPS} robot. Clearly this is one of the most interesting cases, as this type of manipulator is the one the most encountered in practice, and because the 6- \underline{UPS} robot will be the equivalent mechanism for a large number of other mechanical architectures (for example for the *Hexa*, figure 2.38, or the *Hexaglide*, figure 2.35).

We will first start by generalizing the TSSM, known also as the 3-3 robot by reference to the number of attachment points on the platform and on the base, by presenting results for $m-n$ robots where m is the number of attachment points on the base and n on the platform. These results, together with the systematic analysis of Faugère and Lazard (161) are summarized in tables 4.6,4.7.

4.3.3.1 6-5 manipulators

For this robot, presented in figure 4.11, the elimination result proposed by Faugère is a 40th order polynomial. If both the platform and the base are planar, Yin presented an example with 10 assembly modes (637).

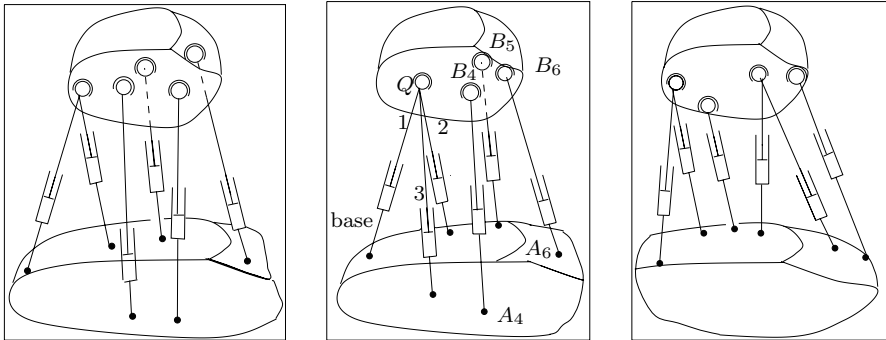


Figure 4.11. On the left, Yin's 6-5 parallel robot: the direct kinematics is established as a 40th order univariate polynomial. In the middle, Innocenti's 6-4 parallel robot: the direct kinematics is reduced to an 8th order polynomial, although with a maximum of 16 assembly modes. On the right, Hunt's 6-4 parallel robot: according to Faugère and Innocenti, this robot has a maximum of 32 assembly modes.

4.3.3.2 6-4 manipulators

The 6-4 manipulator presented in figure 4.11 was studied by Innocenti (264). He first shows that the position of Q may be determined with the help of

the lengths of the links 1, 2, 3, and that there are two solutions. These three links therefore control the location of Q , while the other three are used to control the orientation around this point, and the robot is partially decoupled. Innocenti then establishes that for a given location of Q the orientation of the platform may be obtained by solving an 8th order univariate polynomial; Husain (254) presents an example with 16 real solutions. This analysis allows us also to solve the direct kinematics of spherical wrist (figure 4.9) and of spherical robots, see for example (195)*.

Hunt (251) considered the 6-4 robot described in figure 4.11 and claimed a maximum of 24 assembly modes, while Yin apparently established a 24th order polynomial for the general case (637). But Faugère (161) shows that the correct result was 32; this was confirmed by Innocenti (273).

4.3.3.3 6-3 manipulators

Nanua (442)*, Hunt (251) and Geng (182) established that the 6-3 robot shown in figure 4.12, a particular case of Innocenti's 6-4 robot, still has a maximum of 8 solutions.

4.3.3.4 5-5 manipulators

Innocenti (270) managed to obtain a 40th order polynomial for this mechanism (figure 4.12) and later on suggested, in a personal communication, an example with 24 real solutions. For one of the particular cases of the 5-5 robot, shown in figure 4.12, Hunt (251) has shown that there are at most 24 assembly modes.

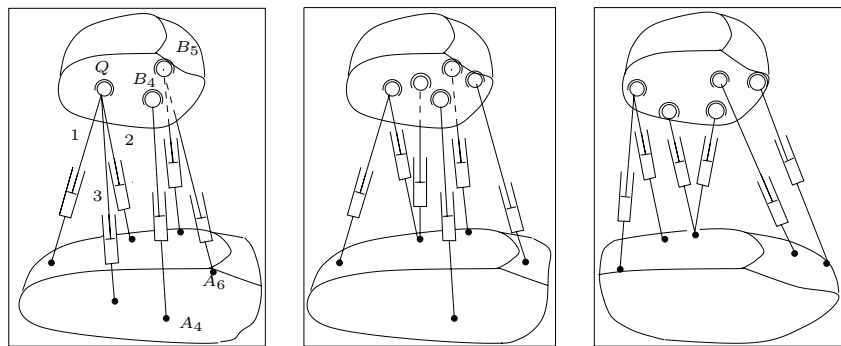


Figure 4.12. On the left, the 6-3 parallel robot with at most 8 assembly modes. In the middle, Innocenti's 5-5 robot: the direct kinematics can be written as a 40 degree polynomial. On the right, Hunt 5-5 robot; this robot has at most 24 assembly modes.

4.3.3.5 5-4 manipulators

The joints may be grouped in a way that leads to different types of 5-4 parallel robots. The direct kinematics of the robot in figure 4.13 was

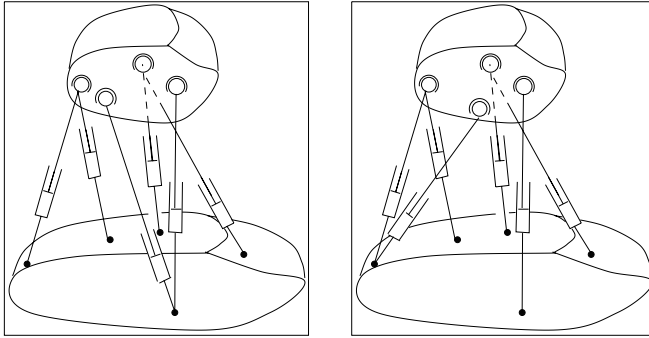


Figure 4.13. On the left, Innocenti's 5-4 parallel robot: the direct kinematics is established as a 16 degree polynomial. On the right, Innocenti and Yin's 5-4 parallel robot: the direct kinematics is established as a 24 degree polynomial.

studied by Innocenti (265), who shows that we can refer to the study done for the TSSM: we then obtain a 16 degree polynomial, and Innocenti gives an example with 8 solutions. As for the 5-4 robot described in figure 4.13, Innocenti (269) has established a 24 degree polynomial, and presented an example with 8 real solutions. Lin (367) presents a complete analysis for 5-4 robots with base and platform planar. All the various cases are illustrated in figure 4.14.

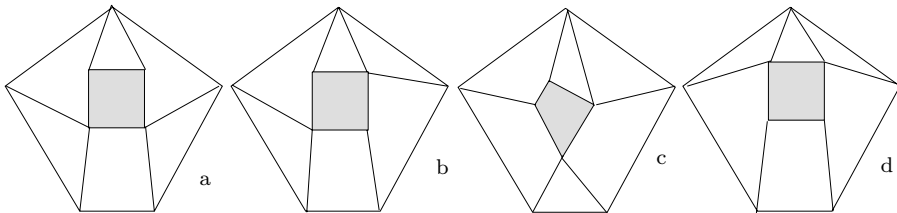


Figure 4.14. The 5-4 parallel robots studied by Lin, seen from above. The upper bounds of the number of solutions are: a) 16, b) 24, c) 16, d) 32.

The a) type is equivalent to that of Innocenti, and Lin also finds a 16th order polynomial. The b) and d) types lead to the same type of analysis, and Lin obtains a 32th order polynomial. Without being too strict in his demonstration, Lin suggests a 24th order polynomial for the b) type which is confirmed by Faugère in the general case. For the d) type, the polynomial has degree 32 while Innocenti (272) analyzes the general case: the polynomial still has degree 32 and Innocenti provides an example with 8 solutions. For the c) type, Lin (366) does an analysis that is similar to that of the

4-4 robot, which has a maximum of 16 solutions, while Faugère suggests a maximum of 24 solutions in the general case.

4.3.3.6 4-4 manipulators

The 4-4 manipulator has four joints on the base and four on the moving platform. Lin (366) studied this case thoroughly. With 4 joints, the faces of the manipulator are of the quadrilateral type (Q) or triangular type (T), and there will be two faces of the Q type and four of the T type. Lin distinguishes several cases, according to the order in the faces: type I for the $TTQTTQ$ order, type II when the order is $TTTQTQ$, type III when the order is $TTTTQQ$ (figure 4.15). Lin shows that type I is equivalent to a

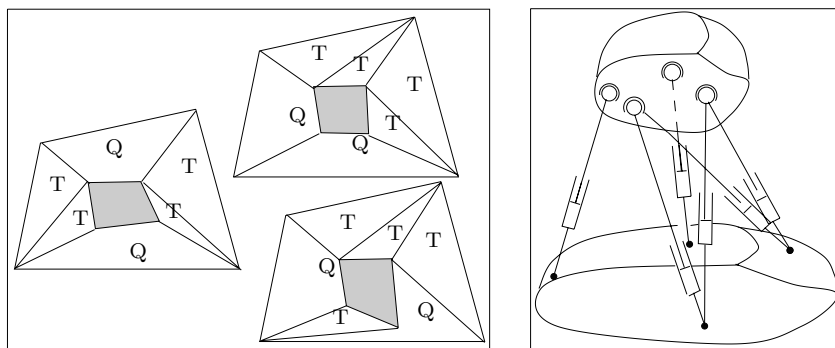


Figure 4.15. The different types of 4-4 robots studied by Lin, seen from above. On the right, the 4-4 robot studied by Innocenti with up to 16 assembly modes.

MSSM with a 16 degree direct kinematics. For the type II, a slightly more complex analysis gives a polynomial formulation of degree 4 but with 16 as the maximum number of solutions, as each of the polynomial solution leads to four solutions overall. As for the type III, Lin obtains a 12th order polynomial leading to a maximum of 24 solutions; he presents an example with 12 real solutions. Lin studies only the case when the joints are coplanar, however Hunt (251) showed that the maximum number of assembly modes remained the same even if this constraint were lifted.

Another type of 4-4 robot was studied by Innocenti (267) (figure 4.15). Presenting an 18th order polynomial form, he shows that this manipulator may have up to 16 solutions; he gives an example with 16 real solutions. For the 4-4 robot in which 3 links have a common joint on the base, we have already determined that with 3 common joints on the platform the maximum number of solution was 8, and this result holds if 3 links have also a common joint on the base (61).

4.3.3.7 Manipulators with 5 aligned points

Zhang (656) considers the 6-*UPS* manipulators in which 5 of the joints, either on the base, or on the moving platform, are aligned (figure 4.16). In this particular case Zhang shows that it is possible to find an explicit

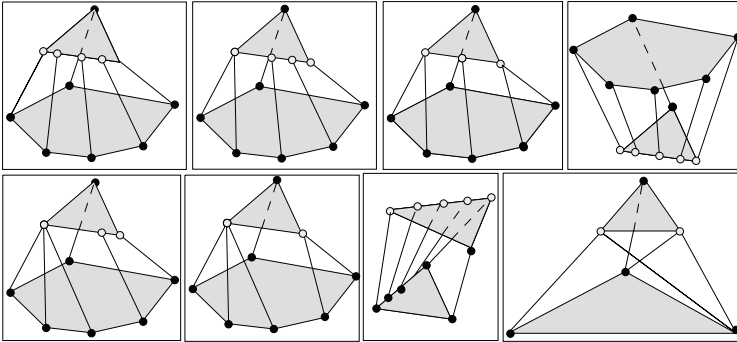


Figure 4.16. 6 degrees of freedom spatial mechanism as suggested by Zhang. Here 5 joints are aligned either on the base, or on the moving platform. A symbolic formulation for the direct kinematics solutions may be found.

formulation for the direct kinematics solutions. We can indeed manipulate the direct kinematics equations to transform the problem into the resolution of a 4th order polynomial, thus obtaining an analytical form for all the solutions; generally there are 16.

4.3.4. THE SSM

The methods presented in the previous sections do not apply to the 6-*UPS* robot with 6 degrees of freedom, nor do they apply to the SSM (planar base and moving platform).

In that case Lazard (345) very smartly established that the maximum number of solutions was 40. He shows that the problem is the same as solving a system of 9 equations (3 linear, 6 quadratic) in 9 unknowns. Given the degree of the equations, Bezout's theorem shows that there can be no more than 64 solutions. Using one of Bezout's theorem and the concept of Gröbner basis, he demonstrates that there are 24 solutions at infinity, and thus at most $64-24=40$ real solutions. However Rouillier, in a private communication, mentions a proof that the number of real solutions cannot exceed 36, as was confirmed by Husty (259). As for the calculation of these solutions, we refer to the next section. Note that special cases of SSM have been considered in the literature. For example Mavroidis (390) examines the case where the base and platform are identical, and shows that there will be at most 24 solutions to the direct kinematics. As similar result has

been obtained by Yang (635) for congruent platforms (the platforms are similar up to a scale factor).

4.3.5. GENERAL CASE OF THE 6- \underline{UPS} ROBOT

4.3.5.1 *Maximum number of assembly modes*

Determining the maximum number of assembly modes and the solution poses for 6- \underline{UPS} parallel robots has been one of the greatest challenges in the field of mechanism theory for the last few years. The exact result is that the maximum number of solutions, whether complex or real, is 40 and was first established by Ronga (511), via a very complex proof, and by Raghavan (492) using an *homotopy* method (see the solving appendix).

This result was later demonstrated by Lazard (346), using Gröbner bases; by Mourrain (431), and by Wampler (602), using a parametrization based on dual quaternions and various mathematical tools. We should also mention the approach of Sarkissyan (519), who attempted to design a robot with the largest possible number of solutions; he used Burmester theory for the coupler curves of 5- SS mechanisms. Another interesting point is that adding a seventh leg may not decrease the number of solutions. For example Husty (260) shows that for symmetrical robots having planar base and platform it is possible to find A_7, B_7 points lying on the base and platform planes, such that the length A_7B_7 will remain the same for the solutions of the direct kinematics obtained for the initial 6-legged robot. Those points are obtained when the A_7 anchor points coordinates satisfy a cubic constraint equation, a unique B_7 point being associated to each A_7 point satisfying the constraint. This feature is related to permanent singular robots, a topic that will be addressed in section 6.8.

4.3.5.2 *Determination of the solutions*

The challenge of finding all the solutions of the direct kinematics problem has attracted a lot of attention. The methods that can be used to solve this problem are intuitively presented in the solving appendix, and may be summarized as follows:

- *elimination*: this method requires algebraic equations for the inverse kinematics, may be fast, but has two main drawbacks: it is not easy to get a final polynomial that has the minimal degree (40 in our case), and the calculation of its coefficients is very sensitive to numerical errors (hence this method does not provide certified solutions)
- *homotopy*: in practice this method requires algebraic equations for the inverse kinematics, may provide certified solutions, but is usually slow

- *Gröbner basis*: this method requires algebraic equations for the inverse kinematics, and provides certified solutions if the coefficients of the inverse kinematics equations are rational numbers
- *interval analysis*: this method may be used even if the inverse kinematics equations are not algebraic, and provides certified solutions. The unknowns must be bounded (this is usually the case for kinematics problems) and the computation time is difficult to estimate
- *ad-hoc methods*: the direct kinematics problem is transformed into another simpler problem that is solved with one of the above method or by an optimization procedure. For example Parenti and Innocenti (266) use the solution for a 5-5 robot mentioned in section 4.3.3.4 to reduce the problem to finding the zeroes of a univariate function. These methods may be fast, but are devoted to a specific architecture, and cannot easily be extended to deal with other problems.

After numerous unsuccessful attempts throughout the world for establishing an univariate polynomial formulation, of degree 40, this remarkable result was finally obtained in 1994 by M. Husty (256). Like Wampler (602), who actually also developed a similar method, Husty uses a method based on dual quaternions. After numerous manipulations of the equations, he successfully proceeds to an elimination that leads to a degree 40 polynomial. Unfortunately, these manipulations requires some intuition, and it is not known if the method can be automated.

Currently the fastest exact solution methods are the Gröbner package developed by Rouillier and Faugère (514), and an interval analysis based method (417). They are able to calculate all the solutions in a computation time that ranges from a few seconds to a minute.

4.3.5.3 Example with 40 real solutions

The determination of an example with 40 real solutions has caused quite a few problems, which is understandable for various reasons. Consider a rigid body that occupies a certain number of different poses, and wonder how many points on this rigid body may be situated, in all the poses, on fixed spheres, one for each point⁸. Roth (513) shows that for 7 different poses, there can be a maximum of 20 such points. For our present problem, this means that for 7 poses of the platform, we can find on it up to 20 points remaining on spheres for all poses. We need only 6 of them though, since in our problem only the B points remain on spheres. However, Roth demonstrates further on in his paper that in general, for 8 different poses of the rigid body, there are no more points of the rigid body that are on spheres for the 8 poses. . . In our case, though, we are looking for 6 points

⁸This is in fact one of the topics of the *Prix Vaillant* suggested by the Académie des Sciences, and for which Borel (52) and Bricard (57) offered partial solutions

that are on spheres for 40 poses. We may have therefore assumed that this case, if it does exist, is extremely special. The first proof of the existence of a robot with 40 real assembly modes was given by Dietmaier (132). His proof is constructive as he produces examples with 40 solutions. Basically the principle is to start with a robot geometry for which all the solutions are computed (for example by using Husty's algorithm). Some of these solutions are real, while the others are complex. Dietmaier then cleverly perturbs the locations of the joints so that the real solutions remain real while the imaginary part of one of the complex solutions decreases; the solutions of the perturbed system are obtained by using an iterative method. After a few iterations this imaginary part vanishes to give a double real solution; after a few more iterations, this double solution splits into two separate real solutions. The process is repeated for each complex solution until all the solutions are real.

4.3.6. SUMMARY OF RESULTS

Tables 4.6, 4.7 summarize the results known for the direct kinematics of different manipulators. The notation used for the designation of the architecture is the following: the first two figures indicate respectively the number of joints on the base and on the moving platform, then a pictogram shows the type of connection, with an exponent indicating the number of times the pattern is reproduced. The letter (P) indicates coplanar points, while the letter (Q) shows that some points are on a quadrilateral.

4.4. Systematic method for *UPS* robots

Although fast solution methods are available, it may be interesting to investigate *ad-hoc* methods that may be faster for specific cases. Nair (440) has proposed a systematic approach to deal with special cases of *UPS* robots. He shows that the inverse kinematics equations may be separated into a set of linear equations, which relate two sets of variables through a matrix A , and a set of non-linear constraint equations. The linear equations are solved and the solutions are transferred to an appropriate subset of the constraints equations. We then obtain a number, hopefully small, of polynomial equations, which are combined to obtain a univariate polynomial.

Nair considers various examples, for instance a 3-*UPS* wrist with a constraint mast, and establish correctly that the direct kinematics may be solved by finding the roots of a 8th order polynomial in one variable.

For 6-*UPS* robots with a planar base and platform Nair establishes interesting results.

Robot	Section	Assembly modes	degree	Real solutions
3-3 $\{X(\mathbf{w})\}$ (Δ ,Star)	4.2	2	explicit	2
Stewart	4.3.2.3	12	2×6	12
6-6 ($ ^6$)	4.3.5	40	40	40 (132)
6-6 SSM	4.3.4	40	40	24
6-6 5 aligned	4.3.3.7	16	4 explicit	16
6-5 ($\wedge ^4$)	(161)	40	40	?
6-5 (P)-(P)	4.3.3.1	40	40	10
6-4 ($\wedge ^3$)	4.3.3.2	16	8	10
6-4 ($\wedge^2 ^2$)	4.3.3.2	32	32	?
6-3($\wedge \wedge $)	4.3.3.3	8	-	?
6-3 (\wedge^3)	4.3.1	16	8	16
5-5 ($\wedge \vee ^3$)	4.3.3.4	40	40	24
5-5($\wedge \vee ^2$)	4.3.3.4	24	-	-
5-4 ($\wedge \wedge ^2$)	(161)	32	-	4 (367)
5-4 ($\wedge \vee ^2$)	(161)	16	-	?
5-4($\wedge \vee $)	(161)	8	-	?
5-4 ($\wedge \vee $)4P-3P	(440)	8	explicit	-
5-4 ($\wedge \wedge $)	(161)	24	-	8 (269)
5-4 ($\wedge^2 \vee$)	4.3.3.5	16	16	8
5-3 ($\wedge \wedge $)	(161)	8	-	?
5-3 ($\wedge \wedge \wedge$)	(161)	16	-	?
5-3 ($\wedge \wedge \vee$)	(161)	8	-	?
5-3 ($\wedge \vee \wedge$)	(161)	8	-	?

TABLE 4.6. Summary of the results known for the direct kinematics. In sequence, robot type, section or reference where the result is presented, maximum number of assembly modes, degree of the known univariate polynomial (the "explicit" mention indicates that closed-form solutions are known), highest number of real solutions found in the literature.

Robot	Section	Assembly modes	degree	Real Solutions
4-4 ($\wedge \vee $)	4.3.3.6	24	12	12
4-4 ($\wedge $)	(161)	8	-	4
4-4 ($\wedge \vee$)	(161)	∞	-	-
4-4 (\vee^2)	4.3.3.6	16	16	?
4-4 (\boxtimes^2)	4.3.3.6	16	16	16
4-4 ($\wedge \vee \vee$)	4.3.3.6	16	4	?
4-4 ($\vee \vee $)	(161)	16	-	?
4-4 ($\wedge \vee \vee$)	(161)	8	-	?
4-3 ($\boxtimes $)	(161)	8	-	?
4-3 ($\wedge \vee \vee$)	(161)	8	-	?
4-3 ($\vee \vee \vee$)	(161)	8	-	?
4-3 ($\wedge \vee \vee$)	(161)	16	-	?
4-3 ($\wedge \vee \vee$)	(161)	8	-	?
4-3 ($\boxtimes \wedge$)	(161)	8	-	?
4-3 ($\wedge \vee \vee$)	(161)	16	-	?
3-3 ($\wedge \vee$)	(161)	8	-	?
3-3 (\boxtimes)	4.3.2.2	16	8	16
3-3 ($\vee \vee$)	(161)	8	-	?
3-3 ($\boxtimes \wedge$)	(161)	8	-	?

TABLE 4.7. Summary of the results known for the direct kinematics: in sequence, robot type, section or reference where the result is presented, maximum number of assembly modes, degree of the known univariate polynomial, highest number of real solutions found in the literature.

4.4.1. MANIPULATORS WITH 9 LEGS

Proposition 1: *Parallel robots with planar base and platform that have 9 links, such that the matrix A has full rank, admit an explicit solution for the direct kinematics. For a set of fixed link lengths, we find two solutions, one of which is the mirror image of the other with respect to the base.*

Such 9 legs robot are of practical interest: figure 4.17 presents a nanopod (i.e. a 9-legged robot) that is available from Physik Instrumente. We note that Nair does not precisely determine the cases in which the 9×9 matrix A has full rank. The analysis of the rank of A may be performed for the

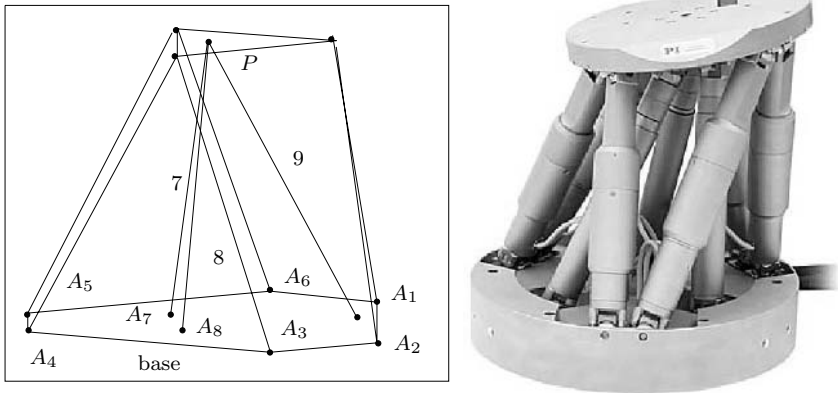


Figure 4.17. 6 degrees of freedom parallel robot with 9 legs. On the left a TSSM in which the links 1-6 are actuated, whereas the links 7-9 are passive and only useful for the direct kinematics. On the right a commercially available nanopod (courtesy Physik Instrumente).

symmetrical TSSM (figure 4.17) in which the links 1 to 6 are actuated, whereas the links 7 to 9 are passive and share the joint at P . In this case, the matrix A will be full rank if the following conditions hold:

- points A_7, A_8, A_9 are not collinear
- P is not on any of the lines B_1B_3, B_3B_5, B_5B_1
- A_4 does not have the same y coordinate as A_3
- A_4 is not on the line joining A_3 to the mid-point of A_1A_2

4.4.2. MANIPULATORS WITH 7 AND 8 LEGS

Proposition 2: *Parallel robots that have 8 links, planar base and platform and a matrix A with full rank, accept up to 8 explicit solutions for the direct kinematics.*

Consider the manipulator shown in figure 4.17. If the point P that attaches the passive links is on the axis y of the moving platform, the matrix A will have a full rank if the following conditions hold:

- P is not on the line B_3B_5
- A_4 does not have the same y coordinate as A_3

- A_4 is not on the line joining A_3 to the mid-point of A_1A_2
- A_7 is not on the line joining A_8 to the mid-point of A_1A_2

Nair studies a specific case for an 8 link robot called the 8-8 robot, for which the coordinates of the attachment points on the planar base are

$$\begin{aligned} A_1 &= (-b, a) & A_2 &= (b, a) & A_3 &= (a, b) & A_4 &= (a, -b) & A_5 &= (b, -a) \\ A_6 &= (-b, -a) & A_7 &= (-a, -b) & A_8 &= (-a, b) \end{aligned}$$

while the coordinates of the attachment points on the planar platform are

$$\begin{aligned} B_1 &= (-y, x) & B_2 &= (y, x) & B_3 &= (x, y) & B_4 &= (x, -y) & B_5 &= (y, -x) \\ B_6 &= (-y, -x) & B_7 &= (-x, -y) & B_8 &= (-x, y) \end{aligned}$$

He shows that the matrix \mathbf{A} will have rank 8 if at least two of the parameters a, b, x, y are not zero and if

$$\frac{a}{b} \neq \frac{y}{x} \quad \frac{a}{b} \neq \frac{x}{y}$$

Under these assumptions Nair finds a maximum of four explicit solutions for the direct kinematics.

Proposition 3: *Parallel robots with planar base and platform, with 7 links, a matrix \mathbf{A} with full rank, and a known length $\|\mathbf{OC}\|$, accept up to 8 explicit solutions for the direct kinematics.*

This result of Nair may be completed by a paper of Innocenti (276) that claims that for a 6-*UPS* robot of general geometry with seven links a unique solution may be found, and proposes an algorithm to find this solution. However Innocenti's result relies on the assumption that a given matrix is of full rank, a claim that is not substantiated and is not true if the base and platform are planar.

In conclusion, Nair's formalism presents interesting results, and sometimes gives polynomial formulations. However, the delicate operation of reducing the closure equations sometimes leads to a result that is not optimal, or even to no result at all, because the calculation becomes extremely complex. For the 6-*UPS* robot, Nair would obtain a polynomial of degree 144, although he did not complete the calculation because of the size of the expressions. This formalism was elaborated for robots with *RRPS* chains only; it deserves to be implemented for other types of chains.

4.5. Conclusion

In the previous sections we have presented methods for obtaining *all* solutions of the direct kinematics. The interest of such calculation may not

be clear as the *real* direct kinematic problem is to determine the *current* pose of the platform i.e. its pose when the joint variables were measured. In the next section we will present methods that may eventually calculate the current pose, but all of them need *a priori* information on this pose. This information may not be available (for example when starting the robot) and hence it is necessary to have a solution method that does not require it. Computing all possible solutions for the direct kinematics is a possible approach in that case. Unfortunately there is no known algorithm that allows one to determine which solution is the current pose in the set of solutions, and this is a challenging kinematic problem.

A possible way to design such an algorithm will be to consider that the robot has been built in a known initial assembly mode. Only the direct kinematic solutions that can be reached from this initial assembly mode with a trajectory that is singularity-free (we will come back later on this subject), that respect the constraints on the joint variables and that is interference-free (i.e. such that the legs, platform and base do not cross) may be valid solutions for the current pose. All these conditions have to be taken into account as the singularity-free condition is not sufficient: it was proved by Innocenti (275), that for planar robot two different direct kinematics solutions may be connected through a singularity-free trajectory, while Chablat (80) and Hunt (252) have shown the same result for spatial robots. Designing an algorithm for a complete verification is extremely difficult, and proving that it will lead to a unique solution is still an open problem.

Apart from solving the real direct kinematic problem, finding all the solutions may have an interest for the singularity and workspace analysis, as we will mention later.

Once again it is important to emphasize the problem of the certification of the result. In some of the proposed methods, the calculation may involve a large number of operations and may therefore be very sensitive to numerical round-off errors. Checking the validity of the solutions with the inverse kinematics may be necessary.

Another problem must also be mentioned. We have assumed in the calculation that all the data that are used for establishing the inverse kinematic equations are exact. This is clearly usually not true: the theoretical geometrical modeling of the robot does not exactly fit the real robot, and the joint variables are measured and therefore uncertain. This problem was mentioned by Guglielmetti (208) for the *Delta* robot.

A final problem for the above methods is that even the fastest one is still too slow for real-time use, for example for control purposes. We will now investigate fast numerical methods.

4.6. Fast numerical methods

In the previous sections, we have mentioned methods for the direct kinematics that allow the determination of all the solutions.

In practice, it is necessary to solve the direct kinematic as often as possible to get the best possible information on the current pose of the robot. This means that the unknown current pose will be close to the pose that was established the last time the direct kinematics was solved. We must therefore find the solution of a system starting with an estimate that is "close" to the current solution. This problem is classical in numerical analysis; this is why this section will suggest various purely numerical methods, and will compare their performances on two aspects: ability to determine the current pose, and computation time. We shall see that under reasonable hypotheses, some of these methods give the "right" solution within a reasonable time. These methods will be exemplified for a 6 d.o.f. robot, but may be used in all cases.

4.6.1. NEWTON SCHEMES

4.6.1.1 Principle

A classical method for solving a non-linear system of equations is the Newton iterative scheme. Formally, assume that the coordinates \mathbf{X} of the moving platform are related to the known joint variables vector Θ by

$$\Theta = \mathbf{G}(\mathbf{X}) , \quad (4.25)$$

and that \mathbf{X}_0 is an estimate of the solution. The iterative Newton scheme at iteration k is

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{A}(\Theta - \mathbf{G}(\mathbf{X}_k)) ,$$

The iterative scheme stops when $\|\Theta - \mathbf{G}(\mathbf{X}_k)\| < \epsilon$ where ϵ is a fixed threshold. There are many variants of the Newton scheme according to the choice of the matrix \mathbf{A} . We will mention:

- *Newton-Raphson scheme* for $\mathbf{A} = \mathbf{J} = \frac{\partial \mathbf{G}^{-1}}{\partial \Theta}(\mathbf{X}_k)$, where \mathbf{J} is the Jacobian matrix of the robot (see next chapter)
- *Damped Newton scheme* for $\mathbf{A} = \alpha^k \frac{\partial \mathbf{G}^{-1}}{\partial \Theta}(\mathbf{X}_k)$ where α^k is a gain factor
- *Quasi-Newton scheme*: in that case the matrix \mathbf{A} of the Newton-Raphson scheme is not computed at each iteration, but only once every p iterations, where p is a number fixed by the user (for example it is possible to use a constant matrix for \mathbf{A})

Let \mathbf{X}_s be the solution of the problem, and define the calculation error e_k at step k of the iteration as the maximum of $|X_s^j - X_k^j|$. Let us assume that, for a given scheme, there are two numbers c, r such that $e_{k+1} \leq c e_k^r$:

the number r will be called the rate of convergence of the scheme. The convergence rate of Newton-Raphson is quadratic ($r = 2$) if we are sufficiently close to a solution (in practice this means that at each iteration we double the number of exact digits), while the convergence rate for the other methods is only linear ($r = 1$). Hence it seems that the Newton-Raphson scheme is the method to be used, but we will see that the choice of the faster method is not that obvious.

4.6.1.2 Implementation for the direct kinematics

In this section we address the implementation of a Newton scheme for 6 d.o.f. robots. For a practical implementation we need first to determine what type of vector \mathbf{X} we should use. The Newton scheme may be used with the minimal set of pose parameters as unknowns (499) (namely 6 for 6 d.o.f. robot) or with an extended vector including the passive joint parameters (231) (surprisingly we will see that increasing the size of \mathbf{X} does not always lead to an increase in the computation time).

For a minimal representation of the pose parameters we will use the location of C and the Euler angles for the orientation part (or any other angular representation) and we will now establish a Newton-Raphson scheme based on this representation. We have to establish what matrix \mathbf{A} will be used in this scheme. We have seen in the "Inverse kinematics" chapter that usually equation (4.25) is available, so that the jacobian matrix of this system may be established and numerically calculated for a given \mathbf{X} . The inverse of this matrix will be the matrix \mathbf{J} used in the scheme; that will be called the *Euler angles jacobian matrix*. An estimate of the generalized coordinates is calculated at each iteration until the corresponding joint coordinates get sufficiently close to Θ , the threshold ϵ being chosen to be compatible with the accuracy of the sensor measurements.

We can also use the following vector in order to represent the position and the orientations:

$$\mathbf{X} = [x_c, y_c, z_c, \alpha \mathbf{v}] ,$$

where the orientation part uses the rotation vector $\delta = \alpha \mathbf{v}$, where \mathbf{v} is the rotation axis unit vector and α is the rotation angle. We know that the instantaneous rotation vector Ω is

$$\Omega = \dot{\alpha} \mathbf{v} + \sin \alpha \dot{\mathbf{v}} + (1 - \cos \alpha) \mathbf{v} \times \dot{\mathbf{v}} ;$$

If α is small, this relation may be written as

$$\Omega \approx \dot{\alpha} \mathbf{v} + \alpha \dot{\mathbf{v}} = \dot{\delta} : \tag{4.26}$$

Thus the derivative of the vector \mathbf{X} is approximately equal to the twist of the end-effector. Consequently the jacobian of \mathbf{G} relates the actuated joint

velocities to the twist of the end-effector, and we will see in the next chapter that this matrix is called the inverse kinematic jacobian. In this scheme we calculate at each iteration the corrections to be made to the location of C , and those to be made to the rotation vector. Using this last part, we have to calculate the rotation matrix δR corresponding to the new orientation of the end-effector. This is done using the following formula:

$$\delta R = I_3 + (1 - \cos \alpha)\hat{v}^2 + \sin \alpha \hat{v} , \quad (4.27)$$

where I_3 is the 3×3 identity matrix and \hat{v} the anti-symmetrical matrix associated with the vector cross-product

$$\hat{v} = \begin{pmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{pmatrix}$$

Whatever the chosen representation, the matrix J depends on the configuration. Furthermore we will see in the next chapter that the matrix J is usually not available, while J^{-1} , the jacobian of the system \mathbf{G} can easily be calculated. Hence to apply the Newton-Raphson scheme we have to numerically invert J^{-1} , and the inversion of this matrix will increase the computation time of the algorithm. The use of the quasi-Newton scheme with J calculated at regular intervals, or constant, may thus be justified. In that case the iterative scheme needs only the calculation of the inverse kinematics at each iteration, and the computation time will therefore be approximately a multiple of the inverse kinematics computation time.

The choice of using angles to represent the orientation of the end-effector may cause a representation problem. For instance, for the Euler angles, the jacobian of the system \mathbf{G} is singular for $\theta = 0$. Choosing another set of angles with a singularity away from the workspace of the robot is possible, but singularity may still occur during the scheme. A better solution is to switch to the quasi-Newton scheme as soon as the determinant of J^{-1} become small.

Another possible implementation for 6-UPS robots consists in using a specific result that we presented in the chapter devoted to the inverse kinematics: we established that we could express the coordinates of the center of the moving platform as a function of the rotation angles and the link lengths. We can then conceive methods where the Newton scheme is used on the system that involves only the rotation angles, the location of the platform being calculated, at each iteration, directly from the link lengths and the new rotation angles. We can thus hope for a decrease in the computation time as the size of the system to be solved is reduced. A numerical study shows however that this approach may have overall a

larger computation time: the computation time of each step is smaller, but the convergence domain decreases notably, thereby leading to a much larger number of steps before convergence.

Another choice for \mathbf{X} consists in representing the pose of the platform (supposed to be planar) by the coordinates of three specific points on it i.e. by 9 unknowns. Indeed if the coordinates of these points in the reference frame are known, then the pose of the platform will be completely determined. Let us assume that we are using the coordinates of the points B_1, B_2, B_3 . We first note that there are always three triplets $(\alpha_i, \beta_i, \lambda_i)$ of constants with $\alpha_i + \beta_i + \lambda_i = 1$, called the *barycentric* coordinates, such that:

$$\mathbf{OB}_k = \alpha_k \mathbf{OB}_1 + \beta_k \mathbf{OB}_2 + \lambda_k \mathbf{OB}_3 \quad k \in [4, 6] \quad (4.28)$$

For a given platform geometry the barycentric coordinates of a point may be calculated by solving the linear system

$$\mathbf{CB}_{k_r} = \alpha_k \mathbf{CB}_{1_r} + \beta_k \mathbf{CB}_{2_r} + \lambda_k \mathbf{CB}_{3_r} \quad (4.29)$$

The positions of the points B_4, B_5, B_6 may therefore be calculated from the positions of B_1, B_2, B_3 . The 9 unknowns must be such that they satisfy the following 9 constraint equations:

$$\|\mathbf{B}_i \mathbf{B}_j\|^2 = l_{ij}^2 \quad i, j \in [1, 3], i \neq j \quad ; \quad G(\mathbf{OB}_k, \Theta_k) = 0 \quad k \in [1, 6]$$

The 3 first equations indicate that the distances between pairs of points in the set B_1, B_2, B_3 shall be known constants. The remaining 6 equations are the inverse kinematic equations of the leg k that connects the actuated joint variable Θ_k to the location of point B_k (for example for a Gough platform we will have $\|\mathbf{A}_k \mathbf{B}_k\|^2 = \rho_k^2$). These 9 equations are all functions of the 9 unknowns if we use the relations (4.28) and may be written as $F(\mathbf{X}) = \mathbf{0}$. One can then use Newton's method on this system, and initialize it by an estimate of the positions of B_1, B_2, B_3 . We must however note that, as in the case of the previous algorithms, it is difficult to calculate the jacobian matrix of the system (see exercise 4.8).

Whatever the type of iterative method, we note that the Newton scheme involves calculations that can be done in parallel: we could therefore use a distributed implementation in order to reduce the computation time. Guglielmetti (208) shows that a method which requires 640 μs on a single processor board may requires only 343 μs on a system with four processors. Other authors have suggested the use of a distributed implementation (198).

4.6.1.3 Drawbacks of the Newton schemes and real-time issues

It is widely believed that the Newton schemes will converge as soon as the initial estimate is sufficiently "close" to a solution, and that this convergence

will lead to the solution that is the closest to the initial guess. Unfortunately both assumptions are wrong: Newton schemes may not converge (as we will show in the examples) and it is easy to find counter-examples for the second assumption (see exercise 4.9). The inversion of the Jacobian matrix may also cause a convergence problem when the matrix $J^{-1}(\mathbf{X}_k)$ is close to being singular. Both problems pose an important reliability problem, especially if the direct kinematics is used in a control loop, as an incorrect answer (or no answer) will lead to incorrect control.

Fortunately, when using Newton's method for control purposes, we have some bounds on the solution parameters: given the sampling time of the controller, the maximal velocity of the platform, and the latest pose, we can derive the extremal values for the new pose parameters. Consequently we may detect if Newton's method has converged towards an incorrect solution, unless more than one solution satisfies the extremal values constraints (for a slow moving robot this should not occur, as such closeness of 2 solutions implies a nearly singular J^{-1} , which will lead to a failure of the calculation of the inverse).

Still, from the control viewpoint, it is important to note that the comparison of the computation time of the various direct kinematics procedures has to take into account the sampling time of the robot controller. Indeed, the controller will make available the joint coordinates measurements only at each sampling time. Similarly a processor devoted to the solution of the direct kinematics will only provide its result according to the controller clock. Hence all direct kinematics procedures that are able to provide the solution at the end of the same number of controller clock cycles are equivalent. Consequently, for control purposes, we may use a direct kinematics procedure that may not be the fastest available in absolute term, but the one that leads to the most reliable result, provided that the additional computation time, compared to the fastest method, does not exceed the sampling time.

4.6.1.4 Convergence of the Newton schemes

The drawbacks of the Newton schemes lead us to investigate more closely the convergence domains of these schemes. We consider a system $\mathbf{f}(\mathbf{X}) = \mathbf{0}$, where \mathbf{f} is a vector of dimension n , and investigate the convergence of the Newton-Raphson scheme for this system.

In the following study, the norms for vectors and matrices are defined thus:

$$\text{if } \mathbf{A} = ((a_{ij})) \text{ then } \|\mathbf{A}\| = \max_i \sum_j |a_{ij}| .$$

We define a starting point \mathbf{X}_0 for the iteration and a closed neighborhood

around this point:

$$\bar{U}(\mathbf{X}_0) = \{\|\mathbf{X} - \mathbf{X}_0\| \leq \mathcal{H}\} .$$

Kantorovitch's theorem (575) states that:

1. if the jacobian matrix of the system $\mathbf{J} = \left(\left(\frac{\partial \mathbf{f}}{\partial x_j}\right)\right)$ has an inverse \mathbf{J}^{-1} at $\mathbf{X} = \mathbf{X}_0$ with $\|\mathbf{J}^{-1}\| \leq A_0$,
2. if there is a constant B_0 such that

$$\|\mathbf{J}^{-1}\mathbf{f}(\mathbf{X}_0)\| \leq B_0 \leq \frac{\mathcal{H}}{2} , \quad (4.30)$$

3. if there is a constant C such that

$$\sum_{k=1}^{k=n} \left| \frac{\partial^2 f_i(\mathbf{X})}{\partial x_i \partial x_j} \right| \leq C .$$

for $i, j = 1, 2, \dots, n$ and all \mathbf{X} in $\bar{U}(\mathbf{X}_0)$,

4. if the constants A_0, B_0, C satisfy the inequality:

$$\mu_0 = 2nA_0B_0C \leq 1 , \quad (4.31)$$

then the iterative scheme:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{J}^{-1}(\mathbf{X}_k)\mathbf{f}(\mathbf{X}_k)$$

with the initial estimate \mathbf{X}_0 converges towards the unique solution of $\mathbf{f}(\mathbf{X}) = \mathbf{0}$ in $\bar{U}(\mathbf{X}_0)$. The size of the convergence domain is therefore given by $\text{Min}(B_0, \frac{1}{2nA_0C})$. A similar result is obtained for the quasi-Newton scheme when the matrix \mathbf{J} is constant, with a smaller convergence domain. Kantorovitch's theorem must be understood as providing a lower bound for the radius of the convergence domain, a bound that may be exact in some cases. It may thus happen that, for a given solution, the convergence domain is larger than the bound provided by the theorem.

We applied this theorem to the Newton-Raphson scheme, in which the pose of the platform is represented by the position of three points B_i , using as robot the INRIA left hand. Choosing the most favorable triplet of points for the pose \mathbf{X}_{nom} defined by $x_c = y_c = 0, z_c = 53.3, \psi = \phi = \theta = 0$, we find $A_0 = 0.214, B_0 = 3.328, C = 4$. The application of the inequality (4.31) leads to a convergence domain with diameter 0.0648. However numerical experiments shows that the Newton-Raphson scheme always converges to \mathbf{X}_{nom} in a much larger domain.

4.6.1.5 Extending the unicity domain: the inflation

We have seen that Kantorovitch's theorem allows one to determine a domain in which there is one and only one solution but also that the size of this domain may only be a lower bound. We will see later on that determining the largest domain around a solution may be of interest. For that purpose we will briefly describe a procedure, called *inflation*, that may enlarge the domain determined by Kantorovitch's theorem (detailed mathematical explanation about this procedure may be found in (445)). Assume that \mathbf{X}_0 is a solution of the system $\mathbf{f}(\mathbf{X}) = \mathbf{0}$ and let \mathcal{D} be a domain that includes \mathbf{X}_0 . It is easy to show that if the Jacobian matrix \mathbf{J} of \mathbf{f} is non singular over \mathcal{D} , then \mathbf{X}_0 is the only solution in \mathcal{D} . The problem is now to determine the largest possible domain \mathcal{D} around \mathbf{X}_0 so that $\mathbf{J}(\mathbf{X})$ is non singular for all \mathbf{X} in \mathcal{D} .

For that purpose, let us define a *diagonally dominant* $n \times n$ matrix $\mathbf{J} = ((J_{ij}))$ as a matrix such that

$$\forall i \in [1, n], |J_{ii}| > \sum_{j=1}^{j \neq i} |J_{ij}|$$

One of the properties of a diagonally dominant matrix is that it is non singular. In the procedure proposed by Neumaier (445), the domain \mathcal{D} is initialized with \mathbf{X}_0 , and its size is iteratively increased by a fixed quantity β . At each step, interval analysis is used to verify that $\mathbf{J}(\mathbf{X})$ is diagonally dominant over \mathcal{D} . But for the direct kinematic solution procedure using the coordinates of 3 points as representation of the pose of the platform, we have been able to show that it is possible to determine directly the largest possible β such that for any \mathbf{X} in the domain $[\mathbf{X}_0 - \beta, \mathbf{X}_0 + \beta]$, \mathbf{J} is diagonally dominant (417).

4.6.2. INTERVAL ANALYSIS SCHEME

As mentioned previously, an interval analysis solution method is appropriate for the direct kinematics (417). A property of this method is that its computation time is sensitive to the size of the domain in which we are looking for solutions. For the real-time direct kinematics we have already seen that, if we know the solution X_0^k at time k , it is easy to determine a hyper-cube search domain \mathcal{S}_1 whose center is X_0^k , and whose size may be determined, so that \mathcal{S}_1 is guaranteed to include the pose at time $k + 1$. Using the direct kinematics solution algorithm based on interval analysis, we are able to determine certified solution(s) in \mathcal{S}_1 . Two failure cases may occur:

- the solution is not certified: this occurs when the matrix \mathbf{J} is nearly singular

- two certified solutions are determined (more certified solutions may exist but the algorithm stops as soon as at least 2 solutions have been found)

In both cases, if the direct kinematics is used in a control loop, it is necessary to stop the robot immediately.

Note that an interesting alternative to interval analysis, that uses the same branch-and-bound principle, is based on the transformation of the inverse kinematic equations into Bernstein polynomials (43).

The certification of the solution will be time consuming. The next section will present some experimental tests of a preliminary version of this method; it is not easy to implement.

4.6.3. METHODS EFFICIENCY AND COMPUTATION TIME

We will compare the different suggested procedures, for the INRIA left hand robot, and a set of test values. The methods are as follows:

- 1: quasi-Newton method with kinematic jacobian. The algorithm calculates the rotation matrix, but not the corresponding Euler angles.
- 2: quasi-Newton method with Euler angles jacobian. Euler angles are therefore computed
- 3: quasi-Newton method using 3 points to represent the pose of the moving platform
- 4: polynomial form. The platform is simplified so that the robot becomes a TSSM. The computation time will be identical for all tests.
- 5: interval analysis algorithm using 3 points to represent the pose of the moving platform

Table 4.8 shows the values chosen for the tests. The jacobians used in the quasi-Newton methods are estimated at $(0, 0, 40, 0, 0, 0)$ for the kinematic jacobian, $(0, 0, 40, 0, 12, 0)$ for the Euler angles jacobian. The error threshold on the link lengths is 0.01 cm. For method 3 the initial estimates of the location of the points are derived from the pose parameters used as initial guesses for the other methods. For method 5, the size of the search domain for each test is fixed as four times the difference between the exact and estimated values given in the test.

Table 4.9 shows the computation times for the various methods established on Dell D400, 1.2 GHz. The results may be summarized thus:

- the quasi-Newton method using the kinematic jacobian presents the best characteristics as far as convergence and computation time are concerned.
- the quasi-Newton method using Euler angles jacobian exhibits good computation time but its convergence domain is smaller than that for the previous method.

- the three point method works almost in a constant time which is larger than those of the previous two methods; its convergence domain seems larger, and it provides a better estimate of the orientation.
- the method using interval analysis approach is much slower than the other methods although largely compatible with real time constraints. This method is penalized by the choice of the search domain and the certification of the result. Note also that its current implementation may be largely improved; still it is the safest method
- for the special case of the TSSM, the polynomial form has a computation that is only 10 to 20 larger than the Newton schemes

test number	x_c	y_c	z_c	ψ	θ	ϕ
(0) exact value	3	3	40	0	0	0
estimated value	2.9	2.9	39.9	0.5	-0.5	0.5
(1) exact value	3	3	40	0	0	0
estimated value	0	0	40	0	0	0
(2) exact value	3	3	40	0	0	0
estimated value	2.95	2.95	39.95	0.1	-0.1	0.1
(3) exact value	3	3	40	0	0	0
estimated value	0	0	45	0	0	0
(4) exact value	4	-5	47	15	-10	20
estimated value	0	0	45	0	0	0
(5) exact value	2	-2	46	15	-10	20
estimated value	0	0	45	0	0	0

TABLE 4.8. Test positions values (angles in degrees).

Finally let us note that all the numerical methods presented in the above sections assume that, even with modeling and measurement uncertainties, there is a solution to the direct kinematics. An alternative approach to deal with uncertainties is to use a nonlinear observer based on the dynamics of the robot, but its final errors seem not to be compatible with fine positioning (313)*.

4.6.4. PATH TRACKING

A problem related to direct kinematics is the path tracking problem: given time laws for the actuated joint coordinates, determine what will be the

method/test number	0	1	2	3	4	5
1	2.4	1.8	1.3	1.9	6.1	4.3
2	2.5	3.2	1.7	3.1	-	-
3	3.87	3.85	3.86	3.87	4.04	4.04
4	39.9					
5	300	340	300	390	460	390

TABLE 4.9. Computation time for direct kinematics resolution (in μ s, sign - indicates a non-convergence of the algorithm).

trajectories of the robot. For that purpose Siciliano (539) has proposed the *CLDK* algorithm that assumes that the initial pose at time $T = 0$ is known. The interval analysis scheme may also be used for that purpose: all initial poses are obtained by the interval algorithm and then a continuation scheme based on Kantorovitch's theorem with an adaptive time step allows us to follow all the kinematic branches very efficiently with zero tracking errors (417).

4.7. Direct kinematics with extra sensors

The results presented in the previous sections show that, with progress in algorithms and processor speed, direct kinematics is less a problem than it was a few years ago. Still we have seen that finding the current pose of the robot may be difficult, even with good information on the location of this pose. Furthermore it may be interesting to investigate how to improve the computation time, as direct kinematics is an important issue for control. One possible approach to solve these problems is to add sensors (i.e. to have more than n sensors for a n d.o.f. robot) to obtain information, allowing a faster calculation of the current pose of the platform, at the cost of more complex hardware. We will also see in the calibration chapter that adding sensors is justified by the possibility that they offer for the auto-calibration of the robot.

This solution was adopted in practice by Arai and Stoughton (14) and Inoue (277). Inoue's robot has rotation sensors measuring the inclination of the parallelograms of his robot, whereas Arai adds an extra arm, with 2 rotation sensors at each end, and a sensor measuring the length of the arm.

Adding extra sensors induces some interesting problems:

- what are the types, locations and minimal number of sensors that are

necessary to obtain a unique solution for the direct kinematics?

- given the measurement errors of the extra sensors, is the accuracy of the computed pose compatible with the desired accuracy for the manipulator ?

In the remaining section we will illustrate these problems on the 6-*UPS* robot. These problems have been investigated for other type of robots: Notash (450) investigated the addition of a fourth chain to a 6 degrees of freedom robot possessing 3 chains, while Parenti studied the problems for the spherical wrist equivalent to the 6-4 robot with 3 legs attached at the same point on the platform (464).

4.7.1. TYPE AND LOCATION OF THE EXTRA SENSORS

There are two types of sensors that can be added:

- angular sensors that may be placed on the passive joints of the manipulator, if possible near the base in order not to make the moving equipment heavier
- linear or angular sensors that are placed on extra passive chains or between existing chains.

The main drawback of adding passive arms is the risk that the workspace may be reduced, because of the even more important risk of link interference. On the other hand, angular sensors placed on the joints may present the same drawbacks for accuracy as for serial robots, since the measurement error will be amplified by the length of the links: one may therefore expect relatively poor accuracy in the estimated pose of the platform.

4.7.2. MAXIMAL NUMBER OF SENSORS

Determining all possible sensor layouts (location, type and number of sensors) that yield a unique solution for the direct kinematics is far from being trivial. Adding a sensor either leads to an additional constraint equation or to a simplification of the initial inverse kinematics equations. This new system may have only one solution, so that the sensor layout is a candidate for an implementation. On the other hand, the new system may still admit theoretically several solutions but the sensor layout may be acceptable if there are multiple solutions only for specific poses or particular robots geometries, that are not physically meaningful. We will now present known results for various sensor layouts.

4.7.2.1 *Addition of angular sensors*

Consider a 6-*UPS* robot to which we add rotation sensors at the level of the universal joints on the base, i.e. one or two sensors are added to

each joint. If we assume that a universal joint has two sensors, we may measure the direction of the link. Furthermore, the joint sensor provides the length of the link. As a consequence, we may calculate the position of the extremity of the link that belongs to the platform.

If three links are instrumented, which means adding 6 sensors, we can calculate the position of three points of the platform and hence solve the direct kinematics problem. Thus, 6 sensors are enough to determine the pose of the moving platform; this sensor configuration will be called 2-2-2, each digit corresponding to the number of sensors placed on a link.

Another interesting configuration is the 2-2, 2 sensors on 2 universal joints: it can be shown that, in general, 2 solutions may be obtained for the direct kinematics, but only for very particular poses of the robot (400), which means that in practice, the solution is unique.

A key result is the one proposed by Parenti (461): with only one additional angular sensor, it is possible to determine the current pose of the platform as long as the base and platform are planar. In the general case, the solutions may be obtained from the common roots of 2 univariate polynomials of degree 40. A drawback of the approach presented by Parenti is that the calculation is too complex to be done in real time. The same author proposes a real-time solution for the 2 sensors case (462)* that leads to one solution. Hence every layout including a 2 configuration, such as 2-2, 2-2-2, 2-2-1, 2-2-1-1, 2-1, 2-1-1, 2-1-1-1, 2-1-1-1-1, will have only one solution. Similarly, for a SSM, any layout including one angular sensor will have only one solution. The extensive study of Tancredi (573)* of each layout may complement the results proposed by Parenti.

Open problems remain for robots having non coplanar attachment points on the base and/or on the platform, and whose sensor layout involves only 1 sensor per leg. If we have at least 3 such sensors, then a TSSM is included in the structure, and consequently the number of solutions cannot exceed 16. For a robot with 1-1 layout an upper bound of the number of solutions is 32, while the layout with one angular sensor has evidently at most 40 solutions. Baron (26)* has proposed an interesting algorithm when the sensors layout allows for a kinematic decoupling between the translation and orientation of the moving platform. In that case the determination of the orientation part can be cast as a linear algebraic system constrained by the orthogonality condition of the rotation matrix.

4.7.2.2 *Addition of linear sensors*

This method consists in adding passive extensible links with measured lengths. It is an interesting method because the estimation of the pose will be less sensitive to measurement errors than with angular sensors.

Section 4.4 showed that for the TSSM, adding three passive links sharing a joint on the moving platform allows us to calculate two explicit solutions,

except for particular geometries, which we presented in section 4.4.1. We also know that it is not necessary to investigate the case of a robot with planar base and platform when less than 3 links are used, since Nair (440) has shown that for 8 links, up to 8 explicit solutions could be found. For a general geometry of the base and platform, Innocenti claims that with 7 legs a unique solution may be found (276), but this claim relies on a given matrix being of full rank, and this assumption is not substantiated. As for the SSM, Nair showed that adding three passive links usually gave two solutions which are symmetrical with respect to the base (and hence one of the solutions can be discarded). Bonev (49) proposes for the SSM a specific arrangement of the 3 additional legs that allows for an analytic calculation of the solution. The same author has also studied the addition of 3 legs when only the platform is planar, but was unable to show that the solution was unique (44).

4.7.2.3 *Combination of angular and linear sensors*

Chiu (95) has proposed a method for the 6-*UPS* robot that involves 3 additional sensors: 2 rotation sensors at the universal joints located at A_1 on the base, and a linear sensor on a passive leg that connects the base and platform, the attachment point on the platform lying on the line B_1, B_2 . This layout leads generally to 4 solutions, but Chiu provides necessary and sufficient conditions to get a unique solution together with an algorithm that allows one to track the current pose during a motion.

4.7.3. RELATIONSHIP BETWEEN SENSORS ACCURACY AND POSE ACCURACY

It is extremely interesting to be able to calculate a unique solution to the direct kinematics, but the calculation accuracy, for given measurement errors, should remain compatible with the accuracy required for the robot. Thus Arai (14) uses only the measurement of the pose that was obtained from a passive arm as an initial estimate for an iterative algorithm. If the measured pose is close to the solution, the certified Newton method will quickly converge toward the solution, provided that we are not close to a singularity.

Another possible approach is to determine the maximal allowed error for the extra sensors so that we can reach a desired accuracy on the estimated pose for any pose within a given workspace. A preliminary study was performed by Stoughton (554), and an extensive study of the relation between the error on the pose and the errors on the extra sensors was carried out by Tancredi (573)*. He explicitly calculated the matrix relating the errors in the pose of the moving platform to the errors in the angular sensors. He did this for the 2-2 layout and could thus draw maps showing the necessary

sensor accuracy. For example, figure 4.18 shows the maximal distance between the real pose and the calculated pose for the 2-2, when the maximal sensor errors are 0.0062 radians, and when the platform moves in the $x - y$ plane. These maps just give indication on the maximal allowed value for

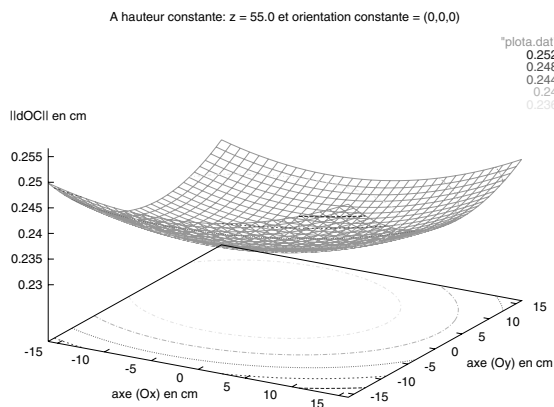


Figure 4.18. Maximal distance between the calculated pose and the real pose in the 2-2 case for maximal sensor errors of 0.0062 radian and when the platform moves in the $x - y$ plane.

the sensor errors for typical robot motion; we will present in section 5.4.2.5 an algorithm that allows one to effectively compute this maximal allowed value over a given workspace.

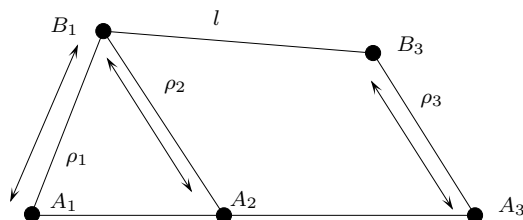
Another approach is suggested by Baron and Angeles (26); they use layouts for which the direct kinematics equations become linear in the coordinates of C . After solving three of such equations, only the rotation matrix has to be calculated; they suggest an estimation method for the rotation matrix that minimizes the positioning errors. Vertechy (594) proposes an alternate method that is less computer intensive, but is valid only for 6-3 robots.

Finally Chiu (95) provides a method for his special layout that allows for the determination of the location of the attachment points of the redundant leg that minimizes the error in the direct kinematics, but only for a specific pose of the platform.

4.8. Exercises

Exercise 4.1: Show that the manipulator of figure 4.4 has at most 4 assembly modes. Look at the value of the polynomial for $T = -1, 1$.

Exercise 4.2: Show that the robot in the following figure has 4 assembly modes that may be obtained by solving a sequence of two quadratic equations.



Exercise 4.3: Determine the direct kinematics of the various planar parallel robots that are presented in the chapter "Architecture".

Exercise 4.4: Consider Ming's over-constrained planar robot (422). It is made of a rectangular platform whose corners are linked to the ground by 4 wires. Show that the maximum number of solutions for its direct kinematics is still 6.

Exercise 4.5: Show that the *Star* robot admits two solutions for the direct kinematics

Exercise 4.6: We consider the following parallel robot, described by Husain (253): the moving platform is linked to the base by 3 chains. Two of these chains are made of three successive revolute joints. The revolute joint that is linked to the base is passive, while the others are actuated. The third chain is made of a universal joint that is attached to the base, each axis of this joint being actuated, and which is followed by a passive prismatic joint. The chains are linked to the moving platform via ball-and-socket joints. Show that the direct kinematics for this type of mechanism has at most 16 solutions. Suggest a way to obtain a univariate polynomial which will give the solutions.

Exercise 4.7: Consider the robot represented on figure 4.19; Nair calls this *W0*. Using Nair's method, show that this robot admits up to 8 solutions for the direct kinematics, and that they may be determined explicitly.

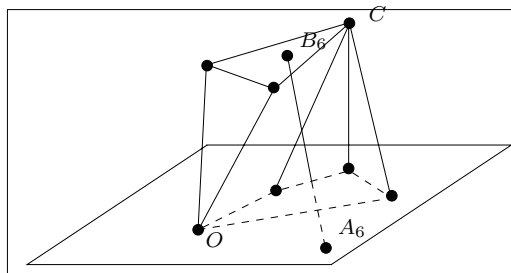


Figure 4.19. The *W0* manipulator

Exercise 4.8: Calculate the inverse of the jacobian matrix that appears in the iterative method using the position of three points B_i to represent the pose of the moving platform.

Exercise 4.9: Let f be the function defined by $f = \sin(x+1)e^{e^x} + 1$. Plot the function f for x in $[-2.2, 2.2]$ to determine the two solutions of $f = 0$ in the range $[-2.2, 2.2]$. What are the solutions given by the Newton scheme with as initial guess $-2.5, -2, -1.5, 0, 1, 2$? Explain why Newton converges with the initial guess 2 to this solution although there is another solution that is much more to the initial guess.

Exercise 4.10: Show that for a TSSM, the addition, under certain conditions, of three extra angular sensors allows us to determine the direct kinematics uniquely.

Exercise 4.11: Show that under the condition of the previous exercise, a TSSM that is equipped with two angular sensors, will in general have only one solution for the direct kinematics.

Exercise 4.12: Under what condition will a TSSM that is instrumented with two angular sensors, as in the previous exercise, have two solutions for the direct kinematics?

Exercise 4.13: Show that a TSSM that is instrumented with one angular sensor, under the conditions found in the previous exercise, will have one solution for the direct kinematics.

Exercise 4.14: Show that the *Nabla 6* robot described in figure 2.36, admits at most 16 solutions for its direct kinematics.

Exercise 4.15: Show that solving the *Hexa* direct kinematics is equivalent to solving that of the *6-UPS* robot.

Problem 4.1: Find the maximum number of solutions to the direct kinematics of the robot proposed by Zoppi (figure 2.31) and find all the solutions

Problem 4.2: Consider the robots for which an upper bound of the number of assembly modes is known. Is it always possible to find a manipulator with this number of assembly modes? If the answer is 'no', give a counter example

Problem 4.3: Is it possible to determine all the special layouts of joints for a given architecture such that an analytic solution of the direct kinematics may be obtained, see for example (325) for the *6-UPS* robot?

Problem 4.4: Are there geometries for the TSSM for which the direct kinematics polynomial may be factored?

Problem 4.5: Can we obtain a condition on the geometry and joint variables of a TSSM so that the direct kinematics will have 16 solutions?

Problem 4.6: May Nair's method be extended to other robot architectures such as the robots that have chains of the *PRRS* type?

Problem 4.7: May the frequency of obtaining various numbers of assembly modes for a given workspace be determined from the architecture and the geometry of a manipulator?

Problem 4.8: Design an algorithm that uses the conditions presented in section 4.5 to eliminate infeasible solutions for the direct kinematics, and determine under what conditions this algorithm leads to a unique feasible solution

Problem 4.9: Determine the minimal number of sensors to add to the various planar parallel robots that are presented in the chapter "Architecture", so that a unique solution is found for the direct kinematics.

Problem 4.10: Determine under what conditions all layouts with only one additional angular sensor in the legs for a 6-*UPS* robot will lead to a unique direct kinematics solution

Problem 4.11: Determine under which geometrical condition a 6-*UPS* robot with seven instrumented legs has a unique solution to its direct kinematic (276)

Problem 4.12: Determine the positioning and the minimal number of linear sensors that should be added to the 6-*UPS* robot to obtain a unique solution for the direct kinematics.

Problem 4.13: Using the results of the previous problem, determine the relation between the error in the estimation of the pose and the sensor errors.

Problem 4.14: Suppose a linear sensor is placed between two fixed points of two links of a 6-*UPS* robot. How many sensors of this type are necessary in order to obtain a unique solution for the direct kinematics?

Velocity, accuracy and acceleration analysis

This chapter will deal with the determination of the relations between the twist of the moving platform and the actuated joint velocities. We will establish how the limits on the joint velocities influence the allowed twist of the end-effector. We will then present the relations between the joint accelerations and the cartesian and angular accelerations of the platform.

5.1. Kinematics relations

Let Θ_a , Θ_p be the actuated and passive joint velocities, and \mathbf{W} the twist of the end-effector constituted of \mathbf{V} , the cartesian velocity of a specific point of the end-effector, called the *operating point*, and of Ω , the angular velocity vector. We assume here that all joints have 1 d.o.f. (higher pairs may be decomposed into a combination of 1 d.o.f. joints) and we will follow Zlatanov (663) by defining the *forward instantaneous kinematic problem (FIKP)* as determining the twist \mathbf{W} and the passive joint velocities Θ_p as functions of the active joint velocities Θ_a . The *inverse instantaneous kinematic problem (IIKP)* is to determine the passive and active joint velocities Θ_p , Θ_a as functions of the twist \mathbf{W} . It must be noted that the twist depends upon the choice of the operating point and hence any relation involved in FIKP and IIKP will be affected by the location of this point.

5.2. Inverse jacobian matrix

We consider here a non-redundant parallel robot with n d.o.f. and having N joint variables, n of which are the actuated joint variables Θ_a . We showed in the "Inverse kinematics" chapter that it is possible to obtain various relations between the joint coordinates and the generalized coordinates with the generic form:

$$\mathbf{H}_2(\mathbf{X}, \Theta) - \mathbf{H}_1(\mathbf{X}) = \mathbf{0} . \quad (5.1)$$

If Θ include all passive and active joint variables there are $N + n$ unknowns in these equations. As the robot should have mobility 0 when the n actuated joints are locked the number of equations in (5.1) should be N . We may also distinguish a particular case of relations (5.1), that will be called the

minimal kinematics set (MKS), in which Θ is reduced to the n actuated joint variables Θ_a , leading to a system of n equations.

We assume that the platform twist \mathbf{W} consists of the p -dimensional cartesian velocity vector \mathbf{V} and of Ω , the q -dimensional angular velocity vector. A differentiation of (5.1) allows us to obtain a relation of the type:

$$\mathbf{A}(\mathbf{X}, \Theta)\dot{\Theta}_a + \mathbf{B}(\mathbf{X}, \Theta)\dot{\mathbf{X}} + \mathbf{C}(\mathbf{X}, \Theta)\dot{\Theta}_p = 0, \quad (5.2)$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are $N \times n$, $N \times n$, $N \times (N - n)$ matrices. In the following analysis we will omit indicating that matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are functions of the pose parameters and joint variables.

If the robot has at most one rotational d.o.f. (i.e. $q \leq 1$), then the orientation representation may be chosen so that its derivative is the angular velocity of the end-effector i.e. $\dot{\mathbf{X}} = \mathbf{W}$. Otherwise the derivatives of the orientation representations are not the components of the angular velocity of the platform. For example if we use the Euler angles to represent the orientations of the end-effector, the angular velocity vector Ω is related to the derivatives of the Euler angles by

$$\Omega = \mathbb{T}_e \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & \cos(\psi) & \sin(\psi) \sin(\theta) \\ 0 & \sin(\psi) & -\cos(\psi) \sin(\theta) \\ 1 & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} \quad (5.3)$$

As a general rule we will have

$$\mathbf{W} = \mathbf{H}\dot{\mathbf{X}} \quad \dot{\mathbf{X}} = \mathbf{H}^{-1}\mathbf{W} \quad (5.4)$$

Clearly equation (5.2) is not unique and will vary according to the choice for \mathbf{X} and Θ (see for example (634) for a relation that involves only $\dot{\mathbf{X}}, \dot{\Theta}_p$). If we use the MKS, then equation (5.2) is

$$\mathbf{A}\dot{\Theta}_a + \mathbf{B}\dot{\mathbf{X}} = 0, \quad (5.5)$$

This equation relates the end-effector cartesian velocities and the derivatives of the orientation representations to the actuated joint velocities. \mathbf{A}, \mathbf{B} are both $n \times n$ square matrices, and provided that \mathbf{A} is invertible we get:

$$\dot{\Theta}_a = -\mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{X}} = \mathbf{J}^{-1}\dot{\mathbf{X}},$$

where \mathbf{J}^{-1} is a matrix that will be called an *inverse jacobian matrix*. If the end-effector has 3 rotational d.o.f. and we are using the Euler angles to represent its orientation, \mathbf{J}^{-1} is a matrix \mathbf{J}_e^{-1} that we will call the *Euler angles inverse jacobian matrix*.

The matrix relating the end-effector velocity vector to the actuated joint variables, that defines the velocity linear *input-output* equations, will

be called the *inverse kinematic jacobian* matrix J_k^{-1} . The inverse kinematic jacobian matrix is essential for the velocity and trajectory control of parallel robots. Using (5.4) we get

$$\mathbf{A}\dot{\Theta}_a + \mathbf{B}\mathbf{H}^{-1}\mathbf{W} + \mathbf{C}\dot{\Theta}_p = 0, \quad (5.6)$$

As noticed by Gosselin and Angeles (189), if we use the MKS we get

$$\mathbf{A}\dot{\Theta}_a + \mathbf{B}\mathbf{H}^{-1}\mathbf{W} = 0, \quad (5.7)$$

Consequently

$$J_k^{-1} = -\mathbf{A}^{-1}\mathbf{B}\mathbf{H}^{-1} = \mathbf{J}^{-1}\mathbf{H}^{-1} \quad (5.8)$$

As \mathbf{H} is a nonsingular matrix, J_k^{-1} , \mathbf{J}^{-1} will be singular or nonsingular at the same time. Equation (5.7) is usually written as:

$$\mathbf{A}_k\dot{\Theta}_a + \mathbf{B}_k\mathbf{W} = 0, \quad (5.9)$$

Note that in general the inverse kinematic jacobian is not a jacobian matrix in the strict mathematical sense of the term, since there is no representation of the orientation of a rigid body, the derivative of which with respect to time corresponds to the rigid body angular velocities.

Another point is that for a manipulator with $n < 6$ d.o.f. it may sound interesting to determine only the $n \times n$ inverse kinematic jacobian that relates the actuated joint velocities to the possible d.o.f. velocities. We will see however in the next chapter that determining an inverse kinematic jacobian that involves $\dot{\Theta}_a$ and the *full twist* of the end-effector may be essential for singularity analysis. This type of matrix is coined a *overall jacobian* by Joshi (293) but we will call it a *full inverse kinematic jacobian*.

5.2.1. EULER ANGLES INVERSE JACOBIAN

As seen previously a simple derivation allows us to calculate the inverse jacobian matrix for a representation of the orientation by the Euler angles, as shown in the following example.

5.2.1.1 Example: 6-UPS manipulator

The square of the leg lengths ρ may be obtained as

$$\|\mathbf{AO}\|^2 + \|\mathbf{CB}_r\|^2 + 2((\mathbf{AO} + \mathbf{RCB}_r) \cdot \mathbf{OC} + \mathbf{AO} \cdot \mathbf{RCB}_r) + \|\mathbf{OC}\|^2 \quad (5.10)$$

Matrix \mathbf{A} is therefore 2 times the diagonal matrix in which the diagonal elements are the ρ_i . The derivatives of the right side of (5.10) with respect to (x_c, y_c, z_c) are the components of $2(\mathbf{AO} + \mathbf{RCB}_r) + 2\mathbf{OC} = 2\mathbf{AB}$. Hence the 3 first components of a row of the matrix \mathbf{B} are the components of \mathbf{AB} .

For the orientation part, the term of the equation (5.10) containing the orientation parameters is $2(\mathbf{OC} + \mathbf{AO})\mathbf{RCB}_r$. If \mathbf{Q} represents the derivative of \mathbf{RCB}_r with respect to one of the Euler angles, then the corresponding component of \mathbf{B} is $2\mathbf{AC}\cdot\mathbf{Q}$. Consequently a row of the Euler inverse jacobian matrix is:

$$\left(\frac{\mathbf{AB}}{\rho}, \frac{\mathbf{AC}\cdot\mathbf{Q}_\psi}{\rho}, \frac{\mathbf{AC}\cdot\mathbf{Q}_\theta}{\rho}, \frac{\mathbf{AC}\cdot\mathbf{Q}_\phi}{\rho} \right)$$

This matrix can be easily computed, especially if the inverse kinematics has been computed beforehand.

5.2.2. INVERSE KINEMATIC JACOBIAN

We have already seen a method for deriving the inverse kinematic jacobian from the matrices \mathbf{A} , \mathbf{B} of the MKS. There are other ways to derive the inverse kinematic jacobian (see for example the "Static" chapter or the use of Grassmann-Cayley algebra for that purpose (142; 549)). In this section we derive the inverse kinematic jacobian from a velocity analysis. Consider the velocity of an attachment point B_i of a given leg on the end-effector. It may be expressed according to the velocity of C and the angular velocity vector $\boldsymbol{\Omega}$ by

$$\mathbf{V}_{B_i} = \mathbf{V}_C + \mathbf{BC} \times \boldsymbol{\Omega} \quad (5.11)$$

In matrix form this equation may be written as a function of the velocity vector of the end-effector \mathbf{W} as

$$\mathbf{V}_{B_i} = \mathbf{J}_{X_i} \mathbf{W}$$

Now let $\boldsymbol{\omega}_i$ be the joint (actuated and passive) velocity vector of leg i . The velocity of B_i may also be expressed as

$$\mathbf{V}_{B_i} = \mathbf{J}_{\Theta_i} \boldsymbol{\omega}_i$$

Equating the two previous equations leads to:

$$\mathbf{J}_{X_i} \mathbf{W} = \mathbf{J}_{\Theta_i} \boldsymbol{\omega}_i \quad (5.12)$$

Combining these equations for all legs leads to an inverse jacobian that relates linearly the velocity vector of the end-effector \mathbf{W} to the velocities of all joint, actuated or not. If we are interested only in the inverse kinematic jacobian, we must eliminate the passive joint velocities from equation (5.12). As the equations (5.12) are linear, this may be done by linear elimination, but the process may be tedious. The concept of *reciprocal screw* may be of some help. This idea, pioneered for parallel robots by Mohamed (425)*, has been put in use for many robots (see for example (293)). The first step of the

method is to write the equation that states that the instantaneous twist of the end-effector is a linear combination of the instantaneous twists of the legs. The second step is to identify, for a leg with g actuated joints, g screws that are reciprocal to the screw system associated with all the passive joints of the legs. The orthogonal product of these screws with both sides of the equation obtained at step 1, leads to a set of g equations that are free from the velocities of the passive joints. Combining these equations for all legs allows one to get the inverse kinematic jacobian.

The difficult step of this method is the second one. A profound knowledge of reciprocal screws is needed to complete this step and we may still need some mechanical intuition. Also, dealing only with equations may hide the geometrical interpretation of the inverse kinematic jacobian (see for example (587)) that will be quite important for singularity analysis.

We will present now the calculation of the full inverse kinematic jacobian matrices for various examples.

5.2.2.1 Example: planar 3-RPR manipulator

Let θ be the rotation angle of the platform around C . Although the robot is planar, we will use 3D vectors. Let us define $\boldsymbol{\Omega}$ as $(0, 0, \dot{\theta})$ and \mathbf{n}_i as the unit vectors of the legs. The velocity \mathbf{V}_B of the point B is

$$\mathbf{V}_{B_i} = \mathbf{V} + \mathbf{B}_i \mathbf{C} \times \boldsymbol{\Omega} \quad \mathbf{V}_{B_i} = \dot{\rho}_i \mathbf{n}_i$$

Equating the dot product by \mathbf{n}_i of the right terms of these equations leads to

$$\dot{\rho}_i = \mathbf{n}_i \cdot \mathbf{V} + (\mathbf{C} \mathbf{B}_i \times \mathbf{n}_i) \cdot \boldsymbol{\Omega} \quad (5.13)$$

The above equation allows us to compute a row of the 3×3 inverse kinematic jacobian matrix relating the joint velocities to the twist $v_x, v_y, \dot{\theta}$. It may also be used to calculate rows of a 6×6 full inverse kinematic jacobian matrix. The other rows are obtained from the constraint equations indicating that the robot motion are planar

$$\mathbf{V} \cdot \mathbf{z} = 0 \quad \boldsymbol{\Omega} \cdot \mathbf{y} = 0 \quad \boldsymbol{\Omega} \cdot \mathbf{x} = 0$$

The full velocity equations may therefore be written as:

$$\begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{J}^{-1} \mathbf{W} = \begin{pmatrix} \mathbf{n}_1 & \mathbf{C} \mathbf{B}_1 \times \mathbf{n}_1 \\ \mathbf{n}_2 & \mathbf{C} \mathbf{B}_2 \times \mathbf{n}_2 \\ \mathbf{n}_3 & \mathbf{C} \mathbf{B}_3 \times \mathbf{n}_3 \\ \mathbf{z} & \mathbf{0} \\ \mathbf{0} & \mathbf{x} \\ \mathbf{0} & \mathbf{y} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \quad (5.14)$$

which establish a full inverse kinematic jacobian.

5.2.2.2 Example: 3 – UPU manipulator

The inverse kinematic equations giving the square ρ^2 of the link lengths are established as:

$$\rho^2 = \|\mathbf{OC}\|^2 + 2(\mathbf{AO} + \mathbf{CB}) \cdot \mathbf{OC} + (\mathbf{AO} + \mathbf{CB})^2 . \quad (5.15)$$

By differentiating this equation we obtain

$$\rho \dot{\rho} = \mathbf{V} \cdot \mathbf{AB} , \quad (5.16)$$

A row of the 3×3 inverse kinematic jacobian matrix is therefore obtained as $\mathbf{A}_i \mathbf{B}_i / \rho_i$. Let us now calculate the full inverse kinematic jacobian matrix. We start by calculating the velocity \mathbf{V}_B of the B points.

$$\mathbf{V}_B = \mathbf{V} + \mathbf{BC} \times \boldsymbol{\Omega} \quad (5.17)$$

Let us define \mathbf{n} as the unit vector of the leg, and compute the dot product of the right and left terms of the previous equation:

$$\mathbf{V}_B \cdot \mathbf{n} = \dot{\rho} \mathbf{n} = \mathbf{V} \cdot \mathbf{n} + (\mathbf{BC} \times \boldsymbol{\Omega}) \cdot \mathbf{n} = \mathbf{V} \cdot \mathbf{n} + (\mathbf{CB} \times \mathbf{n}) \cdot \boldsymbol{\Omega} \quad (5.18)$$

Now let us define $\mathbf{u}_i, \mathbf{v}_i$ the unit vectors of the two joint axis of the U joint. These vectors are the same for the base and platform. The angular velocity of the leg $\boldsymbol{\omega}_l$ with respect to the base is

$$\boldsymbol{\omega}_l = \dot{\theta}_A^i \mathbf{u}_i + \dot{\alpha}_A^i \mathbf{v}_i,$$

while the angular velocity of the platform $\boldsymbol{\omega}_p$ with respect to the leg is

$$\boldsymbol{\omega}_p = \dot{\theta}_B^i \mathbf{u}_i + \dot{\alpha}_B^i \mathbf{v}_i.$$

The angular velocity of the platform is

$$\boldsymbol{\Omega} = \boldsymbol{\omega}_l + \boldsymbol{\omega}_p = K_1^i \mathbf{u}_i + K_2^i \mathbf{v}_i,$$

where K_1^i, K_2^i can be obtained from the previous equations. Now define $\mathbf{s}_i = \mathbf{u}_i \times \mathbf{v}_i$, and compute the dot product of the right and left terms of the previous equation by \mathbf{s}_i :

$$\mathbf{s}_i \cdot \boldsymbol{\Omega} = 0 \quad (5.19)$$

The twist \mathbf{W} of the platform and the joint velocities vector $\dot{\boldsymbol{\Theta}}$ may be defined as

$$\mathbf{W} = (v_x, v_y, v_z, \Omega_x, \Omega_y, \Omega_z) \quad \dot{\boldsymbol{\Theta}} = (\dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, 0, 0, 0)$$

Combining equations (5.18, 5.19) we get the full velocities equations as

$$\begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{J}^{-1}\mathbf{W} = \begin{pmatrix} \mathbf{n}_1 & (\mathbf{CB}_1 \times \mathbf{n}_1) \\ \mathbf{n}_2 & (\mathbf{CB}_2 \times \mathbf{n}_2) \\ \mathbf{n}_3 & (\mathbf{CB}_3 \times \mathbf{n}_3) \\ \mathbf{0} & \mathbf{s}_1 \\ \mathbf{0} & \mathbf{s}_2 \\ \mathbf{0} & \mathbf{s}_3 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \quad (5.20)$$

which establishes the full inverse kinematic jacobian.

5.2.2.3 Example: 3 – PUS rotational wrist

We consider the 3 degrees of freedom rotational wrist of type 3-PUS shown in figure 2.11, and whose kinematics is presented in figure 5.1.

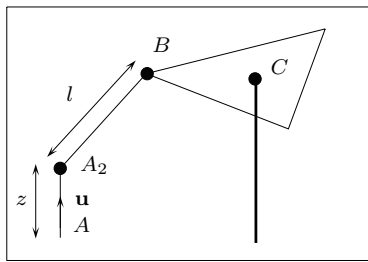


Figure 5.1. One of the links of a 3 degrees of freedom wrist in rotation

With the notation of figure 5.1, the velocity \mathbf{V}_B of point B and \mathbf{V}_{A_2} of point A_2 are:

$$\mathbf{V}_B = \mathbf{BC} \times \boldsymbol{\Omega}, \quad \mathbf{V}_{A_2} = \dot{\lambda} \mathbf{u}$$

The length l of leg A_2, B is constant and leads to

$$\|\mathbf{A}_2\mathbf{B}\|^2 = l^2 \quad (5.21)$$

Differentiating this equations gives

$$(\mathbf{V}_B - \mathbf{V}_{A_2}) \cdot \mathbf{A}_2\mathbf{B} = (\mathbf{BC} \times \boldsymbol{\Omega} - \dot{\lambda} \mathbf{u}) \cdot \mathbf{A}_2\mathbf{B} = 0$$

Hence we get

$$\dot{\lambda} = \frac{(\mathbf{CB} \times \mathbf{A}_2\mathbf{B}) \cdot \boldsymbol{\Omega}}{\mathbf{u} \cdot \mathbf{A}_2\mathbf{B}} \quad (5.22)$$

which gives a row of the 3×3 inverse kinematic jacobian matrix. Let us now calculate the full inverse kinematic jacobian matrix. The velocity \mathbf{V}_B

of the B points is established in equation (5.17). Differentiating equation (5.21) leads to

$$\dot{\lambda} = \frac{\mathbf{A}_2\mathbf{B}.\mathbf{V} + (\mathbf{CB} \times \mathbf{A}_2\mathbf{B}).\boldsymbol{\Omega}}{\mathbf{u}.\mathbf{A}_2\mathbf{B}} \quad (5.23)$$

If we define the joint velocities vector $\dot{\boldsymbol{\Theta}} = (\dot{\lambda}_1, \dot{\lambda}_2, \dot{\lambda}_3, 0, 0, 0)$, equation (5.23) leads to the first three rows of the full inverse kinematic jacobian:

$$J_i^{-1} = \left(\left(\frac{\mathbf{A}_2^i\mathbf{B}_i}{\mathbf{u}_i.\mathbf{A}_2^i\mathbf{B}_i} \quad \frac{(\mathbf{CB} \times \mathbf{A}_2\mathbf{B})}{\mathbf{u}.\mathbf{A}_2\mathbf{B}} \right) \right) \quad i \in [1, 3]$$

We now consider the ball-and-socket joint at C ; that is equivalent to having three R joints with axes meeting at C and whose unit vectors \mathbf{n}_i are linearly independent. The constraint equation is $\mathbf{V}.\mathbf{n}_i = 0$, so the three last rows of the full inverse kinematic jacobian are

$$J_i^{-1} = ((\mathbf{n}_i \quad \mathbf{0})) \quad i \in [4, 6]$$

5.2.2.4 Example: 6-UPS manipulator

We consider the case of the 6-UPS robot, the inverse kinematics being defined by equation (5.24):

$$\rho^2 = \|\mathbf{AO}\|^2 + \|\mathbf{CB}_r\|^2 + 2(\mathbf{AO} + \mathbf{RCB}_r).\mathbf{OC} + 2\mathbf{AO}.\mathbf{RCB}_r + \|\mathbf{OC}\|^2 . \quad (5.24)$$

By differentiating this equation we obtain

$$\rho\dot{\rho} = \mathbf{V}_C.\mathbf{AB} + \dot{\mathbf{C}}\mathbf{B}.\mathbf{AC} , \quad (5.25)$$

we have also

$$\dot{\mathbf{C}}\mathbf{B} = \mathbf{BC} \times \boldsymbol{\Omega} . \quad (5.26)$$

If \mathbf{n}_i denotes the unit vector of link i , we have

$$\mathbf{n}_i = \frac{\mathbf{AB}}{\|\mathbf{AB}\|} = \frac{\mathbf{AB}}{\rho} . \quad (5.27)$$

Using these results, we may write equation (5.25) as

$$\dot{\rho} = \mathbf{V}_C.\mathbf{n}_i + (\mathbf{BC} \times \boldsymbol{\Omega}).\frac{\mathbf{AC}}{\rho} ; \quad (5.28)$$

this can be written as

$$\dot{\rho} = \mathbf{V}_C.\mathbf{n}_i + \boldsymbol{\Omega}.\left(\frac{\mathbf{AC}}{\rho} \times \mathbf{BC}\right) = \mathbf{V}_C.\mathbf{n}_i + \boldsymbol{\Omega}.\left(\mathbf{n}_i \times \mathbf{BC}\right) . \quad (5.29)$$

One row of the inverse kinematic jacobian is therefore:

$$[\mathbf{n}_i \quad , \quad (\mathbf{n}_i \times \mathbf{B}_i \mathbf{C})] \quad . \quad (5.30)$$

As mentioned previously, the inverse jacobian matrix is dependent upon the choice of the operating point. We may for example attach the reference frame to the end-effector, and choose O as operating point (542). In that case a row of the inverse kinematic jacobian matrix may be defined as:

$$[\mathbf{n}_i \quad , \quad (\mathbf{n}_i \times \mathbf{A}_i \mathbf{O})] \quad . \quad (5.31)$$

5.2.2.5 Example: 6-PUS manipulator

Using the inverse kinematics equations, we can write

$$\rho_i^2 = \lambda_i^2 - 2\lambda_i \mathbf{u} \cdot \mathbf{A}_{i0} \mathbf{B}_i + \|\mathbf{A}_{i0} \mathbf{B}_i\|^2 \quad . \quad (5.32)$$

Differentiating this equation, we obtain

$$\dot{\lambda}_i \lambda_i - \lambda_i \mathbf{u} \cdot \mathbf{V}_{\mathbf{B}_i} - \dot{\lambda}_i \mathbf{u} \cdot \mathbf{A}_{i0} \mathbf{B}_i + \mathbf{A}_{i0} \mathbf{B}_i \cdot \mathbf{V}_{\mathbf{B}_i} = 0 \quad , \quad (5.33)$$

where $\mathbf{V}_{\mathbf{B}_i}$ represents the velocity of point B_i , which can be written as

$$\mathbf{V}_{\mathbf{B}_i} = \mathbf{V}_C + \mathbf{B}_i \mathbf{C} \times \boldsymbol{\Omega} \quad . \quad (5.34)$$

Grouping the terms allows us to obtain

$$-\dot{\lambda}_i \mathbf{u} \cdot \mathbf{A}_i \mathbf{B}_i + \mathbf{A}_i \mathbf{B}_i \cdot \mathbf{V}_C + (\mathbf{A}_i \mathbf{B}_i \times \mathbf{B}_i \mathbf{C}) \cdot \boldsymbol{\Omega} = 0 \quad , \quad (5.35)$$

which leads to

$$\dot{\lambda}_i = \frac{\mathbf{A}_i \mathbf{B}_i}{\mathbf{u} \cdot \mathbf{A}_i \mathbf{B}_i} \cdot \mathbf{V}_C + \frac{(\mathbf{A}_i \mathbf{B}_i \times \mathbf{B}_i \mathbf{C})}{\mathbf{u} \cdot \mathbf{A}_i \mathbf{B}_i} \cdot \boldsymbol{\Omega} \quad . \quad (5.36)$$

5.2.3. INVERSE JACOBIAN AND PLÜCKER LINE COORDINATES

Let us consider a line going through points M_1, M_2 , and define the 6-dimensional Plücker vector \mathbf{P} as

$$\mathbf{P} = (\mathbf{M}_1 \mathbf{M}_2, \mathbf{O} \mathbf{M}_1 \times \mathbf{M}_1 \mathbf{M}_2)$$

This vector uniquely defines, up to a multiplicity factor, the line going through M_1, M_2 . The normalized Plücker vector \mathbf{P}_n is defined as

$$\mathbf{P}_n = \left(\frac{\mathbf{M}_1 \mathbf{M}_2}{\|\mathbf{M}_1 \mathbf{M}_2\|}, \frac{\mathbf{O} \mathbf{M}_1 \times \mathbf{M}_1 \mathbf{M}_2}{\|\mathbf{M}_1 \mathbf{M}_2\|} \right)$$

Let us decompose \mathbf{P}_1 into two 3-dimensional vectors $\mathbf{P}_1 = (\mathbf{p}_1, \mathbf{q}_1)$. A necessary condition for \mathbf{P}_1 describing a line is that $\mathbf{p}_1 \cdot \mathbf{q}_1 = 0$. Consider now another Plücker vector \mathbf{P}_2 ; an interesting property is that the line associated with \mathbf{P}_2 intersects the line associated with \mathbf{P}_1 if and only if $\mathbf{p}_1 \cdot \mathbf{q}_2 + \mathbf{q}_1 \cdot \mathbf{p}_2 = 0$. A property of Plücker vectors will be of interest: let us define two arbitrary non zero vectors \mathbf{u}, \mathbf{v} ; if $\mathbf{P}_\mathbf{v}$ is defined as $(\mathbf{p}_1, \mathbf{u} \times \mathbf{p}_1)$, then $\mathbf{P}_\mathbf{v} = (\mathbf{p}_1, \mathbf{v} \times \mathbf{p}_1)$ represents the same line $\mathbf{P}_\mathbf{u}$. We also note that the components of a Plücker vector are not homogeneous in term of units.

If we now look at the full inverse kinematic jacobians established in the previous example, we may see that all of them involve Plücker vectors. More precisely, the Plücker vector of the link connecting the end-effector to the remaining of the leg appears in all the inverse jacobians. We will see in the next chapter that the presence of such vectors in the inverse jacobian plays an important role in singularity analysis.

Hence we may wonder if such vector will appear in the inverse jacobian of any parallel robot. In the general case this has not been formally proved but it is easy to show that this will be true for a large class of spatial and planar manipulators (215):

Theorem: *for any spatial (planar) robot whose legs are attached to the end-effector through a ball-and-socket joint (revolute), the inverse jacobian will involve the Plücker vector of a line associated to the link connecting the leg to the end-effector.*

5.3. Jacobian matrix

Although the inverse jacobian established in the previous sections does not have a very high complexity, the direct analytical determination of the jacobian by inversion of \mathbf{J}^{-1} will be difficult, even with the help of symbolic computation, except in some particular cases (60)*, (165). For planar robots, this inversion is possible, and the jacobian matrix could even be obtained directly by differentiation of the closure equations, as shown by Pennock (469). However, this direct method seems difficult to apply to robots having more than 3 d.o.f. Theoretical analytic formulations of jacobians have been proposed (149; 312) but require complicated matrix inversions.

In the general case, the difficulty of the inversion does not lie in the complexity of the algorithm but in the sheer size of the result. For example, if we start with the calculation of the determinant of the inverse kinematic jacobian matrix for a SSM, we will find that it involves 7488 multiplications, 998 additions and 26 power terms, and to determine \mathbf{J} we will have to perform the same calculation for the 36 minors. For the general 6-UPS robot, even though Mayer St-Onge (391) showed that the computation of

the inverse jacobian determinant is possible, its expression is very large. We shall therefore not give any analytical formulation for jacobian matrices.

The general form of the determinant may be of interest for the singularity analysis as we will see in the next chapter. For the SSM we have

$$|J^{-1}| = A_{z3}z_c^3 + z_c^2(A_{z2} + A_{yz2}y_c + A_{xz2}x_c) + A_{x2}x_c^2 + A_{y2}y_c^2 + A_{yz}z_cy_c + A_{xz}x_cz_c + A_{xy}x_cy_c + A_{xyz}x_cz_cy_c + A_zz_c + A_yy_c + A_xx_c + A_{\psi,\theta,\phi}$$

where the coefficients A only depend on the orientation. Even when we can obtain an analytic expression of the jacobian matrices, a numerical evaluation based on the symbolic expression will usually take a long time, especially for 6 d.o.f. robots, and numerical precision will sometimes be insufficient. Practically, the jacobian matrix will be obtained by relying on a numerical inversion method that will be faster and, in general, less sensitive to round-off errors.

5.4. Kinetostatic performance indices

5.4.1. MANIPULABILITY AND THE KINEMATICS POLYHEDRON

The inverse kinematic jacobian matrix also allows us to establish a linear relation between the manipulator accuracy $\Delta \mathbf{X}$ and the measurement errors $\Delta \Theta$ on Θ . Clearly it is interesting to quantify the influence of the joint measurement errors on the positioning errors of the end-effector, and this is the purpose of the kinetostatic performance indices. In practice we may assume that the measurement errors are bounded i.e. that they satisfy

$$\|\Delta \Theta\| \leq 1 \tag{5.37}$$

If we assume that the Euclidean norm is used in this equation, then

$$\|\Delta \Theta\| \leq 1 \Rightarrow \Delta \mathbf{X}^T J_k^{-T} J_k^{-1} \Delta \mathbf{X} \leq 1 \tag{5.38}$$

Equations (5.37, 5.38) establish that the hyper-sphere in the joint error space is mapped into an *ellipsoid* in the generalized Cartesian error space; this is often called the *manipulability ellipsoid*. It is usually admitted that the shape and volume of this ellipsoid characterizes the manipulator dexterity. Indeed if the ellipsoid has an axis that is very large, then there will be a large amplification factor between the errors in the actuated joint measurements errors and the positioning error for the end-effector for a given combination of sensor errors.

The validity of the concept of manipulability ellipsoid may be questioned. Indeed the use of the Euclidean norm in (5.37) implies that the joints measurement errors are not independent. For example, if one of the

joint measurement error is 1, then, by some mysterious influence, the other measurement errors should all become simultaneously 0 . . . A more realistic norm will be the infinity norm, the maximal absolute value of the vector components. This will indicate that the errors are independent, but that their maximal absolute value is bounded, which corresponds to the real situation. With this norm, the joint errors are restricted to lie in a hyper-cube in the joint error space. We get

$$-1 \leq \sum_{j=1}^{j=n} J_{ij}^{-1} \Delta X_j \leq 1 \quad i \in [1, n] \quad (5.39)$$

Each of these inequalities defines a region in the positioning error space that is bounded by two hyper-planes; the intersection of these regions is a convex polyhedron that we will call the *kinematic polyhedron*. The hyper-cube in the joint error space is thus mapped into the kinematic polyhedron in the positioning errors space. This kinematic polyhedron will include the manipulability ellipsoid, but is larger than this ellipsoid.

The kinematic polyhedron may easily be calculated by geometrical techniques. The vertices of this polyhedron are the twists obtained by considering all combinations of extremal joint velocities; there are 2^n such combinations. An example is given for the INRIA left hand in figure 5.2, in which we have assumed that 4 elements of the twist have a 0 value. Pierrot (477)

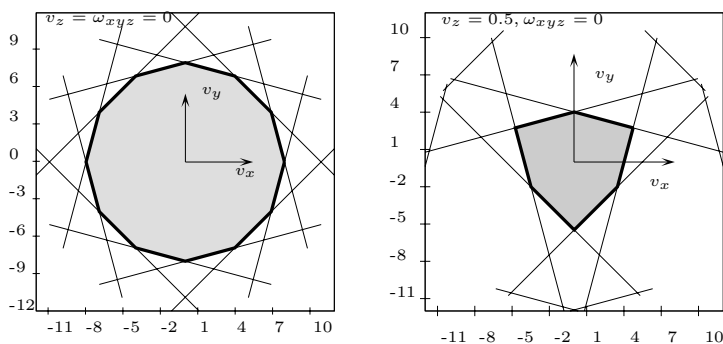


Figure 5.2. Allowed velocities region (in gray) in the (v_x, v_y) plane for INRIA left hand when it is in its nominal position.

addressed this problem for redundant robots. It must be noted that, apart from being more realistic, the previous mapping leads to a geometrical object that can be more easily manipulated than the ellipsoid. For example, assume that one wants to determine what are all the possible end-effector twists that can be obtained in 2 different poses of the end-effector. For

that purpose we will have to calculate the intersection of the 2 kinematic polyhedra obtained for the 2 poses, a well known problem of computational geometry that can be more easily solved than computing the intersection of 2 ellipsoids.

5.4.2. CONDITION NUMBER AND OTHER INDICES

5.4.2.1 Manipulability index and condition number

We will now formalize kinetostatic performance indices that allow us to quantify the dexterity of a robot. Such indices have been defined since a long time for serial robots as, for example, the Yoshikawa's manipulability index $\sqrt{|\mathbf{J}\mathbf{J}^T|}$. As these indices are well known, we will focus on the *condition number* that is often used for parallel robots. Consider the linear system:

$$\mathbf{A}\mathbf{X} = \mathbf{B}$$

for a norm we have

$$\|\mathbf{A}\mathbf{X}\| = \|\mathbf{B}\| \leq \|\mathbf{A}\| \|\mathbf{X}\| \quad \|\mathbf{A}^{-1}\Delta\mathbf{B}\| = \|\Delta\mathbf{X}\| \leq \|\mathbf{A}^{-1}\| \|\Delta\mathbf{B}\|$$

From which we get

$$\frac{\|\Delta\mathbf{X}\|}{\|\mathbf{X}\|} \leq \frac{\|\mathbf{A}^{-1}\| \|\Delta\mathbf{B}\|}{\|\mathbf{X}\|} \leq \frac{\|\mathbf{A}\| \|\mathbf{A}^{-1}\| \|\Delta\mathbf{B}\|}{\|\mathbf{B}\|}$$

This equation indicates how a *relative* error in \mathbf{B} gets multiplied, and leads to a *relative* error in \mathbf{X} . The error amplification factor, called the *condition number* κ , is therefore defined as

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| .$$

For \mathbf{J}_k^{-1} , the condition number expresses how a relative error in Θ gets multiplied and leads to a relative error in \mathbf{X} . It characterizes in some sense the dexterity of the robot, and will be used as a performance index. The condition number is dependent on the choice of the matrix norm. The most used norms are as follows:

- the 2-norm, defined as the square root of the largest eigenvalue of matrix $\mathbf{J}^{-T}\mathbf{J}^{-1}$: the condition number of \mathbf{J}^{-1} is thus the square root of the ratio between the largest and the smallest eigenvalues of $\mathbf{J}^{-T}\mathbf{J}^{-1}$,
- the Euclidean (or Frobenius) norm defined for the $m \times n$ matrix \mathbf{A} by $\|\mathbf{A}\| = \sqrt{\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} |a_{ij}|^2}$, or equivalently as $\sqrt{\text{tr}(\mathbf{A}^T\mathbf{A})}$; if λ_i denotes the eigenvalues of $\mathbf{J}^{-T}\mathbf{J}^{-1}$, then the condition number is the ratio between $\sum \lambda_i^2$ and $\prod \lambda_i$. Note that sometime is used a weighted norm, in which $\mathbf{A}^T\mathbf{A}$ is substituted by $\mathbf{A}^T\mathbf{W}\mathbf{A}$, where \mathbf{W} is a weight matrix

The smallest possible value of the condition number is 1. The inverse of the condition number, which has a value in $[0,1]$, is also often used; a value of 0 indicates that the inverse jacobian matrix is singular. Note that it is not possible to calculate an analytical form for the condition number as long as the robot has more than 4 d.o.f. It is theoretically possible for a robot with 3 or 4 d.o.f. by using the analytical formulation of the roots of the characteristic polynomial, but these formulations are highly unstable numerically. Therefore, in practice, it is in general better to use classical linear algebra software that offer a robust condition number calculation.

But there is major drawback to the condition number. For a robot having at least one translational and one rotational d.o.f., the inverse jacobian will be heterogeneous as far as units are concerned. For instance, for a 6-*UPS* robot, 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. Hence one has to be quite careful when using such an index as an optimality criteria for a parallel robot.

Various proposals have been made to avoid this drawback. Researchers such as Ma and Angeles (380) suggested dividing the rotational elements by a length, called the *characteristic length*, such as the length of the links in a nominal position, or the *natural length*, defined as that which minimizes the condition number for a given pose. Still the choice of the length remains arbitrary as it just allows us to define a correspondence between a rotation and a translation. As mentioned by Park (467), "this arbitrariness is an unavoidable consequence of the geometry of $SE(3)$ " (see also (548) for a thorough discussion on the validity of manipulability indices). Angeles (13) recognizes that point: he shows that the error when approximating a rigid-body displacement by a rotation in a four-dimensional space, the translation being normalized by the characteristic length, is monotonically decreasing with the characteristic length, and hence no minimum value of the error may be found.

Gosselin (191) defines a new inverse jacobian that transforms the linear velocities of two points on the end-effector into actuator velocities. Later on Kim (318), proposes using for a 6-*UPS* robot, the inverse jacobian obtained from the inverse kinematics equations based on the coordinates of 3 points of the end-effector (see section 4.6.1.2). In that case the inverse kinematics consists of 9 equations, but only 6 involve the joint variables. These equations are used to derive a 6×9 inverse jacobian J^{-1} , whose rank is usually 6. Although this inverse jacobian matrix is homogeneous in terms of units, the condition number will no longer describe the intrinsic behavior of the robot as its value will depend on the choice of points on

the end-effector.

It is also mentioned sometimes that two condition numbers should be computed to avoid the unit discrepancy: one for the restriction of the jacobian to the translation motion of the end-effector, and one for its restriction to the orientations. The experimental data presented in the next section seems to indicate that this is not always a valid approach.

As far as control is concerned, it is better to have a condition number for J^{-1} that is as small as possible. This ensures that the errors on the actuated joints affect the moving platform position as little as possible. Note however that the condition number indicates a bound on the *relative* error and not the largest positioning error. For example a cartesian X-Y table robot has the identity as inverse jacobian matrix, with 1 as condition number, but the largest positioning error is $\sqrt{2}$.

5.4.2.2 *Validity of the condition number*

The condition number is mentioned in the literature as the main index for characterizing the accuracy of parallel robots. We will show on a simple example that such statement may be questioned.

Consider the 6-UPS robot whose geometry is defined by the coordinates of the attachment point presented in table 5.1, and which are close to the INRIA *left hand*.

joint	x_a	y_a	z_a	x_b	y_b	z_b
1	-12.758	3.902	0.0	-7.8218	-1.052	0.0
2	-9.758	9.098	0.0	-3	7.3	0.0
3	-3	-13	0.0	-4.8218	-6.248	0.0
4	3	-13	0.0	4.8218	-6.248	0.0
5	9.758	9.098	0.0	3	7.3	0.0
6	12.758	3.902	0.0	7.8218	-1.052	0

TABLE 5.1. Coordinates of the attachment points in cm.

We consider three poses of this robot, defined by $P_1=x_c = y_c = 0, z_c=53$ cm, $\psi = 0, \theta = 0, \phi = 0$, $P_2=x_c = y_c = 0, z_c=53$ cm, $\psi = 30^\circ, \theta = 0, \phi = 0$; and $P_3=x_c = y_c = 10, z_c=53$ cm, $\psi = 0, \theta = 0, \phi = 0$. The accuracy of the robot at these poses is characterized by the maximal errors $\Delta X_{x,y,z,\theta_x,\theta_y,\theta_z}$ obtained as the sum of the absolute values of the corresponding rows of J_k , multiplied by a nominal sensor accuracy (chosen here as $\pm 1/100$ mm). These values are presented in table 5.2; it can be seen that the positioning errors are significantly larger for P_2 and P_3 compared to P_1 . As for P_3 , the errors are usually large compared to P_2 , except for the rotation around z . A reasonable ranking in term of accuracy is therefore $P_1 \gg P_2 > P_3$.

Pose	ΔX_x	ΔX_y	ΔX_z	ΔX_{θ_x}	ΔX_{θ_y}	ΔX_{θ_z}
P_1	0.1184	0.1268	0.010087	0.1185	0.1184	0.697
P_2	0.1189	0.1274	0.01266	0.1333	0.1429	0.808
P_3	0.123	0.1309	0.0372	0.15	0.1663	0.7208

TABLE 5.2. Maximal positioning errors at P_1, P_2, P_3 (mm and rad).

For this robot, we define the normalized inverse jacobian matrix J_n^{-1} obtained by dividing the orientation components of the J_k^{-1} by 53 i.e. roughly the legs lengths at pose P_1 . We will now examine the values of various kinetostatic indices at the 3 poses. The considered indices will be:

- C_d : the determinant of J_k^{-1}
- C_2, C_2^n : the 2-norm condition number of J_k^{-1}, J_n^{-1}
- C_F, C_F^n : the Frobenius-norm condition number of J_k^{-1}, J_n^{-1}
- C_2^3, C_F^3 : the 2-norm and Frobenius norm condition number of the inverse jacobian matrix obtained when the inverse kinematics equations are based on the coordinates of 3 points of the end-effector. The chosen points will be all possible triplets in the set B_i : hence we will provide ranges for these indices.
- M_t, M_o : the manipulability index of the restriction of J_k to its translation, orientation parts (387)

For all indices we expect to have an absolute value that is maximum for P_1 and is larger for P_2 than for P_3 and their values are presented in table 5.3, the values of M_t, M_o being (12.65, 0.004266), (20.45, 0.00754), (13.99, 0.00471) for P_1, P_2, P_3 .

	C_d	C_2	C_2^n	C_F	C_F^n	C_2^3	C_F^3
P_1	-29.219	75.14	63.927	152.84	70.167	[9.55,55.472]	[258.85,3204.91]
P_2	-24.644	75.162	73.847	154.048	80.938	[9.619,43.837]	[218.8,2383.58]
P_3	-23.928	80.646	68.42	158.3	74.70	[10.065,58.95]	[286.46,3617.96]

TABLE 5.3. Performance indices for the poses P_1, P_2, P_3

We may deduce interesting results from these values:

- C_d : this index is coherent with the maximal positioning errors
- C_2 : it may be seen that the difference is surprisingly very small between P_1, P_2 and significant between P_3, P_2 . This is not what we may expect from an accuracy index

- C_2^n : the accuracy ordering between P_2, P_3 is not respected
- C_F : the accuracy ordering is respected, but the change between P_1 and P_2 is relatively small
- C_F^n : the accuracy ordering between P_2, P_3 is not respected
- C_2^3, C_F^3 : for P_3 the condition number is either very close to the one of P_1 (C_2^3) or always larger. On the contrary, for P_2 , the condition number is in general significantly smaller than the condition number for P_1 , and sometime very close, but in all cases smaller, than the condition number for P_3 . This completely disqualifies these condition numbers as accuracy indices
- M_t, M_o : the manipulability indices of P_1, P_3 are close, while that for P_2 is significantly larger. These indices behave oppositely to what was expected and may be disqualified

Hence apart from the manipulability, none of these indices exhibits a consistent behavior with respect to the positioning errors of this robot. We cannot derive a general law from this single example, but it shows clearly that kinetostatic indices based on the kinematic jacobian have to be used with some precaution, especially when addressing optimal design for robots.

5.4.2.3 Isotropy

Poses with a condition number of 1 are called *isotropic poses*, and robots having only such type of poses (e.g. the X-Y table) are called *isotropic robots*. Designing a parallel robot that is isotropic in one pose or is isotropic over its full workspace is often considered as a design objective (11; 27; 73; 160; 584; 619).

The isotropy denomination is somewhat improper as, *stricto sensu*, it implies that the largest magnitude of the positioning errors will be the same for any extremal joint errors, which is not true. For example for the X-Y table, the largest positioning error has a value that varies in the range $[1, \sqrt{2}]$. It is even worse for redundant robots: for example Krut (334) exhibits a robot that is mechanically similar to a X-Y table but whose condition number, although constant, is not 1.

Still, isotropic configurations have an interest as they exhibit the most regular behavior of the positioning errors with respect to the joint errors. Hence, instead of using the name *isotropic robot* we may consider using the name *maximally regular robot*.

5.4.2.4 Global conditioning indices

Kinetostatic indices are, in general, dependent on the pose. Instead of considering the index \mathcal{I} in a specific pose, we may introduce a *global condition-*

ing index ν over the manipulator workspace W by

$$\nu = \frac{\int_W \mathcal{I} dW}{\int_W dW} .$$

as proposed by Gosselin (186) for the condition number. The global conditioning index for the condition number, GCI , is often mentioned by authors as a criterion that can be used for optimal design. A major problem is the calculation of the GCI , as it is almost never possible to calculate it exactly. It is indeed well known that the numerical evaluation of such a multi-dimensional integral is difficult. In our case, most authors just sample the workspace and average the values of the condition number obtained at the sampling poses, although this method does not allow us to determine an error bound on the result. It is sometimes assumed that if the result with m_1 sampling points is close to the result obtained with m_2 points, m_2 being significantly larger than m_1 , then the later result is a good approximation to ν . This assumption may be true if the condition number is smooth enough, a claim that is difficult to support (see exercise 5.8 for a simple counter-example).

Usual integration methods requires, for reaching a given level of accuracy, a number n of evaluations that grows exponentially with the space dimension. Hence a better evaluation will probably be obtained by using Monte-Carlo integration (with an error that decreases as $1/\sqrt{n}$). A certified evaluation of the global conditioning index is therefore an open problem but nevertheless the GCI calculation will probably be computer intensive.

Another global conditioning index is the *uniformity of manipulability* defined as the ratio of the minimum and maximum values of the manipulability index over a given workspace (474). It suffers from the same calculation problems as the GCI .

5.4.2.5 Other accuracy indices

At a given pose, and for a unit joint measurement error, the maximal positioning error (MPE) for the end-effector is obtained by taking the sum of the absolute values of the corresponding row of the jacobian. Apart from the condition number, it may be thought that a good kinetostatic performance index for a robot will be the largest MPE over a given workspace. A theoretical solution to this optimization problem is quite difficult as usually an analytical formulation of the jacobian matrix is not available. To obtain the largest MPE, researchers have proposed the same method as for computing the GCI , i.e. a sampling of the workspace. For instance Clavel (100) uses this approach for the *Delta* robot, and Patel (468) for the 6-*UPS* robot. But the approach is computer intensive and does not allow a robust calculation of the maximum, as it does not provide an error bound on the

result. We have recently proposed an algorithm based on interval analysis that is still computer intensive, but which allows one to calculate this maximum up to a pre-defined accuracy (418).

But computing the largest MPE over a given workspace is not sufficient for comparing two robots. Clearly, calculating the average and variance of the MPE over the workspace will be of interest, but unfortunately there is no known algorithm to compute them.

The above definitions of the kinetostatic indices do not take into account other factor affecting the accuracy of parallel robots, such as uncertainties in the location of the attachment points, or clearance and friction in the joints, an issue that is addressed in section 5.8.

5.5. Determination of the joint velocities and twist

The purpose of this section is to describe numerical procedures for calculating the actuated joint (generalized) velocities, being given the generalized (actuated joint) velocities.

5.5.1. DETERMINATION OF THE JOINT VELOCITIES

In the previous section, we showed that it is usually possible to determine analytically the inverse kinematic jacobian matrix that linearly relates the actuated joint velocities $\dot{\Theta}_a$ to the twist \mathbf{W} of the end-effector as:

$$\dot{\Theta}_a = J_k^{-1} \mathbf{W} \quad (5.40)$$

This equation allows one to compute the actuated joint velocities directly.

5.5.2. DETERMINATION OF THE TWIST

In a previous section we showed that it is usually difficult, even for robots with less than 6 d.o.f. robots, to invert J_k^{-1} analytically: a numerical procedure will therefore generally be used to calculate the twist from the joint velocities. For a given pose of the end-effector, we may, for example, use a numerical inversion algorithm to determine the jacobian from its inverse, or a linear equations solver to get \mathbf{W} from (5.40). Reboulet (499) suggests using the quasi-Newton scheme (see section 4.6.1):

$$\mathbf{W}_{k+1} = \mathbf{W}_k + J_0(\dot{\Theta}_a - J^{-1} \mathbf{W}_k) . \quad (5.41)$$

where 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.

Convergence and robustness of such a scheme have been discussed in section (4.6.1). Reboulet actually shows that it is convergent in all the

workspace of the robot he uses, and we have observed the same phenomenon for the robot defined in table 5.1. Numerical tests have shown that this algorithm is fast; on a DELL D400, 1.2 GHz, convergence occurs after 1 to 2 iterations, which leads to a computation time of about 0.11 ms, while it is established at 0.2 ms for a linear solver, and 0.4 ms for a matrix inversion.

5.6. Extrema of the velocities in a workspace

When designing a robot it may be interesting to determine what are the extrema of the end-effector twist velocities for fixed values of the extrema of the actuated joint velocities. We consider a n d.o.f. robot and suppose that the actuated joint velocities are bounded for all links, and that the bounds are the same for all joints. As the relations between the joint velocities and the twist are linear we may assume, without loss of generality, that

$$|\dot{\Theta}_i| \leq 1, \quad \forall i \in [1, n]. \quad (5.42)$$

5.6.1. EXTREMA OF THE TWIST

At a given pose, we define the *full independent velocity (FIV)* \dot{X}_i as

$$\dot{X}_i = \sum_{j=1}^{j=n} |J_{k_{ij}}|$$

For design purposes, it may be interesting to determine what are the maximal and minimal *FIV* \dot{X}_i^M, \dot{X}_i^m for all poses over a given workspace. For example the minimal *FIV* may be used to determine what should be the minimal actuator velocity so that the end-effector is able to reach a given velocity around/along a given axis for any pose. Note also that the maximal *FIV* allows one to determine the largest positioning error of the end-effector.

Such minimax optimization problems are in general difficult, but are more so for parallel robots, because the kinematic jacobian is usually not available. However we have proposed in a recently published paper (418) an interval analysis-based algorithm that allows for the calculation of \dot{X}_i^m, \dot{X}_i^M up to a given accuracy over almost any arbitrary workspace. This procedure is based on the possibility of computing an enclosure of all solutions in $\dot{\mathbf{X}}$ of the interval linear systems $\dot{\Theta}_a = J_{fk}^{-1} \dot{\mathbf{X}}$ where J_{fk}^{-1} is an interval matrix (see the interval appendix). Still, this algorithm is computer intensive, and calculating efficiently \dot{X}_i^m, \dot{X}_i^M is an open problem.

For parallel robots, we may also consider the maximal translational/angular velocity V_{max} at a given pose as the Euclidean norm of the corresponding components of the end-effector's twist. Determining the minimal and maximal values V_{max}^m, V_{max}^M of V_{max} for all poses in a given workspace is also

interesting for design purposes. It may be shown that if J_t is the restriction of the kinematic jacobian to its translation/rotation part, then V_{max}^2 is bounded by the sum of the squares of the singular values of J_t , multiplied by n (see exercise 5.9).

We have proposed in (409) an algorithm for a 6-*UPS* robot that determines V_{max}^m, V_{max}^M along a given direction for any pose within a translation workspace (i.e. the orientation of the end-effector is supposed to be constant). By-products of this algorithm are the extremal values of the angle between the link and a fixed direction, and therefore the maximal motion of the passive joints when C describes its workspace. But solving the general case is still an open problem.

5.6.2. EXTREMA OF THE JOINT VELOCITIES

We may be interested in determining what should be the minimal joint velocities so that a minimal translation/angular velocity V of the end-effector may be reached for any pose within a given workspace. This problem is basically equivalent to determining V_{max}^m and then scaling the joint velocities by the factor V_{max}^m , and is thus an open issue.

We have proposed in (408) an algorithm that allows one to compute the minimal value of the joint velocities for a 6-*UPS* robot so that a given translational velocity along a fixed direction may be obtained for any pose within a translation workspace.

5.7. Accelerations analysis

We will now determine what are the relations between the joint accelerations and the cartesian and angular accelerations of the end-effector. We note that parallel robots may present excellent characteristics as to acceleration: the *Delta* robot, for example, presents a maximal acceleration of about 500 m/s². General methods exist to obtain accelerations for closed-loop mechanisms (441) although for parallel robots it is generally easy to obtain these relations directly. Indeed, from equation (5.40) we obtain by differentiation

$$\ddot{\Theta} = J_k^{-1} \dot{\mathbf{W}} + J_k^{-1} \mathbf{W} . \quad (5.43)$$

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; the problem is more complex for redundant robots, see (453; 648). We will present some examples in the next sections.

5.7.1. 6-UPS ROBOT

In this case, one row of the inverse kinematic jacobian matrix is

$$\left[\frac{\mathbf{AB}}{\rho}, \frac{(\mathbf{CB} \times \mathbf{AB})}{\rho} \right]. \quad (5.44)$$

We first consider the first 3 elements of this row. We have

$$\frac{d\left(\frac{\mathbf{AB}}{\rho}\right)}{dt} = \frac{1}{\rho^2}(\rho\dot{\mathbf{A}}\mathbf{B} - \dot{\rho}\mathbf{AB}), \quad (5.45)$$

with

$$\dot{\mathbf{A}}\mathbf{B} = \mathbf{V} + \mathbf{CB} \times \boldsymbol{\Omega} \quad (5.46)$$

which allows us to complete the differentiation of these elements. For the last three elements, we have

$$\frac{d\left(\frac{\mathbf{CB} \times \mathbf{AB}}{\rho}\right)}{dt} = \frac{\rho \frac{d(\mathbf{CB} \times \mathbf{AB})}{dt} - \dot{\rho}(\mathbf{CB} \times \mathbf{AB})}{\rho^2}. \quad (5.47)$$

Moreover, we have

$$\frac{d(\mathbf{CB} \times \mathbf{AB})}{dt} = \dot{\mathbf{C}}\mathbf{B} \times \mathbf{AB} + \mathbf{CB} \times \dot{\mathbf{A}}\mathbf{B}, \quad (5.48)$$

and

$$\dot{\mathbf{C}}\mathbf{B} = \mathbf{CB} \times \boldsymbol{\Omega}; \quad (5.49)$$

thus:

$$\frac{d(\mathbf{CB} \times \mathbf{AB})}{dt} = (\mathbf{CB} \times \boldsymbol{\Omega}) \times \mathbf{AB} + \mathbf{CB} \times \mathbf{V}. \quad (5.50)$$

One row of the derivative of the inverse kinematic jacobian matrix is thus:

$$\frac{[\rho(\mathbf{V} + \mathbf{CB} \times \boldsymbol{\Omega}) - \dot{\rho}\mathbf{AB}, \rho((\mathbf{CB} \times \boldsymbol{\Omega}) \times \mathbf{AB} + \mathbf{CB} \times \mathbf{V}) - \dot{\rho}(\mathbf{CB} \times \mathbf{AB})]}{\rho^2} \quad (5.51)$$

We have considered here the inverse kinematics jacobian but we could consider as well another inverse jacobian. Simaan proves that there is an inverse jacobian whose derivative consists of Plücker coordinates of a line (542).

5.7.2. 6-PUS ROBOT

For this type of manipulator, the lengths of the links are fixed and the linear actuator moves the point A_i along a constant direction \mathbf{u} . In this case, one row of the inverse jacobian matrix is

$$\left[\frac{\mathbf{AB}}{\mathbf{AB} \cdot \mathbf{u}}, \frac{(\mathbf{CB} \times \mathbf{AB})}{\mathbf{AB} \cdot \mathbf{u}} \right]. \quad (5.52)$$

We have:

$$\frac{d\left(\frac{\mathbf{AB}}{\mathbf{AB}\cdot\mathbf{u}}\right)}{dt} = \frac{\dot{\mathbf{A}}\mathbf{B}(\mathbf{AB}\cdot\mathbf{u}) - \mathbf{AB}\frac{d(\mathbf{AB}\cdot\mathbf{u})}{dt}}{(\mathbf{AB}\cdot\mathbf{u})^2} \quad (5.53)$$

The previous differentiation is entirely determined by using the following relations:

$$\frac{d(\mathbf{AB}\cdot\mathbf{u})}{dt} = \dot{\mathbf{A}}\mathbf{B}\cdot\mathbf{u} , \quad (5.54)$$

$$\dot{\mathbf{A}}\mathbf{B} = \mathbf{V} + \mathbf{CB} \times \boldsymbol{\Omega} . \quad (5.55)$$

For the last three elements of the row, we have

$$\frac{d\left(\frac{\mathbf{CB}\times\mathbf{AB}}{\mathbf{AB}\cdot\mathbf{u}}\right)}{dt} = \frac{(\mathbf{AB}\cdot\mathbf{u})\frac{d(\mathbf{CB}\times\mathbf{AB})}{dt} - \frac{d(\mathbf{AB}\cdot\mathbf{u})}{dt}(\mathbf{CB}\times\mathbf{AB})}{(\mathbf{AB}\cdot\mathbf{u})^2} . \quad (5.56)$$

Using equations (5.54) and (5.50) we may conclude the differentiation of the row.

5.8. Accuracy analysis

Sources of positioning errors for parallel robots are diverse, and we will now mention the main one.

5.8.1. GEOMETRICAL ERRORS

For 6-*UPS* robots, Masory (386)* has studied the influence of not only the sensor errors but also the manufacturing tolerances on the locations of the joints centers. A thorough analysis has been proposed by Ehmann and co-workers (468; 606); this includes location errors of A, B , error in the leg lengths, and imperfect motion of the ball joints. Ehmann proposes a first and a second order error analysis, but shows that the first order is sufficient, although the differences between these two models increase when the robots become smaller. A software package written in Mathematica allows one to determine, for a given orientation of the platform, the workspace regions where the error is larger than a given threshold, to depict the error distribution in planar cross sections, and to perform a sensitivity analysis. Ehmann models are used by Jelenkovic (283) for an error analysis of 6-*UPS* and 6-*PUS* robots that is available on-line as a Web service[□] *AWE*.

Tischler (579) proposed a numerical approach for determining the influence of the backlash of the joints. The sensitivity to manufacturing tolerances of the 3-*UPU* robot has also been investigated in depth (212; 463; 626), while Parenti (465) and Wohlhart (623) propose an analysis of the effect of joint clearances on trajectories followed by serial and parallel robots.

A general approach to evaluating the positioning errors in one pose has been proposed by Pott (484). It relies however on a numerical evaluation that requires solving the direct kinematics, and is thus computer intensive. According to these works, it is impossible to indicate trends for the influence of the geometrical errors: a case by case study has to be performed, as this influence will be highly dependent upon the mechanical architecture, dimensioning and workspace of the robot

5.8.2. THERMAL ERRORS

Thermal effects may play a role in the positioning accuracy of fine positioning robots. For example Clavel (102) and Niaritsiry (449)* mentions that a $0.01\text{ }^{\circ}\text{C}$ regulated temperature is necessary to obtain nanometer accurate motions. For heavy duty robots, thermal effects are sometime mentioned as possible source of inaccuracy, although few works substantiate this claim (582). Sellgren (530) proposes using thermal sensors to correct this effect, and poses the location of these sensors as a design problem. Pritschow (489), however, states that cooling is the most effective measure, as the thermal model of parallel robots is complex. Sellgren (530) shows that for a 6-*UPS* robot, internal cooling may reduce the error in the leg lengths by 50 %.

5.8.3. GRAVITY INDUCED ERRORS

For micro-robots, Niaritsiry (449)* mentions that the deformation due to gravity may be significant and close to errors due to geometrical errors for a small workspace. For machine-tools, Pritschow (489) mentions also that gravity induces deformation in the machine kinematics. He also notes that this effect is almost constant for serial machines over the workspace, but that this is not the case for parallel robots. On the other hand, he notes that even simple flexible models leads to very good improvement, provided that the stiffnesses of the components have been measured beforehand.

5.8.4. DYNAMICS ERRORS

Pritschow (489) notes that, for high-speed machining, the dynamic errors have a much stronger impact than the static errors. It may be believed that this will also be the case for very fast parallel robots. Pritschow identifies elastic deformations, natural vibrations and drive errors as potential sources of positioning errors.

5.8.5. WORST POSES FOR ACCURACY

Another aspect of accuracy analysis will be to be able to locate *a priori*, even roughly, the poses with the worst positioning accuracy, so that the computation time of the accuracy analysis will be reduced. According to Hay (222), who relies on numerical calculation, for planar robots the worst values of the condition number are obtained for poses on the boundary of the workspace. Open problems are to determine if this also true for the worst positioning errors, and if this is true for spatial robots also.

5.9. Exercises

Exercise 5.1: We consider a 3-RRR planar robot with

$$OA_1(0,0) \quad OA_2(5,0) \quad OA_3(2,0) \quad CB_1(0,0) \quad CB_2(0,0) \quad CB_3(1,0)$$

All the lengths of the links are equal to 5. We denote the joint velocities vector by Θ . Let x, y be the coordinates of B_1 in the reference frame, θ be the angle between B_1B_2 and the x axis, and $W = (\dot{x}, \dot{y}, \dot{\theta})$. Determine the matrices A, B so that $A\Theta + BW = 0$

Exercise 5.2: Determine the full inverse kinematic jacobian of the planar 3 - RRR robot

Exercise 5.3: Determine the full inverse kinematic jacobian of the robot presented in figure 2.44

Exercise 5.4: Calculate the inverse kinematic jacobian matrix of a SSM and its determinant, when the base and platform are regular hexagons and when the centers of the hexagons are placed along the z axis.

Exercise 5.5: Calculate the H matrix of equation (5.4) for a spatial parallel robot having a planar platform when the pose of the robot is defined by the coordinates of its points B_1, B_2, B_3

Exercise 5.6: Prove the theorem presented in page 162

Exercise 5.7: Show that the manipulability ellipsoid is included in the kinematics polyhedron

Exercise 5.8: Consider a serial 2R robot with link length equal to 10. Prove that the 2-norm condition number $\kappa(\theta_2)$ of this robot is a function only of the second joint angle θ_2 . Then compute the global conditioning index GCI of the robot when this angle varies between 0 and 2π by using a numerical method. Let $w(n)$ be

$$w(n) = \left(\sum_{j=0}^{j=n-1} 1/\kappa(2j\pi/(n-1)) \right) / (2n\pi)$$

Clearly $w(n)$ is equal to the GCI when n goes to infinity. It is assumed that $w(n+10)$ is a good approximation of the GCI for n such that $100(w(n+10))$

$10) - w(n))/w(n+10)$ is lower than 0.5. Starting from $n = 10$, the sampling size is increased by 10 until the above condition is satisfied. Determine the value n_1 of n at which this situation occurs. Determine the relative error between the *GCI* and $w(n_1)$.

Exercise 5.9: Shows that for a n d.o.f. parallel robot, the square of the maximal translational velocity is equal to the sum of the squares of the singular values of J_t , multiplied by n , where J_t is the restriction of J_k to the translation components

Exercise 5.10: For 6-*PUS* robot, the lengths of the links are fixed, and the actuation is done by moving point A_i along a constant direction defined by the vector \mathbf{u} . Show how we can calculate the extrema of the joint velocities that are needed for the execution of a given twist, when C moves along a line segment, and the orientation is constant.

Exercise 5.11: Determine the needed extrema of the acceleration of the actuators so that a 6-*UPS* robot starting with a zero velocity reaches a fixed acceleration in a given direction, for any pose in a given workspace.

Problem 5.1: Determine exactly the extrema of the joint velocities of a 6-*UPS* robot for a given twist when C lies in a cartesian box.

Problem 5.2: Determine the extremal velocity in a given direction that a 6-*UPS* robot may reach when the amplitudes of its joint velocities are bounded by ± 1 . Determine the pose for which this extremal velocity may be reached.

Problem 5.3: Determine the absolute extremal velocity that a 6-*UPS* robot may reach in a given workspace, when the amplitudes of its joint velocities are bounded by ± 1 . Determine the poses where this extremal velocity is reached.

Problem 5.4: Find a method for calculating the extrema of the sum of the absolute values of the row components of a jacobian for all the configurations of the moving platform within a given workspace.

Problem 5.5: Under what conditions on the geometry of a robot and on a desired workspace will the largest maximal positioning errors be obtained for a pose on the border of the workspace ?

Problem 5.6: Is there an explicit relation between the coordinates of the base and the moving platform joint centers of a parallel robot which will ensure that it will be isotropic at a given configuration?

Problem 5.7: Determine the extremal acceleration in a given direction that a 6-*UPS* robot may reach in a given pose, when the magnitudes of the joint velocities and accelerations are bounded by ± 1 .

Problem 5.8: Consider the previous problem and find the extrema of the acceleration for any position within a given workspace.

Problem 5.9: Determine *a priori* (i.e. just by looking at the geometry of the robot) regions that contains the poses with the worst positioning accuracies.

Singular configurations

This chapter introduces the notion of *singular configurations*. Singular configurations are 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. We will explain why such poses should generally be avoided, and how these configurations may be characterized. We will then describe a geometric method that allows us to systematically determine the conditions for singularity, and obtain analytical relations between the parameters of the end-effector pose, corresponding to the different singularity cases. We will introduce indices that indicate how close a pose is from a singularity, and we present practical methods for determining whether a singular configuration exists within a given workspace.

6.1. Introduction

We consider a non-redundant n d.o.f. parallel robot having n actuated joints for a total of N joints. The motion of the end-effector is described by a set of parameters \mathbf{X} , while the joint motion are described by Θ , which includes the passive and active joint parameters Θ_p, Θ_a . We have seen that the kinematic equations may be written as:

$$F(\mathbf{X}, \Theta) = 0, \quad (6.1)$$

which is a set of N equations in $N + n$ unknowns. A *kinematic singularity* will exist when there are less than N independent equations in (6.1), or equivalently when the Jacobian matrix of the system has a rank less than N . Determining such equation dependency at first glance is not trivial: see for example the singularity analysis of Husty for a patented robot (257). We will now examine what are the consequences of such kinematic singularity on the behavior of the robot.

6.2. Singularity influence and classification

6.2.1. SINGULARITIES AND VELOCITIES

An initial study of singularity was presented by Gosselin and Angeles (189) who use a restriction of (6.1) to the active joints Θ_a and to the parameter vector \mathbf{X}_n that describes the n desired end-effector motion. The full twist

\mathbf{W} of the end-effector is decomposed into $\mathbf{W}_n, \mathbf{W}_e$ where \mathbf{W}_n corresponds to the velocities of the n d.o.f. of the robot, and \mathbf{W}_e is the complement of \mathbf{W}_n with respect to \mathbf{W} . By differentiation Gosselin and Angeles obtain a relation between the actuated joint velocities $\dot{\Theta}_a$ and the n -dimensional twist \mathbf{W}_n of the following type:

$$A\dot{\Theta}_a + B\mathbf{W}_n = 0 .$$

They then distinguish 3 different types of kinematic singularity:

- A is singular (called *type 1* or *serial* singularity): there will be a non-zero velocity vector $\dot{\Theta}_a$ for which the platform does not move.
- B is singular (called *type 2* or *parallel* singularity): there will then be a non-zero twist \mathbf{W}_n for which the joint velocities are zero. In the neighborhood of such a configuration, the robot will be able to have an infinitesimal motion while the actuators are locked (as in this case the mobility of the end-effector should be 0 it is said that the robot *gains* some d.o.f.). As a consequence, certain degrees of freedom of the end-effector cannot be controlled, and this is a major problem. The poses satisfying this condition will be called *singular configurations* or *singularities*, and will be the main study of this chapter. Clearly, for a given mechanism, such a singularity depends on the actuation scheme (388)
- both A and B are singular (*type 3* singularity): the end-effector may be moved while the actuators are locked, and *vice versa*.

A more general study was proposed by Zlatanov (663)* and we will follow his lead. Our definition of kinematic singularity corresponds to his *increased instantaneous mobility (IIM)* singularity (also called an *uncertain configuration* by Hunt (248)). In the "Velocity" chapter we establish that the relations between the passive and active joint velocities and the *full* twist \mathbf{W} of the end-effector may be written as:

$$A\dot{\Theta}_a + B\mathbf{W} + C\dot{\Theta}_p = 0 . \quad (6.2)$$

This relation is used to solve the *forward instantaneous kinematic problem (FIKP)*, determining \mathbf{W} and $\dot{\Theta}_p$ as functions of the active joint velocities $\dot{\Theta}_a$, and the *inverse instantaneous kinematic problem (IIKP)*, determining $\dot{\Theta}_p, \dot{\Theta}_a$ as functions of the twist \mathbf{W} . Equation (6.2) may be written as

$$L(\dot{\Theta}_a, \dot{\Theta}_p, \mathbf{W})^T = 0 \quad (6.3)$$

where L is a $N \times (N+n)$ matrix. The following singularities may be defined:

- *redundant input singularity (RI)*: a zero twist of the end-effector is obtained for non-zero actuated joint velocities. In that case the IIKP is not solvable; this corresponds to Gosselin's type 1 singularity.

- *redundant output singularity (RO)*: a non-zero twist \mathbf{W} of the end-effector may be obtained although the actuators are locked and the FIKP is not solvable. This type of singularity covers Gosselin’s type 2 singularity (if \mathbf{W} is such that $\mathbf{W}_n \neq 0$), but is broader, because \mathbf{W} may be such that $\mathbf{W}_n = 0, \mathbf{W}_e \neq 0$, a case which cannot be observed if we are looking only at input-output velocities equations; this has been coined *constraint singularity* by Bonev and Zlatanov (48; 666)*. Some authors distinguish a special case of constraint singularities where the motions of the end-effector are finite. Such singularities are called *architectural singularities*, and lead to a so-called *self-motion* robot. We will devote section 6.8 to this type of robot.
- *redundant passive motion singularity (RPM)*: non-zero passive joint velocities may be observed although the actuators are locked and the twist of the end-effector is 0. This singularity is called *actuator singularity* by Han and Park (212). Such a situation may occur, for example, for a Gough platform if S joints are used at both extremities of the leg; a rotation of the leg around the line joining the centers of the two S joints will in theory produce no velocity of the end-effector. In practice however this will not be true as the joints will not be perfect, and such a configuration should be avoided.

An even finer distinction addressing second order singularity is presented by Liu (369) and Wohlhart (623), but it is difficult to put in practice. A n -order singularity corresponds to the case where the n -th order time derivative of the singularity motion is not 0, while the first to n -th time derivative of the actuated joints variables are 0 (our definition of singularity hence corresponds to a first order singularity). As the inverse kinematic equations describe a variety of finite order, there is a finite value of n such that if all the n time derivative are not 0, then the mechanism will exhibit a finite singular motion and will become permanently singular (see section 6.8).

Note that other singularities may be defined. For example Chen (88) presents a *control singularity* that occurs when a single motor actuates two joints. A relation similar to (6.2), in which the joint velocities Θ_a are replaced by the motor velocities \mathbf{u} , may be written, and the singular cases of this relation have to be studied. For parallel wire robots, it is possible to define a singularity as a pose at which tension in a wire vanishes.

6.2.2. SINGULARITIES AND STATICS

It is possible to introduce singularities by studying the mechanical equilibrium of a parallel robot, as we will do more thoroughly in the "Statics" chapter. We denote by $\boldsymbol{\tau}$ the forces/torques exerted by the legs on the end-effector. The vector $\boldsymbol{\tau}$ includes the force/torque provided by the actuated

joints and possibly force/torque exerted by some passive joints. If a wrench \mathcal{F} is applied on the moving platform, the mechanical system will be in equilibrium if the resultant of the joint forces which act on the platform is the opposite to \mathcal{F} . If this is not the case, the end-effector will move until an equilibrium position is reached. We know, however, that in an equilibrium position there is a relation between $\boldsymbol{\tau}$ and \mathcal{F} :

$$\mathcal{F} = \mathbf{J}_{\text{fk}}^{-\text{T}} \boldsymbol{\tau} , \quad (6.4)$$

where $\mathbf{J}_{\text{fk}}^{-\text{T}}$ is the transpose of the full inverse kinematic jacobian. This equation describes a linear system in terms of the components of $\boldsymbol{\tau}$; it will generally admit a unique solution for a given \mathcal{F} , except when matrix $\mathbf{J}_{\text{fk}}^{-1}$ is singular: then the linear system will not have a unique solution and the mechanical system cannot be in equilibrium. Another practical consequence is that in the neighborhood of a singular configuration, the joint forces may become quite large since they are expressed as a quotient in which the denominator is the determinant of $\mathbf{J}_{\text{fk}}^{-1}$. There is therefore a significant risk of breakdown of the mechanism; it is therefore important to be able to find such singular configurations. Note that the inverse kinematic jacobian involves only the actuated joints (and possibly forces/torques in some passive joints). A more general singularity analysis in view of detecting infinite forces/torques in the joints should involve all possible joints.

6.2.3. SINGULARITIES AND KINEMATICS

Another way to introduce singular configurations is to study the uniqueness of the solution to the direct kinematics around a given solution. The actuated joint variables being fixed, the inverse kinematic equations may be written as:

$$\mathbf{F}(\mathbf{X}) = \mathbf{0} \quad (6.5)$$

If \mathbf{X}_0 is a solution of this system, then the rank theorem states that there is only one solution in the neighborhood of \mathbf{X}_0 if the jacobian of the system is nonsingular, otherwise multiple solutions are obtained, and although the actuated joints are locked the platform may be infinitesimally movable. Singularities are related to the study of the *kinematic branches*. For a given set of actuated joint variables Θ_a^0 , the system (6.5) admits a given finite set of m solution $\mathbf{X}_1, \dots, \mathbf{X}_m$. If the actuated joint variables vary continuously from Θ_a^0 to another set Θ_a^1 , then the solutions of (6.5) vary continuously: the poses of the end-effector follow different trajectories, the *kinematic branches*. A singularity will occur if two branches collapse, or if a branch goes to infinity. Note that such singularity does not cover the constraint singularities.

6.2.4. SERIAL SINGULARITY

In a serial singularity, a zero twist of the end-effector is obtained for non-zero actuated joint velocities. It has been often claimed that such a singularity corresponds to a "workspace limit". But such a statement has never been fully explained and therefore it is necessary to clarify this point.

For a robot having revolute joints between the attachment points A, B of the legs it is possible to show that if a pose is on the boundary of the workspace, then we have a serial singularity, but the converse is not true. Indeed a counter-example has been proposed by Bonev (51) with the 3- \underline{PRP} planar robot. A serial singularity is obtained when the axes of the two P joints in a chain are aligned. In that case, a pair of equal and opposite velocities of the prismatic joints leads to a zero velocity for the chain extremity, whatever its location, and thus even for poses of the end-effector that are inside the workspace of the robot.

But it is possible to have a pose on the boundary of the workspace without there being a serial singularity. For example the \mathbf{A} matrix for a 6- \underline{UPS} robot is a diagonal matrix with the leg lengths as elements. Assuming that the leg lengths are not zero, \mathbf{A} is never singular, although the robot has a workspace boundary due to the limited stroke of the actuators.

The definition of serial singularity makes it sensitive to the form of the kinematic equations. For instance, for a 6- \underline{UPS} robot, a diagonal \mathbf{A} matrix (whose determinant may be 0) appears when using the square of the leg lengths to form the kinematic equations, but will be replaced by the non-singular identity matrix if we use the square root of the leg lengths to form the kinematic equations.

Serial singularity has an interesting property: close to such a singularity the velocity of the end-effector will be very small even for large actuator velocities and hence the robot will be very accurate. This property has been studied by Zabalza (644) but is difficult to use.

6.3. Parallel singularities

6.3.1. MOTIVATIONS FOR THE STUDY OF SINGULARITY

The study of parallel singular configurations, or more exactly of redundant output singularity, is important for

- control issues: the pose of the end-effector is no longer controllable
- safety issues: elements of the robot may be submitted to very large forces, causing a breakdown of the robot. It is sometime believed that this is only an academic problem, but after experimenting such breakdown with our "left hand" prototype, and having seen another one with an industrial prototype we cannot agree with this statement ...

Singularity analysis may be studied along several directions that we will address in the remainder of the chapter:

- for a better geometrical understanding of the singularities, which may lead to a systematic discovery of these singularities and to determining what will be the possible motion of the end-effector at a singularity
- for defining an index that allows one to determine how close a pose is to a singularity
- for investigating the relation between a set of poses and singularities. For example we may have to check if the robot workspace or a robot trajectory does not include a singularity, or to find the largest cube in the robot workspace that is singularity-free

Parallel singularities may also be useful in some cases. A large amplification factor between the end-effector motion and the actuated joint motion may be interesting for fine positioning devices with a very small workspace. Another example is the force sensor proposed in (495) which is close to a singular configuration in order to improve its sensitivity along some measurement directions. We will also mention in a later section (6.8) a possible application of robots whose singularity leads to a finite motion.

Another possible use of singularities will be to sort the solutions of the direct kinematics. We have seen that powerful methods are available to determine all solutions of the problem, but that we were unable to determine which solution corresponds to the current pose. Now assume that the initial assembly mode of the robot is perfectly known: a direct kinematics solution will be a candidate to be the current pose of the platform only if it can be reached from the initial assembly by a singularity-free trajectory. In other words, the solution must belong to the same *aspect* (see section 6.9) as the initial assembly mode. Although determining if such a trajectory exists is still an open problem, it may be thought that singularity consideration may help to sort the solutions of the direct kinematics. Singularity consideration may not lead to a unique solution (i.e. eliminating all solutions of the direct kinematics except the current pose) because Innocenti (275) showed that it was possible, for planar parallel robots, to go from one solution of the direct kinematics to another without encountering any singularity. The same result has been reported by Chablat (80) and Hunt (252) for spatial robots. Note that the separation of the workspace into aspects implies that the useful workspace may be reduced to the aspect that includes the initial assembly mode. We will address this problem in the "Workspace" chapter.

6.3.2. SINGULARITY ANALYSIS

The search for singular configurations rests on the study of the singularities of the full inverse kinematic jacobian matrix, which is usually ob-

tained through a velocity analysis (see the previous chapter) or through an analysis of the mechanical equilibrium (see the "Statics" chapter). A direct analysis will thus involve the calculation of the determinant of matrix the J_{fk}^{-1} , which may be a complicated task, even with symbolic computation software. Calculating a closed-form for the determinant will usually be possible for robot with 3 d.o.f. (planar (524) or spherical (525)) but will be more difficult for a larger number of d.o.f. It was shown to be possible first for specific 6 d.o.f. robots (165; 569) and then by Mayer St-Onge (391) for 6-*UPS* robots. A compact form of the determinant may sometimes be obtained by using Grassmann-Cayley algebra (33; 142; 549). In some cases the analytical form of the determinant may be used to obtain interesting information; for example Pernkopf (473) shows that for a Gough platform with planar base and platform there will be a singularity-free ball centered at $x_c = y_c = 0$ when the base and platform frames have the same orientation, provided that the base and platform are not coplanar.

Note also that the elements of the inverse kinematic jacobian are usually functions of the pose parameters \mathbf{X} , and are difficult to express as functions of the actuated joint variables Θ_a . This implies that usually singularity conditions cannot be obtained in the actuated joint space, although it has been possible in one case (611). Calculating the determinant is only the first stage; the second stage, finding the poses that cancel the determinant, is even more difficult.

Researchers like Fichter (164) and Hunt (248) have intuitively analyzed particular cases of singularity for the inverse jacobian matrix of a 6-*UPS* robot, and have obtained a certain number of cases that will be presented in this chapter. But intuition has its limits, and since the direct method is not satisfactory, researchers have suggested methods based on the degeneracy of the screws that are associated with the robot links (337; 560). These methods are equivalent to the geometric approach that we will present, but may be more difficult to use. A preliminary result on the geometry of the singularities was obtained by Cauchy (75); this shows that the singularity of an articulated octahedron could be obtained only for concave configurations. We now discuss a geometrical method that allows us to obtain a better understanding of the geometry of the singularities.

6.4. Grassmann geometry

We recall the definition of a Plücker vector for a line; for two points M_1 , M_2 on the line and a reference point O , the Plücker vector, of dimension 6, is

$$\mathbf{P}_r = [\mathbf{M}_1\mathbf{M}_2, \mathbf{OM}_1 \times \mathbf{OM}_2] = [\mathbf{M}_1\mathbf{M}_2, \mathbf{M}_2\mathbf{M}_1 \times \mathbf{OM}_2] = [\mathbf{p}, \mathbf{q}]$$

A line is represented by any vector $\lambda \mathbf{P}_r$ where λ is an arbitrary non-zero scalar. A 6-dimensional vector will represent a line if $\mathbf{p} \cdot \mathbf{q} = 0$ and \mathbf{q} is not the zero vector. Two lines with Plücker vectors $\mathbf{P}_r^1 = [\mathbf{p}_1, \mathbf{q}_1]$, $\mathbf{P}_r^2 = [\mathbf{p}_2, \mathbf{q}_2]$ intersect if and only if $\mathbf{p}_1 \cdot \mathbf{q}_2 + \mathbf{q}_1 \cdot \mathbf{p}_2 = 0$. Plücker vectors with $\mathbf{p} = \mathbf{0}$ 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 $(\mathbf{0}, \mathbf{q})$ if and only if $\mathbf{r} \cdot \mathbf{q} = 0$. Consequently a point at infinity that belongs to the two lines at infinity $(\mathbf{0}, \mathbf{s}_1)$, $(\mathbf{0}, \mathbf{s}_2)$ has coordinates $(0, \mathbf{s}_1 \times \mathbf{s}_2)$.

The representation of a line by its Plücker coordinates is redundant because the dimension of this vector is 6, although only 4 parameters are needed to define a line. This redundancy may be decreased by introducing the *normalized* Plücker vector \mathbf{P}_{rn} defined by:

$$\mathbf{P}_{rn} = \left[\frac{\mathbf{M}_1 \mathbf{M}_2}{\|\mathbf{M}_1 \mathbf{M}_2\|}, \frac{\mathbf{OM}_1 \times \mathbf{OM}_2}{\|\mathbf{M}_1 \mathbf{M}_2\|} \right]$$

We have seen in the previous chapter that the columns of the full inverse kinematic jacobian matrices of most parallel robots are constructed from the Plücker vectors, normalized or not, 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. If n Plücker vectors are linearly independent, they will span a *variety* with dimension $n \leq 6$; if some of them are linearly dependent, the dimension of the *variety* will be less than n . H. Grassmann (1809-1877) 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 $m < n$. He established the geometrical conditions on a set of $n + 1$ lines so that the induced variety has dimension n . For a thorough introduction to Grassmann geometry, see (117; 486; 592), and for its application to the singularity analysis of parallel robots see (107)*,(215).

Grassmann's geometrical conditions allows us to design an algorithm for finding the singular configurations of *any type* of parallel robot whose full inverse kinematic jacobian consists of Plücker vectors. We will consider all sets of n lines that are associated with the vectors, and then determine what should be the pose of the moving platform 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.

6.4.1. VARIETY AND GEOMETRY

We now 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. Note that we will often mention the case of intersecting lines; these include parallel lines, which intersect at infinity.

We shall start with the dimensions going from 1 to 3 (figure 6.1); 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 \mathcal{F} that are dependent upon 3 lines $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$. It is possible to show that all the points belonging to the lines \mathcal{F} lie on a quadric surface \mathcal{Q} . This quadric degenerates to a pair of planes $\mathcal{P}_1, \mathcal{P}_2$ if any two of the three lines $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ intersect.

- condition 3d: all the lines are coplanar, but do not constitute a planar pencil of lines; $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ are coplanar and $\mathcal{P}_1, \mathcal{P}_2$ are coincident .
- condition 3c: all the lines possess a common point, but they are not coplanar; $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ 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 \mathcal{L} in common; $\mathcal{F}_2, \mathcal{F}_3$ intersect at a point \mathbf{p} , and \mathcal{L} intersects \mathcal{F}_1 at \mathbf{a} . Two different cases may occur:
 - $\mathcal{P}_1, \mathcal{P}_2$ are distinct and intersect along the line \mathcal{L} . The set of dependent lines are the lines in \mathcal{P}_1 that go through \mathbf{a} , and the lines in \mathcal{P}_2 that go through \mathbf{p}
 - $\mathcal{P}_1, \mathcal{P}_2$ are distinct and parallel. This occurs if two of the lines \mathcal{F}_i are parallel; \mathcal{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*; $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ are skew

Condition 3a must be explained in more detail. Consider three pairwise skew lines in space. The lines which intersect all three of these skew lines define a set, called a *regulus*. This set of lines, the regulus, constitutes a surface, a hyperboloid; the three skew lines are said to be the *generators* or *transversals* of the surface. In 1645, Sir Christopher Wren showed that apart from the regulus, a second set of lines generates the same hyperboloid; this second set is called the *complementary regulus*. The hyperboloid therefore is a surface that is doubly ruled. The lines of the reguli possess an interesting

characteristic: all the lines of a regulus intersect all the lines of the other regulus and none of its own regulus.

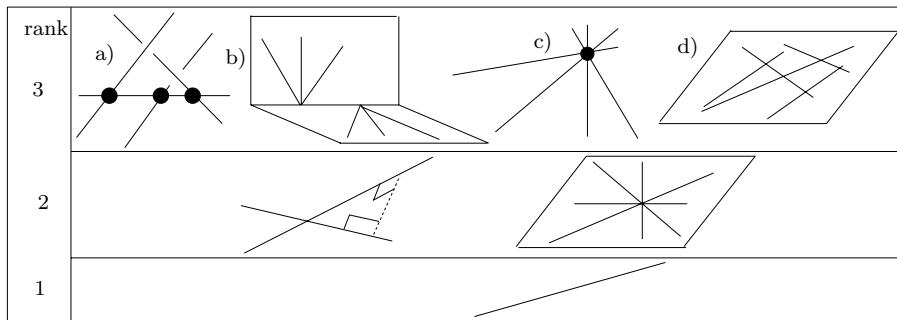


Figure 6.1. Grassmann varieties of dimension 1,2,3.

We now list the geometrical conditions that characterize the varieties of dimension 4 and 5 (figure 6.2). 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*.

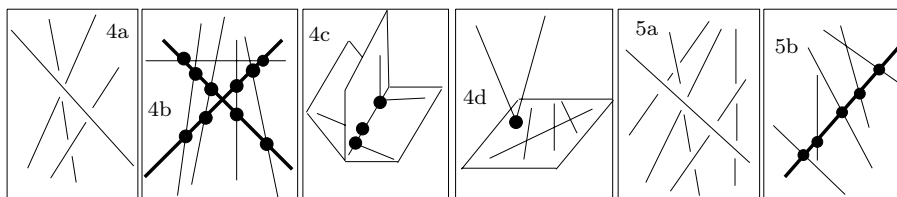


Figure 6.2. Grassmann varieties with dimension 4 and 5. The varieties are generated by the thin lines.

A variety C 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

- *singular* (5b) if $\bar{\mathbf{c}} \cdot \mathbf{c} = 0$. All the lines of the complex intersect the line with Plücker coordinates $(\mathbf{c}, \bar{\mathbf{c}})$.
- *general or non singular* (5a) if $\bar{\mathbf{c}} \cdot \mathbf{c} \neq 0$

The degree of freedom associated with a linear complex is a screw motion (248) with axis defined by the line with Plücker vector $(\mathbf{c}, \bar{\mathbf{c}} - p\mathbf{c})/||\mathbf{c}||$, where $p = \bar{\mathbf{c}} \cdot \mathbf{c}/||\mathbf{c}||$ is the pitch of the motion. All coplanar lines of a non singular complex are in a plane that is normal to the helical motion at a point, and intersect this point, thereby constituting a planar pencil of lines.

6.4.2. EXAMPLES OF GEOMETRICAL ANALYSIS

Line geometry has been used for the singularity analysis of numerous robots (see for example (427) and the references Web page). We will present here some typical examples: the planar 3- $R\underline{P}R$, the 3- $U\underline{P}U$ and the MSSM (figure 6.3).

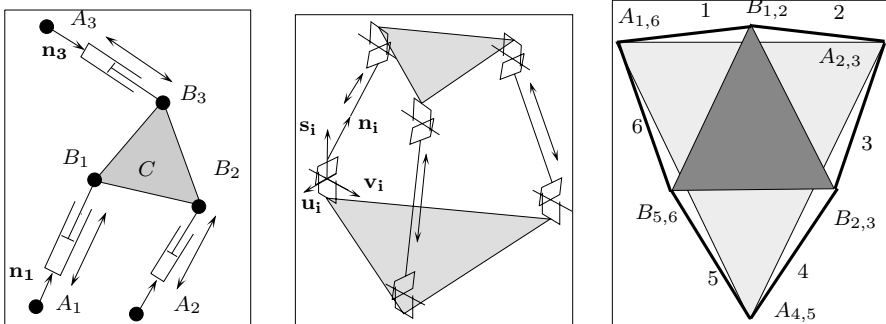


Figure 6.3. The planar 3- $R\underline{P}R$ robot, the 3- $U\underline{P}U$ robot and the MSSM.

6.4.2.1 Planar 3- $R\underline{P}R$ manipulator

We have established in section 5.2.2.1 the full inverse kinematic jacobian of this type of robot as

$$\begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{n}_1 & \mathbf{CB}_1 \times \mathbf{n}_1 \\ \mathbf{n}_2 & \mathbf{CB}_2 \times \mathbf{n}_2 \\ \mathbf{n}_3 & \mathbf{CB}_3 \times \mathbf{n}_3 \\ \mathbf{z} & \mathbf{0} \\ \mathbf{0} & \mathbf{x} \\ \mathbf{0} & \mathbf{y} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \tag{6.6}$$

The 3 first rows of the full inverse kinematic jacobian J_{fk}^{-1} are the Plücker vectors of the lines associated with the legs of the robot. In these vectors the

third to fifth components will be 0. The fourth row is a line perpendicular to the robot plane and going through the origin. The remaining rows are lines at infinity.

A row of the 3×3 inverse kinematic jacobian matrix J_k^{-1} is constituted of the 2 first components of \mathbf{n}_i and the last component of $\mathbf{CB}_i \times \mathbf{n}_i$. By expanding the determinant of J_{fk}^{-1} with respect to its last three rows, we find that the determinant of this matrix is equal to the determinant of J_k^{-1} . If A_k denotes the set of row vectors of J_k^{-1} , and A_{fk} the first three row vectors of J_{fk}^{-1} , a linear dependence between the vectors of A_k (A_{fk}) implies a linear dependence between the vectors of A_{fk} (A_k). Hence the full inverse kinematic jacobian will be singular if and only if the Plücker vectors in A_{fk} (which describe the lines associated with the legs of the robot) are linearly dependent. Hence according to the Grassmann conditions, the robot will be singular if the three legs are coplanar, and their associated lines meet the same point (possibly at infinity i.e. if the legs are parallel).

Sefrioui (526) did a thorough analysis of the singularity condition of planar robots, and showed that, in general, it is a quadratic equation. Bonev (51) presents an exhaustive analysis of the singularity condition for all types of planar parallel robots.

6.4.2.2 3-UPU manipulator

We will consider here the case of the 3-UPU robot as a translation mechanism (figure 6.3). The full inverse kinematic jacobian has been established:

$$\begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{n}_1 & (\mathbf{CB}_1 \times \mathbf{n}_1) \\ \mathbf{n}_2 & (\mathbf{CB}_2 \times \mathbf{n}_2) \\ \mathbf{n}_3 & (\mathbf{CB}_3 \times \mathbf{n}_3) \\ \mathbf{0} & \mathbf{s}_1 \\ \mathbf{0} & \mathbf{s}_2 \\ \mathbf{0} & \mathbf{s}_3 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \quad (6.7)$$

where the vector \mathbf{s}_i are the vectors obtained as the cross-product of the vectors associated with the axes of the base U joint of the legs. The inverse kinematic jacobian J_k reduces to a 3×3 matrix constituted of the \mathbf{n}_i vectors. If T is the 3×3 matrix constituted of the \mathbf{s}_i vectors, the determinant of J_{fk} is equal to $|J_k||T|$. Hence a singularity is obtained when one of the following conditions holds:

1. the legs of the robot lie in the base plane (or are parallel, which may occur only for a particular geometry of the robot)
2. the vectors \mathbf{s}_i are coplanar
3. two of the vectors \mathbf{s}_i are parallel

These singularity conditions may also be explained by using Grassmann geometry. Assume that the vectors \mathbf{s}_i are not coplanar or parallel, then the

variety spanned by the three last rows of J_{fk} includes the line at infinity $(0, \mathbf{n}_1 \times \mathbf{n}_2)$. If the lines associated with the legs of the robot lie in a plane, then the line at infinity also belongs to this plane. Hence we have 4 lines in the same plane which correspond to the singular configuration 3d, and explain singularity 1.

The three last rows of J_{fk} represent line at infinity. The lines at infinity $(0, \mathbf{s}_1)$, $(0, \mathbf{s}_2)$ intersect at the point $(0, \mathbf{s}_1 \times \mathbf{s}_2)$ and this point belong to the line at infinity $(0, \mathbf{s}_3)$ if $(\mathbf{s}_1 \times \mathbf{s}_2) \cdot \mathbf{s}_3 = 0$. Consequently if this condition is fulfilled the three lines are coplanar and intersect at the same point: this is a singularity case justifying singularity 2. If two vectors $\mathbf{s}_i, \mathbf{s}_j$ are parallel then $(0, \mathbf{s}_i)$, $(0, \mathbf{s}_j)$ represent the same line, which explain singularity 3.

Singularity 2 has often been neglected as it does not appear if only J_k is considered. This singularity was discovered during a CK meeting at the Seoul National University (SNU) where a prototype was built that exhibited large rotational motion, although J_k was not singular. In this prototype, the first axis of the U joints on the base and platform intersected at the same point, and in a given configuration the U joints axis were all horizontal. Bonev and Zlatanov (48), and later on Di Gregorio (135), discovered that this configuration leads to a *constraint singularity* as the vectors \mathbf{s}_i are all parallel. The associated infinitesimal motion is a 2 d.o.f rotational motion along the horizontal axis that go through the intersection point of the legs. In a later study Bonev and Zlatanov established that a constraint singularity was also possible for the wrist version of the 3- UPU (666).

But a further question was prompted by the SNU prototype rotational motion. The singularity 3 exists theoretically only in a given configuration while the prototype exhibits very large rotational motion well away from this singularity configuration. Bonev and Zlatanov (48) established that if a system of 9 quadratic equations in 9 unknowns (the coordinates of the B_i points) has multiple solutions, then the robot singular motion may become finite. But these authors were not able to establish if there was geometric design of the 3- UPU such that the system has multiple solutions. They conjecture that this is not the case, and that the large motions were due to the clearances in the U joints. The effect of a clearance in the U joint of the robot on the motion of the end-effector was indeed emphasized by Han and co-workers (212), who showed that the 3- UPU is indeed highly sensitive to these clearances; this may explain the large motion of the SNU prototype. A similar result was obtained by Wohlhart for the 6- UPS robot (623).

6.4.2.3 MSSM

This section presents an extensive analysis for the MSSM (figure 6.3) and was presented in (397) together with an analysis of the TSSM (for the TSSM another interesting geometrical approach based on the singular configura-

tion of a characteristic tetrahedron was proposed by Ebert-Uphoff (154)). We assume that no link is coplanar with the base, and that, as a consequence, the platforms are not coplanar. The case when two lines are identical is therefore excluded, especially as the geometry of the MSSM would not accept it. The plane containing links i, j will be denoted P_{ij} .

6.4.2.3.1 Variety of dimension 2

Consider the case when 3 Plücker vectors are linearly dependent, and, as a consequence, the 3 associated lines are coplanar and have a common point. We note that for any triplet of lines that may be coplanar (1, 2, 3 for example, see figure 6.4), two of these lines will have a common joint center on the moving platform (B_1). This point must therefore be the

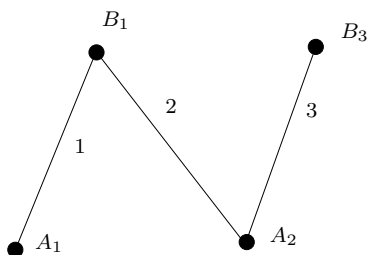


Figure 6.4. Three lines of a MSSM in the same plane.

center of the pencil of lines. Moreover, the third line has a common joint (A_2) with one of the other two lines although, now, the joint is connected to the base. If this third line includes the center of the pencil, it would then have two common points with one of the other two lines and would be identical to it (in our example, line 3 should be identical to line 2); but this was excluded in our hypothesis. Therefore no set of 3 Plücker vectors may span a dimension 2 variety.

6.4.2.3.2 Variety of dimension 3, case 3d

Let us now consider 4 lines and see whether they may satisfy condition 3d, i.e. that the 4 lines are coplanar. Using the previous paragraph, we know that at least two of these lines have a common point on the base, or the moving platform, and that another pair has a common point on the moving platform or the base. Under these conditions, the only possible line quadruplets are (1,2,3,6), (2,3,4,5), (1,4,5,6). Simple permutations of the frames of the base and mobile allow us to study just the quadruplet (1,2,3,6). We note that if (1,2,3,6) are coplanar, then the moving platform also lies in this plane. We will thus obtain Hunt's singular configuration shown in figure 6.5. We will now outline how to derive a singularity condition in terms

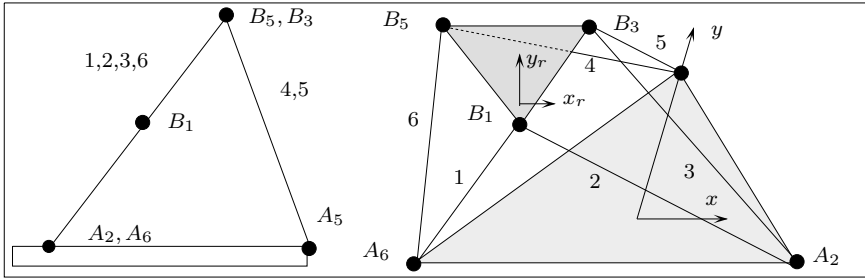


Figure 6.5. Singular configuration of the 3d type for a MSSM.

of x_c, y_c, z_c , the coordinates of C , and of ψ, θ, ϕ , the Euler angles. The singularity condition of the the co-planarity of the lines (1,2,3,6) is expressed with the following equations:

$$(\mathbf{A}_1\mathbf{B}_1 \times \mathbf{A}_2\mathbf{B}_2) \cdot \mathbf{A}_3\mathbf{B}_3 = 0 , \tag{6.8}$$

$$(\mathbf{A}_1\mathbf{B}_1 \times \mathbf{A}_2\mathbf{B}_2) \cdot \mathbf{A}_6\mathbf{B}_6 = 0 . \tag{6.9}$$

Equations (6.8), (6.9) constitute a linear system in y_c, z_c ; x_c is absent. The determinant Δ of this system is $\Delta = K \sin \theta \sin \psi$, where K is a non-zero constant. If $\Delta \neq 0$, y_c, z_c may then be calculated, but these values lead to $(\mathbf{A}_1\mathbf{B}_1 \times \mathbf{A}_2\mathbf{B}_2) = \mathbf{0}$. Links 1, 2 then are collinear and lie in the plane of the base, which was excluded in our hypothesis. We therefore have to consider $\Delta = 0$. This condition will be satisfied if $\sin \theta = 0$, i.e. $\theta = 0$ or $\theta = \pi$. If $\theta = 0$, then equations (6.8), (6.9) can be satisfied simultaneously only if $z_c = 0$ and all the links then lie in the base plane, which was excluded in our hypothesis. For $\theta = \pi$ a similar result is obtained.

The determinant Δ is also zero when $\sin \psi = 0$, i.e. $\psi = 0$ or $\psi = \pi$. In both cases, equations (6.8), (6.9) are identical. For $\psi = 0$ we get

$$z_c = -\frac{(y_{a_2} - y_c) \sin \theta}{\cos \theta} = H_{3d_3}(y_c, \theta) \tag{6.10}$$

while for $\psi = \pi$ we get $z_c = -H_{3d_3}(y_c, \theta)$. The 3d singularity condition is therefore

$$\psi = 0 \quad z_c = H_{3d_3}(y_c, \theta) \quad \forall x_c, \phi \tag{6.11}$$

$$\psi = \pi \quad z_c = -H_{3d_3}(y_c, \theta) \quad \forall x_c, \phi \tag{6.12}$$

It is noteworthy that in this 3d configuration, the 6 lines intersect the line B_3B_5 : this is therefore also a 5b configuration and the lines constitute a singular complex.

6.4.2.3.3 Variety of dimension 3, case 3c

Consider case 3c, when the 4 lines have a common point. Within a quadruplet of lines, two lines have one common point A_i on the base, whereas the other two share B_j on the moving platform. For each of these pairs, the lines cannot have any other common point, or they will be identical. The only case when these 4 lines could share a point is when the points A_i and B_j are the same, which was excluded in our hypothesis. This configuration therefore is not possible.

6.4.2.3.4 Variety of dimension 3, case 3b

Now consider case 3b, where the 4 lines belong to the union of two planar pencils of lines. The different types of possible quadruplets lead to the study of the following quadruplets of lines: (1, 2, 3, 4), (1, 2, 3, 5), (1, 2, 3, 6).

- quadruplet (1, 2, 3, 4): we know the centers B_1, B_3 , of the two pencils, as well as their plane, P_{12}, P_{34} . If a line belongs to the two pencils, it will be supported by $B_1 B_3$. As a consequence, $B_1 B_3$ belongs to the intersection of the two pencils planes, i.e. P_{12}, P_{34} . However, point A_2 belongs to this intersection and thus the points A_2, B_1, B_3 would be on a line; this means that lines 2 and 3 are identical, which we excluded.
- quadruplet (1, 2, 3, 5): the line shared by the two pencils belongs to the intersection of P_{12}, P_{35} , and thus A_2 belongs to this line. The center B_1 of the pencil that is spanned by 1 and 2 also belongs to this common line, and therefore this line is line 2. One point of line 3 also belongs to this line and therefore lines 2 and 3 are identical, which we excluded.
- quadruplet (1, 2, 3, 6): the planes P_{12}, P_{36} share the points A_1, A_2 that therefore support the common line, although this line does not go through the center B_1 of the pencil that is spanned by 1 and 2. This condition therefore cannot be satisfied.

6.4.2.3.5 Variety of dimension 3, case 3a

In case 3a, the 4 lines belong to the same regulus. We know that all the lines of a regulus are skew: due to the special geometry of the MSSM there cannot be a quadruplet of links in which all the associated lines are skew, and consequently no lines may belong to the same regulus.

6.4.2.3.6 Variety of dimension 4, case 4d

Consider 5 lines in the 4d configuration where the 5 lines lie in the same plane or have a common point in this plane. First, we must note that it is impossible to have 5 coplanar lines; we already examined this case for the 3d configuration. This leaves us with the case when three lines are coplanar; there are two situations. The first is when two of the three lines

have a common joint on the moving platform. We may then show that for each remaining pair of lines, the points which belong to the plane of configuration 4d will be the pairs A_i, B_j , and by hypothesis these cannot be identical. For example, for the triplet 2, 3, 4 the intersection between the plane 234 and the pair 1, 5 will be the points A_5, B_2 . The second is when none of the three lines, say none of 2, 3, 5, has a common joint point on the moving platform. Then one of the remaining lines, 4 in the last case, will be coplanar with the considered triplet. We are then able to apply the result obtained for 4 coplanar lines.

The remaining case is when 3 lines have a common point lying in the planes of the two remaining lines. Without loss of generality, we assume that the coplanar lines are the lines 1-2. The lines 3-6 intersect this plane at distinct points (A_1 and A_2) and therefore only the 2 quintuplets (1,2,3,4,5) and (1,2,4,5,6) may possibly satisfy the geometric condition. For these quintuplets, the point intersecting the plane is A_3 or A_6 . However, as for the line triplets (3,4,5), (4,5,6), two lines already have a common point (B_3 or B_5). They therefore cannot have another common point; this concludes the study of this case.

6.4.2.3.7 Variety of dimension 4, case 4c, 4b

Case 4c may quickly be set aside, as it involves 3 lines belonging to one pencil of lines. Case 4b, in which the 5 lines all meet two skew lines, needs to be studied. Without loss of generality, we may consider the quintuplet (1,2,3,4,5). First note that it is impossible for both of two skew lines to intersect two (or more) coplanar lines: 1,2,3 should therefore not be coplanar. If a line intersects the lines 1, 2, 3, then we have 3 possibilities:

- (4b1) the line lies in the plane 2-3 and goes through the center of the joint on the moving platform common to 1-2 (line D1, figure 6.6).
- (4b2) the line lies in the plane 1-2 and intersects 3 in the joint on the base common to 2 and 3 (line D2, figure 6.6).
- (4b3) the line intersects both B_1 and B_3 .

A line having a common point with 4-5 either lies in the plane 4-5 or goes through the center of the joint on the base common to 4 and 5 (A_4). In the 4b1 case if a line intersects 1-2-3-4-5 the line is either in the plane 1-2 or in the plane 2-3 while it lies in the plane 4-5, which is possible only if 2-3-4 are coplanar, in which case we cannot find another skew line which meets the conditions. Note that if the line intersects 5, then it goes through A_4 ; we must exclude Hunt's configuration in which 5 is coplanar with 2-3-4.

In case 4b2 there is no line in the 1-2 or 2-3 plane which goes through A_4 and hence no line can intersect at the same time 1-2-3-4-5. In case 4b3 if the line B_1B_3 intersects 5, then 2-3-4-5 are coplanar (Hunt's configuration)

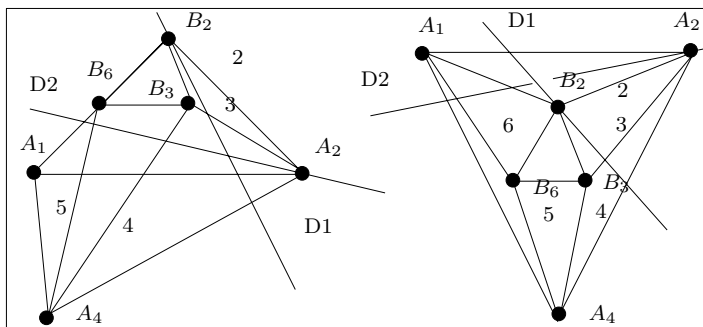


Figure 6.6. Two skew lines (D1, D2) intersect the line associated to links 1, 2, 3.

and there will be no other skew line intersecting 1-2-3-4-5. To summarize, we cannot find two skew lines intersecting all the 5 lines of the quintuplet.

To conclude the quintuplet of lines, we have to consider the case 4a. Here, the problem is to determine whether two of the lines may intersect the regulus that is generated by the other three in a proper point. Consider the lines (1,2,3,4,5); the only triplet of skew lines in this quintuplet is (1,3,5), although lines 2 and 4 intersect the lines of this triplet. The condition therefore cannot be met.

6.4.2.3.8 Variety of dimension 5, case 5b

Consider a sextuplet of lines in a singular complex configuration 5b. In this case, the 6 lines all intersect a line. We have seen (case 4b) that all lines of a quintuplet can meet one line only in two cases:

- the line contains one edge of the moving platform, and the 4 successive links are coplanar; this is Hunt's configuration.
- three links are coplanar, and the line goes through joints on the base and the moving platform that are not common to two of the three links, as shown in figure 6.7.

Let us study the first case. Consider the line B_4B_5 , that intersects lines 3,4,5,6 by construction. If it shares a point with 1, then the links 1,2,3,6 are coplanar, and all the lines then intersect one line; this is equivalent to Hunt's configuration.

In the second case, we assume that the coplanar links are 2,3,4 and consider the line B_2A_4 , (D). This line intersects links 1,2,3,4,5; let us now rotate the moving platform around the edge B_4B_2 . Let M be the intersection of line 6 with the plane 2-3-4: figure 6.7 clearly shows that one may find a rotation that takes M onto the line (D). This line will then share a point with all 6 lines that are associated with the links.

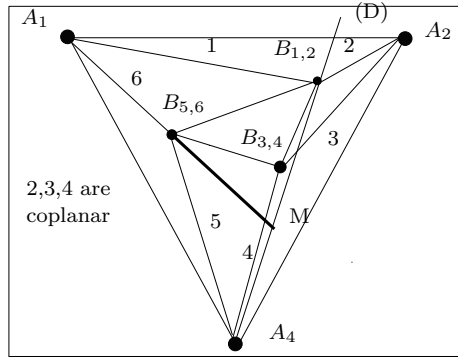


Figure 6.7. 2-3-4 being coplanar, line (D) here intersects the 5 lines (1,2,3,4,5) by construction. When the moving platform rotates around B_2B_4 , the sixth line may then intersect (D) at point M.

Such a configuration is characterized by its triplet of coplanar lines. It is easy to show that if there are not 4 coplanar lines, the only triplets to consider are (1,2,3), (1,5,6), (2,3,4), (3,4,5), (4,5,6). As for the case 3d, one may transform the study of these triplets into the study of the triplet (1,2,3) via *ad hoc* frame modifications. In this case, line A_1B_3 , (D), intersects lines (1, 2, 3, 4, 6) by construction. We then have to state the condition indicating that line 5 intersects (D). Co-planarity of (1,2,3) is expressed by the following equation:

$$(\mathbf{A}_1\mathbf{B}_1 \times \mathbf{A}_2\mathbf{B}_2) \cdot \mathbf{A}_3\mathbf{B}_3 = 0 \tag{6.13}$$

The intersection condition of 5 with the line A_1B_3 is:

$$\mathbf{A}_1\mathbf{B}_3 \cdot (\mathbf{OA}_5 \times \mathbf{OB}_5) + \mathbf{A}_5\mathbf{B}_5 \cdot (\mathbf{OA}_1 \times \mathbf{OB}_3) = 0 \tag{6.14}$$

Equations (6.13)(6.14) constitute a linear system in y_c, z_c with determinant $\Delta = K \sin \theta F(\psi, \theta, \phi)$. When $\Delta \neq 0$, the singularity condition is:

$$y_c = H_{5b_1}(x_c, \psi, \theta, \phi) \quad z_c = H_{5b_2}(x_c, \psi, \theta, \phi) \tag{6.15}$$

Figure 6.8 shows an example of this type of singular configuration. When $\Delta = 0$ with $\theta = 0$, equations (6.13)(6.14) leads to $z_c = 0$, i.e. the links lie in the base plane or,

$$\tan \psi = \frac{y_{b_1}}{x_{b_3}} = -\frac{y_{a_4}}{x_{a_1}}, \tag{6.16}$$

which may be satisfied only for particular robot geometries. If $\theta = \pi$ equations (6.13)(6.14) leads to $z_c = 0$, i.e. the links lie in the base plane or

$$\tan(\psi - \phi) = \frac{y_{b_1}}{x_{b_3}} = -\frac{y_{a_4}}{x_{a_1}} \tag{6.17}$$

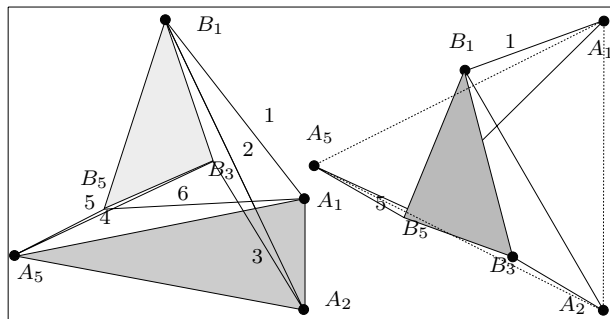


Figure 6.8. An example of singular configuration 5b for the MSSM. Lines 1,2,3 are coplanar, and therefore line A_1B_3 meets links 1, 2, 3, 4, 6. Moreover, line A_5B_5 meets line A_1B_3 by construction.

that may be satisfied only for particular robot geometries. The determinant is also zero when $F(\psi, \theta, \phi) = 0$; this allow us to calculate ψ as a function of θ, ϕ . Equations (6.13),(6.14), then give z_c and x_c . The singularity condition may therefore be written as

$$\psi = H_{5b3}(\theta, \phi), \quad z_c = H_{5b4}(y_c, \theta, \phi), \quad x_c = H_{5b5}(y_c, \theta, \phi). \quad (6.18)$$

6.4.2.3.9 Variety of dimension 5, case 5a

Having studied the case of the singular complex, we may now consider that of the general complex (case 5a).

We have seen that one characteristic of a general complex is that all its lines that are coplanar belong to a planar pencil, i.e. they have a common point. Consider the planar pencils of lines that are spanned by the following pairs of lines (1,6), (2,3) and (4,5). Each of these pencils possesses, in the general case, a line D_i that is coplanar with the base platform (figure 6.9). If the 6 lines belong to a non singular complex, then lines D_i , that belong to pencils of lines that are spanned by lines of the non singular complex, also belong to this complex. The lines D_i must therefore have a common point M , with coordinates $(x, y, 0)$. Let \mathbf{v}_{ij} be the vector perpendicular to the pencil of lines that is spanned by lines i, j . The necessary conditions so that lines D_i belong to the pencil and intersect the point M are

$$\mathbf{A}_1\mathbf{M} \cdot \mathbf{v}_{16} = 0 \quad \mathbf{A}_3\mathbf{M} \cdot \mathbf{v}_{23} = 0 \quad \mathbf{A}_5\mathbf{M} \cdot \mathbf{v}_{45} = 0 \quad (6.19)$$

These three equations are linear in x, y . It is therefore possible to use two of these equations to obtain x, y , transfer their values to the third and thereby get the constraint equation. This constraint equation will have degree 3 in z_c , degree 2 in each of x_c, y_c , and coefficients that are functions

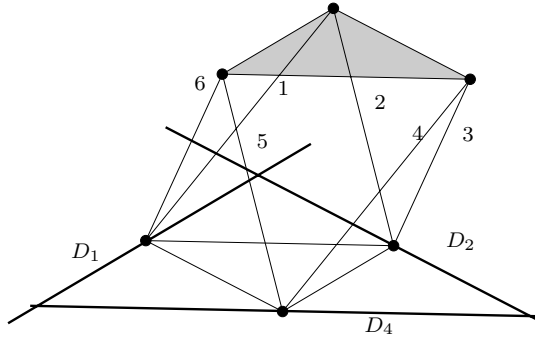


Figure 6.9. The lines D_1, D_2, D_4 belonging to the pencils spanned by $(1,6), (2,3), (4,5)$ are coplanar with the base platform. If the lines $(1,2,3,4,5,6)$ constitute a non singular complex, then the lines D belong to the same pencil.

of ψ, θ, ϕ . If $\theta = \phi = 0$, the constraint equation simplifies to $\tan \psi = U$, where U is a function of the coordinates of A_i, B_i . This equation becomes even simpler if the moving platform is symmetrical, since then we obtain $\cos \psi = 0$ which corresponds to a rotation of the moving platform around the z axis by $\pm\pi/2$. This particular singularity was observed by Fichter experimentally (163) and then justified theoretically (164). The constraint equation therefore generalizes Fichter's configuration. Note that we have solved a linear system to get this condition: if the determinant of this linear system is 0, we then obtain equations that cannot be satisfied.

Let us consider an example of singular configurations of type 5a for the standard MSSM, using the following parameters values: $x_c = 0, y_c = 0, \psi = 40^\circ, \theta = 40^\circ, \phi = 40^\circ$. The polynomial will have three real roots, leading to three singular configurations, as described in figures 6.10,6.11,6.12.

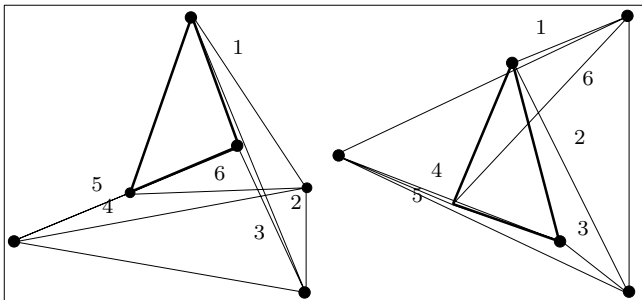


Figure 6.10. Perspective and plan view of the first singularity 5a.

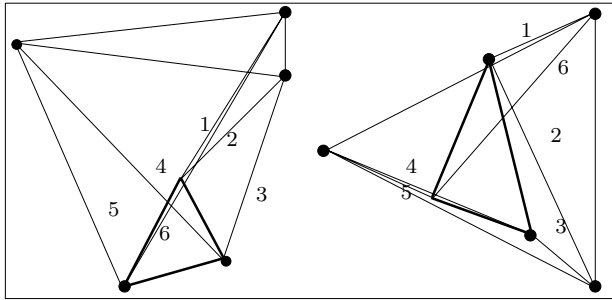


Figure 6.11. Perspective and plan view of the second singularity 5a.

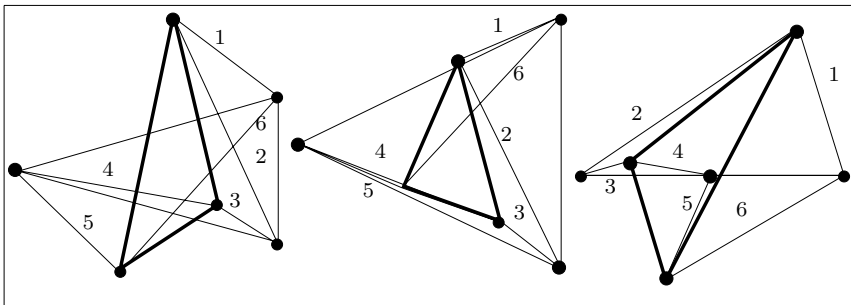


Figure 6.12. Perspective, plan and side views of the third singularity 5a.

We note that each singularity case of an MSSM amounts to the generation of a linear complex. Condition 5a therefore is the general equation that defines the singular configurations, and simply takes a particular form in cases 3b, 5b; the fact that the development of the determinant of the inverse jacobian leads to the equation obtained for case 5a has actually already been demonstrated (391). We note that the geometric approach allows us to obtain singularity conditions more easily than direct development of the determinant.

Table 6.1 presents a summary of the conditions that the pose parameters of the moving platform must satisfy so that the MSSM is singular. In this table, we denote $\mathbf{X}_c = (x_c, y_c, z_c)$, $\mathbf{\Omega}_c = (\psi, \theta, \phi)$.

case	singularity conditions	
3d $ \psi = 0$	$z_c = H_{3d_3}(y_c, \theta)$	$\forall x_c, \phi$
$ \psi = \pi$	$z_c = -H_{3d_3}(y_c, \theta)$	$\forall x_c, \phi$
5a $ \sum_{i=0}^{i=3} a_i(x_c, y_c, \Omega_c) z_c^i = 0$		
$ \sum_{i=0}^{i=2} b_i(x_c, z_c, \Omega_c) y_c^i = 0$		
$ \sum_{i=0}^{i=2} c_i(y_c, z_c, \Omega_c) x_c^i = 0$		
$ \theta = \phi = 0$	$\psi = \pm \frac{\pi}{2}$	$\forall(\mathbf{X}_c)$
5b $ \psi = H_{5b_3}(\theta, \phi)$	$z_c = H_{5b_2}(x_c, \Omega_c)$	
$ \psi = H_{5b_3}(\theta, \phi)$	$z_c = H_{5b_4}(y_c, \theta, \phi)$	$x_c = H_{5b_5}(y_c, \theta, \phi)$

TABLE 6.1. MSSM singularity conditions.

6.5. Motion associated with singularities

6.5.1. DETERMINATION OF THE SINGULARITY MOTION

The study of singular configurations presented in the previous section allowed us to bring families of singularities to the fore. However we did not examine the type and number of extra degrees of freedom that we obtained in these configurations. This may easily be done by considering the eigenvectors that are associated with the null eigenvalues of the full inverse kinematic jacobian matrix. These eigenvectors define the motion of the manipulator that occurs with a null vector of joint velocities. According to the reciprocity principle, they also give the external wrenches that cannot be balanced by the joint forces.

In order to determine the eigenvectors, we substitute the relations determined during the singularity analysis into the full inverse kinematic jacobian matrix, and calculate symbolically a basis of the kernel of the linear system defined by this matrix. This allows us to determine the rank of the kernel (and therefore the number of d.o.f. gained by the robot at the singularity) and to characterize geometrically the infinitesimal motions that are associated with each type of singular configurations.

6.5.2. DETERMINATION OF THE INSTANTANEOUS ROTATION AXIS

Any rigid body displacement can be realized by a rotation about an axis, the *instantaneous rotation axis* (IRA), combined with a translation parallel

to that axis.

The determination of the instantaneous rotation axis (IRA) may be made as follows. Suppose that a point $M(x, y, z)$ belongs to the IRA; its velocity \mathbf{V}_M then is collinear with $\boldsymbol{\Omega}$:

$$\mathbf{V}_M = \alpha \boldsymbol{\Omega} . \quad (6.20)$$

We have

$$\mathbf{V}_M = \mathbf{V}_C + \mathbf{M}\mathbf{C} \times \boldsymbol{\Omega} , \quad (6.21)$$

which leads us to

$$\mathbf{V}_M \times \boldsymbol{\Omega} = \mathbf{V}_C \times \boldsymbol{\Omega} + (\mathbf{M}\mathbf{C} \times \boldsymbol{\Omega}) \times \boldsymbol{\Omega} = 0 , \quad (6.22)$$

this leads to 2 independent linear equations for x, y, z , thus defining two planes. The IRA is the intersection of these two planes. The *pitch* h of the motion is defined as

$$\mathbf{V}_M \cdot \boldsymbol{\Omega} = h . \quad (6.23)$$

If the pitch is 0 the motion is a pure rotation while it will be infinite for a translation motion.

6.5.3. EXAMPLE: THE MSSM

We examine the different singularity cases for the MSSM. The vectors of the basis of the kernel of the full inverse kinematic jacobian are denoted \mathbf{A}_i (by \mathbf{A} , if the dimension of the kernel is 1). If $\mathbf{A}_i = (\mathbf{p}_i, \mathbf{q}_i)$, then \mathbf{p}_i represents the translational velocity of the singular motion and \mathbf{q}_i its angular velocity while $(\mathbf{q}_i, \mathbf{p}_i)$ will be the Plücker vector of the IRA.

6.5.3.1 Type 3d configuration

We have seen that singularity conditions may be written

$$\psi = 0 , \quad z_c = H_{3b_3}(y_c, \theta) , \quad \forall x_c, \phi; \quad (6.24)$$

$$\psi = \pi , \quad z_c = -H_{3b_3}(y_c, \theta) , \quad \forall x_c, \phi. \quad (6.25)$$

We first assume that $\psi = 0$. If $\phi \neq 0$ the vector of the basis of the kernel of the inverse jacobian is:

$$\mathbf{A} = \left(0, \frac{y_{b_3} \sin \theta}{\sin \phi \cos \theta}, -\frac{y_{b_3}}{\sin \phi}, \frac{\cos \phi}{\sin \phi \cos \theta}, 1, \frac{\sin \theta}{\cos \theta} \right) \quad (6.26)$$

When $\phi = 0$, we obtain:

$$\mathbf{A} = (0, y_{b_3} \sin \theta, -y_{b_3} \cos \theta, 1, 0, 0) \quad (6.27)$$

When $\psi = \pi$, $\phi \neq 0$, the vector of the basis is

$$\mathbf{A} = \left(0, \frac{y_{b_3} \sin \theta}{\sin \phi \cos \theta}, \frac{y_{b_3}}{\sin \phi}, \frac{\cos \phi}{\sin \phi \cos \theta}, 1, -\frac{\sin \theta}{\cos \theta} \right), \quad (6.28)$$

when $\phi = 0$ we obtain

$$\mathbf{A} = (0, y_{b_3} \sin \theta, y_{b_3} \cos \theta, 1, 0, 0). \quad (6.29)$$

In all cases we see that the pitch of the motion is 0, which indicates a pure rotation and the IRA is the line B_3B_5 .

6.5.3.2 Type 5a and 5b configuration

We know that the infinitesimal motion that is associated with a complex is a screw motion (248).

This study is limited to the case where $\theta = \phi = 0$ and the moving platform is symmetrical. In this case, we know that the singularities are obtained for Fichter's configuration, $\psi = \pm \frac{\pi}{2}$. If $\psi = \frac{\pi}{2}$ matrix \mathbf{A} is too large to be given here but can be found in (397). It has the general form:

$$\mathbf{A} = \left(1, U_{5a}^1, \frac{F_{5a}^1(y_c)}{U_{5a}^2 z_c}, \frac{F_{5a}^2(x_c)}{U_{5a}^3 z_c}, \frac{F_{5a}^3(y_c)}{U_{5a}^4 z_c}, 1 \right)$$

where the U_{5a}^i are functions of the location of the A_i, B_i . However, if the platforms are equilateral triangles that are inscribed in circles with radii R_m , for the moving platform and R_b , for the base, we find

$$\mathbf{A} = \left(0, 0, -\frac{R_m R_b}{2x_c}, \frac{y_c}{x_c}, \frac{z_c}{x_c}, 1 \right), \quad (6.30)$$

if $x_c \neq 0$; otherwise

$$\mathbf{A} = \left(0, 0, -\frac{R_m R_b}{2z_c}, 0, \frac{y_c}{z_c}, 1 \right). \quad (6.31)$$

If $y_c = 0$, we find a screw motion about the z axis, as described by Fichter, whose pitch h is

$$h = -\frac{R_m R_b}{2z_c}. \quad (6.32)$$

If $x_c = 0$ and $y_c \neq 0$, we find a screw motion about an axis that is defined by

$$x = -\frac{R_m R_b y_c}{2(y_c^2 + z_c^2)} \quad y = \frac{y_c z}{z_c}; \quad (6.33)$$

the pitch is

$$h = -\frac{R_m R_b}{2\sqrt{y_c^2 + z_c^2}}. \quad (6.34)$$

If $x_c \neq 0$ and $y_c \neq 0$, we find a screw motion about an axis defined by

$$\begin{aligned} x &= -\frac{x_c^2 R_m R_b - 2x_c y_c^2 y - 2x_c z_c^2 y - 2x_c^3 y + y_c^2 R_m R_b}{2y_c(x_c^2 + y_c^2 + z_c^2)}, \\ z &= -\frac{z_c(R_m R_b x_c - 2y_c^2 y - 2z_c^2 y - 2x_c^2 y)}{2y_c(x_c^2 + y_c^2 + z_c^2)}, \end{aligned} \quad (6.35)$$

with pitch

$$h = -\frac{z_c R_m R_b}{2\sqrt{x_c^2 + y_c^2 + z_c^2}}. \quad (6.36)$$

Similar result may be obtained for $\psi = -\frac{\pi}{2}$.

Type 5b is a special case of 5a with all legs intersecting the same line. The associated motion is a pure rotation around this line.

6.6. Singularity indices

In the previous sections we have gained a better understanding of the geometrical nature of the singularities. Now assume that the robot is at a pose that is not singular i.e. for which the determinant of the full inverse kinematic jacobian is not 0. It may be interesting to define a "distance" to measure how "far" the pose is from a singularity. We quote the word "distance" as we may distinguish two different cases:

- the d.o.f. of the robot are only either translational or rotational
- the d.o.f. of the robot are a mix of translation and orientation

It is possible to define a *distance*, in the mathematical sense of a metric, between two poses only in the first case. In the second case we may rely only on a weaker notion, a *singularity index*.

Assume that we have a translational robot in a given pose M . We may indeed measure the closeness of M to a singularity by determining the closest singular pose M_s (an issue that has never been addressed to the best of our knowledge) and use the distance between M, M_s for determining the closeness to the singularity. But singularity has also a physical meaning, and indicates a change in the kinematic behavior of the robot (e.g. that some end-effector motion may become large for small changes in the actuated joints variables) and it may happen that two poses with the same distance to a singularity will exhibit different changes in kinematic behavior (see exercise 6.6 for the 3-UPU robot). Hence even for such robot it may be interesting to define a singularity index that is more indicative of such changes, an issue that we will address in this section.

Vogelwede (597) defines three conditions that a singularity index $\mathcal{S}(\mathbf{X})$ should fulfill:

- \mathcal{C}_1 : $\mathcal{S}(\mathbf{X}) = 0$ if and only if \mathbf{X} is a singularity

- \mathcal{C}_2 : if \mathbf{X} is non singular, then $\mathcal{S}(\mathbf{X}) > 0$
- \mathcal{C}_3 : $\mathcal{S}(\mathbf{X})$ must have a clear physical meaning

As singularity is an invariant kinematic property, we believe that we should add a fourth item:

- \mathcal{C}_4 : $\mathcal{S}(\mathbf{X})$ should be invariant under a change of units

We have presented in the previous chapter kinematic indices that may be used as singularity indices: the manipulability index $\sqrt{|\mathbf{J}\mathbf{J}^T|}$ as defined by Yoshikawa (640), the inverse of the condition number $1/\kappa$, and the smallest singular value of \mathbf{J}_{fk}^{-1} , which will be 0 at a singularity. All these indices have variants according to the inverse jacobian matrix on which they are applied, and to the matrix norm that is used for their calculation.

These indices, although they are commonly used, satisfy condition $\mathcal{C}_1, \mathcal{C}_2$ but not $\mathcal{C}_3, \mathcal{C}_4$. Furthermore, we have observed in the previous chapter, section 5.4.2.2, some coherence problems with these indices. Also we have also observed on the INRIA "left hand" an interesting phenomenon regarding the manipulability index: it peaks just before the singularity. This peak corresponds to an extremely stable position of the manipulator, at which large external forces may be balanced by small joint forces. However, this favorable characteristic exists only in a small part of the workspace, and a small motion away from this configuration will lead to a very bad conditioning for the robot.

Voglewede (597) defines various singularity indices M , either in the velocity or force domain: we will focus on the velocity domain and will consider the indices related to force in the "Statics" chapter. In the velocity domain Voglewede proposes a generic form of \mathcal{S} as

$$\mathcal{S}(\mathbf{X}) = \begin{cases} \min \dot{\mathbf{\Theta}}^T \mathbf{U} \dot{\mathbf{\Theta}} \\ \text{subject to } \dot{\mathbf{\Theta}} = \mathbf{J}^{-1} \mathbf{W}, \quad \mathbf{W}^T \mathbf{T} \mathbf{W} = 1 \end{cases}$$

He presents various choices for the matrices \mathbf{U}, \mathbf{T} so that $\dot{\mathbf{\Theta}}^T \mathbf{U} \dot{\mathbf{\Theta}}$ and $\mathbf{W}^T \mathbf{T} \mathbf{W}$ have a physical meaning. He shows also that the calculation of the index \mathcal{S} is equivalent to solving a generalized eigenvalue problem, a well known problem. However, for a robot exhibiting both translational and rotational d.o.f. it appears that the choice of the matrices \mathbf{U}, \mathbf{T} is equivalent to making a balance between translational and rotational motion, a balance that is somewhat arbitrary. Voglewede considers that the most meaningful choice for \mathbf{U} is a diagonal matrix whose diagonal element are the stiffnesses of the legs, while \mathbf{T} is the inertia matrix of the end-effector. With that choice, \mathcal{S} become the lowest natural frequency of the robot, a physical characteristic that makes sense from a control viewpoint. However this choice introduces the dynamics of the robot, while a singularity index may be interesting only

at the kinematic level (for example for characterizing the relation between the positioning accuracy of the robot and the active joint measurement errors).

Another approach is proposed by Wolf and Shoham (626) for robots whose full inverse kinematics jacobian is constituted of Plücker vectors. The principle is to determine the linear complex that is the closest to the variety spanned by the Plücker vectors of the inverse jacobian. This linear complex is found by using a method proposed by Pottmann (485). Physically it is shown that this index corresponds to finding the minimum of the instantaneous work generated by the wrenches provided by the legs while the end-effector is undergoing an infinitesimal twist. In a singular configuration the wrench will produce no work while the end-effector is moving and this index will be 0.

Apart from the manipulability index, a major drawback of all proposed singularity indices is that they usually do not have a closed form. They may be calculated numerically for a given robot and pose, but their minimum, average value and standard deviation over a given workspace or trajectory are difficult to ascertain, although they are important in practice. Indeed, when designing a robot, it is important to determine if it is singularity-free over a given workspace, or alternatively if a given trajectory of the robot is singularity-free. We will address this issue in the next section.

6.7. Singularity test

Previous sections have shown that Grassmann geometry allows us to obtain relations between the pose parameters in a singularity. In practice, however, an important problem is to determine whether there are singularities within a given workspace or trajectory or not. For example this problem is crucial during the design phase of a robot (checking if the whole useful workspace of the robot is singularity-free) or during trajectory planning. In both cases we are first interested by a fast straight yes-no answer; the location of the eventual singularities is of lesser importance.

In certain cases, the location of the singularity may be represented jointly with the workspace, thereby allowing a visual check. For instance, for planar parallel robots, Sefrioui (526) suggested a method for representing the workspace together with the singularity loci.

Our purpose is to present a generic method that will allow us to check the presence of a singularity for any robot type and for any parametric trajectory, or any workspace workspace defined in terms of constraints on the pose parameters. The workspace that has to be tested for the presence of singularity is therefore a m -dimensional finite variety, and the end-effector poses that belong to the variety have parameters that are continuous para-

metric functions of m parameters $\boldsymbol{\lambda} = \{\lambda_1, \dots, \lambda_m\}$. For example

- for a trajectory, the pose parameters will be defined as functions of the time parameter.
- for a translational robot, the poses belonging to a spherical volume will be defined by 3 parameters (e.g. the radius of the sphere and the latitude and longitude angles)

Note that, as the variety is finite, the parameters $\lambda_1, \dots, \lambda_m$ are bounded, i.e. each of them must belong to a known interval.

We have seen that a singularity is characterized by the cancellation of the determinant of the 6×6 full inverse kinematic jacobian (however for a n d.o.f. robot with $n < 6$ it may be sufficient to check only the cancellation of the $n \times n$ inverse kinematic jacobian if no constraint singularities may occur). Using symbolic computation, it is possible to calculate the determinant as an analytic function of the parameters in $\boldsymbol{\lambda}$. Furthermore it is reasonable to assume that one pose belonging to the variety is known, and that it is possible to compute numerically the determinant of J_{fk}^{-1} at this pose. Without loss of generality we may assume that this determinant is positive. The singularity test amounts to determining if there is another pose belonging to the variety, such that the determinant is less than or equal to 0, in which case any path between the two poses has to cross a singularity.

For that purpose we will use interval analysis, as summarized in the interval appendix. Given ranges for the parameters $\lambda_1, \dots, \lambda_m$, interval arithmetic allows us to compute a range $[a, b]$ such that the value of the determinant is included in this range for any value of the λ_i in their respective ranges. Consequently, if a is strictly positive, the value of the determinant cannot be negative, while if b is strictly negative, then the determinant will always be negative. Interval arithmetic may lead to $a < 0, b > 0$: in this case a bisection process is used on the parameters ranges. This process is summarized in the interval appendix and a detailed implementation of the algorithm is given in (413). Note however that some expertise in interval analysis is needed to get an efficient implementation. Our current implementation allows us to check a 6D workspace for a 6-UPS robot in a computation time that ranges from less than 1 second to 30 seconds.

One of the advantage of this algorithm is that it allows us to deal with uncertainties in the robot geometry, or with trajectory tracking errors. The determinant may be interval evaluated even if its analytical form has interval coefficients. Another advantage is that interval analysis may also be extended to determine the presence of a singularity within a workspace which is defined in term of joint coordinates under two assumptions:

- the workspace of the robot has only one connected component

- it is possible to determine if the region in the joint space corresponding to a hypercube in the task space belongs to the workspace

Another approach to singularity analysis is to determine some type of largest component that is included in the robot workspace and which is either singularity-free, or for which the inverse of a singularity index has a value in a given range. A variant of the previous algorithm allows us to determine, for example, the largest singularity-free box included in the workspace, or such that for any pose in the box a singularity index lies in a given range (for example see (82) for the Orthoglide robot).

A problem related to singularity indices is the classification, or *stratification*, problem: can we establish a classification of the singularities with respect to some properties in a way that is coordinate and dimension free?

6.8. Mechanisms in permanent singularity

Up to now we have considered that singularity should be avoided. However, we may consider a completely different approach: we may design a mechanism whose singular motion will be finite and controlled by additional actuators. For example the helical motion that is obtained for a linear complex may be controlled with only one actuator, and that may be of interest for various applications such as the manufacturing of complex shapes.

The problem is therefore to determine mechanisms whose singular motion is finite. Such a singularity is sometime called *structural* or *architectural*. For the 6-UPS mechanism, this problem was the subject of the *Prix Vaillant*, which was won by Borel (52) and Bricard (57). The problem was to determine under what conditions a rigid body may show continuous motion when some points of the rigid body are compelled to remain upon fixed spheres. Bricard (56) used a result of Chasles (1851) that demonstrated that if the anchor points A_i on the base are constrained to lie on a conic section and if the anchor points B_i on the platform are in projective correspondence with the A_i points, then the lines connecting the A_i, B_i belong to a linear complex (624). Bricard mechanisms were studied thoroughly by Lebesgue (348) and Baker (23).

A thorough analysis of the permanent singularity of the 6-UPS has been performed by Husty and Karger (258),(300)* and by Wohlhart (625)*, for planar parallel robots, a equiform planar platform, and finally for the general case of non planar base and platform. For this later case, Husty and Karger expand the determinant of the full inverse kinematic jacobian which is a function of the pose of the robot. They then examine for which condition all the constant coefficients of the determinant may be equal to 0. They show that a necessary condition for permanent singularity (called *self-motion* by these authors) is that 4 anchor points on the base or on the

platform must be collinear and then describe all possible singularity cases with their associated motion (which is a variety whose rank may be 1, 2 or 3). A noticeable point is that in many cases we may add legs to the robot without modifying the self motion and its nature.

Finally, as mentioned by Wohlhart (623), singularity motion for robot cannot be disconnected from the effects of clearance. We have already seen with the SNU 3-*UPU* robot that these clearances may extend the motion initiated in a singularity from infinitesimal to finite, or enlarge the range of the motion. In his paper, Wohlhart considers a robot in a singular position which admits arbitrary first, second, \dots , n th time derivative inputs, and defines their *degree of shakiness* as n . In terms of forward kinematics, the maximum degree of shakiness corresponds to the cancellation of all the coefficients of the direct kinematics polynomial. He then shows that the uncertainties in the pose of the robot grow considerably with the degree of shakiness.

6.9. Singularity-free path-planning and workspace enlargement

The singularity problem may be considered for a *trajectory*. We saw in section 6.7 that it is possible to detect if a singularity occurs on a given arbitrary trajectory. If this is the case, it is necessary to modify the trajectory to avoid the singularities.

Dasgupta (128) proposed an algorithm to obtain a singularity-free trajectory between two poses. The pose of the end-effector is taken as a function of one variable; the condition number is examined at discrete steps on the corresponding straight line trajectory. If it is lower than a fixed threshold, then the trajectory is modified in order to have a larger condition number. Later on, Dasgupta and co-workers suggested another approach (531) which is an adaptation of the well known potential energy method that is used for path planning of serial robots. They first define a function that is the sum of a kinetic energy like term (that attracts the end-effector to the goal pose) and a penalty term that increases as the robot comes close to the boundary of its workspace or to a singularity. They then use a numerical procedure to find a path that minimizes the integral of the function over a finite time horizon. Dash (130) has proposed another approach, based first on a sampling of the workspace, that allows us to determine singular points. These points are regrouped in clusters that are enclosed in polygons that are considered as obstacles for a global path planner. Then a local planner may modify the nominal path locally to avoid singularity.

Nenchev (443) considered this problem by using an optimal control approach that minimizes the error between the nominal path and the followed one. Bhattacharya (41) proposed a similar approach for a 6-*UPS* robot

where the velocity of the end-effector is determined by minimizing the distance to the nominal path under constraints on the leg forces. In a second approach Bhattacharya proposed to use as constraint that the determinant of the inverse kinematic jacobian is kept constant. Although the second approach may lead to leg forces that are larger than pre-set limits, it is significantly faster than the first approach. A similar optimization procedure was also proposed by Perng (472), with another penalty constraint, to avoid the singularity. Jui (295) and Ider (261) have shown that if at a singularity the drop of rank of the inverse kinematic jacobian is only one, then it may be possible to control the acceleration so that the forces in the legs remain bounded. An alternative is proposed by O'Brien (452) with passive joints braking: when locking some joints in the neighborhood of a singularity it is possible to avoid crossing it, at the expense of a loss of manipulability.

Another interesting problem related to singularities is to consider that they may limit the useful workspace of the robot. Wenger (617) shows how to manage the assembly mode of a planar robot to use the best of the robot workspace. The workspace of a robot may be separated into different regions, called *aspects* by Wenger, that are singularity-free and separated by singularity varieties. Usually the robot operates within the aspect which includes its initial assembly mode. But Hesselbach and co-workers propose using the dynamics of 2 and 3 d.o.f. robots to cross the singularity between two different aspects (232; 234). Another possibility is to use the kinematic redundancy to avoid singularities (370; 565). Determination of the aspects for spatial robots, sometime called the *partitioning problem*, is still an open problem. A special case for the generation of a singularity-free path is for $n - 1$ -axis machining with a n d.o.f. robot: in that case only $n - 1$ d.o.f. of the robot among the n possible are used. The free pose parameter may be used to avoid singularity (see (532; 412) for examples).

6.10. Singularity and design

In the previous sections we have presented some methods for dealing with the singularity problem:

- by checking that a given workspace or trajectory is singularity-free (section 6.7)
- by modifying the trajectory to avoid a singularity (section 6.9)

We have also seen in section 6.8 that it was possible in some cases to design a robot so that it exhibits finite motion during a singularity. The next logical step will be to determine the dimensioning of a robot so that it is singularity-free, at least in a given workspace. To the best of our knowledge the issue of using the dependence between J_{fk} and the design parameters to avoid the degeneracy of this matrix has never been fully addressed for

a parallel robot (i.e. by considering all design parameters of a robot as potential sources of change for the location of the singularity). We may just mention the related work of Zhou (658) that shows the importance of one design parameter on the location of the singularity of a four d.o.f. robot. Another approach is to find the design parameters that maximize the minimal value of a singularity index over a given workspace: we will address this approach in the "Design" chapter.

6.11. Exercises

Exercise 6.1: Determine the characteristics of the infinitesimal motion in a singular configuration of a 3- $R\underline{P}R$ planar parallel robot for a non-zero orientation angle.

Exercise 6.2: Determine the instantaneous rotation center of the motion of a 3- $R\underline{P}R$ planar parallel robot when the robot is in a singular configuration.

Exercise 6.3: Consider a 3- $R\underline{P}R$ planar parallel robot with joint centers coordinates:

$$\begin{aligned}x_{a_1} &= 0, & y_{a_1} &= 0, & x_{b_1} &= 0, & y_{b_1} &= 0, \\x_{a_2} &= 5, & y_{a_2} &= 0, & x_{b_2} &= 3, & y_{b_2} &= 0, \\x_{a_3} &= 1, & y_{a_3} &= 5, & x_{b_3} &= 2, & y_{b_3} &= 2.\end{aligned}$$

Establish, for a non-zero orientation angle, what is the value of y_c as a function of x_c , so that the robot is in a singular configuration. Show that joint forces generally tend to infinity if we apply a torque around C in these singular configurations.

Exercise 6.4: Explain, using Grassmann geometry, why the following configurations are singular for the *Delta* robot (99)* (a link denotes a member of a parallelogram):

1. when all the 6 links are parallel
2. when 4 links are parallel
3. when 4 links lie within one plane
4. when the 6 links lie within one plane

Indicate how many degrees of freedom will be found in each case.

Exercise 6.5: Using the previous exercise, determine the singular pose closest to a given pose

Exercise 6.6: Consider a 3- UPU robot designed so that the vectors \mathbf{s}_i in the full inverse kinematic jacobian, equation (6.7), are never coplanar or parallel. Assume that none of the legs may become vertical. For a given pose, determine the closest singular pose, and show that any pose in an horizontal plane is at a constant distance from its closest singular pose. Is the distance to the closest singular pose a convenient singularity index, according to the criteria presented in section 6.6 ?

Exercise 6.7: Consider a SSM with congruent platforms with a ratio k . Determine under what condition the manipulator is always in a singular configuration. Use the inverse kinematics equations, defining the rotation matrix by the following columns (x_1, x_2, x_3) (x_4, x_5, x_6) , (x_{10}, x_{11}, x_{12}) , and the position of the reference point C by (x_7, x_8, x_9) . Define the intermediate variables $U = x_1 x_7 + x_2 x_8 + x_3 x_9$, $V = x_4 x_7 + x_5 x_8 + x_6 x_9$, $W = x_7^2 + x_8^2 + x_9^2$, and take $a_i, b_i, 0$ as the coordinates of Bi . This exercise was inspired by Guozhen (209).

Exercise 6.8: Consider a SSM with joint coordinate:

$$\begin{aligned} A_1(-9, 13, 0) , \quad A_2(9, 13, 0) , \quad A_3(6, 5, 0) , \\ A_4(2, -10, 0) , \quad A_5(-2, 10, 0) , \quad A_6(-6, 5, 0) , \\ B_1(-1, 5, 0) , \quad B_2(1, 5, 0) , \quad B_3(4, -1, 0) , \\ B_4(2, -3, 0) , \quad B_5(-2, -3, 0) , \quad B_6(-1, 5, 0) . \end{aligned}$$

Calculate the numerator of the determinant of J^{-1} in terms of x_c, y_c, z_c when the orientation is $\psi = 0, \theta = \pi/2, \phi = 0$. What is the nature of the singularity curve in a plane for which z_c is constant?

Exercise 6.9: Using the previous exercise, express $|J^{-1}|$ in terms of the link lengths only.

Exercise 6.10: Explain how singularity conditions of the *Hexa* robot may be determined.

Problem 6.1: Determine whether the singular configurations define connected components for the solutions of the direct kinematics of spatial robot.

Problem 6.2: Determine the singular pose that is the closest to a given pose for a 3 d.o.f. parallel wrist

Problem 6.3: Classify the singularities of a MSSM, and distinguish those that do not lead to infinite joint forces.

Problem 6.4: Find the singularity condition of a MSSM as functions of the actuated joint variables (i.e. the length of the leg)

Problem 6.5: Express the determinant of the inverse kinematic jacobian of a MSSM as a function of the leg lengths and of the orientation angles (use the result of the previous problem).

Problem 6.6: Determine mechanisms that may exhibit permanent singularity apart of the 6-*UPS* and planar robots.

Problem 6.7: Determine mechanisms whose motion in permanent singularity is a variety of order greater than 3

Problem 6.8: Determine how to calculate the dimensioning of a given robot so that it is singularity-free in a given workspace

Problem 6.9: Determine the aspects of a given 6-*UPS* robot

Workspace

This chapter will present various methods for the determination of the workspace of parallel robots. Different types of workspaces will be defined, and algorithms for calculating them will be presented. The chapter will conclude with algorithms for the trajectory verification and motion planning of parallel robots within the workspace.

7.1. Workspace limits, representation and type

Parallel manipulators motions can be restricted by different factors: mechanical limits on passive joints, self-collision between the elements of the robot, limitations due to the actuators and singularity varieties that may split the workspace into separate components.

The main problem with the workspace representation of parallel robots is that the limitations on the d.o.f. are all usually coupled. Hence for robots having more than 3 d.o.f. there will be no possible graphical illustration of the robot workspace. This is not usually the case with serial robots. For example, the workspace of a 6 d.o.f. serial robot with a concurrent axis wrist may be represented by the 3D volume that may be reached by the center of the wrist - this illustrates the translations, and by the surface that may be reached by the extremity of the end-effector (which illustrates 2 degrees of freedom in rotation). The 3D volume depends only on the motion capability of the first three actuated joints, while the orientation uses only the last three joints. A graphical representation of the workspace of parallel robots will be possible only for 3 d.o.f robots. For robots with $n > 3$ d.o.f., workspace representation will be possible only if we fix $n - 3$ pose parameters. According to which types of parameters are fixed or to the constraint we impose on the parameter, we will obtain different *types of workspace*.

7.1.1. THE DIFFERENT TYPES OF WORKSPACES

The most usual types of workspace are:

- *constant orientation workspace* or *translation workspace*: all possible locations of the operating point C of the robot that can be reached with a given orientation

- *orientation workspace*: all the possible orientations that can be reached while C is in a fixed location
- *maximal workspace* or *reachable workspace*: all the locations of C that may be reached with at least one orientation of the platform.
- *inclusive orientation workspace*: all the locations of C that may be reached with at least one orientation among a set defined by ranges on the orientation angles. The maximal workspace is a particular case of inclusive orientation workspace for which the ranges for the orientation angles are $[0, 2\pi]$
- *total orientation workspace*: all the locations of C that may be reached with all the orientations among a set defined by ranges on the orientation angles
- *dextrous workspace*: all the locations of C for which all orientations are possible. The dextrous workspace is a particular case of total orientation workspace, the ranges for the rotation angles being $[0, 2\pi]$.
- *reduced total orientation workspace*: all the locations of C that may be reached with a subset of the orientation angles that may have any value in defined ranges, while the others may have arbitrary values. Such a workspace may be important for applications that do not involve all the d.o.f. of the robot. For example, if a 6 d.o.f. robot is used as a 5-axis machine-tool, the ϕ orientation parameter is not important

In this chapter we will focus on workspaces that are limited only by geometrical constraints. But other constraints may limit the useful robot workspace: for instance we have already seen that a threshold on the value of a singularity index may play a role in limiting the workspace of the robot (82); Kumar (336) introduces the notion of *controllably dextrous workspace* as the subset of the dextrous workspace which does not contain any singular configuration. The workspaces we will calculate in this chapter will always include such specialized workspaces. Even if we assume a perfect knowledge of the robot geometry, calculation of the workspaces is in general a complex task. We will thus suppose that this geometry is perfectly known, but we must remember that manufacturing errors may have an influence on the workspace (see for example (598) for an analysis of this influence).

7.1.2. ORIENTATION REPRESENTATION

The most usual orientation parameters are the standard Euler angles ψ, θ, ϕ and the pitch, yaw, roll angles. A problem with the Euler angles is that they exhibit a representation singularity for $\theta = 0$ as in this configuration the rotation angle is $\psi + \phi$. To solve this problem, Bonev (46) propose using modified Euler angles, the *tilt and torsion angles*, describing the orientation

by a first rotation of angle ϕ around the z axis, followed by a rotation of angle θ around y , and then a rotation of angle ψ , the torsion, around the new z axis. Bonev then discussed the best way to represent an orientation workspace. He considered 3 possibilities:

- a coordinate system whose axes represent the three Euler angles
- a spherical coordinate system, where ϕ, θ are the azimuth and zenith angles and ψ is the ray length
- a cylindrical coordinate system, where ϕ, θ are the circular coordinates and ψ is the z coordinate

Bonev concluded that the last choice offers the best orientation workspace representation. As an alternative Pernkopf proposed the use of the Euler parameters which may offer a better visibility for some motions (473).

7.2. Workspace calculation methods

Various approaches may be used to calculate the workspace of a parallel robot. We will summarize these methods in the next sections.

7.2.1. GEOMETRICAL APPROACH

The purpose of this method is to determine geometrically the boundary of the robot workspace. The principle is to deduce from the constraints on each leg a geometrical object \mathcal{W}_i that describes all the possible locations of \mathbf{X} that satisfy the leg constraints. One such object is obtained for each leg and the robot workspace is constituted of the intersection of all \mathcal{W}_i .

Consider for example a chain of a 6-*UPS* robot, and assume that the only constraint is that the length of the leg has to lie in the range $[\rho_{min}, \rho_{max}]$: the extremity B of the chain is constrained to lie in a volume which is comprised between the spheres with center A and radii ρ_{min}, ρ_{max} (figure 7.1). If we further assume that the orientation of the end-effector is constant, then each leg constrains C to lie in a similar region \mathcal{W}_i^i whose center is obtained by translating A_i by the constant vector $\mathbf{B}_i\mathbf{C}$. The workspace of the robot is then obtained as the intersection of the six \mathcal{W}_i^i .

We may also consider a *PRRS* chain in which the prismatic actuator is connected by a universal joint to a link of length l : the volume \mathcal{V} that may be reached by B is constituted of part of a cylinder with radius l and height $\rho_{max} - \rho_{min}$, topped by two hemispheres with radius l (figure 7.1). If the orientation is constant we can get \mathcal{W}_i^i from \mathcal{V} by a simple translation.

In some case such as the *Delta*, calculation of the workspace may be done directly by CAD (100) since this calculation is equivalent to the intersection of simple 3D volumes (see (138) for an analytical description of the workspace boundary). The geometrical approach has also been used for

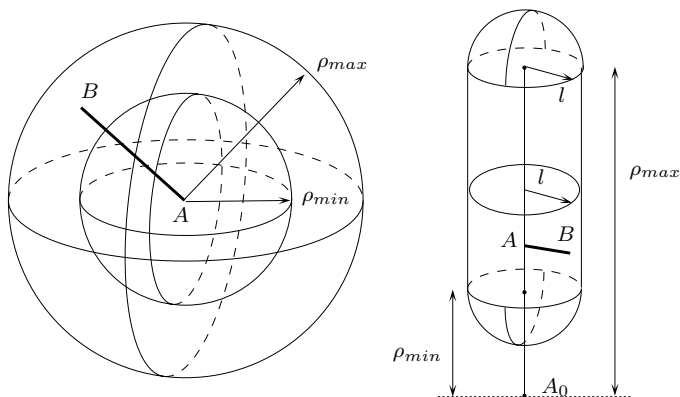


Figure 7.1. The volumes that can be reached by the B point of the chain of a parallel robot. On the left we have considered a chain of type $RRPS$ in which the P actuator has a length between ρ_{min}, ρ_{max} . On the right we have considered a chain of the type $PRRS$ with the same constraint on the P actuator and a link of length l connecting the RR and S joints.

spherical robots (6; 64) and by Arun(18) for the robot of figure 2.57. The choice of the task space may be important to simplify the calculation. For example Husty (255), shows that homogeneous coordinates with an appropriate kinematic mapping may be appropriate, since they allow a simple representation of the constraints for a planar 3- RPR robot.

This approach is usually restricted to 3D workspace and is able to deal mostly with the constraints on the joint coordinates, although we will see that joint limits and interference constraints may be taken into account in some cases.

The main interest of the geometrical approach is that it is usually very fast and accurate, and provides a minimal representation of the workspace which may be used to calculate efficiently some characteristics of the workspace, such as its volume. Its drawbacks are that it must be tailored to the considered robot, it may be difficult to take all constraints into account, and the minimal representation of the workspace may not be the most appropriate for tasks such as motion planning. A possible simplified approach is to compute only slices of the workspace, and to approximate the section of \mathcal{W}_l by polygons. This approach requires a good *computational geometry* library that is able to execute Boolean operations, whether intersection, union, or difference, on polygons with arbitrarily many edges.

7.2.2. DISCRETISATION METHOD

Numerous papers dealing with workspace calculation use methods based on the discretisation of the pose parameters, in order to determine the workspace boundary. In this discretisation approach, the workspace is covered by a regular grid, either cartesian or polar, of *nodes*. Each node is then tested to see whether it belongs to the workspace. The boundary of the workspace is constituted of the set of valid nodes, where at least one close neighbor does not belong to the workspace.

The advantage of this method is that it allows one to take into account all constraints. But this approach has many drawbacks:

- the accuracy of the boundary depends on the sampling step that is used to create the grid, the computation time grows exponentially with the sampling step, so that there is a limit on the accuracy.
- problems occur when the workspace possesses voids.
- the boundary representation may involve a large number of nodes
- the boundary is used for different operations such as determination of the workspace volume, inclusion of a trajectory in the workspace, etc. When performed on a boundary represented by a discrete set of poses, these operations are computer intensive. To avoid this drawback, Chablat (79) proposed storing the workspace representation as an octree structure that allows faster motion planning and volume calculation. Still, getting the structure is computer and memory intensive.

A web page allows us to use this method to calculate on-line the workspace of 6-UPS and 6-PUS robots^{□ AWE}.

7.2.3. NUMERICAL METHODS

Another approach to workspace calculation was suggested by Jo (292). Taking the constraints on the joint coordinates into consideration, he transformed the inequalities that are imposed by these constraints into equalities by introducing extra variables. He then considered the generalized coordinates (vector \mathbf{X}), the joint coordinates (vector Θ) and the variables (vector \mathbf{w}) that are introduced by the transformation of the inequalities into equalities. Let \mathbf{q} be the vector that is constituted of all of these unknowns. The structure of the mechanism leads to constraint equations on the components of \mathbf{q} which may be written in implicit form as $\Phi(\mathbf{q}) = 0$. Let J_Φ be the jacobian of the system, i.e. the matrix:

$$J_\Phi = \frac{\partial \Phi}{\partial \mathbf{q}} = \left(\left(\frac{\partial \Phi}{\partial \mathbf{X}}, \frac{\partial \Phi}{\partial \Theta}, \frac{\partial \Phi}{\partial \mathbf{w}} \right) \right)$$

The workspace boundary is obtained as the set of vectors \mathbf{q} , such that for a given \mathbf{X} , there will be not be a unique set of vectors Θ, \mathbf{w} . In other words,

the rank of the matrix:

$$\left(\left(\frac{\partial \Phi}{\partial \Theta}, \frac{\partial \Phi}{\partial \mathbf{w}} \right) \right)$$

is lower than its dimension. A numerical procedure is then used to calculate the pose of the platform where this condition is satisfied. We note, however, that Jo illustrated this approach only for the simple case of the calculation of the *constant orientation workspace* of a 6-*UPS* robot. The introduction of other constraints limiting the workspace would lead to a jacobian so large as to render the procedure quite difficult to manage. Adkins (1) and Haugh (221) manage to find a point on the boundary, and use a numerical continuation method to follow the boundary. Although this approach is general, Adkins and Haugh restrict their calculation for a constant orientation workspace, as the general problem will be very complicated. But these authors were able to predict the singularity barriers that may split the workspace into different *aspects* (an aspect is a maximal singularity-free component of the workspace). Instead of using a continuation method to follow the boundary, some authors have proposed formulating that point as a constrained optimization problem (544).

Another method is based on the principle that, for a pose on the boundary of the workspace, the velocity vector of the moving platform cannot have a component along the normal of the boundary. Agrawal (2) and Kumar (336) use this method for manipulators with only revolute joints, and applied it to various planar robots to calculate the maximal workspace and the dextrous workspace. The main drawbacks of this method are that prismatic actuators cannot be considered, and it is quite difficult to introduce the notions of mechanical limits and of interference between links.

An efficient method based on interval analysis is described in the interval analysis appendix. It has the advantages of being able to deal with almost any constraint and any number of d.o.f., and has proved to be efficient in computing the most difficult case of 6D workspaces of 6-*UPS* robots (411). However it provides only an approximation to the workspace (but up to an arbitrary accuracy and with an error bound on the error) and is relatively computer intensive.

Finally let us mention two special cases: micro-robots and wire robots. Arai (15) considers the micro-robot case; he supposes that the actuator motions are small enough so that he may consider the inverse jacobian to be constant. Moreover, as the orientation changes are small, he considers only the part \mathbf{J}_t^{-1} of the inverse kinematic jacobian corresponding to the translatory motions, and writes for each link j $\Delta^2 \Theta_a^j = \Delta \mathbf{X}^T \mathbf{J}_t^{-T} \mathbf{J}_t^{-1} \Delta \mathbf{X}$. This relation means that, for each link and for a fixed maximal $\Delta \Theta_a^j$, the center of the platform is inside a zone which is bounded by a quadric, and the workspace is the intersection of these quadrics.

For wire robots the tension in the wire should be positive and bounded above. Hence the computation of the workspace is no longer a geometrical problem but involves statics. This difficult problem is addressed in (285), (342)*, (422; 491; 508), (593)*.

The following sections will favor geometrical approaches that generally are very efficient for the determination of various types of workspace; algebraic geometry will play an important role in some of the algorithms.

7.3. Planar manipulators

In this section we will consider 3-RPR planar manipulators¹, as described in figure 7.2. The other types of planar robots, namely the 3-RRR, 3-PRR robots, will be studied only in the exercises, since their workspaces may easily be calculated with variants of the algorithms presented in this section.

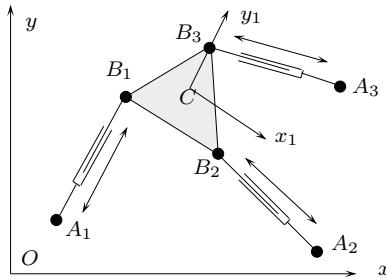


Figure 7.2. A 3-RPR planar robot.

7.3.1. CONSTANT ORIENTATION WORKSPACE

We will suppose that the orientation of the moving platform is fixed, and our goal is to determine all the possible locations for a reference point C of the platform. The suggested algorithms represent a direct application of the general method described in the previous section. We will successively examine the influence of various factors on all these possible locations.

7.3.1.1 Joint coordinates limits

We assume that the minimum length of the linear actuators is ρ_{min} while their maximum is ρ_{max} . As a result, the points B_i are located within annular zones: the external boundary is a circle C_{e_i} with center A_i and radius ρ_{max} , while the internal boundary is a concentric circle C_{i_i} with radius ρ_{min} .

¹Most algorithms that are described in this section are available by anonymous ftp in the directory coprin/Workspace/3-RPR

Since the orientation of the platform remains constant, when the point B_i moves within its zone, the point C moves within an analogous zone \mathcal{W}_i with boundaries which are circles $\mathcal{C}_{e_i}, \mathcal{C}_{i_i}$, represented by broken circles in figure 7.3, the center S_i of which may be obtained by translating A_i by the vector $\mathbf{B}_i\mathbf{C}$. If the constraints on link i are satisfied, then C must be inside

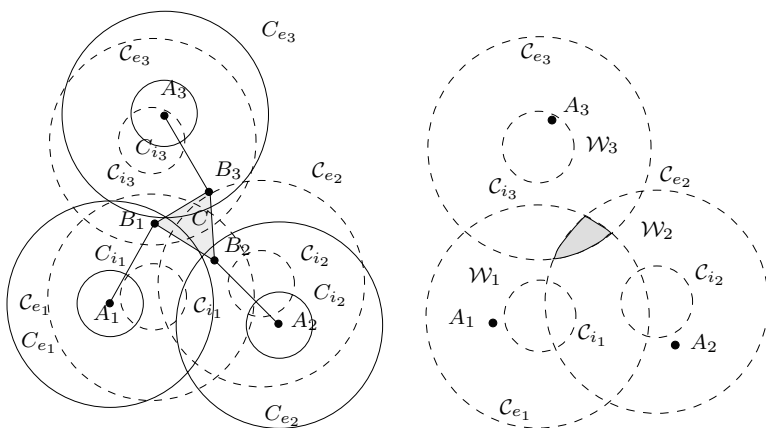


Figure 7.3. On the left, the geometrical elements that take part in the determination of the constant orientation workspace of a 3-*RPR* planar robot. On the right, the workspace as deduced from these objects (in grey).

\mathcal{W}_i . The workspace is obtained when the constraints on all links are satisfied and will therefore be the intersection of the three annuli $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3$; this is easily calculated. The boundary of the workspace will be constituted of circular arcs from the three annuli (figure 7.3).

7.3.1.2 Mechanical limits on the passive joints

In this section we suppose that in addition to the joint limitations, the joints located on the base have limited rotational capacities. We assume that the links may move only within an angular sector with amplitude α_i . The allowed zones \mathcal{W}_i therefore are now annular sectors; their intersection constitutes the workspace. The boundary of the workspace is therefore constituted of circular arcs and segments (figure 7.4). We may also impose similar constraints on the joints located on the platform (see exercise 7.1).

7.3.1.3 Leg interference

In this section we will assume that the legs cannot intersect. Assume that M is an intersection point between the legs A_iB_i, A_jB_j such that

$$\mathbf{A}_i\mathbf{M} = \lambda_1\mathbf{A}_i\mathbf{B}_i \quad \mathbf{A}_j\mathbf{M} = \lambda_2\mathbf{A}_j\mathbf{B}_j$$

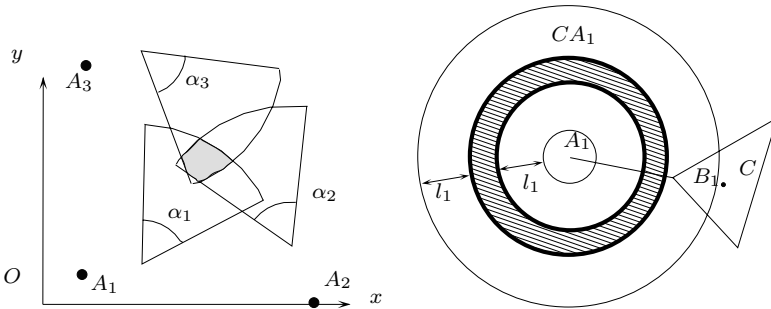


Figure 7.4. On the left in grey, the workspace when the constraints are the joint coordinate limits and the mechanical limits placed on the joints located on the base. On the right the dextrous workspace of C , taking only the constraints on link 1 into account (in hatched lines, $l_1 = \|\mathbf{CB}_1\|$).

For M to be an intersection point of the line segments A_iB_i, A_jB_j , we must have $\lambda_1, \lambda_2 \in [0, 1]$. Solving the above system in the coordinates of M and in λ_1, λ_2 we get that $\lambda_m = N_m/D_m$ with $m = 1, 2$. The equations $N_m = 0, N_m = D_m, D_m = 0$ define three lines for the location of C that share a common point. Each of these lines splits the $x_c - y_c$ plane into 2 regions: for example the line $N_m = 0$ leads to regions such that $N_m > 0$ and $N_m < 0$. Consider now the 2 infinite polygons $\mathcal{R}_1^m, \mathcal{R}_2^m$ obtained as the intersection of the regions $(N_m > 0, D_m > 0, N_m < D_m), (N_m < 0, D_m < 0, N_m > D_m)$: for each location of C in these polygons, we have $\lambda_m \in [0, 1]$. The intersection of all the pairs pairs of infinite regions $(\mathcal{R}_k^1, \mathcal{R}_l^2)$ with k, l equal to 1 or 2 will define all the locations of C at which the legs i, j will intersect.

7.3.2. ORIENTATION WORKSPACE

We will now suggest the basis for an algorithm that would provide a convenient representation of the workspace that may be reached in rotation for a 3-*RPR* planar manipulator with center in a fixed position; we represent a possible orientation by the displacements of a point that is fixed on the platform, and which is not its center. This approach may be generalized to other types of planar robots.

We first determine the possible locations of the points B_i . These points must be located within the annular zones imposed by the constraints on the link lengths. However, for a fixed center, each of these points is also situated on the circle C_{B_i} with center C and radius $\|\mathbf{CB}_i\|$ (figure 7.5). As a result, the point B_i is located on the portions $C_{B_i}^j$ of C_{B_i} that belong to the annular zone. We now consider a point M that is rigidly connected to the platform.

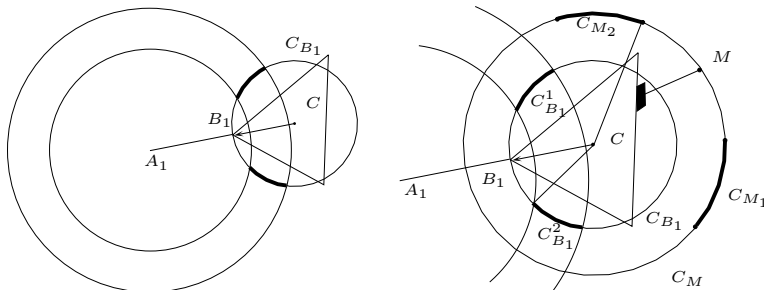


Figure 7.5. On the left, the possible displacements of a point B_i when point C is fixed (in bold). On the right, the possible displacements of a point M that is linked to the platform, when point C is fixed, taking into account the constraints on the links i.e. the parts C_{M_1}, C_{M_2} of the circle with center C , and radius $\|\mathbf{CM}\|$, and that were obtained from the allowed zones for $B_1, C_{B_1}^1, C_{B_1}^2$.

This point is located on a circle C_M with center C . The angle between B_i and M being fixed, we may calculate from the allowed zones for B_i , the zones C_{M_j} allowed for M , by considering only the constraints on the link i (figure 7.5). The allowed motions of M are the intersection of all the arcs C_{M_j} when all the links are taken into account. We may easily introduce constraints on the joints, as studied in the previous section. As for the interference between the legs i, j , the equations $N_m = 0, N_m = D_m, D_m = 0$ defined in section 7.3.1.3 lead, for a fixed value of x_c, y_c , to up to two solutions in the rotation angle. Consequently the circle on which B_i lies will be split into circular arcs. The interesting arcs are the ones obtained for $(N_m > 0, D_m > 0, N_m < D_m), (N_m < 0, D_m < 0, N_m > D_m)$ that lead to λ_1, λ_2 in the range $[0, 1]$. These arcs are reported on the circle C_m and if they intersect, then the intersection corresponds to the location of M for which legs i, j intersect.

7.3.3. DEXTROUS WORKSPACE

The *dextrous workspace* is the set of locations of the reference point C for which any orientations can be reached. We consider a 3-RPR robot and a particular point C_1 of this workspace; as all orientations are possible, a point B_i must be able to describe a circle with center C_1 and radius $\|\mathbf{CB}_i\|$. This circle must be contained within the annular zone of link i . Therefore if we take into account only the constraints on link i , the dextrous workspace is the circular annular region CA_i with center A_i and internal and external radii $\rho_{min_i} + \|\mathbf{CB}_i\|$ and $\rho_{max_i} - \|\mathbf{CB}_i\|$. There is such a region only if $\rho_{max_i} - \rho_{min_i} \geq 2\|\mathbf{CB}_i\|$ (figure 7.4); the dextrous workspace is the intersection of the three zones CA_i . This method may also include

constraints on the joints and legs interference.

7.3.4. MAXIMAL WORKSPACE

The *maximal workspace* is defined as all the locations of C that may be reached with at least one orientation. The problem of determining the maximal workspace for 3-*RPR* robots was first mentioned by Kassner (304); he noticed that the boundary of this workspace is constituted of circular and sextic arcs, but determined these elements by a discretisation method. We will present the main points of a method for calculating the boundary of the maximal workspace of 3-*RPR* robots; details of this algorithm are described in (408). Other types of robots are considered in the exercises.

We first note that it is easy to determine whether a point does belong to the maximal workspace or not. We need to determine only whether at least one orientation is possible for this position. Let us first consider point B_1 for the position of C that we are studying. This point may move on the circle C_1 with center C and radius $\|\mathbf{CB}_1\|$. We then calculate the intersection of C_1 with the annular region that corresponds to the constraints on the length of link 1; this is bounded by two circles C_{max}^1, C_{min}^1 centered in A_1 and with radii $\rho_{max}^1, \rho_{min}^1$.

If there is no intersection point, we test whether the circle C_1 is inside C_{max}^1 and outside C_{min}^1 ; this may easily be done by testing whether point C belongs to C_{max}^1 but not to C_{min}^1 . If this is the case, then all orientations around C are allowable for the moving platform, as far as the constraints on link 1 are concerned. If C_1 is outside C_{max}^1 or inside C_{min}^1 , then no orientation is possible and C does not belong to the maximal workspace. The same method is repeated for links 2 and 3.

Let us now suppose that at least one of C_1, C_2, C_3 possesses an intersection with its annular region. To each intersection point there corresponds a rotation angle, and we sort all these angles, in order to obtain intervals I_n^i (figure 7.6). We thus obtain 3 lists of possible intervals for the 3 links. We calculate the intersection of all the triplets of intervals made from elements belonging to each list. If this intersection is not empty, then the point belongs to the maximal workspace. We note that this procedure allows us to determine both whether the point belongs to the maximal workspace and what the possible values of the rotation angle are.

We will now determine the maximal workspace for a specific point of the end-effector, namely point B_3 ; for a different point, the algorithm uses the same principle although it is slightly more complex. This point must lie within its corresponding annular region. We then note that if B_3 is located on the boundary of the maximal workspace, then at least one of the links will possess a length that will correspond to an extremum. We may then

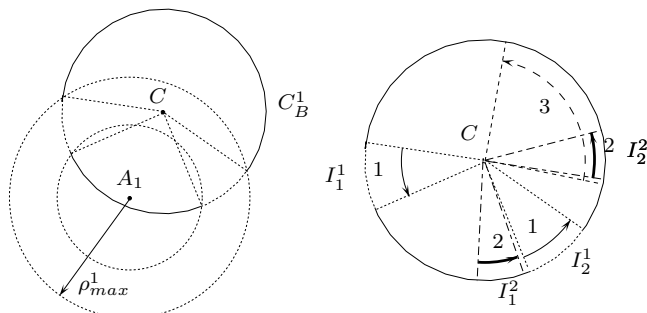


Figure 7.6. On the left, if C is in a fixed position, then the point B_i may describe a circle C_B^i centered at C . The possible positions of B_i , and thus the rotation angles around C are the intersections of the circles C_B^i with the circles in dashed lines, with centers A_i and radii $\rho_{max}^i, \rho_{min}^i$. We thus obtain, for the three links, three lists of intervals that define the possible rotation angles. If these three lists have an intersection that is not empty, then point C belongs to the maximal workspace. In the figure, the intersection is empty: the considered point does not belong to the maximal workspace.

distinguish various cases according to the number of links that have a length at an extremum:

- a link is at an extremum: if the link i is at an extremum and B_3 on the boundary, the robot must be in a configuration such that points A_i, B_i, B_3 are aligned. Three cases are possible according to the order of these points: $A_i B_i B_3, A_i B_3 B_i, B_3 A_i B_i$. Point B_3 then lies on the circles $C_{B_3}^i$, centered at A_i . We thus obtain a list of circles that could potentially be part of the boundary of the workspace.
- two links are at extrema: when the lengths of links 1 and 2 are fixed, then B_3 lies on the coupler curve of a 4-bar mechanism, i.e. a sextic, as studied in the "Direct kinematics" chapter. There are 4 sextics to consider; they correspond to the different possible combinations for the lengths of links 1 and 2: $(\rho_{max}^1, \rho_{max}^2), (\rho_{max}^1, \rho_{min}^2), (\rho_{min}^1, \rho_{min}^2), (\rho_{min}^1, \rho_{max}^2)$.

Figure 7.7 presents the different geometric elements that play a role in a typical example of maximal workspace calculation. All these elements are put together in a single list. The algorithm then proceeds by calculating the intersection between all pairs of element of the list. We note that this stage implies the calculation of the intersection of the coupler curves of two 4-bar mechanisms: a solution to this problem was presented by Innocenti (271). After having calculated the intersection points, we consider each element of the list and its component between two successive intersection points. The third stage consists in determining what components belong to the boundary. For each component, we determine whether it is part of the boundary of

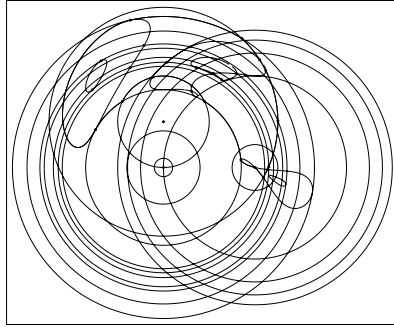


Figure 7.7. All the geometric elements that play a role for maximal workspace calculation: we have circles which are obtained when a link has a length at an extremum and sextics that we get when two links are at an extrema. We will have to calculate all the intersection points between these elements

the maximal workspace by taking an arbitrary point on the component (for example its middle point). We then compute the inverse jacobian kinematic matrix at this point, the unit normal vectors $\mathbf{N}_1, \mathbf{N}_2$ of the component at this point together with the joint velocities that correspond to a cartesian velocities of the end-effector directed according to $\mathbf{N}_1, \mathbf{N}_2$. The signs of the joint velocities that are obtained allow us to conclude whether the component belongs to the boundary or not. A component of the boundary will be such that a displacement along the outward normal will lead to a violation of the constraints on the lengths, whereas a displacement in the opposite direction will not violate these constraints. For example, for the component of the sextic corresponding to $\rho_{max}^1, \rho_{max}^2$, if the joint velocities $\dot{\rho}_1, \dot{\rho}_2$ are both positive for the normal \mathbf{N}_1 (and therefore negative for \mathbf{N}_2), then the component belongs to the boundary, since a motion according to \mathbf{N}_1 would lead to an increase of lengths ρ_1, ρ_2 , which already are at their maximum. We consider, for example, the planar robots with dimensions defined in figure 7.8; figure 7.9 presents their maximal workspace. It must be noted that we get a zone \mathcal{Z} which may be split by arcs of sextics into connected sub-zones $\mathcal{Z}_1, \mathcal{Z}_2, \dots$. According to the initial assembly mode of the robot, the actual maximal workspace will be either the zone \mathcal{Z} or will be reduced to the sub-zone \mathcal{Z}_i in which the initial assembly mode is located.

7.3.5. INCLUSIVE ORIENTATION WORKSPACE

It may also be interesting to calculate all the locations of the reference point that may be reached by the robot with at least one orientation within a given interval. The maximal workspace that we presented in our last section

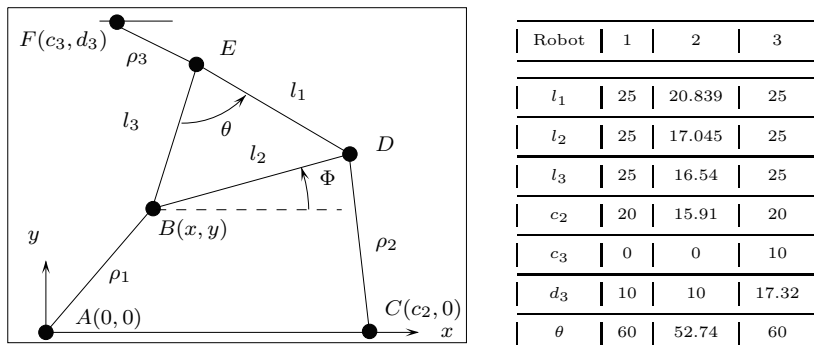


Figure 7.8. Notation for planar robots

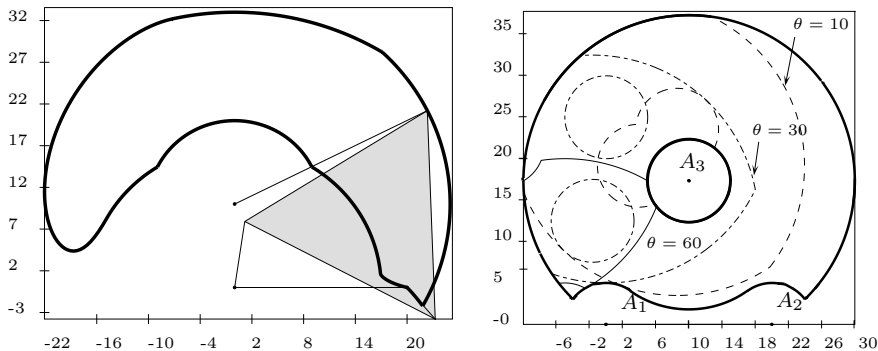


Figure 7.9. On the left, maximal workspace of robot 1, when the links have a length within $[2,8]$, $[5,25]$, $[10, 25]$. On the right and in bold, maximal workspace of robot 3 when the links have a length within $[5, 20]$, $[5, 20]$, $[5, 20]$. The dotted areas represent constant orientation workspaces for various orientations.

is only one particular case of this, with an interval of $[0, 2\pi]$.

We assume that the reference point is B_3 . As for the principle, the algorithm for calculating this workspace is similar to the previous one. We can determine whether a point is situated within this workspace, since we can calculate the possible orientation angle intervals for each point. We have to check only whether the intersection of these intervals and the orientation interval is not empty.

The second stage is different, since for determining the components we consider not only the intersection points but also those for which the orientation is equal to one of the extremities of the orientation interval; note that for each element of the list, there is only one platform orientation for a given position of B_3 on the element.

The third stage consists in determining whether the component belongs to the boundary. We therefore test whether a displacement along the normal to the component leads to a violation of the constraints on the lengths, and also test whether the orientation of the middle point actually belongs to the orientation interval. Figure 7.10 presents workspaces that are calculated for various orientation intervals.

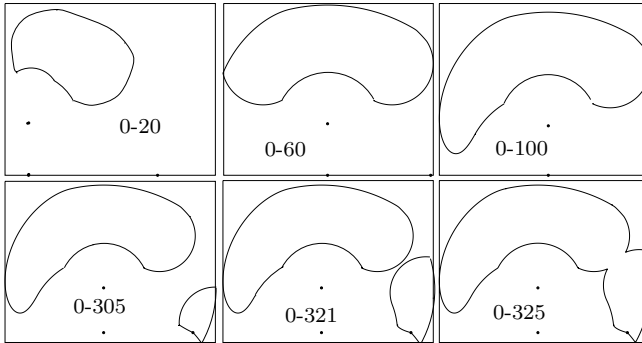


Figure 7.10. Inclusive orientation workspaces for different orientation intervals for robot 1 (links lengths [2,8], [5,25], [10, 25])

7.3.6. TOTAL ORIENTATION WORKSPACE

The *total orientation workspace* consists of all the locations of the reference point that may be reached by the robot with all possible orientations within a given interval $[\theta_i, \theta_j]$. The dextrous workspace is an example of total orientation workspace, the interval being $[0, 2\pi]$.

We can determine whether a point belongs to this workspace: we need only calculate the orientation intervals that are allowed at this point, and to check that one of these intervals contains the orientation interval. For a point on the boundary of the total orientation workspace, a link of the robot will have an extreme length. Clearly, at least one link must have an extreme length, but two or three of them cannot have extreme lengths at the same time, as in that case the orientation of the manipulator would be unique.

A point may be on the boundary of the workspace because one of the ends of the interval which gives the possible orientations at this point corresponds to one end of the orientation interval θ_i, θ_j . For a given orientation of the platform, when point B_i moves on one of the circular boundaries \mathcal{G}_i of the annular regions, the point B_3 moves on the corresponding circle of the annular region obtained by translating the first by the vector $\mathbf{B}_1\mathbf{B}_3$; vector $\mathbf{B}_1\mathbf{B}_3$ is constant since the orientation is fixed. For

each extremity θ_i, θ_j of the orientation interval, we thus obtain 6 circles as potential boundary elements, i.e. a total of 12 circles. Their centers and radii are $(A_3, \rho_{max}^3), (A_3, \rho_{min}^3), (A_1 + \mathbf{B}_1\mathbf{B}_3, \rho_{max}^1), (A_1 + \mathbf{B}_1\mathbf{B}_3, \rho_{min}^1), (A_2 + \mathbf{B}_2\mathbf{B}_3, \rho_{max}^2), (A_2 + \mathbf{B}_2\mathbf{B}_3, \rho_{min}^2)$.

A point may also lie on the boundary of the workspace because for an orientation θ in the interval $[\theta_i, \theta_j]$, one of the links reaches an extreme length. As the point belongs to the boundary, the circular arc with center B_3 , radius $\|\mathbf{B}_3\mathbf{B}_1\|$, and angle $|\theta_i - \theta_j|$, which is described by B_1 when the orientation of the platform varies between θ_i and θ_j , must be included within \mathcal{G}_1 ; furthermore this arc will be tangent to this region at one point, for example to the circle with radius ρ_{max}^1 (figure 7.11). This tangency

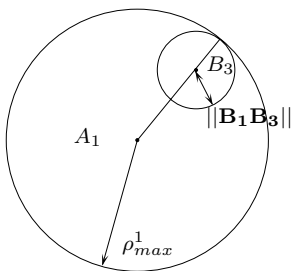


Figure 7.11. The point B_3 may belong to the boundary of the total orientation workspace if the arc of circle described by the point B_1 when the end-effector rotates around B_3 with an angle from θ_i to θ_j is included in the annular region \mathcal{G}_1 of point B_1 , and is tangent at some point to one of the circles of the annular region.

implies that in this configuration B_3 is located on a circle with center A_1 , and radius $\rho_{max}^1 - \|\mathbf{B}_1\mathbf{B}_3\|$. This therefore introduces the circles, with centers and radii given below, as potential elements of the workspace boundary: $(A_1, \rho_{max}^1 - \|\mathbf{B}_1\mathbf{B}_3\|), (A_1, \rho_{min}^1 - \|\mathbf{B}_1\mathbf{B}_3\|), (A_2, \rho_{max}^2 - \|\mathbf{B}_2\mathbf{B}_3\|), (A_2, \rho_{min}^2 - \|\mathbf{B}_2\mathbf{B}_3\|)$, i.e. 4 extra circles. The total orientation workspace is then obtained as the intersection of the 16 circles. Figure 7.12 presents examples of total orientation workspaces.

7.4. 3-UPU manipulator

We consider a translational 3-UPU robot. Given a range for the stroke of the prismatic actuator, the location of the B_i point is a spherical shell centered at A_i as seen in figure 7.1. As the orientation of the end-effector is constant, point C lies inside a similar spherical shell centered at $A_i + \mathbf{B}_i\mathbf{C}$. The workspace of the robot is thus the intersection of three spherical shells. We will detail the calculation of this intersection in the next section.

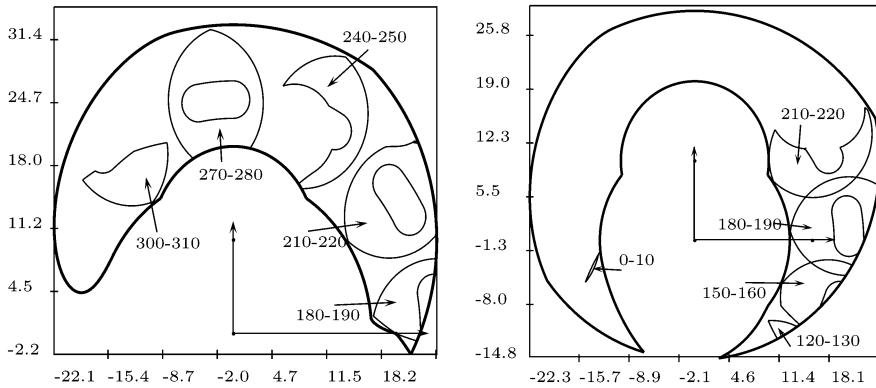


Figure 7.12. Examples of total orientation workspaces for robots 1 and 2. The maximal workspace is shown in bold.

7.5. 6-UPS manipulator

We now consider a 6-UPS, and examine the various types of workspaces². Other examples, such as the *Hexa* robot, are presented in the exercises, while Bonev (45) presents an extensive study of workspace calculation for 6-PUS robots.

7.5.1. CROSS-SECTIONS OF THE CONSTANT ORIENTATION WORKSPACE

The general calculation principle presented in section 7.2.1 was applied for the 6-UPS robot first by Gosselin (188) and is used here to examine the possible displacements of the joint centers on the moving platform. We note that points B_i are located within the volume bounded by the two concentric spheres centered at A_i , having radii which are the maximal and minimal link lengths.

The intersection of this volume with the cross-section plane will therefore be either an empty region (and the workspace will then also be empty), or a region that is bounded by one or two concentric circles i.e. an annular region. The zone that is allowed for C with respect to link i is then obtained by translating this region by the vector $\mathbf{B}_i\mathbf{C}$. If all the links are taken into account, the intersection of the plane and the workspace is a region that corresponds to the intersection of 6 annular regions. Figure 7.13 presents two examples; for simplification we have limited the number of annular regions to 3. Consequently the boundary of a cross-section of the

²Most of the algorithms described in this section are available by anonymous ftp, directory `coprin/Workspace/Gough`

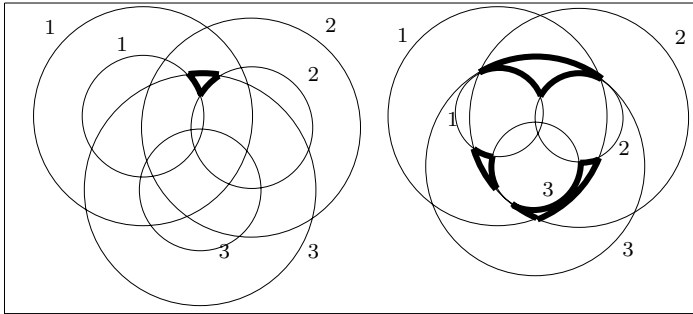


Figure 7.13. In bold, the boundary of a constant orientation workspace in a plane. The boundary is obtained as the intersection of 3 annular regions.

workspace will be constituted of circular arcs. This structure leads to a rapid determination of the boundary of the workspace without having to rely on a discretisation method. It will also allow us to calculate the area of a cross-section of the workspace.

Such an approach however takes into account only the limits on the joint coordinates. We will see that this method may be generalized so that it may also consider the other factors having an influence on the workspace.

Note that sometimes it may be interesting to approximate the boundaries of the region described by B_i by polygons, to obtain a rapid first estimate of the workspace as the intersection calculation is easier.

Figure 7.15 shows cross-sections of the workspace in a horizontal plane for INRIA *left hand*, for which the stroke of the linear actuators is 3 cm, when the rotation matrix is the identity. The shape and area of the cross-

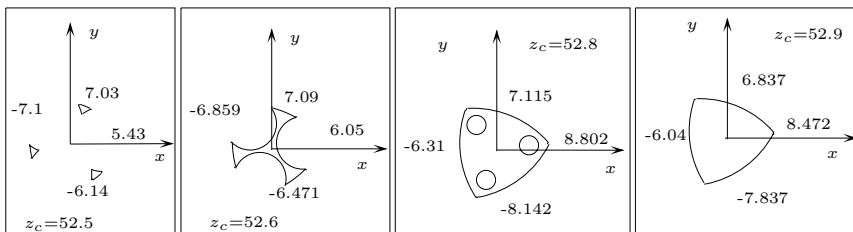


Figure 7.14. Horizontal cross-sections for various z_c of the workspace for INRIA *left hand* with an orientation defined by $\psi = 30^\circ, \theta = \phi = 0^\circ$.

section are very sensitive to the orientation angles. Figure 7.14 presents new sections for an orientation defined by the Euler angles $\psi = 30^\circ, \theta = \phi = 0^\circ$.

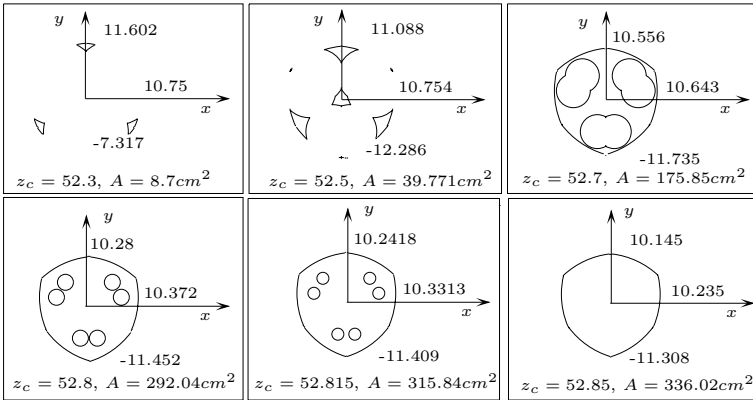


Figure 7.15. Horizontal cross-sections for various z_c of the workspace for INRIA left hand with rotation matrix $R = I_3$.

7.5.2. 3D CONSTANT ORIENTATION WORKSPACE

The workspace may also be calculated directly as a volume, although this is more complicated. Thus, for the 6-UPS robot, we will have to calculate the intersection of volumes that are limited by two concentric spheres, as noted by Gosselin (193). Calculation of this intersection is carried out in the following manner:

- calculate the intersection circles of each pair of spheres among the 12 spheres that bound the volumes.
- calculate the intersection of all of the previous circles which belong to one same sphere.
- determine the various circular arcs for each intersection circle, as defined from the intersection points at the previous stage
- test each arc to determine whether it belongs to the boundary of the workspace or not. This is done by considering the middle point of each arc and checking whether the constraints are satisfied for this location.

The 3D graphical representation is obtained by drawing the circular arcs that do belong to the boundary. Figure 7.16 presents an example of 3D workspace.

7.5.2.1 Workspace area and volume

It will generally be easy to calculate the area of a cross-section of the workspace with the help of the analytical description of the boundary, or by using a polygonal approximation if necessary. For the 6-UPS robots, we may apply Gauss’s divergence theorem that gives the area A of a planar

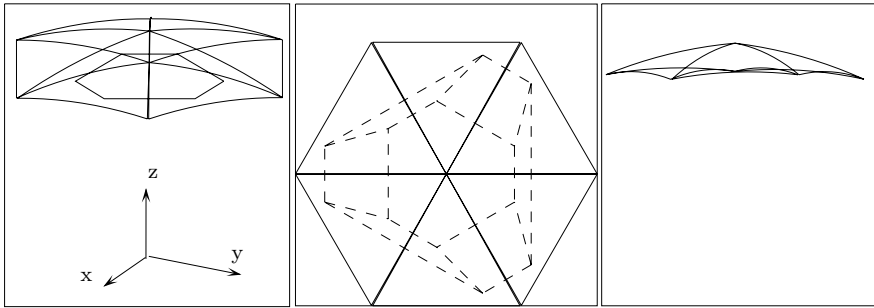


Figure 7.16. 3D workspace for INRIA left hand ($\psi = \theta = \phi = 0^\circ$).

cross-section by

$$A = \frac{1}{2} \int_{\partial\Omega} \mathbf{s} \cdot \mathbf{n} d\partial\Omega \quad (7.1)$$

where $\partial\Omega$ represents the boundary of the region, \mathbf{s} the position vector of an arbitrary point on $\partial\Omega$ and \mathbf{n} the normal unit vector that is oriented toward the exterior of the surface delimited by the curve $\partial\Omega$. As the workspace is here defined by a list of circular arcs, we may write equation (7.1) as

$$A = \frac{1}{2} \sum_{i=1}^{N_a} A_i, \quad (7.2)$$

with

$$A_i = \int_{\partial\Omega_i} \mathbf{s} \cdot \mathbf{n} d\partial\Omega_i, \quad (7.3)$$

where N_a represents the number of arcs constituting the boundary, and $\partial\Omega_i$ is the i -th arc. If the arc has center with coordinates $[h, g]$, and radius r , and if its extremities are defined by the angles θ_1 and θ_2 , we may then express s and \mathbf{n} by

$$\mathbf{s} = \begin{bmatrix} h \\ g \end{bmatrix} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}, \quad (7.4)$$

$$\mathbf{n} = \begin{cases} [\cos \theta, \sin \theta]^T & , \text{ if the arc is on the external boundary,} \\ [-\cos \theta, -\sin \theta]^T & , \text{ if the arc is on an internal boundary.} \end{cases} \quad (7.5)$$

We then obtain

$$A_i = hr[\sin \theta_2 - \sin \theta_1] + gr[\cos \theta_1 - \cos \theta_2] + r^2[\theta_2 - \theta_1], \quad (7.6)$$

for an external arc and, for an internal arc

$$A_i = -hr[\sin \theta_2 - \sin \theta_1] - gr[\cos \theta_1 - \cos \theta_2] - r^2[\theta_2 - \theta_1] \quad (7.7)$$

Calculation of the workspace volume may be easily obtained from the area of cross-section by assuming that the volume varies linearly between two cross-sections if the distance between the two cutting planes is small enough. Figure 7.17 presents the variations of the volume of the workspace according to the actuator strokes. It seems that the volume is approximately proportional to the cube of the stroke, as conjectured by Masory (385). Such a conjecture is reasonable, as if the stroke is very large, the workspace will be approximately a sphere whose radius will be the stroke.

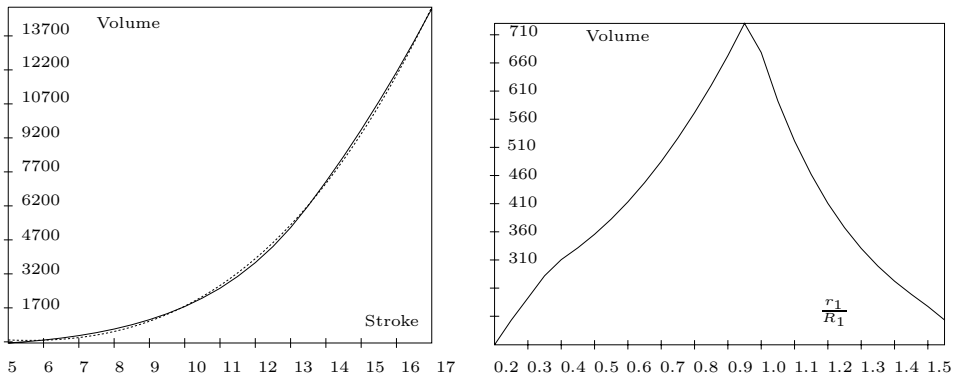


Figure 7.17. On the left, the volume of the constant orientation workspace for INRIA left hand as a function of the stroke of the actuators (in thin lines). The dotted lines represent the degree 3 polynomial that approximates this variation. On the right, the volume of the constant orientation workspace as a function of the ratio (radius of the platform)/(radius of the base).

Figure 7.17 shows the volume of the workspace according to the ratio between the radius of the platform and the radius of the base. We notice that this volume is maximal if the radius of the platform is approximately identical to the radius of the base.

7.5.2.2 Mechanical limits on the joints

This section will generalize the method presented in the previous section to include mechanical limits on the passive U, S joints. Our aim is to model the mechanical limits geometrically so that we can use it for workspace calculation and to allow flexibility in the modeling. We will first examine how the constraints on the joints that are attached to the base may be modeled.

7.5.2.2.1 Model for the mechanical limits

The mechanical limits may be defined by a surface that bounds the volume in which the axis of the link that is connected to the joint must be situated. For an U joint, this surface is usually quite complicated. For an S joint, classical commercially available joints have a range of motion that is restricted by a cone. But many robots use a combination of U and R joints that offer a larger range of motion. Hence we have to look for a model that is flexible enough to deal with all cases, while sufficiently simple to enable the calculation. An appropriate model for this surface is a pyramid with an appropriate number of planar faces. Indeed this model offers a high flexibility: for example a cone may be approximated in a conservative way by a pyramid with many faces. On the other hand, the pyramid model allows for a very simple workspace constraint surface which is a polyhedron (or a polygon if we have to calculate a cross-section). Note that this model allows us to deal with an U joint having skew axes, a topic addressed by Pernkopf (473).

For a given joint we have therefore to define this pyramid according to the constraint on the joint. We define a reference frame $(A_i, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r)$ that is attached to the pyramid with apex A_i (figure 7.18). Each face i is defined by the unit vector \mathbf{n}_i normal to the face.

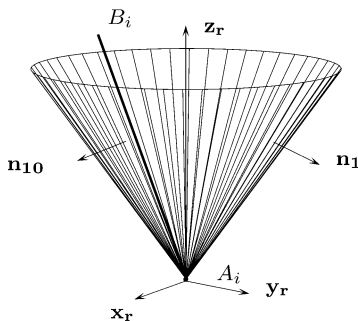


Figure 7.18. A pyramid with planar facets that models the constraints on the joint at A_i . In this example the pyramid is an approximation to a cone. Each face i is defined by its normal vectors \mathbf{n}_i . If the constraints on the joint are satisfied, the segment $A_i B_i$ must lie in the interior of the pyramid.

7.5.2.2.2 Possible locations of the centers of the joints

We assume that an analysis of the parallel robot has allowed us to establish that the points B_i are located within a volume \mathcal{V}_a if the constraints on the joint coordinates are satisfied. The intersection of this volume with the pyramid then defines the volume for point B_i for which the constraints on the joint coordinates, as well as the constraints on the passive joints, are

satisfied. If we look only for cross-sections of the workspace, we will first have to calculate the intersection \mathcal{P}_a of \mathcal{V}_a with the cutting plane, then the intersection \mathcal{P}_p of the pyramid with the cutting plane (i.e. a polygon). The allowed region for point B_i is then the intersection of \mathcal{P}_a with \mathcal{P}_p . Having determined the possible zones for the B_i , we obtain, by a simple translation, the allowed zones for C , the workspace then being the intersection of these zones. For example, we saw that for a 6-UPS robot, the zones \mathcal{P}_a are annular regions. The intersection of \mathcal{P}_a with \mathcal{P}_p will thus be a geometrical object, the boundary of which will be constituted of segments and circular arcs; this is commonly called a *generalized polygon*. The workspace is obtained as the intersection of 6 generalized polygons; this is easily calculable (402). Figure 7.19 shows a 3D representation of the constraints on the location of the B_i , either due to the constraints on the lengths, or the mechanical limits on the joints attached to the base.

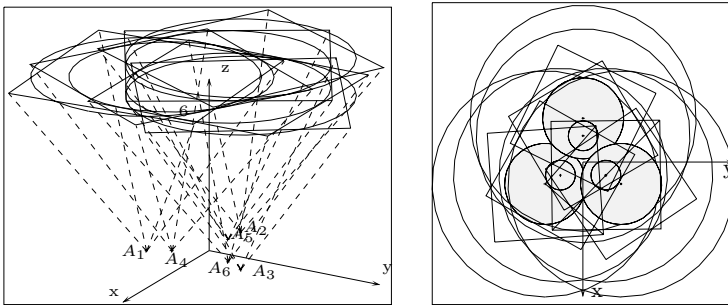


Figure 7.19. On the left, the possible zones for the points B_i . The pyramids are shown in dots. On the right, the constraints, seen from above: the *blank circles* correspond to the external circles, the *grey circles* to the internal circles, and the *squares* to the zones that are due to the mechanical limits on the joints attached to the base.

Figure 7.20 shows the possibly important influence of the mechanical limits on the passive joints: the workspace volume of INRIA *left hand* is drawn according to the constraints on the length of the linear actuators with and without taking the mechanical limits on the joints into account.

7.5.2.2.3 Joints on the moving platform

For the constraints on the joints attached to the moving platform we may choose the same model as the one we used for the joints attached to the base. We may therefore define a pyramid P_i with apex B_i and which is such that if the constraints on the joint are satisfied, the segment line B_iA_i will be located inside the pyramid. We may then define a pyramid that will be called *equivalent* to P_i , P'_i , with apex A_i situated so that if A_i is inside

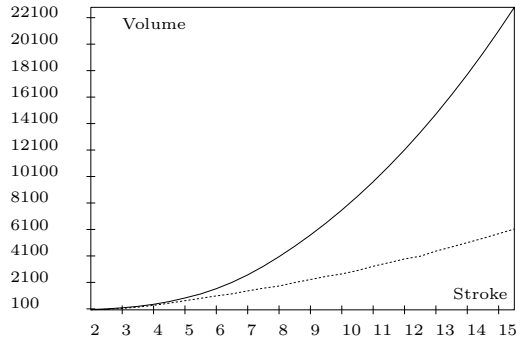


Figure 7.20. The thin lines represent the workspace volume according to the constraint on the links lengths, without constraints on the passive joints. The dotted line shows the same volume when the constraints on the joints are modeled by a 4 faced pyramid.

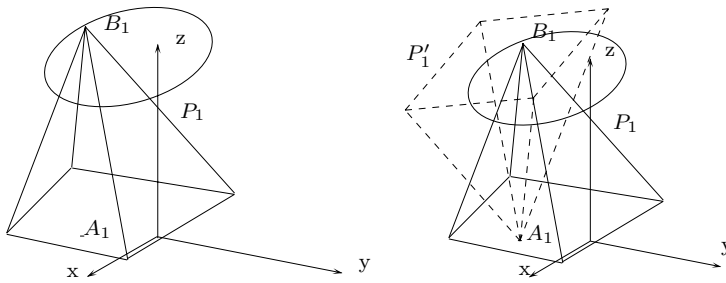


Figure 7.21. On the left, definition of the pyramid that characterizes the constraints on the joints at B_i . Point A_i is located inside the pyramid if the constraints are satisfied. On the right, definition of the equivalent pyramid (in dotted lines).

P_i then B_i is inside P'_i (figure 7.21). This is therefore similar to having a new joint at A_i and the previous workspace calculation can be used.

To illustrate the workspace calculation, we consider the example of a robot developed by Arai (14) at MEL in Tsukuba, whose joints at A_i are situated underneath the base, while the links come out of a square opening. It is hence possible to model the constraints on these joints by 4 face pyramids. We show views of the constant orientation workspace, first taking into account only the limits on the link lengths, and then, for the same orientations, taking the constraints on the joints into account. We see that the constraints on the joints noticeably influence the workspace (figures 7.22); the working volume decreases roughly by a factor of 6. Matory (385) mentions that the volume of the workspace is maximal if the main axes of the joint have the same directions as the links when the robot is in a nominal position in which the actuators are at mid-stroke. As a

matter of comparison, figure 7.23 presents the half workspace of the INRIA left hand calculated with Maple, taking into account only the limits on the link lengths, and then, for the same orientations, taking the constraints on the joints into account. With the geometrical approach, such calculation is almost instantaneous, while several minutes are needed by Maple. Furthermore the geometrical method provides a very compact result on which calculation like area or volume are easy while the result provided by Maple is large and not very appropriate for volume calculation.

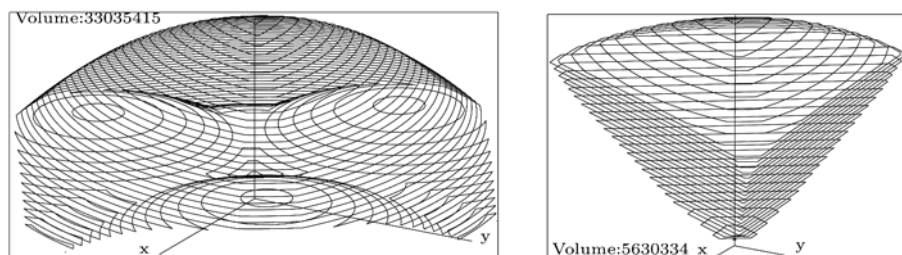


Figure 7.22. View in perspective of the Arai robot workspace, orientation $\psi = \theta = \phi = 0^\circ$. On the left, the constraints are the links lengths, while on the right the mechanical limits of the passive joints are also taken into account.

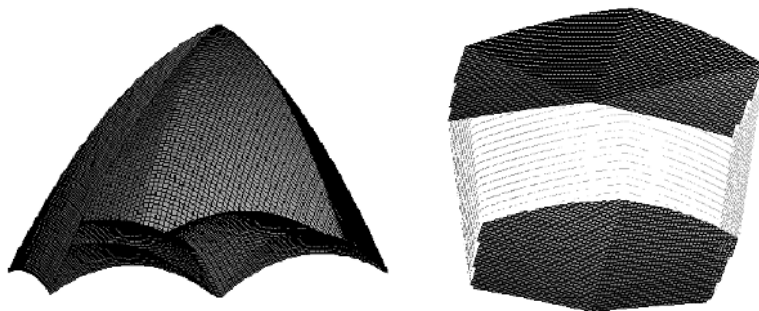


Figure 7.23. View in perspective of INRIA left hand half workspace, orientation $\psi = \theta = \phi = 0^\circ$, as computed with Maple. On the left, the constraints are the links lengths, while on the right the mechanical limits of the passive joints are also taken into account. A large scaling on the z axis has been applied for better visibility

7.5.2.3 Interference between links

The last limiting factor for the workspace is the risk of interference between the links. We will consider a 6-*UPS* robot.

7.5.2.3.1 Notion of distance between links

To take link interference into account, we will consider the locations within the workspace for which the minimal distance between any pair of

points on two separate links is equal to a constant d ; this is called the *safety distance*. We will assume that this distance is smaller than the minimum of the distances between the pairs (A_i, A_j) , (B_i, B_j) of the pair of links i, j . Without loss of generality, we will consider the particular couple of links 1 and 2. Such an approach will allow us to deal with the case of interference between cylindrical links. Indeed, if we impose as safety distance the sum of the radii of the cylindrical links, and if we are able to determine the poses for which the distance between the pair of links is equal to the safety distance, then we will have determined the poses for which the cylindrical links interfere.

We will say that all links are at a safe distance, if the distances between all the pairs of links are greater than or equal to their safety distance. To calculate the distance between two line segments, there are different cases that have to be considered (figure 7.24), detailed in (402). Each of these cases may be distinguished by inequalities that are functions of the pose parameters. Strictly speaking, the above distance between links is

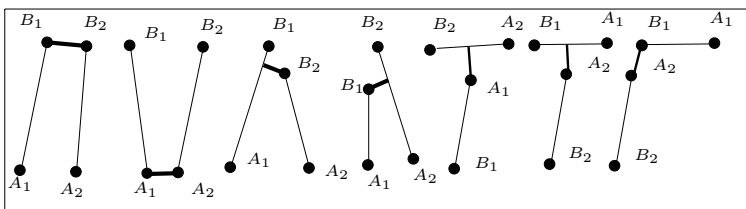


Figure 7.24. In bold, the minimal distance between two links.

conservative for cylindrical legs: it may be seen on the right of figure 7.24 that the distance between 2 links may be lower than the sum of the radii, even though while the corresponding cylinders do not intersect. But such cases will not occur in general for parallel robots.

7.5.2.3.2 Interference loci

Using the notion of distance between links, we may determine the loci of C where the distance between one pair of links is equal to the safety distance. If we consider only horizontal cross-sections of the workspace, these loci are conics that divide the plane into zones, where the distance between links is either greater or smaller than the safety distance (402). We need to calculate the zones for each pair of links, for which the distance between links is greater than the safety distance, and then calculate the intersection of these zones to obtain the workspace for which the interference between links is taken into account. Figure 7.25 presents a few examples of workspaces with and without constraints on mechanical limits on the

joint attached to the base, as well as with and without taking interference between links into account.

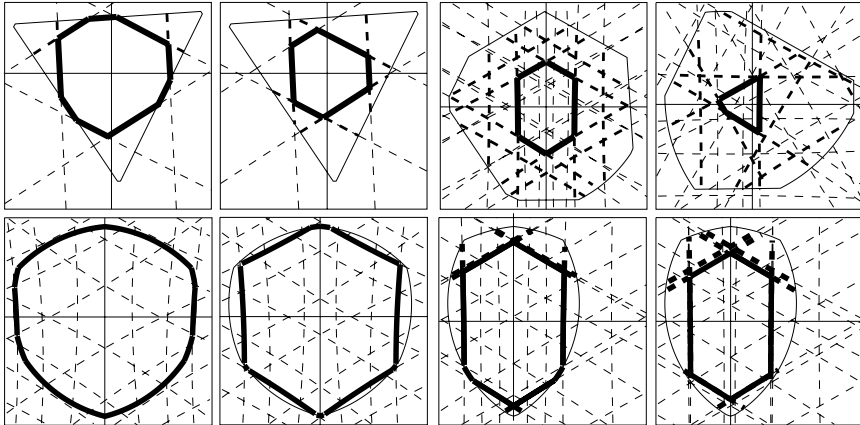


Figure 7.25. Constant orientation workspace determination. The *conics* that appear when taking into account the interference between the links are shown in dots: in these particular cases we obtain hyperbole and degenerate conics i.e. two parallel lines. The workspace, taking the interference between links into account, is represented in bold, and with thin lines when the interference between links is not taken into account.

7.5.3. ORIENTATION WORKSPACE

We have already seen that representing an orientation workspace in a way that has a physical meaning is difficult. In addition, most of the methods presented in the literature rely on a discretisation approach (46).

We propose an algorithm that allows us to represent only 2 orientation d.o.f. of a 6-UPS robot, but that uses discretisation only for one of them. This algorithm will be especially suitable for the representation of the orientation abilities of a 5 d.o.f. robot.

We suppose that point C of a 6-UPS robot is fixed in the reference frame. We consider a unit length link CN_e which is attached to the moving platform at C . When the moving platform rotates around C , the extremity N_e of the link moves on the *unit sphere* centered at C . Hence by representing the regions of the unit sphere where the end of the link may be located, we characterize two rotary degrees of freedom of the manipulator. Only the rotation around the link axis is not illustrated. By a careful choice of the direction of the link we may obtain all the allowed rotations for the end-effector around two perpendicular axis.

We first assume that the moving platform rotates through an angle θ_1 around a vector \mathbf{X}_1 fixed in the reference frame. Once the platform is

in this position, we look at the possible rotation around a vector \mathbf{X}_2 in the reference frame. For a fixed value of θ_1 , and if the manipulator is not submitted to any constraint, the extremity N_e describes a circle C_e on the unit sphere (figure 7.26). The constraints on the manipulator are actually

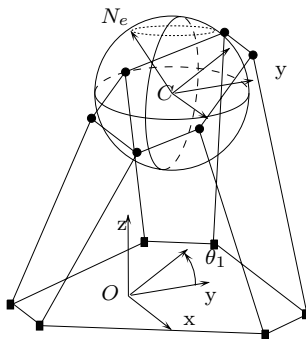


Figure 7.26. Orientation representation: the platform first rotates through an angle θ_1 around the x axis, thus $\mathbf{X}_1 = [1, 0, 0]$. When the platform rotates around the z axis, i.e. $\mathbf{X}_2 = [0, 0, 1]$, N_e describes a circle, drawn with dotted line, on the unit sphere.

such that N_e has to lie only on some circular arcs which are part of the circle, and we have shown that these circular arcs may be calculated exactly (403). By using regularly spaced values for the angle θ_1 in a range of diameter π , we may span the whole unit sphere, and obtain the possible regions for N_e on it. Figures 7.28, 7.27 show examples of orientation workspaces.

7.5.4. DEXTROUS WORKSPACE

Determination of the *dextrous workspace* may be carried out with the same type of algorithm as for planar robots. With respect to the constraint imposed by leg i the possible locations of C , if they exist, are located in the spherical annulus CS_i with center A_i and internal and external radii $\rho_{i_{max}} - \|\mathbf{CB}_i\|$, $\rho_{i_{min}} + \|\mathbf{CB}_i\|$ (figure 7.29). Indeed, in these poses for C , B_i may freely rotate around C while respecting the leg length constraint. Note that CS_i exists only if $\rho_{i_{max}} - \rho_{i_{min}} \geq 2\|\mathbf{CB}_i\|$.

The dextrous workspace, if it exists, is the intersection of the 6 zones CS_i . We could also consider the mechanical limits on the joints; we would have to find the intersections of the spherical region and the pyramids.

7.5.5. MAXIMAL WORKSPACE

There are many calculation methods that have been proposed for the calculation of the maximal workspace. The method proposed by Adkins and Haugh (221) may in theory calculate the boundary of the maximal

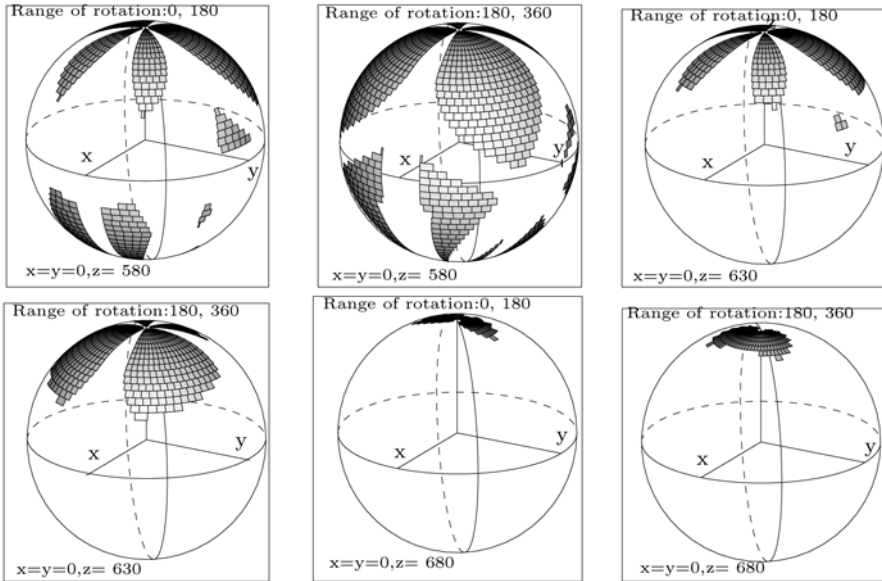


Figure 7.27. Orientation workspace. Representation of the allowed zones for the normal to the moving platform, for rotations around the x axis followed by rotations about the z axis. The constraints are the links lengths, the mechanical limits of the joints attached to the base and the interference between the links.

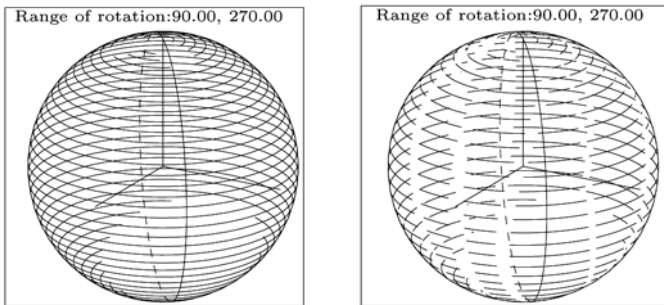


Figure 7.28. The allowed zones for axis $-y_r$ of the moving platform for the rotations around the x axis with an angle in the interval $[\pi/2-3\pi/2]$ followed by rotations around the z axis. On the left, only constraints on the link lengths are taken into account, while on the right, interference between links is considered; some rotations satisfying the lengths constraints are actually impossible because of the interference between the links.

workspace but is complex, and these authors use it just for the calculation of the constant orientation workspace. A very rough approximation of maximal workspace of a 6 d.o.f. robot was proposed by Kim (311).

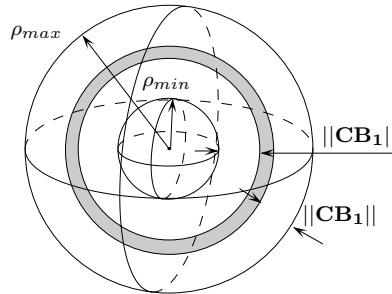


Figure 7.29. The dextrous zone allowed for C , only taking the constraints on link i into account.

Interval analysis is an appropriate tool for computing an approximation of the maximal workspace. The algorithm is briefly presented in the interval appendix and detailed in (411). For a robot with n d.o.f., the output of the algorithm is a list \mathcal{L}_1 of n -dimensional boxes that are included in the maximal workspace, and another list \mathcal{L}_2 of boxes that may include poses of the maximal workspace. The minimal width w_2 of the boxes in \mathcal{L}_2 is fixed in advance and determines the quality of the approximation. The maximal workspace volume is greater than or equal to the total volume of the boxes in \mathcal{L}_1 and less than or equal to the total volume of the boxes in $\mathcal{L}_1, \mathcal{L}_2$. The quality of the approximation may be improved incrementally: the starting point is the list \mathcal{L}_2 obtained for a given value of w_2 , and the calculation is restarted with a smaller value of w_2 . The result will be new boxes included in the maximal workspace, and a lower total volume for the boxes of the new \mathcal{L}_2 .

Figure 7.30 presents cross-sections of the maximal workspace of INRIA *left hand* for $z = 50$ with various accuracies. Note that this algorithm may be generalized to deal with inclusive or total orientation workspace. For example figure 7.31 presents an inclusive orientation workspace of INRIA *left hand* for z in $[50, 60]$, the orientation interval being $[0, 20]$. This algorithm may also be appropriate for computing a reduced total orientation workspace as, for example, the location of C for which the normal of the platform should be able to be in any position in a cone (607).

7.5.6. WORKSPACE FOR MACHINE-TOOL

One main interest of the reduced total orientation workspace lies in the field of machine-tools, where only five-axis motions are required; the rotation of the platform around its normal is ensured by the spindle. In that case the yaw angle θ will define the angle between the spindle axis and the z axis, and

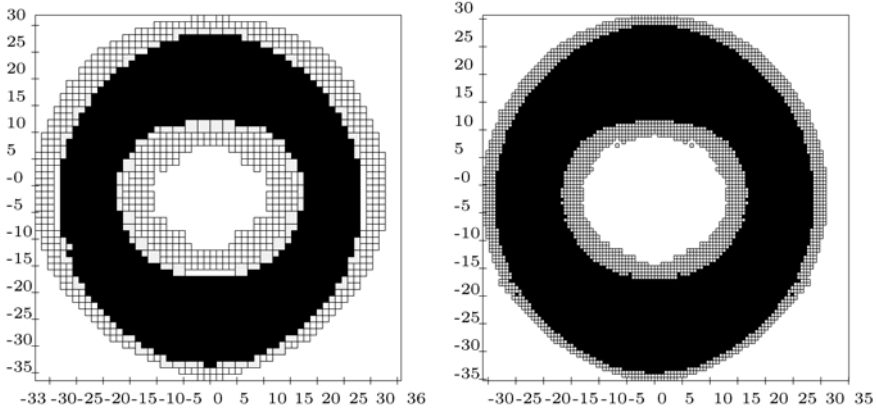


Figure 7.30. Cross-sections at $z = 50$ of the maximal workspace with accuracy 0.84, 0.42 (the accuracy is defined as the distance between the center of a box and a vertex). The black area lies completely in the workspace.

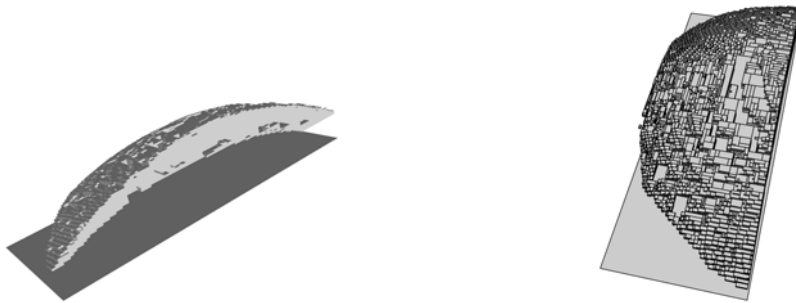


Figure 7.31. Inclusive orientation workspace of INRIA *left hand* for z in $[50, 60]$, the orientation intervals are $[0, 20]$.

its limit will indicate the maximal tilting angle of the spindle. Huang (241)* assumes that ϕ is equal to $-\psi$, and defines various workspaces:

- *minimum reachable yaw angle for a given point*: for a given C this is the minimum value of θ for any value of ψ in $[0, 2\pi]$
- *position-orientation workspace*: all the locations of C together with their minimum reachable yaw angle

Huang shows that the minimum reachable yaw angle for a given point may be calculated exactly if we consider constraints on the legs lengths, and assume that the mechanical constraints on the passive joints may be modeled by a cone. For the calculation of the position-orientation workspace, Huang uses a discretisation method. Similar workspace definition and assumption on the value of ϕ have been introduced by Wang (607). A more rigorous calculation based on interval analysis has been proposed by Pott (483) for

the 6-*PUS* robot. The choice of imposing $\phi = -\psi$ may not lead to the largest value of θ . To show that a better choice for ϕ is possible we considered the INRIA left hand, assuming that the smallest and largest lengths are 50, 57, and computed the largest possible θ angle by assuming first that $\phi = -\psi$ and then allowing any value of ϕ in the range $[0, 2\pi]$. In the first case we found out that θ will lie in the range $[-29, 29]$ degree, while in the second case the range is $[-31.261, 31.261]$. The increase in the maximum tilting is therefore significant (about 7.22%). With interval analysis it is possible to design an algorithm that allows us to increase the minimum reachable yaw angle for a fixed location of C by allowing us to adjust the ϕ angle (see exercise 7.12), or compute a larger translation workspace for a given maximum tilting angle.

7.5.7. COMPARISON BETWEEN ARCHITECTURES

We will now examine the influence of the joint layout on the workspace volume. We will consider a MSSM, a TSSM and a SSM of similar dimensions and with the same minimal and maximal leg lengths; these are the only workspace constraints we will consider. We will also consider the MSP robot presented by Stoughton (555): this robot presents a crossed layout of joint centers obtained by using an optimization process in which the cost function was a weighted sum of the workspace volume and the dexterity, the dexterity having a larger weight than the workspace volume.

Table 7.1 indicates the volume of the constant orientation workspace for different orientations of the moving platform. The SSM has the larger workspace, followed by the TSSM and then the MSSM. It must be noted that the workspace volume of the SSM is always approximately 30 % greater than that of the TSSM. Similarly the SSM volume is 70 % greater than that of the MSSM, thus the TSSM volume is approximately 25 % greater than that of the MSSM. The MSP has a workspace volume which is somewhat smaller than the MSSM, and much smaller than that of the SSM, as claimed by Stoughton.

A similar comparison can be made between the maximal workspaces. Here we are able to compute only an approximation to the maximal workspace (see section 7.5.5). The volume of the approximation of the maximal workspace is V_c with an error margin that is $[0, V_n]$. Table 7.2 indicates the volume of the maximal workspace for the four different robots, the error margin, and the minimal and maximal ratios between the different workspace volumes.

According to these results it is clear that joint layout is important for the workspace volume: for example the volume for the SSM is about twice the volume of the MSP; this factor may be slightly reduced if we consider

Orientation (ψ, θ, ϕ)	MSP	MSSM	TSSM	SSM	$\frac{TSSM}{MSSM}$	$\frac{SSM}{MSSM}$	$\frac{SSM}{TSSM}$
0,0,0	861	950	1214	1576	1.277	1.659	1.298
0,5,0	748	766	968	1271	1.264	1.659	1.313
5,0,0	765	924	1177	1529	1.274	1.655	1.299
5,5,0	685	749	947	1245	1.264	1.662	1.315
5,5,5	613	706	894	1180	1.266	1.671	1.32
0,10,0	502	434	542	745	1.249	1.716	1.374
10,0,0	670	848	1082	1406	1.276	1.658	1.299
10,10,0	405	412	522	720	1.267	1.748	1.379
10,10,10	355	363	468	661	1.289	1.821	1.412

TABLE 7.1. The volume of the constant orientation workspace for various orientations

	SSM	TSSM	MSSM	MSP
Volume	7848+[0,1950]	5182+[0,1174]	4144+[0,750]	3524+[0,615]
Ratio				
$\frac{TSSM}{MSSM}$	$\frac{SSM}{MSSM}$	$\frac{SSM}{TSSM}$	$\frac{TSSM}{MSP}$	$\frac{SSM}{MSP}$
1.0587-1.534	1.603-2.364	1.235-1.89	1.22-1.9	1.896-2.78

TABLE 7.2. The volume of the maximal workspace, the maximal error, the minimal/maximal ratios of the maximal workspace volumes for the four robots.

the mechanical limits on the joints.

7.6. Workspace performance indices

Various indices may be used to characterize the workspace of parallel robots. The most used is the workspace volume, which causes a unit problem if the d.o.f. of the robot mix translations and orientations. Other indices have been proposed, such as the *workspace volume index*, the ratio between the workspace volume and the volume of the robot. For planar robots, Heerah (224) proposes the ratio between the footprint area and maximal workspace area, and compares possible architectures using this ratio.

Liu (375) geometrically studied a few limit poses for the TSSM that may be used to characterize a robot workspace: highest, lowest and most tilted position. He established their values from the robot geometry, and from the limits on the lengths of the links. For a *Delta* robot Stock (553) defines a *space utilization* index that reflects the ratio of the workspace size to the physical size of the robot.

7.7. Trajectory verification

The purpose of trajectory verification is to determine if a given trajectory may be performed by a parallel robot. A necessary condition for a trajectory to be valid is that it lies within the workspace of the robot. But other criteria may have to be checked, either related to the robot performances (singularity, accuracy, ...) or to the surrounding (e.g. obstacle avoidance). In this section we will focus on necessary conditions, although some other criteria will be mentioned.

7.7.1. LINE SEGMENT VERIFICATION

A basic problem in trajectory verification is to check whether a trajectory defined as a line segment for C and a constant orientation of the platform lies completely in the workspace or not. We limit our study to 6-*UPS* robots but the proposed approach may be extended to other robots as well³.

If M_1, M_2 are the beginning and end points of the trajectory, then for any C lying on the trajectory we have

$$\mathbf{OC} = \mathbf{OM}_1 + \lambda \mathbf{M}_1 \mathbf{M}_2 \quad \text{with} \quad \lambda \in [0, 1] . \quad (7.8)$$

7.7.1.1 Constraints on the link lengths

Let us calculate a length of a link for any point on the trajectory between M_1 and M_2 . We have

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + \mathbf{CB} , \quad (7.9)$$

where $\mathbf{CB} = \mathbf{RCB}_r$ is a constant vector. The length ρ of a link is given by

$$\rho^2 = \|\mathbf{AO}\|^2 + \|\mathbf{OC}\|^2 + \|\mathbf{CB}\|^2 + 2(\mathbf{AO} + \mathbf{CB}) \cdot \mathbf{OC} + 2\mathbf{OA} \cdot \mathbf{CB} \quad (7.10)$$

Using equation (7.8) we obtain

$$\rho^2 = \lambda^2 \|\mathbf{M}_1 \mathbf{M}_2\|^2 + 2\lambda (\mathbf{AM}_1 + \mathbf{CB}) \cdot \mathbf{M}_1 \mathbf{M}_2 + \|\mathbf{AM}_1 + \mathbf{CB}\|^2 . \quad (7.11)$$

³The algorithms described in this section are integrated in the workspace calculation software, available by anonymous ftp, directory `coprin/Workspace/Gough`

We therefore have an equation of the type $\rho^2 = a\lambda^2 + b\lambda + c$, where a, b, c are coefficients that depend only on the trajectory and the geometry of the robot:

$$a = \|\mathbf{M}_1\mathbf{M}_2\|^2 \quad b = 2(\mathbf{A}\mathbf{M}_1 + \mathbf{C}\mathbf{B}) \cdot \mathbf{M}_1\mathbf{M}_2 \quad c = \|\mathbf{A}\mathbf{M}_1 + \mathbf{C}\mathbf{B}\|^2$$

We note that $a > 0, c > 0, a + b + c > 0$. We now look at the equation

$$a\lambda^2 + b\lambda + c - \rho_{max}^2 = 0 \tag{7.12}$$

Let $f(\lambda) = \rho^2(\lambda) - \rho_{max}^2$ be the left-hand term of this equation. If equation (7.12) has no real root, and since $a > 0$, then $f(\lambda)$ will be positive for all λ . Consequently the lengths of the link will be larger than the maximal length for any point on the trajectory and the trajectory will not be feasible.

We now assume that the equation has two real roots x_1, x_2 with $x_1 \leq x_2$. Since $a > 0$, $f(\lambda)$ will be positive in the intervals $] -\infty, x_1[$, $]x_2, +\infty[$. The intersection of these intervals with the interval $[0, 1]$ will give the intervals of λ (i.e. the components of the trajectory) where the length of the link is larger than the maximal length (note that it is easy to prove that if the leg lengths at M_1, M_2 are lower than ρ_{max} , then this will be the case for any point between M_1, M_2 , see exercise 7.13). Repeating this algorithm for the 6 links enables us to find the parts of the trajectory where at least one of the links has a length which is larger than its maximal value i.e. the parts of the trajectory that are not feasible. We now look at the equation:

$$a\lambda^2 + b\lambda + c - \rho_{min}^2 = 0 \tag{7.13}$$

Let $g(\lambda) = \rho^2(\lambda) - \rho_{min}^2$ be the left-hand term of this equation. If equation (7.13) has no real roots, and since $a > 0$, then $g(\lambda)$ is positive for any λ and the length of the link for any point on the trajectory is larger than the minimal length. Let us assume that the equation has two roots x_1, x_2 with $x_1 \leq x_2$. Since $a > 0$, $g(\lambda)$ will be negative in the interval $]x_1, x_2[$ and the intersection of this interval with the interval $[0, 1]$ will give us the interval where the length of the link will be lower than the minimal length. Repeating this algorithm for the 6 links we can find the parts of the trajectory where at least one of the links has a length which is lower than its minimal value.

In summary, combining the procedures for $f(\lambda), g(\lambda)$, we are able to determine not only if a trajectory is feasible but also which parts are not feasible and which constraints are violated.

Analyzing these equations more thoroughly, we can obtain simplifying rules that sometimes allow us to check the feasibility of the trajectory directly without having to analyze the intervals (see exercises and (401)). It is possible also to determine the changes to be made to the extremal lengths of the links so that the trajectory will be feasible. Determining such changes will generally be possible for all the considered constraints (401).

7.7.1.2 Mechanical limits on the joints

As in the section dealing with workspace calculation, we assume that the mechanical limits on the passive joints may be modeled by a pyramid with planar faces and apex A_i .

Let \mathbf{n}_i be the external normal to face i of the pyramid that is associated with the joint at A . If point B is located on the inner side of face i we should have

$$\mathbf{AB} \cdot \mathbf{n}_i \leq 0 \quad (7.14)$$

Using equation (7.8) we obtain

$$\lambda \mathbf{M}_2 \mathbf{M}_1 \cdot \mathbf{n}_i + (\mathbf{OM}_1 + \mathbf{AO} + \mathbf{CB}) \cdot \mathbf{n}_i \leq 0 \quad (7.15)$$

As this is a linear equation in λ we may easily calculate the intervals for λ in $[0,1]$ for which equation (7.15) is not satisfied.

The intersection of the interval so obtained with the interval $[0,1]$ will give the components of the trajectory where the constraints on the base joints are not satisfied. This algorithm will have to be used for the faces of all the pyramids that are associated with the joints.

In conclusion we note that we can take the intersections between links into consideration under the same hypotheses as for workspace calculation.

7.7.1.3 Example

The computation times for the above algorithms are extremely low; this allows us to consider a real-time use of the method. Figures 7.32,7.33 show a few trajectories for the Arai robot (14); the forbidden components of the trajectories are indicated.

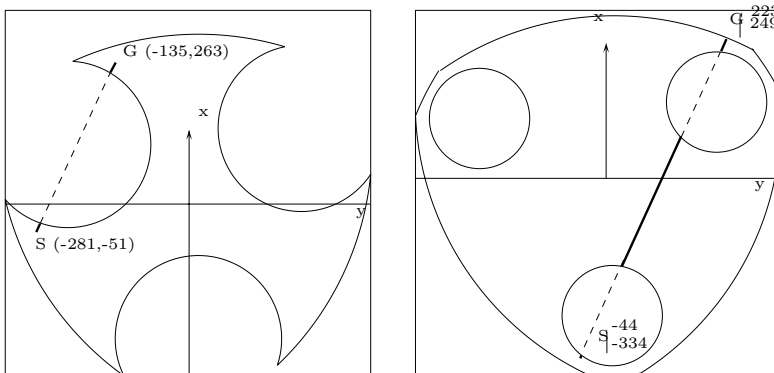


Figure 7.32. Trajectory verification examples: for the trajectory the forbidden components are shown in dots, while the allowed components are in bold. The workspace boundary is drawn in thin lines.

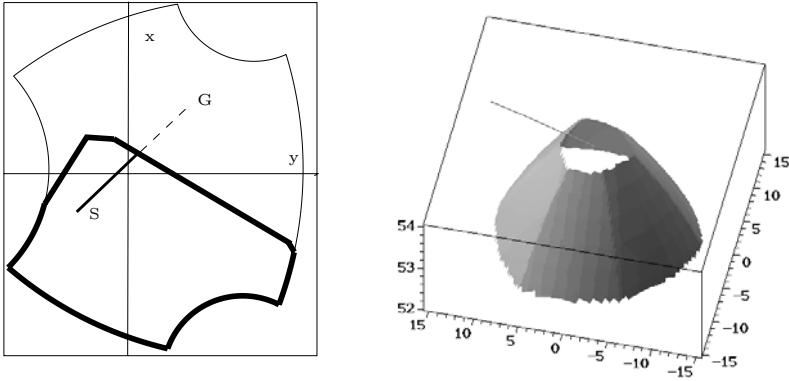


Figure 7.33. Trajectory testing examples: on the left, the boundary of the workspace where there is no interference between the links is shown in bold. For the trajectory the forbidden components are represented by the dotted line.

7.7.2. PARAMETRIC TRAJECTORY VERIFICATION

We now assume that the trajectory is defined by a set of one-parameter functions, one for each pose parameter. Typically the parameter will be the time T , restricted to lie in the range $[0,1]$ and we have

$$x_c = f_x(T) \quad y_c = f_y(T) \quad z_c = f_z(T) \quad \psi = f_p(T) \quad \theta = f_t(T) \quad \phi = f_h(T)$$

No assumption is made on the nature of the f functions except that they are continuous. In that case an algorithm based on interval analysis is able to verify if the constraints on the leg lengths, motion of the passive joints and eventually leg interferences are satisfied on the whole trajectory. This algorithm may also check if the trajectory is singularity free, or check any other constraints that can be expressed (implicitly or explicitly) as functions of T . Furthermore it may take into account small variations around the trajectory due to control errors, as well as small errors on the geometrical modeling of the robot. This algorithm is briefly summarized in the interval appendix, and presented in more detail in (415).

7.8. Motion-planning

We have already seen in the "Singularity" chapter methods that allow us to modify a given trajectory locally to avoid singularities. In this section we will look at other aspects of motion planning.

7.8.1. GLOBAL MOTION PLANNING

The purpose of global motion planning is to determine a trajectory that joins two poses and that satisfy at least the necessary conditions for lying fully within the workspace of the robot. Harris (219) dealt with motion planning between two poses, looking for the parameters of the screw motion linking the two poses, and reckoning that this motion should be able to minimize the changes in the link lengths. For planar robots, Chablat and Wenger suggested a motion-planning algorithm which includes the singularities, and interference between the links and the platforms (79; 617). The use of the condition number in order to manage the path of parallel robots that are redundant with respect to the task they must perform should also be mentioned. Thus, Gosselin and Angeles (190) present an algorithm, illustrated on both a planar and a spherical robot, that helps to find the orientation of the robot so that it presents the best accuracy at some specific poses along the path. Another approach to motion planning is to specify some precision poses, and to synthesize a trajectory that respects the robot constraints and is close as possible to the precision poses. Su (556) considers this approach for parallel robots with 2 to 5 d.o.f. and computes the trajectory by an optimization method that minimizes the distance to the precision poses using a dual quaternion metric. The problematic parts of this approach is clearly the optimization, and the difficulty of introducing singularity and self-collision constraints.

Motion planning is a classical problem for serial robots in view of avoiding obstacles. In our case workspace boundary, singularities, . . . may be considered as obstacles to be avoided. Roughly speaking, there are two types of motion planning methods: *potential fields* and *configuration space* methods. In the first approach an attractive potential is attached to the goal and a repulsive one to the obstacle. The motion of the end-effector results from the action of this fields. To the best of our knowledge, this approach has never been used for motion planning of parallel robots. Our own tests have shown that the problem of local minima, which is the major problem for potential fields, is very real for motion planning of parallel robots.

In the configurations space approach, a preliminary step is to determine a description of the free space, either as a collection of cells (the *cell decomposition approach*) or of reachable poses (the *roadmaps approach*). Once this description is constructed, a motion planning query consists in connecting the start and goal poses through cells or reachable poses, a local planner ensuring the local trajectory between two successive elements.

This approach was initially designed for serial robots with a closed form for the direct kinematics. Hence the determination of the reachable cells of poses starts in the joint space, and is then transposed to the task space.

This approach has been extended to closed chains (583; 630; 631) but a major difficulty is that a random choice of variables in the joint space has a zero probability satisfying the closure equations. Furthermore self-intersection and singularities are usually not considered. The only work dealing effectively with parallel robots is a probabilistic roadmap approach proposed by Cortés and Siméon (111). They improve the creation of the reachable poses of the roadmap by using the structure of the robot, and they deal with self-collision, and obtain a very efficient planner. But the local planner does not take into account either singularity or multiple solutions for the direct kinematics, that may prohibit the use of the trajectory.

We will now present some examples of simple motion planning approaches.

7.8.1.1 Cell decomposition for planar robots

The workspace of a planar parallel robot may be represented in 3D by using the parameters x_c, y_c, θ . We first compute cross-sections of the workspace for different values of θ using the procedure described in section 7.3.1 (figure 7.34). We then calculate a polygonal approximation to each cross-

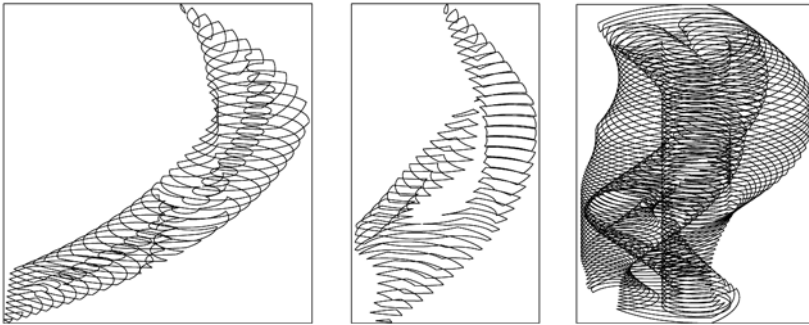


Figure 7.34. Cross-sections of the workspace of robot 1, for various allowed lengths of the links. Each horizontal cross-section corresponds to a constant orientation workspace i.e. a fixed value for θ .

section. From these polygonal approximations, we obtain a polyhedral representation of the workspace as a set of tetrahedra. Figure 7.35 presents an example of such a reconstruction. Motion planning consists in looking for the shortest-path trajectory joining the beginning and end points and going through the centers of the tetrahedra, using an A^* algorithm. After obtaining this trajectory, we use a smoothing stage. Figure 7.36 presents an example of a trajectory obtained by this method. The interest in this technique is that it allows us to discover trajectories which may present any type of orientation control law: linear or polynomial interpolation between

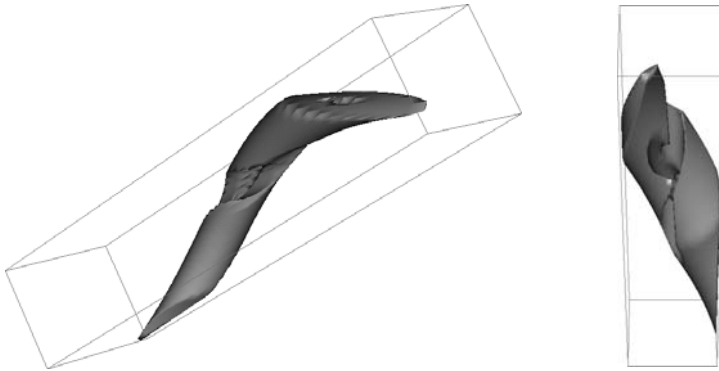


Figure 7.35. 3D workspace of a planar parallel robot in the x, y, θ space.

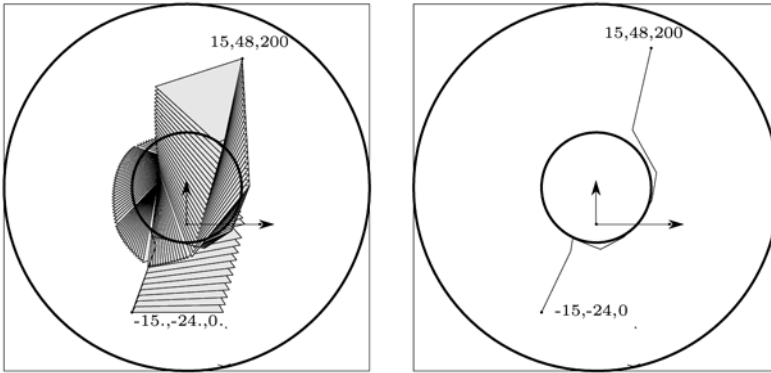


Figure 7.36. A trajectory obtained by using the 3D reconstruction of the workspace. The beginning and end points are indicated on the drawing.

the orientation of the beginning and end poses, etc. 3D reconstruction may actually be done once and for all, and we can then use it later to generate any trajectory.

7.8.1.2 Cell decomposition for spatial robots

We will first illustrate the cell decomposition approach on a simple example in which the orientation of the end-effector has to be constant, while the start and end points lie in the same horizontal plane. Under that assumption we know how to calculate the constant orientation workspace, see section 7.5.1. As we know the boundary of the workspace, we may describe the interior of the workspace as the union of small square cells, the layouts of which are calculated from the boundary of the workspace. Among these cells are the two cells which include the beginning and end points of

the trajectory. We then consider a valued graph with nodes at the centers of the cells, and linked to adjacent nodes by arcs; any node will generally have 8 adjacent nodes, less if the cell is on the boundary of the workspace. The value of the arc joining two nodes is the distance between the nodes if the node is within the workspace and if the straight line joining the nodes is inside the workspace. If either of these two conditions is not satisfied, the value of the arc is assigned an arbitrarily large value. A shortest path algorithm, such as the A^* algorithm, will provide a trajectory joining the beginning and end points and going through the nodes (figure 7.37). The

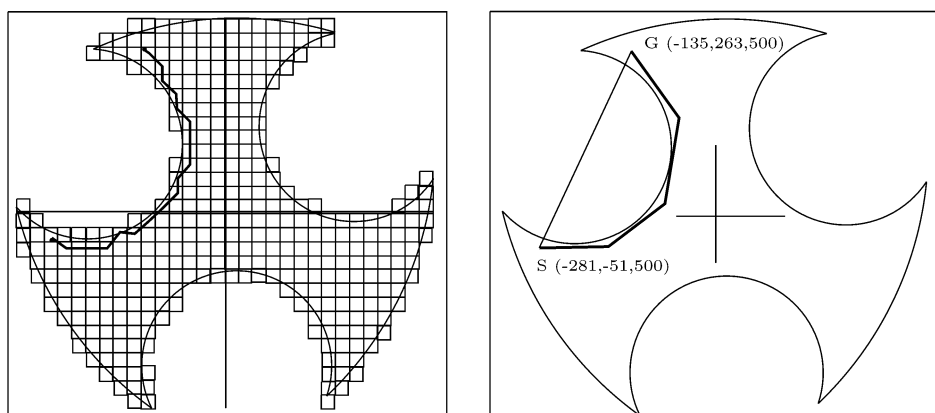


Figure 7.37. On the left, the A^* algorithm has allowed to find a trajectory joining the beginning and end points. On the right, the trajectory obtained after smoothing.

trajectory given by the algorithm may be quite rough because of the algorithm itself's; but it may easily be smoothed out (figure 7.37).

It is easy to cover the workspace with cells, but there are a few problems:

- the knowledge of the shape of the workspace is only partially taken into account, as it is used only to limit the number of cells
- we cannot guarantee that a trajectory will be found.

The technique used for planar motion-planning may be extended to space, either with a constant or non constant orientation. We will use various horizontal cross-sections; if the orientations at the beginning and end points are not the same, the orientation for each calculated cross-section may be obtained by linear interpolation. We tile the volume so obtained with cubic cells, as shown on figure 7.38. An A^* shortest-path algorithm may then be used. We present an example of trajectory determination in figure 7.39. The number of cells in the spatial case is much higher than in the planar case; trajectory verification will require much computation time. Such a

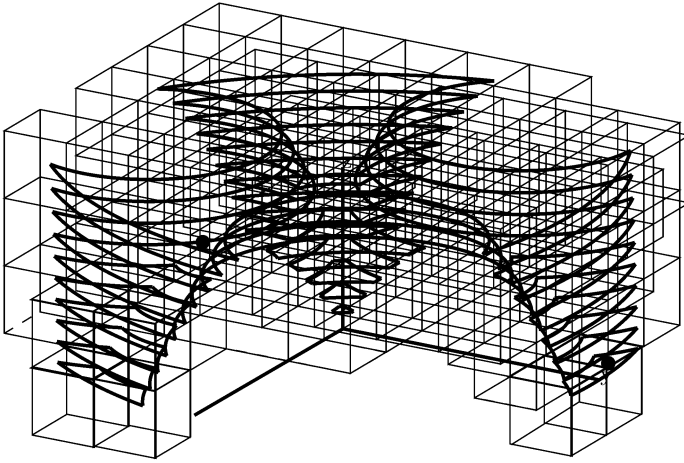


Figure 7.38. Since we know the boundary of the workspace, we may build a tiling of the workspace based on cubic cells. The boundary of the workspace is represented in bold.

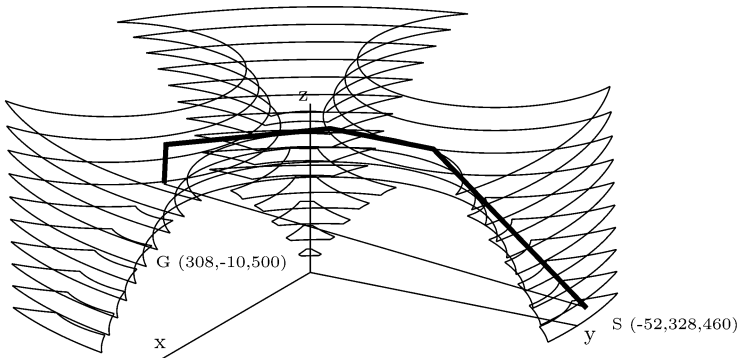


Figure 7.39. A trajectory that is obtained with the help of the A^* algorithm after smoothing (in bold, view in perspective). The direct trajectory (the thin line) is partially outside the workspace

method will not work if the beginning and end points are within the same horizontal plane, but have different orientations.

7.8.1.3 Roadmaps

The roadmap approach will be illustrated on the same spatial example. When the interference between links is ignored, the boundary of the workspace will be made of circular arcs and segments; this constitutes a generalized polygon, as defined by Laumond (344).

We may then use the *visibility graph method* in order to find the shortest

trajectory between the two points (344). The *visibility graph* G between two points A, B is defined in the following manner:

- A, B and all the *boundary convex vertices* are nodes of G .
- let X, X' be two nodes of G . If the segment joining X, X' is completely inside the workspace and is tangent to the boundary of the workspace, it constitutes an edge of G .
- let X be a node, and E' a circular edge of the workspace. If there is a point X' of E' so that the segment XX' is inside the workspace and is tangent to the boundary of the workspace at X and X' , then point X' is a node of G and the segment XX' is an edge of G .
- let E, E' be two circular edges of the boundary of the workspace. If a point X of E and a point X' of E' are located so that the segment XX' is inside the workspace and is tangent to the boundary of the workspace at X and at X' then X, X' are nodes of G and the link XX' is one of its edges.
- all the components of the boundary of the workspace joining two convex vertices of the boundary are edges of G .
- two nodes X, X' on the same circular edge are joined by an edge of G if there is no other node of G between X and X' on the same circular edge.

Figure 7.40 presents an example of a visibility graph. The main result is that if there is a trajectory from node A to node B , there will then be a trajectory in the graph G between these two nodes, and if a shortest path exists, this path will be contained in G . An A^* algorithm may be used to find this trajectory.

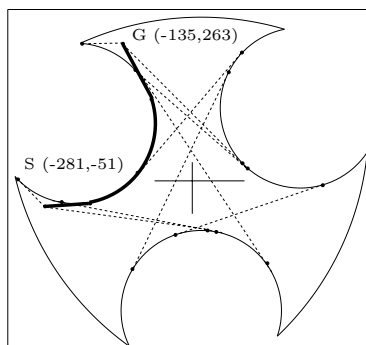


Figure 7.40. Planning in the visibility graph. Some of the edges of the graph are shown in dots, and the trajectory is in bold.

7.8.2. MACHINE-TOOL MOTION PLANNING AND PART POSITIONING

Machine-tool motion planning may present some specificities, as less than the n d.o.f. of the machine may be used: for example, for a Gough platform, the rotation of the planar platform around the plane normal will not be used as it is not necessary for machining tasks.

We showed in (412) that, for a Gough platform, it was possible to determine ranges for the free d.o.f. to ensure that a given manufacturing trajectory lies in the workspace. Hence we have still some freedom on the free d.o.f. to optimize another performance criterion. In (412) we choose to optimize the stiffness along the trajectory with an average increase between 5 and 25%. Another approach is proposed by Chen (90), who partitioned the d.o.f. into critical and secondary, and synthesized a control law that ensured the tracking of the critical d.o.f. while minimizing a velocity based secondary criterion. In any case, classical CAD systems for machine-tools are not appropriate to make the best use of the potentiality of parallel robots, as they generate approximated trajectories whose differences from the desired trajectory may already amount to 10 to 20 % of the final positioning errors.

Another aspect of motion planning for machine-tool is *part positioning*: being given a machining to be performed the problem is to determine the positioning of the part so that the machining trajectory will lie within the machine workspace. This problem, which is more difficult for parallel robots than for serial ones, has been addressed by Wang (607) with an optimization method, and by Pugazhenti (490) by discretisation. Note that part positioning may play a role in medical applications also (613).

7.8.3. PROSPECTIVE FOR GLOBAL MOTION PLANNING

As seen previously, apart from the work of Cortés and Siméon, the works presented for closed-chain motion planning may not be the most appropriate for parallel robots; they use neither the simple inverse kinematics of these structures nor the more complex forward kinematics.

However, we have seen in the previous sections that we are able to get an almost complete description of the full workspace of a parallel robot with n d.o.f. as a set of n -dimensional boxes. We have also seen in the "Singularities" chapter that we have an algorithm that allows us to test if a box is singularity-free. With some effort we believe that it will also be possible to determine if all poses in a box are self-collision free. We will thus be able to have a description of the workspace in term of boxes that satisfy all the constraints. With such a description of the workspace, it will then be possible to use a classical motion planner. The computation time of the workspace description may be large, but has to be done only once. The

major difficulty is that this description may be quite large: the calculation of a trajectory in this structure may thus be time consuming.

In any case, motion planning for parallel robots is an interesting and open challenge.

7.9. Exercises

Exercise 7.1: Show how a limitation on the rotation of the passive joints attached to the moving platform of a 3-RPR planar robot could be taken into account in calculating its constant orientation workspace.

Exercise 7.2: Describe an algorithm for calculating the *constant orientation workspace* of a 3-RRR planar robot.

Exercise 7.3: Describe an algorithm for calculating the *constant orientation workspace* of a 3-PRR planar robot.

Exercise 7.4: Describe an algorithm for calculating the *orientation workspace* of a 3-RRR planar robot.

Exercise 7.5: Describe an algorithm for calculating the *orientation workspace* of a 3-PRR planar robot.

Exercise 7.6: Describe an algorithm for calculating the *dextrous workspace* of a 3-RRR planar robot.

Exercise 7.7: Describe an algorithm for calculating the *dextrous workspace* of a 3-PRR planar robot.

Exercise 7.8: Describe an algorithm for calculating the *maximal workspace* of a 3-RRR planar robot (for point B_3).

Exercise 7.9: Describe an algorithm for calculating the horizontal cross-sections of the constant orientation workspace of the *Hexa* robot (figure 2.38).

Exercise 7.10: Show how the calculation algorithm of the cross-sections of the constant orientation workspace of a 6-UPS robot could be changed to adapt it to a 6-PUS robot; include only the limitations on the stroke of the linear actuators. Show that the boundary of the workspace is constituted of circular arcs and of arcs of ellipses.

Exercise 7.11: Show how the algorithm of the previous exercise could be modified to take into account the fact that the universal joints that are attached to the linear actuators constrain the links to be in an angular cone with an opening angle of θ_m , with $|\theta_m| < \frac{\pi}{2}$.

Exercise 7.12: Consider a 6-UPS robot having known minimal and maximal leg lengths, and with C at a fixed location. Design an algorithm for determining the maximum tilting angle θ that may be reached by the platform for any value of ψ in the range $[0, 2\pi]$, allowing any value of ϕ in the same range.

Exercise 7.13: Assume that the point C of a 6-*UPS* robot describes a line segment M_1M_2 , with a constant orientation. Show that if at each of the points M_1, M_2 , the link length is lower than ρ_{max} , then it will remain lower than this limit for the whole line segment. Establish a similar result when the limit is ρ_{min} .

Exercise 7.14: Consider a line segment M_1M_2 described by the point C of a 6-*UPS* robot with a variable orientation. Let \mathbf{U}_2 be the second order approximation of \mathbf{CB} , and \mathbf{n}_i the external normal to the face i of a pyramid describing the constraints on the joints on the base. Show that if $\mathbf{AB} \cdot \mathbf{n}_i \leq 0$ for M_1, M_2 and if $\mathbf{U}_2 \cdot \mathbf{n}_i$ is positive then $\mathbf{AB} \cdot \mathbf{n}_i \leq 0$ for any of the points on the trajectory.

Problem 7.1: How is it possible to model the mechanical limits of a ball-and-socket joint made of an universal joint set on a revolute joint?

Problem 7.2: Is it possible to extend the checking method for the interference between links to more complicated shapes of the links (parallelepipeds, compound shapes) ?

Problem 7.3: Is it possible to determine the minimal radius and the location of the center of a sphere that includes any reachable location of C for a 6-*UPS* robot?

Problem 7.4: Is it possible to calculate geometrically the exact boundary of the maximal workspace of a 6 d.o.f. robot?

Problem 7.5: Is it possible to determine the sphere of maximal radius, or the box of maximal volume, included in a *constant orientation* workspace, a *total orientation* workspace or in the *maximal* workspace?

Problem 7.6: Is it possible to determine if the *maximal* workspace of a given robot is separated into distinct components?

Problem 7.7: Is it possible to determine constraints on the geometry of a 6-*UPS* robot so that its *maximal* workspace is separated into distinct components?

Problem 7.8: Is it possible to determine the coefficients of polynomials representing the time-functions of the pose parameters so that the corresponding trajectory lies within the workspace of the robot, and is singularity and self-collision free ?

Static analysis

This chapter deals with the relations existing between the joint forces of the robot and the wrench that is applied on the end-effector. We will then study the limits that are imposed on the wrench that may be applied on the platform when the joint forces are bounded, and then reciprocally the extrema of the joint forces when the wrench applied on the robot is bounded. We will then study the stiffness of parallel manipulators.

8.1. Relations between wrench and joint forces

8.1.1. FUNDAMENTAL RELATIONS

The fundamental relation between the joint and the wrench, which is valid for serial manipulators as well as for parallel manipulators, is the following:

$$\boldsymbol{\tau} = \mathbf{J}^T \mathcal{F} \quad (8.1)$$

where $\boldsymbol{\tau}$ is the vector of the joint forces, \mathcal{F} the wrench and J the kinematic jacobian matrix. From this relation we get

$$\mathcal{F} = \mathbf{J}^{-T} \boldsymbol{\tau} \quad (8.2)$$

A consequence of this relation is that the analysis of the mechanical equilibrium proves another mean for obtaining the full inverse kinematic jacobian (see for example the analysis of the 3-*UPU* robot in (626)) and the choice of the operating point will change the matrix involved in (8.1,8.2) (542). Note that only the actuated joints forces/torques are involved in that calculation. A full static analysis should involve all joints, either actuated or passive, and hence all possible inverse kinematic jacobians.

8.1.2. DETERMINATION OF THE WRENCH

Being given the pose of the moving platform and the joint forces, we may calculate the inverse kinematic jacobian matrix, therefore allowing us to determine the wrench that acts on the moving platform.

8.1.3. DETERMINATION OF THE JOINT FORCES

We now face the problem of determining \mathbf{J} ; we have seen in the "Velocity" chapter that except for simple architectures this usually can be expressed only numerically. For determining the joint forces at a given pose we may use either the jacobian matrix numerically determined from its inverse, or use a numerical method for solving the linear system.

Another method is based on the use of an iterative scheme similar to that used for the direct kinematics. This method needs to have an estimate of the solution. The scheme is defined as

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{k-1} + \mathbf{J}_0^T (\mathcal{F} - \mathbf{J}^{-T} \boldsymbol{\tau}_{k-1}) . \quad (8.3)$$

The iteration is stopped when the difference between the wrench \mathcal{F} and the forces that are calculated from the joint forces, $\mathbf{J}^{-T} \boldsymbol{\tau}_{k-1}$, is lower than a fixed threshold. For the INRIA 6-*UPS left hand*, the algorithm seems convergent throughout the workspace, and requires between one and three iterations for an accurate result. The iterative method will usually be faster than other methods, provided that the change in the poses is small.

We have assumed that the bodies constituting the robot are rigid. The problem actually becomes more complex if we add elasticities to certain parts of the robot. A very thorough analysis of the constraints in each of the robot's bodies may be done with the help of a finite element method, as suggested by Ramachandran (493). But such an analysis is difficult to complete on the workspace of the robot. To simplify we may use a lumped model in order to account for joint and link compliances, see for example (155; 243; 639),(653)*, and obtain results that are good approximations. These results show that the compliance in the joint may play a negative influence on the stiffness of the robot. This approach allows to determine also which element stiffness play a major role in the overall stiffness of the robot, although this may change according to the pose of the robot in the workspace.

8.2. Maximal joint forces and maximal wrench

When designing a parallel robot, it is quite common to know some limits on the wrench that will be applied on the moving platform. It will therefore be useful to calculate the extremal values of the joint forces in order to choose the linear actuators and passive joints. On the other hand, we may have limited possibilities for the actuators and passive joints, which determine the maximal value of the joint forces, and may want to calculate what will be the corresponding maximal wrench.

8.2.1. MAXIMAL JOINT FORCES IN A POSE

Let us assume that the wrench is bounded by $\|\mathcal{F}\| \leq 1$, our objective being to determine the maximal joint forces for a given pose of the manipulator. Using equation (8.2) and the Euclidean norm for \mathcal{F} , we obtain

$$\boldsymbol{\tau}^T \mathbf{J}^{-1} \mathbf{J}^{-T} \boldsymbol{\tau} \leq 1 \tag{8.4}$$

This shows that the joint forces are included within an ellipsoid, called the *force ellipsoid*. But this notion is somewhat fallacious if the d.o.f. of the robot mixes translational and rotational motions as the limits on the wrench force and torques will be usually different. The choice of the Euclidean norm for \mathcal{F} may also be discussed, and the infinity norm may be more appropriate. In that case the allowed region for the task forces/torques is a parallelepiped that is mapped by \mathbf{J}^T into a convex polyhedron in the joint forces. The maximal joint forces are thus obtained from the infinity norm of this matrix. The example of figure 8.1 presents the joint forces when a wrench constituting of a cartesian force \mathbf{F} lying in the range $[-1,1]$, and no torque is applied on the moving platform of the INRIA *left hand*.

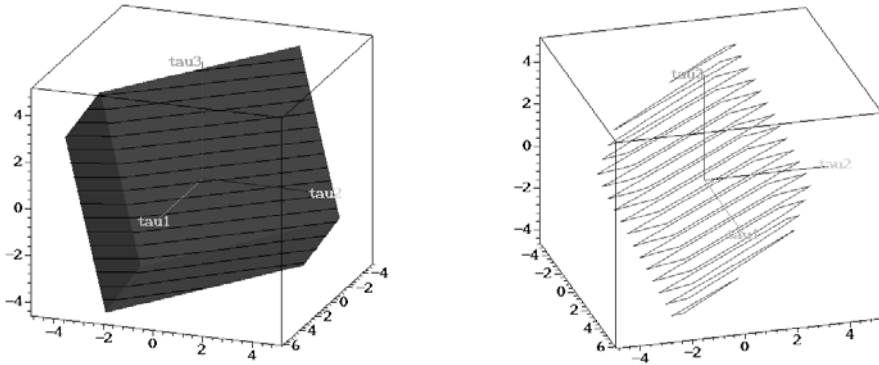


Figure 8.1. Maximal joint forces in the τ_1, τ_2, τ_3 space, when a cartesian force bounded by $|\mathbf{F}_{x,y,z}| \leq 1$ is applied on the moving platform of INRIA *left hand* in its nominal position. On the left, the joint force polyhedron, on the right, some contours

8.2.2. MAXIMAL JOINT FORCES IN A TRANSLATION WORKSPACE

In the chapter "Velocity" we showed that the kinematic jacobian matrix is usually difficult to obtain, particularly for 6 d.o.f. robots. As a consequence, obtaining the maximal joint forces for a given wrench \mathcal{F} on the moving platform is difficult, except for simple architectures such as planar robots (see exercise 8.4).

We described in (410) an algorithm that calculates the extremum of the joint forces for a 6- UPS robot, up to a pre-defined accuracy ϵ , in the case in which the pose of the end-effector has to lie within a translation workspace, i.e. the orientation of the platform is fixed. This algorithm can be extended for other mechanical architectures, and we will summarize it. First we restrict the translation workspace to be a box, as using the principle described in section 13.3 we may then analyze any other type of workspace.

The first component of the algorithm is able to determine the extremum of the joint force when C moves on a line segment. It may be shown that this can be done exactly by solving a seventh-order univariate polynomial equation. The second component compute the extremum for any pose in a horizontal rectangle up to the accuracy ϵ . This is done by sweeping the rectangle by successive parallel line segments. On each of them the extremum is computed, and after each line segment the current calculated minimum and maximum joint forces τ_{min}, τ_{max} are updated. The distance between two successive segments is calculated in such way that the minimum (maximum) joint forces for the new segment will certainly not be lower (greater) than $\tau_{min} - \epsilon$ ($\tau_{max} + \epsilon$). Assuming that this distance is small, a conservative value of the distance is obtained by solving a 22nd order polynomial equation. A similar procedure is used to sweep the box by horizontal rectangles at different altitudes, the altitude between two successive rectangles being calculated so that the change in the extremum is lower than ϵ .

Figure 8.2 shows the sensitivity of the computation time as a function of the accuracy with which the extremum of the joint forces are computed for a typical run.

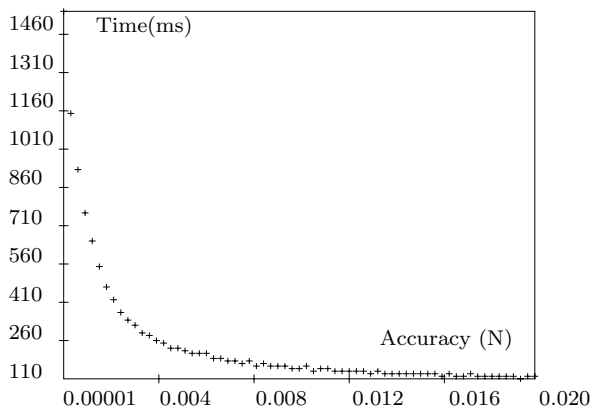


Figure 8.2. Computation time of the extrema of the joint forces versus the desired accuracy on these forces for a box workspace on a DELL D400, 1.2 GHz.

8.2.3. MAXIMAL JOINT FORCES IN A GENERAL WORKSPACE

In section 5.6.1 we presented an algorithm that computes the minimal and maximal values for the twist components, and for fixed bound on each joint velocities. The calculation is performed over a workspace defined as a box, and the extremum is computed up to a pre-defined accuracy. The procedure is based on an interval analysis algorithm that computes the enclosure of the solutions in $\dot{\mathbf{X}}$ of the linear systems $\dot{\Theta}_a = J_{fk}^{-1} \dot{\mathbf{X}}$ where J_{fk}^{-1} is an interval matrix. In our case we have a similar problem with the linear interval systems $\mathcal{F} = J_{fk}^{-T} \boldsymbol{\tau}$ and the algorithm may be used as well (see the interval appendix for the details).

8.2.4. MAXIMAL WRENCH IN A POSE

We may want to determine the distribution of the forces that are produced by the manipulator when the joint forces are bounded. The classical hypothesis is to assume that $\boldsymbol{\tau} \boldsymbol{\tau}^T \leq 1$, so that we obtain, with the help of equation (8.1), that $\mathcal{F}^T J J^T \mathcal{F} \leq 1$. This relation changes the hypersphere of the joint forces into a hyperellipsoid that is called the *resistivity ellipsoid*.

As usual, the ellipsoid concept is not entirely satisfying. It implies that all joint forces are of the same type (force or torque) and a dependence between joint forces that are in reality independently bounded. A better hypothesis is to assume that each joint force has an absolute value bounded by 1. In that case the hyper-cube in the joint force is mapped by J^{-T} into the *wrench polyhedron* in the wrench space. Figures 8.3,8.4 present projections of this polyhedron in the wrench space for the INRIA *left hand* in its nominal position. The wrench polyhedron changes slightly for different

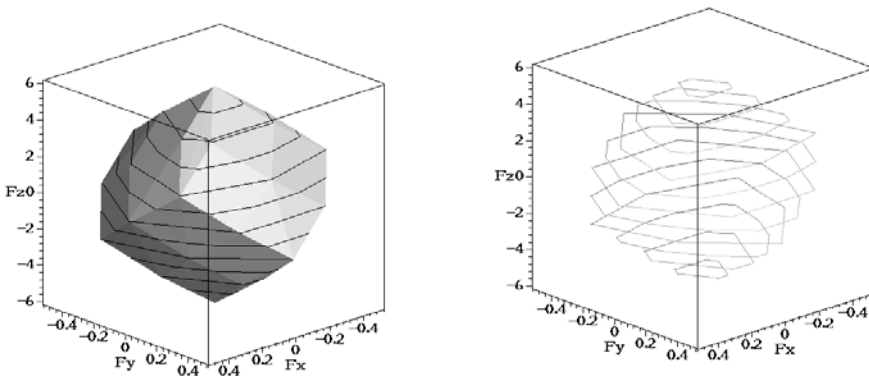


Figure 8.3. Maximal forces for the INRIA *left hand* in its nominal position when all the absolute values of the components of $\boldsymbol{\tau}$ are bounded by 1. On the left, the wrench polyhedron, on the right, some contours.

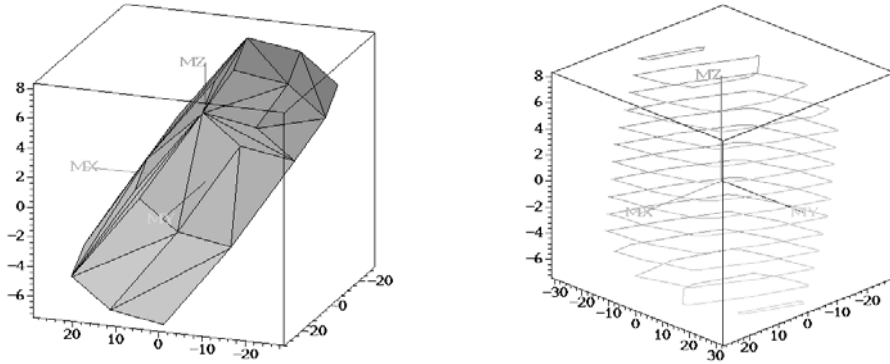


Figure 8.4. Maximal torques for the INRIA *left hand* in its nominal position when all the absolute values of the components of τ are bounded by 1. On the left, the torque polyhedron, on the right, some contours

poses. Figure 8.5 presents the intersection of the force polyhedra obtained for various poses that differ from the nominal pose only by one change in the pose parameters (the amount of change is 5 for the translation and 15 degrees for the rotation). At the nominal pose, the volume and surface of the force polyhedron are 3.836, 22.81, while the intersection has a volume and surface of 1.675, 10.92. Note also that the component that is the most reduced is F_z , whose maximal value is halved.

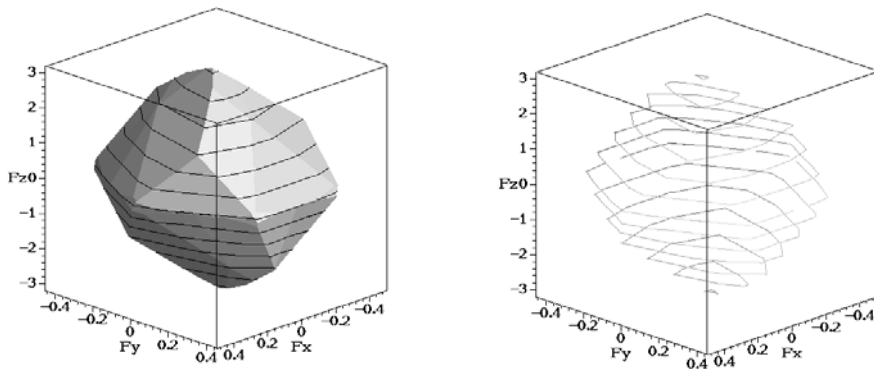


Figure 8.5. Intersection of the force polyhedra for the INRIA *left hand* obtained at various poses. It gives the minimal set of forces that may be applied by the end-effector when all the absolute values of the components of τ are bounded by 1. On the left, the intersection of force polyhedra, on the right, some contours.

8.2.5. MAXIMAL WRENCH IN A WORKSPACE

Using the relation $\mathcal{F} = J^{-T}\boldsymbol{\tau}$ and a global optimization numerical procedure, we can calculate the extremal values of each component \mathcal{F}_i of the wrench that may be applied on the platform for bounded joint forces.

We consider, as an example, the INRIA *left hand*, in which the joint forces are bounded by 300 N because there are force sensors in the links. Figure 8.6 shows the maximal forces and torques that can be applied when the platform moves in the $x-y$ plane.

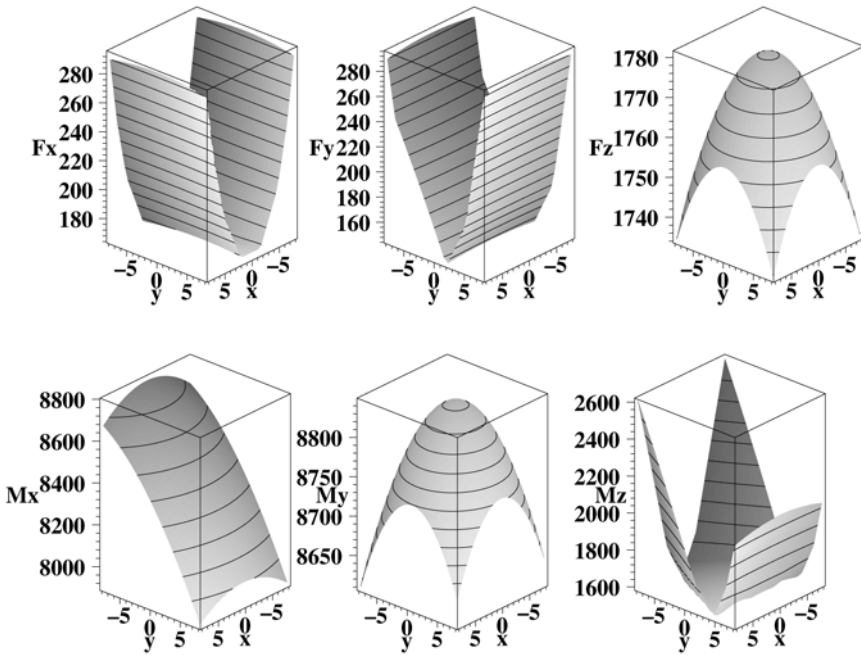


Figure 8.6. Maximal forces and torques that can be applied on the moving platform of the INRIA *left hand* when the platform moves in the $x - y$ plane. All joint forces are bounded by 300 N.

8.3. Force performances indices

Force performance indices may be defined to characterize the overall static behavior of the robot. As equation (8.2) involves the inverse kinematic jacobian, the indices presented in the "Velocity" chapter can be used (hence they are often coined *kinetostatic indices*). Note that, as for the velocity indices, the choice of the norm is important: for example the Euclidean norm should not be used to indicate that the actuated joint forces are

bounded(314). Classical indices are the *force transmission index* (173), based for leg i on the angle between the velocity of the end-effector induced by $\dot{\Theta}_i$, and the force produced on the end-effector by the leg i and the *joint force index* (86) (ratio of maximal joint forces to external load). Chang (86) provides a review of these indices, and proposes a *mean force transmission index* that is however quite difficult to calculate.

Zhang (654) suggests using the mean value and the standard deviation of the trace of the compliance matrix, measures that have the drawback of being very difficult to calculate. Krut (335) suggests using the radius of the largest sphere centered at the origin and included in the task force polyhedron (i.e. the largest force than can be balanced in any direction). Although interesting, this measure does not cover the torque ability, and does not describe the ability of the robot to resist a combined load.

8.4. Parallel robots as force sensors

The simple relation existing between the wrench and the joint forces has enticed numerous researchers to use parallel structures as force sensors. For example, a 6-*UPS* robot with segments that are submitted almost only to traction-compression stresses, will require only a force cell in each link to get the measurement of the joint forces. Then we may calculate the wrench acting on the moving platform with the help of the inverse kinematics jacobian matrix. This principle was suggested as early as 1979 by Rees Jones (504), and is used in many commercial force sensors. Research in this field is still very active (see the references Web page): for example Ranganat (495) proposes using a near singular configuration to improve the sensitivity of the force sensor.

The INRIA *left hand* was actually designed for forces/torques measurements. Sensors with strain gages were placed in each link; these allowed us to measure the vertical forces with an accuracy of about 0.15 N, and the horizontal forces with an accuracy of about 0.03 N. This anisotropy in the measurement accuracy derives from the geometry of the robot.

8.5. Stiffness and compliance

As mentioned in the introduction, many elements may influence the stiffness of a robot, and a full analysis is difficult. The stiffness of a manipulator has many consequences for its control since it conditions its bandwidth. For serial manipulators, the bandwidth is low, only reaching a few Hz at best, and is usually much better for parallel structures. In this section we will assume that only the actuated joints are compliant, while all the other elements are perfectly rigid. We now establish the stiffness matrix of a parallel robot and focus on the 6-*UPS* robot (the stiffness matrix for other

robots may be established using the same procedure, see for example (652) for robots with 3, 4 and 5 d.o.f.).

8.5.1. STIFFNESS MATRIX OF A PARALLEL ROBOT

8.5.1.1 Elastic model

The stiffness of a parallel robot may be evaluated by using an elastic model for the variations of the joint variables as functions of the forces that are applied to the link. In this model, the change $\Delta\Theta$ in the joint variable Θ when an joint force τ is applied on the link is

$$\Delta\tau = k\Delta\Theta \tag{8.5}$$

where k is the elastic stiffness of the link, supposed to be identical for all legs. We also have

$$\Delta\Theta = J^{-1}\Delta\mathbf{X}, \quad \Delta\mathcal{F} = J^{-T}\Delta\boldsymbol{\tau}, \tag{8.6}$$

which leads to

$$\Delta\mathcal{F} = kJ^{-T}J^{-1}\Delta\mathbf{X}. \tag{8.7}$$

The *stiffness matrix* \mathbf{K} and the *compliance matrix* \mathbf{C} are therefore

$$\mathbf{K} = kJ^{-T}J^{-1}, \quad \mathbf{C} = \frac{1}{k}JJ^T. \tag{8.8}$$

Duffy (144) notes that this previous derivation supposes that there is no initial load on the elastic element of the link. If we indeed suppose that the length of the unloaded link is ρ_i^0 we will have

$$\begin{aligned} \Delta\mathbf{F} &= \sum_{i=1}^{i=6} k\Delta\rho_i\mathbf{n}_i + k(\rho_i - \rho_i^0)\Delta\mathbf{n}_i, \\ \Delta\mathbf{M} &= \sum_{i=1}^{i=6} k\Delta\rho_i\mathbf{CB}_i \times \mathbf{n}_i + k(\rho_i - \rho_i^0)\Delta(\mathbf{CB}_i \times \mathbf{n}_i), \end{aligned}$$

where \mathbf{n}_i represents the unit vector of link i , and \mathbf{F}, \mathbf{M} the forces and torques that are applied to the platform. Consequently, the stiffness matrix as defined in equation (8.8) is valid only if $\rho_i = \rho_i^0$, and is coined the *passive stiffness*. Duffy actually presents the exact formulation of the stiffness matrix for a 3-RPR planar robot, while a more general analysis may be found in (239; 542; 564). In this analysis the effect of preloading term is included through a term $\partial J^{-T}/\partial x \boldsymbol{\tau}$ referred as the *active stiffness* (187). Note that the above theoretical stiffness matrix has been validated through experimental results using a parallel wire measuring system (78).

To conclude, let us mention the special case of wire robots: numerous works have addressed this problem (see the references Web page). The stiffness matrix in that case is dependent upon the tension in the wires, and theoretical results are usually very close to experimental data.

Let us now illustrate the stiffness matrix on some examples, starting with the stiffness matrix of a 6-UPS in its nominal position. Given the symmetries, we obtain

$$K_{12} = K_{13} = K_{14} = K_{25} = K_{26} = K_{35} = K_{36} = K_{45} = K_{46} = 0 .$$

In general, the stiffness matrix will not be a diagonal matrix. For the 6-UPS INRIA *left hand*, the stiffness matrix is

$$K = k \begin{pmatrix} 0.05077 & 0.0 & 0.0 & 0.0 & 1.848 & 10^{-6} \\ 0.0 & 0.05077 & -10^{-7} & -1.848 & 0.0 & 0.0 \\ 0.0 & -10^{-7} & 5.898 & 0.00002 & 0.0 & 0.0 \\ 0.0 & -1.848 & 0.00002 & 183.71 & 0.0 & 0.0 \\ 1.848 & 0.0 & 0.0 & 0.0 & 183.7 & 0.0008 \\ 10^{-6} & 0.0 & 0.0 & 0.0 & 0.0008 & 4.0 \end{pmatrix} . \quad (8.9)$$

For the 6-PUS *active wrist* we get

$$K_{12} = K_{13} = K_{14} = K_{25} = K_{26} = K_{34} = K_{35} = K_{36} = K_{45} = K_{46} = 0 ,$$

and the stiffness matrix is

$$K = k \begin{pmatrix} 0.1957 & 0.0 & 0.0 & 0.0 & 0.3451 & -0.0004 \\ 0.0 & 0.1957 & 3e^{-6} & -0.343 & 0.0 & 0.0 \\ 0.0 & 3e^{-6} & 6.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.343 & 0.0 & 48.0 & 0.0 & 0.0 \\ 0.3451 & 0.0 & 0.0 & 0.0 & 47.886 & 0.0061 \\ -0.0004 & 0.0 & 0.0 & 0.0 & 0.0061 & 6.175 \end{pmatrix} . \quad (8.10)$$

8.5.1.2 Beam model

It is also possible to model the deformations of the links of a parallel robot by simulating them by beams. The stiffness of a link will then be

$$k_i = \frac{ES}{\rho} ,$$

where E is the Young modulus, S is the surface of the cross-section of the beam and ρ is its length. If K_a is the diagonal matrix such that $K_{a_{ii}} = E_i S_i / \rho_i$, the stiffness matrix K is

$$K = J^{-T} K_a J^{-1} .$$

8.5.2. PASSIVE COMPLIANCE AND FORCE-FEEDBACK CONTROL

The high rigidity of parallel robots is one of the characteristics for which they are selected for some applications. However, there are cases where we may need a certain structural compliance without losing the knowledge of the position of the end-effector. This is a very common case in robotics, especially for tasks when the end-effector may interact with the environment. In that case, having a compliant mechanism allows us to obtain a slower variation of the contact forces. Moreover, this compliance will allow the contact forces to modify the position of the end-effector, which is needed for certain tasks such as assembly. It is thus possible to show that the contact forces due to alignment errors during an assembly may, if the stiffness matrix has been carefully chosen, generate change in the pose of the end-effector that will correct the errors. This principle lies at the basis of assembly devices such as Whitney's RCC (620) which is in fact a 6-*UPS* robot without actuator, and with elastic links.

In the late 90's parallel robots were developed for tasks which involve contact with the surroundings (e.g. assembly). This was motivated by their high positioning accuracy together with the possibility of using them as force sensors. Numerous works have addressed this issue (see the references Web page). It has been shown that closed loop control schemes using force sensors generally improve if there is a passive compliance in the system. Such passive compliance is always present in serial manipulators, whereas it is reduced in parallel manipulators, for structural reasons. Furthermore, for serial robots, the deformations due to the wrench action cannot usually be measured by the internal sensors of the robot. Hence we will ignore the exact position of the end effector during a contact.

As passive compliance is needed for certain applications of parallel robots, we may choose to introduce it by adding elastic elements in the links. For example, we intentionally placed axial elastic elements in each link of the INRIA *left hand*. As opposed to the serial manipulator, the presence of this passive compliance does not affect the calculation of the pose of the end-effector if we are careful enough to assemble the internal sensors so that the deformations of the elastic elements are taken into account. Figure 8.7 shows how the internal sensors of the *left hand* are assembled in order to reach this aim. With this disposition we always measure the real value of the joint coordinates, and, as a consequence, we are able to compute the almost exact pose of the moving platform (otherwise the forward kinematics will have to take into account the mechanical equilibrium, and will be more complex (321)*,(377),(605)). However we note that the stiffness of the elastic elements needs to be large enough, otherwise an instability might occur in the closed-loop control of the robot. Research in

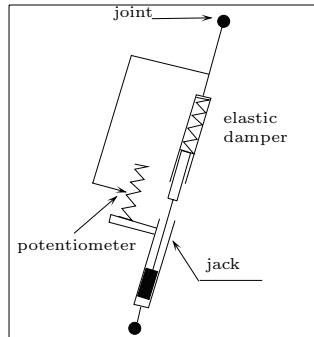


Figure 8.7. Assembly of the internal sensors for the *left hand*, which allows us to measure the real joint coordinates, although the link is compliant.

this area is still active (559).

8.5.3. STIFFNESS MAPS

It might be interesting to have, for a given manipulator, an atlas of the stiffness matrix according to the pose of the end-effector. From this atlas, we would be able to choose a working configuration where the stiffness matrix will fit the needs of the task.

8.5.3.1 *Iso-stiffness maps*

An *iso-stiffness curve* is the locus of C on which a component of the stiffness matrix has a constant value. An *iso-stiffness map* is a set of iso-stiffness curves for different values of a particular component of the stiffness matrix. We may illustrate this concept with the two particular cases of the 6- \underline{UPS} *left hand* and the 6- \underline{PUS} *active wrist*. We present the results of a program that was written by C. Gosselin (187), and which calculates iso-stiffness curves in planar horizontal cross-sections of the workspace, the orientation of the end-effector being fixed. Figure 8.8 presents a few iso-stiffness curves obtained when the stiffness of the elastic elements is 1. We can also trace both iso-stiffness surfaces and iso-stiffness volumes, as shown on figure 8.10.

Figure 8.9 shows a few iso-stiffness curves for the *active wrist*. These results prove that it is possible, within certain limits, to adjust the stiffness of the manipulator according to the task to be done, simply by changing its pose.

8.5.3.2 *Iso-stiffness of 6- \underline{UPS} robot*

In this section we assume that the orientation of the moving platform is constant, and we specifically look at the iso-stiffness loci according to the

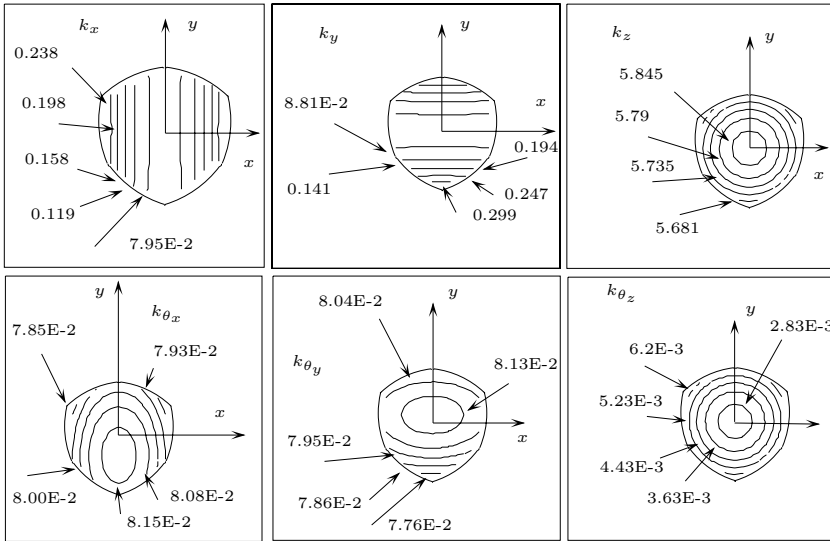


Figure 8.8. Iso-stiffness for the INRIA left hand ($z_c=53.3$, $\psi = \theta = \phi = 0$).

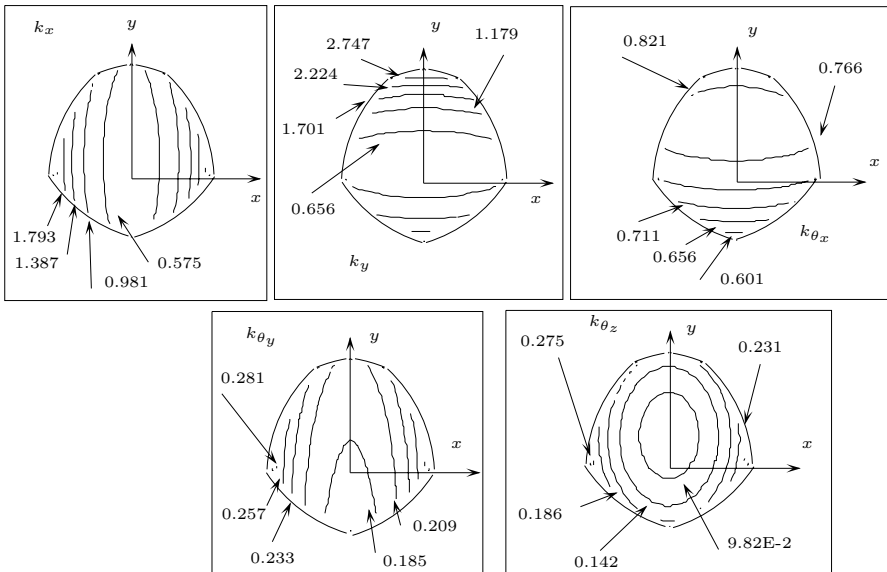


Figure 8.9. Iso-stiffness curve for the active wrist ($z_c = 19$, $\psi = \theta = \phi = 0$).

position of C .

Let us first consider the k_x term of the stiffness matrix:

$$k_x = k \sum_{i=1}^{i=6} \frac{(x_c - x_{a_i} + x_{b_i})^2}{(x_c - x_{a_i} + x_{b_i})^2 + (y_c - y_{a_i} + y_{b_i})^2 + z_c^2} = k \sum_{i=1}^{i=6} T_i(x_c) \quad (8.11)$$

For an SSM, symmetry shows that

$$T_1(x_c) = T_2(-x_c) , \quad T_3(x_c) = T_6(-x_c) , \quad T_4(x_c) = T_5(-x_c) ,$$

Therefore $k_x(x_c) = k_x(-x_c)$ and, as a consequence, the iso-stiffness curves for k_x in an horizontal plane are symmetrical with respect to the y axis. Let us now suppose that the robot is quite slender and that therefore

$$z_c \gg x_c - x_{a_i} + x_{b_i} , \quad z_c \gg y_c - y_{a_i} + y_{b_i} .$$

Let

$$x_i = x_c - x_{a_i} + x_{b_i} , \quad y_i = y_c - y_{a_i} + y_{b_i} .$$

The first order approximation of equation (8.11) is

$$k_x \approx k \sum_{i=1}^{i=6} \frac{x_i^2}{z_c^2} - \frac{x_i^4}{z_c^4} - \frac{x_i^2 y_i^2}{z_c^4} \quad (8.12)$$

so that the first order approximation to the iso-stiffness surface $k_x = K_x$ is

$$-6x_c^4 - 6x_c^2 y_c^2 + a_2 x_c^2 + a_3 y_c^2 + a_4 y_c + a_5 = \frac{K_x z_c^4}{k} \quad (8.13)$$

The coefficients a_i depend only on the geometry of the mechanism. The term k_y is similar

$$k_y = k \sum_{i=1}^{i=6} \frac{y_i^2}{x_i^2 + y_i^2 + z_c^2} = k \sum_{i=1}^{i=6} S_i(x_c) . \quad (8.14)$$

Given the symmetries of a SSM, we have

$$S_1(x_c) = S_2(-x_c) , \quad S_3(x_c) = S_6(-x_c) , \quad S_4(x_c) = S_5(-x_c) ,$$

The iso-stiffness curve in a horizontal plane will therefore be symmetrical with respect to the y axis. Developing equation (8.14) to the first order, we obtain

$$k_y \approx k \sum_{i=1}^{i=6} \frac{y_i^2}{z_c^2} - \frac{y_i^4}{z_c^4} - \frac{x_i^2 y_i^2}{z_c^4} \quad (8.15)$$

Developing and taking the symmetries into account, we find the iso-stiffness surface $k_y = K_y$ as

$$-6y_c^4 - 6x_c^2y_c^2 + b_2x_c^2 + b_3y_c^2 + b_4y_c + b_5 = \frac{K_y z_c^4}{k} \tag{8.16}$$

Lastly, for the term k_z , we have

$$k_z = k \sum_{i=1}^{i=6} \frac{z_c^2}{x_i^2 + y_i^2 + z_c^2} = k \sum_{i=1}^{i=6} S_i(x_c) , \tag{8.17}$$

which gives, when it is developed to the first order,

$$k_z \approx 6k - k \sum_{i=1}^{i=6} \frac{x_i^2 + y_i^2}{z_c^2} . \tag{8.18}$$

The iso-stiffness surface is therefore a cone with circular section and vertical axis (figure 8.10).

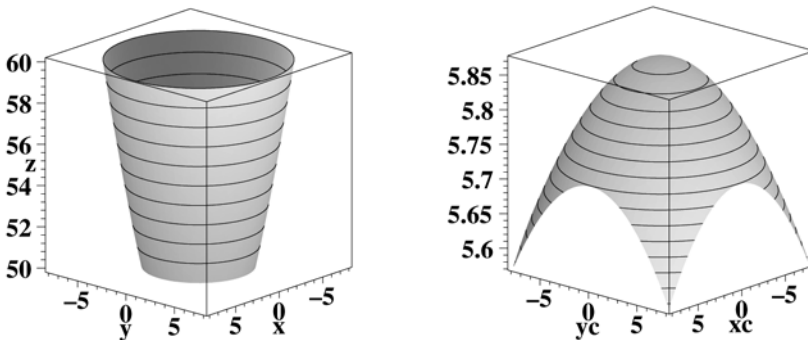


Figure 8.10. On the left, the iso-stiffness surface $k_z = 5.8$ of INRIA left hand for a constant orientation. On the right, the stiffness k_z according to the platform position in the $x - y$ plane

8.5.4. EXTREMA OF THE STIFFNESS IN A WORKSPACE

The purpose of this section is to calculate the extrema of the stiffness for a slender 6-UPS robot when C lies in a given workspace. We specifically look for the extremal values of the diagonal elements of the matrix K , which will be called the *main stiffnesses*. We showed in (407) that if the orientation of the end-effector was constant it is possible to calculate exactly the stiffness extrema over a box workspace (and consequently for any other type of workspace defined in terms of the end-effector parameters, by using the principle described in section 13.3). Note that the extrema of the stiffnesses

for a box will always be obtained on the horizontal faces of the box. For a general box workspace we will have to rely usually on a numerical global optimization procedure.

8.5.5. STIFFNESS AND DESIGN

It may be interesting to design a robot so that its stiffness matrix is close to a desired one, at least for some poses. This may be done either by choosing an appropriate mechanical structure, by an appropriate dimensioning or by control. Artigue (17) suggested a specific robot architecture where the stiffness matrix is diagonal (which is particularly interesting for a force sensor). Hashimoto (220) suggested a method for designing Lee's 3 d.o.f. robot (figure 2.17) so that it possesses a fixed stiffness matrix. Goswami (202) proposes a very general approach for synthesizing passive parallel mechanisms so that they have a fixed *accommodation matrix* (this matrix relates the wrench acting on the end-effector to its twist). Kim (319) calculated the stiffness matrix of a 3-*RRR* planar robot and determined the geometry of the robot so that this matrix is diagonal. Dimensioning may change drastically the values in the stiffness matrix but not so much its general structure. A restricted set of stiffness matrices may be obtained by dimensioning (97; 240),(509)* and designing for stiffness requirements is a difficult problem. For example Simaan shows that the stiffness matrix may be modified with redundant robots but that there are specific configurations for which this control will be lost (541).

Bhattacharya (38) introduces various measures to qualify the stiffness of a robot over a given workspace. Assuming that the Euclidean norm of the wrench is bounded by 1 all wrenches lie on a unit sphere that is mapped by the transformation $K^T K$ to an ellipsoid, the *flexibility ellipsoid*, in the motion space of the end-effector. Bhattacharya uses classical indices derived from $K^T K$ such as the determinant and the average of the minimal eigenvalue, to qualify the stiffness of the robot over a given workspace. The drawbacks of these indices is that they are quite difficult to calculate and that forces and torques are not distinguished.

Maeda (382) shows that for a robot using pneumatic actuators it is possible to modify the compliance of the manipulator by an appropriate control of the jacks. The stability of such control has been studied for planar robot by Svinin (563).

8.6. Static balancing

An interesting problem is that of statically balancing parallel robots. Consider a planar parallel robot in a vertical plane. The masses of the links induce wrench on the end-effector, and the actuators have to produce a

wrench in order to keep a given pose. The aim of static balancing is to reduce (or ideally to cancel) this wrench by adding mechanical elements to the robot. Static balancing can be done by using either counterweights or springs, so that the robot is in mechanical equilibrium in all poses, while the actuators do not provide any force or torque. The drawback of counterweights is that they have a negative influence on the dynamic performances; on the other hand, using springs introduces more unknowns than counterweights.

The problem of static balancing is to determine first if a given architecture may be balanced over its workspace, and then to calculate the necessary counterweight masses and their positions on the links (or the attachment points and stiffness of the springs). This problem was addressed early by Dunlop (145) who suggested the use of counterweights to balance a 2 d.o.f. parallel robot used for antenna aiming, and by Jean (282) for planar robots. In that case the balancing conditions are sufficiently simple to be managed by hand or through a numerical algorithm.

The problem of statically balancing spatial parallel robots is much more complex. The most active team in this field is led by Gosselin and co-workers; they have produced many papers on this subject (153)*, (201)*. An important result was proved by this team (349): it is not possible to balance a 6-UPS robot in all poses by using only counterweights. Gosselin and his co-workers then suggest other mechanisms using parallelograms which can be balanced using springs (153; 428; 610).

Dynamic balancing may also be considered, the problem being to minimize the changes in the inertia over the workspace (628). This issue will be addressed in the "Dynamics" chapter.

8.7. Exercises

Exercise 8.1: Consider a robot with bounded joint forces submitted to a pure horizontal force. Determine the maximal forces that may be applied to it in a given pose by drawing the allowed forces region in the F_x-F_y plane.

Exercise 8.2: Consider a slender 6-UPS robot with a platform moving within a horizontal plane, with a constant orientation and with individually bounded joint forces. Determine the locus of the poses in the $x-y$ plane where the maximum force F_z has a constant value.

Exercise 8.3: Consider a slender 6-UPS robot, with a planar platform moving within a horizontal plane, with a constant orientation ($\psi = 0, \theta = 0, \phi = 0$), and with individually bounded joint forces. The platform is submitted only to a torque M_y . Determine the locus of the pose in the $x-y$ plane where the torque M_y is maximum.

Exercise 8.4: We consider a 3- $R\underline{P}R$ planar robot with points A_i, B_i given by

$$A_1 = (0, 0), A_2 = (5, 0), A_3 = (2, 4), B_1 = (0, 0), B_2 = (2, 0), B_3 = (1, 2).$$

This robot is submitted to a constant wrench $F_x = 1, F_y = 0, M_z = 0$, while point A_1 moves on the line segment going from $(3,3)$ to $(5,3)$, the mobile platform having a constant orientation that is defined by $\theta = 0$. Determine the extrema of the joint forces that act on the links.

Exercise 8.5: Resolve the previous problem, supposing that the robot moves within a horizontal rectangle with corners at $(3,3), (5,5)$.

Exercise 8.6: Find the center of the iso-stiffness curves $k_z = K_z$ in the horizontal plane of a slender 6- UPS robot. Show that for a SSM with $\psi = 0, \theta = 0, \phi = 0$ the iso-stiffness surface axis intersects the point with coordinates $(0,0,0)$.

Problem 8.1: Determine the extrema of the joint forces that act upon a 6- UPS robot which is submitted to a given wrench, while the platform moves within a given 6D workspace.

Problem 8.2: Determine the extrema of the wrench that act on a 6- UPS robot with bounded joint forces, while the platform moves within a given 6D workspace.

Problem 8.3: Establish the stiffness matrix of a 6- UPS robot with segments with unloaded lengths ρ_i^0 and current lengths ρ .

Problem 8.4: Determine the extrema of the stiffnesses for a non slender 6- UPS robot and for a given 6D workspace.

Problem 8.5: Determine if it is possible to balance a horizontal 6- UPS robot using springs.

Problem 8.6: Determine for a wire-driven robot which has no force closure (e.g. a planar robot with only 3 wires) the equilibrium pose for a given load.

Dynamics

This chapter will deal with the direct and inverse dynamics i.e. the determination of the relations between the generalized accelerations, velocities, coordinates of the end-effector and the joint forces. We will also consider their use for simulation and control.

9.1. Introduction

Dynamics plays an important role in the control of parallel robots for some applications:

1. *fast and/or heavily loaded robots*: they operate over a relatively large workspace at a speed such that the dynamic effects have a substantial role in the end-effector motion. Flight simulators or pick-and-place are examples of such applications
2. *high bandwidth robots*: such robots operate over a very small workspace but at a high frequency. A typical application of such a robot is vibration simulation (394)
3. *structurally sensitive robots*: the structure of these robots is such that dynamic effects, even at low speed, may significantly modify their behavior. Typical of this category are wire robots (306; 535) and flexible robots (296)*,(353). Pritschow (489) notes also that for high-speed machining the dynamic errors have a much stronger impact than the static errors.

A first issue has to be addressed: do we need to establish the dynamic models of the robot for control purposes ? There is indeed one school of thought that recommends that dynamic models should not be used because modeling errors are too numerous (some parameters which appear in the dynamics relations indeed are difficult to estimate, even though on-line estimation methods have been proposed (39)). Thus instead of considering the whole system, some suggest that each actuator be controlled independently with a control law more robust than a simple PID (93). It is also argued that establishing a usable control law based on the dynamics relations is difficult not only because of the complexity of the dynamics relations, but also because of the difficulty of managing the direct kinematics which is always encountered in the dynamics relations.

Still, these arguments that may indeed be discussed for category 1 robots are less valid for the other categories. Hence we will present in this chapter a state of the art for the calculation of the dynamic models. This complex subject has been extensively studied (320; 441; 456) and there is an extensive literature specifically devoted to parallel robots (350; 500; 561; 604). There are two different types of dynamic models:

- *inverse dynamics*: being given the trajectory, velocities and acceleration \mathbf{X} , $\dot{\mathbf{W}}$, $\ddot{\mathbf{W}}$ of the end-effector, determine the actuated joint forces $\boldsymbol{\tau}$. Note that the general formulation the inverse model will be

$$\mathbf{M}(\mathbf{X})\ddot{\mathbf{W}} + \mathbf{C}(\dot{\mathbf{W}}, \mathbf{X}, (\Theta)) + \mathbf{G} = \boldsymbol{\tau}$$

where \mathbf{M} is the positive-definite inertia matrix, \mathbf{G} the gravitational term and \mathbf{C} the centrifugal and Coriolis term. This formulation is the same as for serial robots.

- *direct dynamics*: being given the actuated joints forces/torques, determine the trajectory, velocities and acceleration of the end-effector

A classical method for calculating the dynamic models of closed-chains is to consider first an equivalent tree-structure, and then to consider the system constraints by the use of Lagrange multipliers or d'Alembert's principle (3; 372), a principle that has proved to be valid in (371). We will illustrate this method in section 9.2. Other approaches include the use of virtual work (100; 174; 588; 609), Lagrange formalism (172; 179; 376; 419; 447), Hamilton's principle (420)*, Newton-Euler equations (105; 129; 140; 183; 208; 218; 500; 547).

In some cases formalisms may be mixed: for example Zanganeh (650) uses both the Lagrangian and Newton-Euler approaches. An original approach is proposed by Khalil (309): closed-form inverse and direct dynamics models are presented in terms of the dynamic models of the legs which can be calculated by any formalism. Furthermore the *base inertial parameters*, i.e. the minimum number of parameters to compute the models, is presented in closed-form. Computation cost is indeed a key issue for dynamic modeling, as although all formalisms are in theory equivalent (638) their calculation cost may be different. Zhang (657) and Hesselbach (233) have made a thorough analysis of the computation cost of the different formalisms. Geike has investigated how a combined symbolic/numeric approach may be used to automate the calculation of the inverse dynamic model (177), and Bhattacharya (40) shows that the inverse of the inertia matrix, that is needed for the direct dynamics, may be computed recursively.

Frequently, simplifying assumptions are made. For example Clavel (100), Pierrot (476)* and Codourey (105) neglected the rotary inertia of the links, and assumed their masses to be concentrated at the ends. Codourey successfully implemented the models for a *Delta* robot. Do (140) assumes that the

center of mass of each link is located in the middle of the link and that the moving platform is a disk; this simplifies its inertia matrix. Reboulet (500) assumes reasonably that for a 6-*UPS* robot the location of the center of mass of a link does not vary along the link. He obtains a direct dynamics that is simpler than Do's. Ji (287) looked mainly at the influence of the link inertias in the calculation of the dynamics equations by showing that for a Gough platform, the link inertias could be attributed to the platform, which means that the link inertia may be neglected. Nguyen (447), and later Li (364) shows that in practice it is possible to find, for manipulators of the 6-*UPS* type, a bound on the velocity of the linear actuator that allows one to neglect the Coriolis and inertia forces. He also noticed that the layout of the joint centers on the platforms is extremely important: the closer the base is to a triangle, and the platform to a regular hexagon, the less powerful the actuators will have to be. Wendlandt (616) investigates the influence of a spring component on the dynamics equations of a micro-robot. We must also mention the approach of *dynamic balancing* by Xi (628): the tool holder is used as a counterweight to reduce the changes in the inertia matrix. The location and mass of this tool holder are determined for a given task so that the change in the inertia matrix is minimized.

Using dynamic models for control purposes has been studied by Tadoroko (565) and Honneger (237), who suggested using them in an adaptive control scheme, in which the tracking errors are used on-line to correct the parameters used in the dynamics equations. Burdet (65) has tested control laws using dynamic compensation on the 6 d.o.f. Hexaglide, and has shown improvement compared to a linear controller although the velocity of the robot was relatively low. Vibration control with dynamic models has also been addressed, see for example (297)*, (332) and the "Vibration" section in the references Web page. The dynamic effect of an impact on a platform used for spacecraft docking has also been studied (354). Di Gregorio (136) proposes using the eigenvalues of the generalized inertia matrix referred to the end-effector as a dynamic performance index.

We will now present some dynamic model calculation examples.

9.2. MSSM inverse dynamics

We will use the classical method of considering first an equivalent tree-structure, and introducing the system constraints by the use of the Lagrange multipliers (320; 441). In the classical approach for a mechanism containing N rigid bodies and L links, the *tree* mechanism is obtained from the original mechanism by opening the loops at certain passive links so that the obtained tree mechanism has as many independent branches $B = L - N$ as the original mechanism. Thus, for an MSSM which has 31 rigid bodies

and 36 links, we obtain a tree mechanism with 5 independent branches by opening the loops of the MSSM at the level of the ball-and-socket joints on the platform. The number of links of the tree mechanism is equal to the number of rigid bodies of the MSSM i.e. 31 (figure 9.1). Amongst these

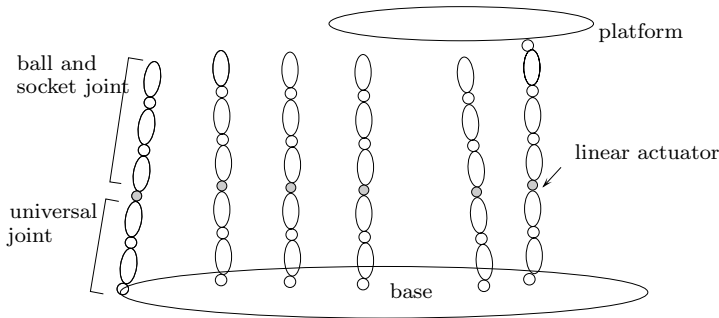


Figure 9.1. The classical tree mechanism used for obtaining the dynamics equations (from Ait-Ahmed (3))

links, we have 6 actuated links and therefore 25 elementary passive links for which we must write a constraint equation. These equations are introduced into the dynamics equations of the tree mechanism with the help of Lagrange multipliers, to obtain the dynamic equations of the original mechanism. We can thus see that the number of multipliers is large, which makes resolution a rather delicate task.

Ait-Ahmed(3) has suggested opening the loops in a larger number of passive links than for the classical method. Thus, for the MSSM, the tree mechanism may be obtained by taking off the platform and the revolute joints, i.e. by taking off 18 passive joints (figure 9.2). The resulting mechanism contains only 18 links, 12 of which require constraint equations, i.e. less than half of those required by the classical method. As for the bodies that are removed, their inertias will be added to the external wrench. Adding the constraints to the tree mechanism may be done by introducing 12 Lagrange multipliers, that may be obtained from the inverse of a 12×12 matrix ω_b , with:

$$\omega_b = \frac{\partial S}{\partial q_b},$$

where q_b represents the parameters of the 12 links and S the 12 constraint equations vector. For the MSSM, Ait-Ahmed manages to establish a symbolic form of the inverse of ω_b , because of the particular structure of the matrix. Thus, we obtain the inverse dynamics for an MSSM, without using simplifying hypotheses; it may be calculated in a relatively small computation time, although still too large for real-time use.

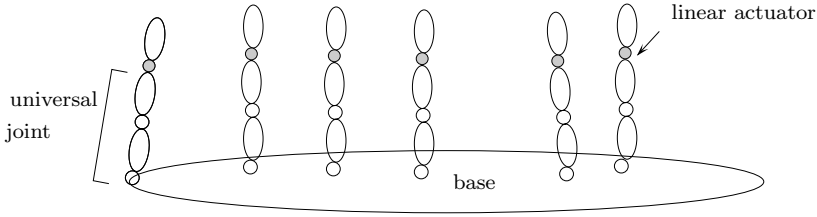


Figure 9.2. The tree model that is used in Ait-Ahmed method for obtaining the dynamics equations

The limits of this method are due to the necessity of calculating the inverse of the 12×12 matrix. Thus, Ait-Ahmed notices that, for the general 6-UPS robot, the matrix ω_b no longer has the particular structure that was found for the MSSM, and he was unable to calculate its inverse. It is therefore necessary to do a costly numerical inversion.

9.3. 6-UPS manipulator dynamics

The complexity of the dynamics equations may be illustrated by the 6-UPS robot, starting with the direct dynamics.

9.3.1. HYPOTHESIS AND NOTATION

The following hypotheses are made for this calculation:

- the inertia matrix I_i of the i -th link, expressed in the frame with origin A_i , and z axis along the link axis, is

$$I_i = \begin{pmatrix} J_i & 0 & 0 \\ 0 & J_i & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- the masses of the links is neglected.

The center of mass of the end-effector is denoted by G , its mass by m and its inertia matrix by I . The acceleration of point G is written as γ_G . For a vector \mathbf{v} with components (x, y, z) we denote by $\bar{\mathbf{v}}$ the matrix such that:

$$\mathbf{v} \times \mathbf{a} = \bar{\mathbf{v}}\mathbf{a} \quad \text{with} \quad \bar{\mathbf{v}} = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

where \mathbf{a} is an arbitrary vector

9.3.2. ALGORITHM PRINCIPLE

The force \mathbf{f}_i that acts at point B_i may be decomposed into two components:

- the given joint force τ_i directed along the unit vector \mathbf{n}_i of the link
- one perpendicular to \mathbf{n}_i , due to inertia, which will be denoted \mathbf{f}_{N_i} (figure 9.3).

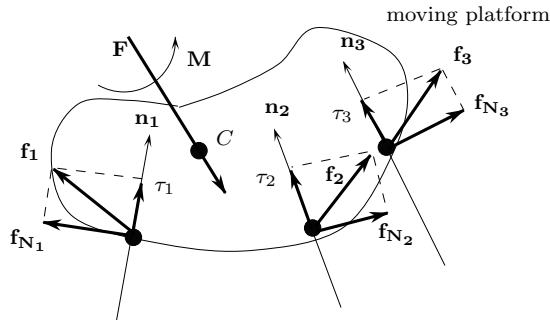


Figure 9.3. External, joint and inertia forces applied on the platform of a parallel robot.

We first assume that we know the vector \mathbf{f}_i , or equivalently the vector \mathbf{f}_{N_i} . We have:

$$\mathbf{f}_i = \tau_i \mathbf{n}_i + \mathbf{f}_{N_i} \quad (9.1)$$

Let \mathbf{F}_N be the resultant force of the forces \mathbf{f}_{N_i} and \mathbf{M}_N their resultant torque about C . If \mathbf{F} and \mathbf{M} are the force and torque applied on the end-effector at point C , the equilibrium equations may be written as

$$\mathbf{F} = \sum_{i=1}^{i=6} \tau_i \mathbf{n}_i + \mathbf{F}_N, \quad (9.2)$$

$$\mathbf{M} = \sum_{i=1}^{i=6} \tau_i (\mathbf{CB}_i \times \mathbf{n}_i) + \mathbf{M}_N. \quad (9.3)$$

Let $\boldsymbol{\sigma}_N$ and $\boldsymbol{\sigma}$ be the dimension 6 vectors:

$$\boldsymbol{\sigma}_N = \begin{bmatrix} \mathbf{F}_N \\ \mathbf{M}_N \end{bmatrix}, \quad \boldsymbol{\sigma} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}. \quad (9.4)$$

Equation (9.3) may be written as

$$\boldsymbol{\sigma} = \mathbf{J}^{-T} \boldsymbol{\tau} + \boldsymbol{\sigma}_N. \quad (9.5)$$

The torque \mathbf{M}_G applied on the end-effector with respect to point G is

$$\mathbf{M}_G = \mathbf{M} + \mathbf{GC} \times \mathbf{F}. \quad (9.6)$$

The Newton-Euler equations are

$$\begin{cases} \mathbf{F} + m\mathbf{g} &= m\gamma_{\mathbf{G}} , \\ \mathbf{M}_{\mathbf{G}} &= \mathbf{l}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{l}\boldsymbol{\Omega} . \end{cases} \quad (9.7)$$

The acceleration of C , denoted by $\gamma_{\mathbf{C}}$, may be obtained from the acceleration $\gamma_{\mathbf{G}}$ by

$$\gamma_{\mathbf{C}} = \gamma_{\mathbf{G}} + \dot{\boldsymbol{\Omega}} \times \mathbf{GC} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{GC}) , \quad (9.8)$$

therefore

$$\mathbf{F} + m\mathbf{g} = m(\gamma_{\mathbf{C}} + \overline{\mathbf{GC}}\dot{\boldsymbol{\Omega}} + (\boldsymbol{\Omega} \times \mathbf{GC}) \times \boldsymbol{\Omega}) . \quad (9.9)$$

From this equation and from equations (9.6,9.7,9.8) we get

$$\mathbf{M} = (1 - m\overline{\mathbf{GC}}^2)\dot{\boldsymbol{\Omega}} - m\overline{\mathbf{GC}}\gamma_{\mathbf{C}} + \boldsymbol{\Omega} \times \mathbf{l}\boldsymbol{\Omega} - m\overline{\mathbf{GC}}(\mathbf{g} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{GC})) . \quad (9.10)$$

We have

$$\boldsymbol{\omega} = ((\boldsymbol{\Omega} \times \mathbf{GC}) \times \boldsymbol{\Omega}) .$$

We may write equations (9.9), (9.10) in matrix form as

$$\boldsymbol{\sigma} = \mathbf{T}_1 \dot{\mathbf{W}} + \mathbf{T}_2 . \quad (9.11)$$

where $\dot{\mathbf{W}}$ represents the derivatives of the twist, \mathbf{T}_1 is a 6×6 matrix and \mathbf{T}_2 is a vector of dimension 6, defined by

$$\mathbf{T}_1 = \begin{pmatrix} mI_3 & m\overline{\mathbf{GC}} \\ -m\overline{\mathbf{GC}} & 1 - m\overline{\mathbf{GC}}^2 \end{pmatrix} , \quad \mathbf{T}_2 = \begin{bmatrix} m\boldsymbol{\omega} - m\mathbf{g} \\ \boldsymbol{\Omega} \times \mathbf{l}\boldsymbol{\Omega} + m\overline{\mathbf{GC}}(\boldsymbol{\omega} - \mathbf{g}) \end{bmatrix} , \quad (9.12)$$

where I_3 represents the 3×3 identity matrix. From (9.5),(9.11) we get

$$\mathbf{T}_1 \dot{\mathbf{W}} + \mathbf{T}_2 = \mathbf{J}^{-\mathbf{T}} \boldsymbol{\tau} + \boldsymbol{\sigma}_N . \quad (9.13)$$

This equation concludes the first stage of our calculation. We have established the derivatives of the twist at C as functions of the joint forces and of the wrench produced by the forces perpendicular to the links. This wrench may also be expressed as a function of the derivatives of the twist: we will now determine this relation.

Let $\gamma_{\mathbf{i}}$ be the acceleration of point B_i . We have

$$\gamma_{\mathbf{i}} = \gamma_{\mathbf{C}} + \dot{\boldsymbol{\Omega}} \times \mathbf{CB}_i + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{CB}_i) , \quad (9.14)$$

which we write in matrix form as

$$\gamma_{\mathbf{i}} = \mathbf{U}_1 \dot{\mathbf{W}} + \mathbf{U}_2 , \quad (9.15)$$

where \mathbf{U}_{1_i} is a 3×6 matrix and \mathbf{U}_{2_i} a vector of dimension 3, defined by

$$\mathbf{U}_{1_i} = \left(\begin{array}{cc} 1_3 & -\overline{\mathbf{CB}_i} \end{array} \right) \quad \mathbf{U}_{2_i} = \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{CB}_i) . \quad (9.16)$$

The projection $\gamma_{\mathbf{N}_i}$ of γ_i on a plane perpendicular to \mathbf{n}_i is given by

$$\gamma_{\mathbf{N}_i} = (\mathbf{n}_i \times \gamma_i) \times \mathbf{n}_i , \quad (9.17)$$

which we write in matrix form, using equation (9.15), as

$$\gamma_{\mathbf{N}_i} = -\overline{\mathbf{n}_i}^2 \mathbf{U}_{1_i} \dot{\mathbf{W}} - \overline{\mathbf{n}_i}^2 \mathbf{U}_{2_i} . \quad (9.18)$$

Besides, we have

$$\mathbf{f}_{\mathbf{N}_i} = -\frac{J_i}{\rho_i^2} \gamma_{\mathbf{N}_i} . \quad (9.19)$$

The components of the vector $\boldsymbol{\sigma}_N$ are

$$\mathbf{F}_N = \sum_{i=1}^{i=6} \mathbf{f}_{\mathbf{N}_i} , \quad \mathbf{M}_N = \sum_{i=1}^{i=6} \mathbf{CB}_i \times \mathbf{f}_{\mathbf{N}_i} . \quad (9.20)$$

Using equations (9.18),(9.19) we get

$$\mathbf{F}_N = \left(\sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{n}_i}^2 \mathbf{U}_{1_i} \right) \dot{\mathbf{W}} + \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} (\overline{\mathbf{n}_i}^2 \mathbf{U}_{2_i}) , \quad (9.21)$$

and

$$\mathbf{M}_N = \left(\sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{CB}_i} \overline{\mathbf{n}_i}^2 \mathbf{U}_{1_i} \right) \dot{\mathbf{W}} + \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{CB}_i} \overline{\mathbf{n}_i}^2 \mathbf{U}_{2_i} . \quad (9.22)$$

We combine these last two equations to determine $\boldsymbol{\sigma}_N$, which we write in matrix form:

$$\boldsymbol{\sigma}_N = \mathbf{V}_1 \dot{\mathbf{W}} + \mathbf{V}_2 , \quad (9.23)$$

where \mathbf{V}_1 is a 6×6 matrix and \mathbf{V}_2 a vector of dimension 6:

$$\mathbf{V}_1 = \left(\begin{array}{c} \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{n}_i}^2 \mathbf{U}_{1_i} \\ \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{CB}_i} \overline{\mathbf{n}_i}^2 \mathbf{U}_{1_i} \end{array} \right) , \quad \mathbf{V}_2 = \left[\begin{array}{c} \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} (\overline{\mathbf{n}_i}^2 \mathbf{U}_{2_i}) \\ \sum_{i=1}^{i=6} \frac{J_i}{\rho_i^2} \overline{\mathbf{CB}_i} \overline{\mathbf{n}_i}^2 \mathbf{U}_{2_i} \end{array} \right] . \quad (9.24)$$

From equations (9.13),(9.23) we get

$$(\mathbf{T}_1 - \mathbf{V}_1) \dot{\mathbf{W}} + (\mathbf{T}_2 - \mathbf{V}_2) = \mathbf{J}^{-\mathbf{T}} \boldsymbol{\tau} ; \quad (9.25)$$

we have

$$\dot{\mathbf{W}} = (\mathbf{T}_1 - \mathbf{V}_1)^{-1} \mathbf{J}^{-\top} \boldsymbol{\tau} - (\mathbf{T}_1 - \mathbf{V}_1)^{-1} (\mathbf{T}_2 - \mathbf{V}_2) . \quad (9.26)$$

This equation represents the direct dynamics of a 6-UPS robot. The inverse dynamics, which is more interesting as far as control is concerned, may simply be established by

$$\boldsymbol{\tau} = \mathbf{J}^\top (\mathbf{T}_1 - \mathbf{V}_1) \dot{\mathbf{W}} + \mathbf{J}^\top (\mathbf{T}_2 - \mathbf{V}_2) \quad (9.27)$$

9.4. 6-PUS robot dynamics

We use the same approach as in the previous section. The force \mathbf{f}_i that is applied at point B_i may be decomposed:

- \mathbf{f}_{s_i} which is directed along the vector \mathbf{n}_i
- \mathbf{f}_{N_i} perpendicular to \mathbf{n}_i , which is due to inertia (figure 9.4)

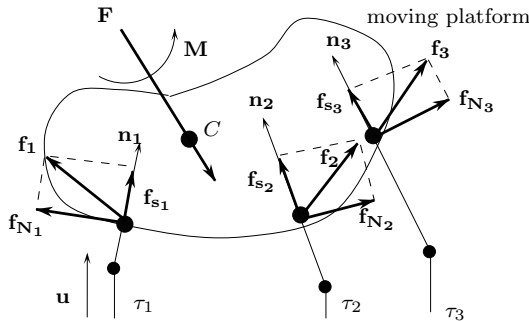


Figure 9.4. External, joint and inertia forces that are applied to the active wrist

The force \mathbf{f}_{s_i} is related to the joint force τ_i by

$$\mathbf{f}_{s_i} = \tau_i (\mathbf{n}_i \cdot \mathbf{u}) \mathbf{n}_i . \quad (9.28)$$

Using the notation of the previous section, we may write

$$\mathbf{F} = \sum_{i=1}^{i=6} \tau_i (\mathbf{n}_i \cdot \mathbf{u}) \mathbf{n}_i + \mathbf{F}_N , \quad (9.29)$$

$$\mathbf{M} = \sum_{i=1}^{i=6} \tau_i (\mathbf{n}_i \cdot \mathbf{u}) (\mathbf{C}\mathbf{B}_i \times \mathbf{n}_i) + \mathbf{M}_N . \quad (9.30)$$

Let \mathbf{J}_α^{-1} be the matrix defined by the rows

$$[(\mathbf{n}_i \cdot \mathbf{u}) \mathbf{n}_i , (\mathbf{n}_i \cdot \mathbf{u}) (\mathbf{C}\mathbf{B}_i \times \mathbf{n}_i)] . \quad (9.31)$$

We note that this matrix is not the transpose of the inverse of the kinematic jacobian matrix of the active wrist. Equations (9.30) may be written in matrix form as

$$\boldsymbol{\sigma} = \mathbf{J}_\alpha^{-1} \boldsymbol{\tau} + \boldsymbol{\sigma}_N . \quad (9.32)$$

The calculation of $\boldsymbol{\sigma}_N$ is identical to that which we did in the previous section. The dynamic model is obtained simply by replacing the matrix \mathbf{J}^T by the matrix \mathbf{J}_α in equations (9.26), (9.27). We therefore have

$$\dot{\mathbf{W}} = (\mathbf{T}_1 - \mathbf{V}_1)^{-1} \mathbf{J}_\alpha^{-1} \boldsymbol{\tau} - (\mathbf{T}_1 - \mathbf{V}_1)^{-1} (\mathbf{T}_2 - \mathbf{V}_2) , \quad (9.33)$$

$$\boldsymbol{\tau} = \mathbf{J}_\alpha (\mathbf{T}_1 - \mathbf{V}_1) \dot{\mathbf{W}} + \mathbf{J}_\alpha (\mathbf{T}_2 - \mathbf{V}_2) . \quad (9.34)$$

9.5. Computation time

Using the inverse dynamics within a control loop requires much calculation. After having measured the joint coordinates, we use the direct kinematics to obtain the pose of the end-effector. We then evaluate vector \mathbf{GC} (and therefore $\overline{\mathbf{GC}}$) defined by $\mathbf{GC} = \mathbf{RGC}_r$. At this stage, we calculate matrix \mathbf{T}_1 , and then estimate the vector $\boldsymbol{\Omega}$ from the orientation of the end-effector. This allows us to calculate ω , and therefore to obtain the vector \mathbf{T}_2 .

Using the orientation of the end-effector, we calculate \mathbf{CB}_i , which gives us the matrices \mathbf{U}_{1i} and the vectors \mathbf{U}_{2i} . We may also determine the unit vectors \mathbf{n}_i of the links, and therefore matrix \mathbf{V}_1 and vector \mathbf{V}_2 . Lastly, we calculate matrix \mathbf{J}^{-1} (or \mathbf{J}_α^{-1}), which we invert numerically.

Using the pose of the end-effector the joint coordinates and the current velocities as input, we need a computation time of 0.0185 ms for the calculation of the joint forces on a DELL D400, 1.2 Ghz. Using the joint forces as input, we may calculate the accelerations of the end-effector in approximately the same time. To this time should be added the communication time; the sampling time of the inner control loop of the actuator should also be taken into account. In practice, the use of dynamics equations (either inverse or direct) for control purposes seems to have been limited to the *Delta* robot (105) and to the Hexaglide (65).

9.6. Examples

9.6.1. INVERSE DYNAMICS

We present an example of the calculation of the joint forces for the 6-*UPS* INRIA *left hand*, using the accelerations imposed on the moving platform as input. In this simulation we use the following parameters

$$m = 1 \text{ kg} , J_i = 2500 \text{ kg cm}^2 , I_{11} = I_{22} = I_{33} = 400 \text{ kg cm}^2 , I_{ij, i \neq j} = 0 .$$

The accelerations inputs are

$$\gamma_x = 1 \text{ cm/s}^2, t \in [0, 2]; \gamma_x = -1 \text{ cm/s}^2, t \in]2, 4]; \gamma_y = 5 \text{ cm/s}^2, t \in [0, 4];$$

the other components of the acceleration vector are zero. The initial cartesian and angular velocities of the end-effector are zero, and the initial position is $[0,0,53.3]$, while the initial orientation is $[0,0,0]$.

The x, y coordinates of the moving platform are presented in figure 9.5, and the joint forces for the links in figure 9.6.

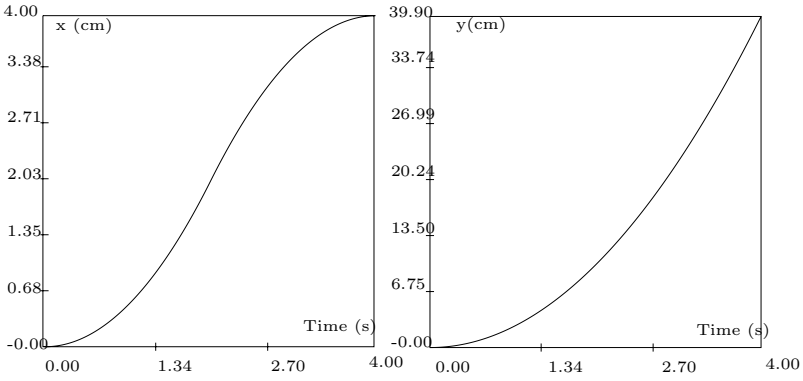


Figure 9.5. The x, y coordinates of the end-effector obtained for given acceleration inputs, obtained by using the inverse dynamics.

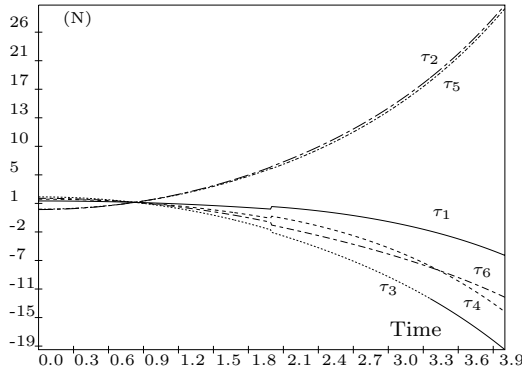


Figure 9.6. Joint forces on the trajectory as calculated with the inverse dynamics.

9.6.2. DIRECT DYNAMICS

We consider the example presented in the previous section to illustrate an application of the direct dynamics. We assume that the end-effector has

to realize a vertical motion, starting from its nominal position. The inverse dynamics shows that all the linear actuators must provide an identical force, that we will suppose constant and equal to 1.65 N. However, an error in the model of one of the actuators, leads to actual applied forces

$$\tau_1 = 1.6665 \text{ N} , \quad \tau_i = 1.65 \text{ N} , \quad \forall i \in [2, 6] .$$

The initial cartesian and angular velocities are assumed to be zero, the initial position of the end-effector is (0,0,53.3), while all of the Euler angles are zero.

Using the direct dynamics, we compute the x, y, z coordinates of the moving platform; these are shown in figure 9.7. We note that the small

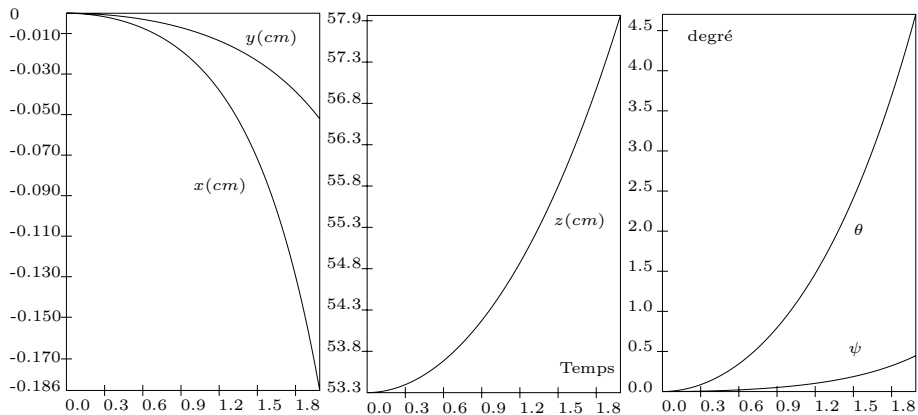


Figure 9.7. Trajectory described by the platform as computed by the direct dynamics

error in the control-loop of the joint forces of link 1 leads to a path that clearly diverges from the required path. The final position that is reached is (-0.186, -0.0521, 57.864), the Euler angles being $\psi=0.447$ degrees, $\theta=4.7031$ degrees, $\phi=0$ degree. With a perfect control, the position would be (0,0,54.57) and $\psi = \theta = \phi = 0$. This example clearly shows that a naive use of the direct dynamics may be quite dangerous.

9.7. Exercises

Exercise 9.1: Show how Lagrange's principle may be used to obtain the inverse dynamics of a *Delta* robot. Note: rather than using only the joint coordinates, we may include the pose parameters of the platform.

Problem 9.1: How can we establish a bound on the computation time of the dynamics equations so that a control law based on these equations will be more efficient than the classical control law that neglects the dynamics?

Calibration

This chapter deals with the identification problem of the parameters that are used for parallel robot modeling. Up to now the analysis presented in the previous sections has assumed a perfect knowledge of the values of structurally intrinsic parameters of the robot, an assumption that does not hold in practice. In this chapter we will focus on methods that allow one to improve the knowledge of the geometry of the actual robots¹.

10.1. Introduction

Practical use of the various analyses presented in the previous sections requires a perfect knowledge of structurally intrinsic parameters of the robot, such as its geometry, dynamic characteristics . . . Thus, position control of a Gough platform requires the knowledge of the location of the centers of the passive joints, as well as the offsets of the links, i.e. the lengths of the legs for a 0 value of the sensor measuring the length changes. Even if estimates of these parameters are available, deviations are unavoidable (e.g. due to manufacturing tolerances) and hence identification procedures may be necessary. In this chapter we will focus on the identification of the geometry of the robot; this is usually called the *kinematic calibration* of the robot. Note that, most of the time, calibration is understood as identifying small deviations of the robot geometry around nominal values, but it may also be used when almost no information on the geometry is available. For example Tadokoro (566) mentions the use of calibration for the identification of the geometry of a wire robot used as a crane during search and rescue operations, and Ji (289) mentions its use for modular robots, i.e. robots whose geometry may be radically changed to adapt their performances to the task.

It must also be noted that there is a fundamental antinomy between optimal design of parallel robot (an issue that will be addressed in the "Design" chapter) and calibration. Indeed one of the purposes of optimal design may be to determine a dimensioning of the robot so that its kinematic behavior is the least sensitive to changes of the robot geometry, while

¹This chapter has benefited from numerous discussions with D. Daney, who is here specially acknowledged

the principle of calibration on the other hand, is on the opposite to use this sensitivity to improve the knowledge of the geometry. Calibration results will hence be sensitive to the quality of design, and therefore comparison of calibration methods based on simulation results will be difficult if the calibrated robots do not have the same sensitivity.

10.2. Types and principle of calibration methods

There are three main types of calibration methods:

1. *external calibration*: methods based on total or partial measurements of the platform poses or of other geometrical elements of the robot through an external device
2. *constrained calibration*: methods that rely on a devoted mechanical system that constrains the robot motion during the calibration process
3. *auto-calibration* or *self-calibration*: methods that rely on the measurements of the internal sensors of the robots. In that case it is required that a n d.o.f. robot has $m > n$ internal sensors

10.2.1. CALIBRATION PRINCIPLE

All calibration methods impose virtual or real constraints on the poses of the end-effector. The principle of the calibration process will be to get these constraints at various poses of the end-effector (called the *calibration poses*) until the number of constraints is large enough to determine which geometry of the robot satisfies them best. This principle is the one that has been used successfully for serial robot calibration, but it was recognized early by Everett (157) that the calibration problem of closed chains was very different from that for serial chains, and requires a specific treatment in term of measurements (for example measuring devices with a small workspace classically used for serial robot are not appropriate for parallel robots) and in terms of processing.

In practice, all the constraints that must be fulfilled at each calibration pose will be written as a set of c constraint equations that are functions of m geometrical parameters \mathcal{P} of the robot, of the measurements obtained at the pose, and of the n pose parameters of the calibration poses. Hence for a set of N calibration poses we get a set of $N \times c$ constraint equations with $m + N \times n$ unknowns. A calibration method is designed in such way that there always is a limit N_l on N such that $N_l \times c \geq m + N_l \times n$ i.e. the number of constraint equations is greater than or equal to the number of unknowns. The constraint equations are usually non-linear, except in some special cases. For example, for micro-robots, linearized constraint equations may be used, as mentioned by Arai (15) and Ojala (454). In an ideal world and

for a perfect calibration method, \mathcal{P} will be the only solution of the constraint equations. In practice, however noise in the measurements and imperfection in the robot modeling will be such that the constraint equations will never be exactly satisfied. Hence a numerical procedure will be used to find a \mathcal{P} that most nearly satisfies the constraint equations. A typical example of such procedure will be to find the \mathcal{P} that minimizes the sum of the squares of the constraint equations.

10.2.2. GENERAL COMPARISON OF CALIBRATION METHODS

External and constrained calibrations share the properties that the robot must be calibrated off-line, and that a specific hardware processing has to be done at each calibration pose (or for each class of calibration poses). Hence these methods will usually be lengthy in time. Furthermore, calibration poses will often be restricted to lie within a small region of the workspace.

On the other hand, for auto-calibration each pose of the robot may be used as a calibration pose, and thus calibration may be performed on-line over the whole workspace of the robot. Furthermore, the additional sensors may be used for other purposes such as, for example, enabling a faster solution of the direct kinematics problem (see section 4.7). But not all parameters may be observable, as we see in next section.

In terms of cost, we will see that external calibration may requires expensive measuring devices to get decent calibration results (although we will see that low-cost measurement devices may be used within an appropriate framework). Constrained calibration is probably the least expensive, while auto-calibration is relatively low-cost, provided that the robot has been designed with this possibility in mind. External and auto calibrations are compared in detail in (123).

10.2.3. ISSUES IN CALIBRATION METHODS

Different issues must be addressed when examining a calibration method:

- *observability*: the possibility of identifying all geometry parameters of the robot. Indeed for some calibration methods some parameters have no effect on the model while some others have an influence only as a group. An observability measure is derived from the $N \times m$ jacobian matrix of the constraint equations, called the *observation matrix*, whose rank will determine the number of observable parameters (308). The observation matrix may sometime be calculated analytically, otherwise it has to be estimated numerically. The set of observable parameters may be obtained by a *QR* decomposition of the observation matrix. The condition number of the observation matrix is usually considered a good index to quantify the calibration process (the lower the

condition number, the better is the calibration process). Here again we should use this information with care as the condition number may mix data with different physical units: in general the condition number cannot be used to compare different calibration methods

- *sensitivity to measurement noise*: noisy sensor measurements will be used during the calibration, and this noise will affect the result
- *choice of the calibration poses*: calibration results may differ according to the poses that are used for creating the constraints equations that will be used for the calibration process. Hence it is necessary to determine which calibration poses will lead to the most accurate calibration
- *accuracy of the calibration*: clearly the best validation of a calibration method will be experimental measurements of the deviation between desired poses and real ones. However such measurements involve also the control of the robot, which may introduce disturbances that are not due to modeling errors. Hence simulation will also play an important role, as it allows one to focus on the efficiency of the calibration process on its own. This implies the need to define indices to characterize the accuracy of the calibration. In most papers not all measured calibration poses are used for the calibration process: some of them are used after the calibration to validate the results. Still, a difficulty occurs when comparing different calibration methods proposed in the literature, as there is no standard calibration index nor comparison benchmark
- *unicity*: there must not be very different sets of geometrical parameters that satisfy the constraint equations in a similar way

These issues will be illustrated on the 6-*UPS* robot. Calibration of redundant robots will not be addressed: internal constraint forces may modify the pose of the end-effector, and must be taken into account (284).

10.3. External calibration

In this method, the moving platform is placed in a certain number of poses by using the actuated joints control, then some geometric features of the robot are measured, with the help of an external measurement device. Note that these measurements may also be used for designing pose corrective laws that modify the control inputs locally (641).

10.3.1. TYPE OF EXTERNAL MEASUREMENTS

A first type of measurement consists in measuring, completely or partially, the pose parameters of the platform. Complete measurements of the location and orientation of the platform may be done with a laser tracker (395), theodolite (662), coordinate measuring machine (123) or op-

tical system (22). All these systems measure the location of 3 points on the platform and reconstruct the location and orientation of the platform from these coordinates; they are all expensive. Less expensive devices have been proposed, as for example a set of LVDT (170); the most promising trend is to use a vision system, a field that was pioneered by Amirat (9) and Maurine (389), and recently revisited with the use of a dedicated calibration board fixed on the platform (125)(506)*.

It is often claimed that a full measure of the pose parameters of the moving platform is difficult to obtain. Hence it has been proposed to measure only some of the pose parameters, for example only 2 orientation parameters, by using 2 inclinometers (37). Partial measurement methods need more calibration poses, and may not allow one to identify all kinematic parameters.

But other geometrical features may be used for calibration purposes: measured distances to a fixed point using a double-ball-bar (DBB) measuring device (571; 615), flatness of planar motion of the robot together with squareness of the motion according to 2 orthogonal axes (244), and deviation of the trajectory with respect to a line (83). An original approach consists in using a vision system not only to measure the pose of the platform but also the leg directions (506).

10.3.2. CALIBRATION WITH DIRECT KINEMATICS

For the modeling of a 6-*UPS* robot, Masory (386) identifies 132 geometric parameters (22 parameters per leg), and Vischer(595) 138 with a closer analysis. This set may be reduced to 42 (the coordinates of the A_i, B_i and the 6 length offsets) if the passive joints are assumed to be perfect. The reduced model is motivated by the work of Wang (608), which shows that the platform positioning is only slightly sensitive to errors in the positions of the joints, for instance those due to manufacturing tolerances. It must be noted however that the work of Wang was performed for a specific robot geometry and is difficult to generalize as it is unclear whether an optimal design (see the "Design" chapter) may not lead to another conclusion.

In 1993 Masory proposed a calibration method (that we will call the *DK method*) which is directly adapted from the one used for serial robot calibration: the constraint equations are $\mathbf{F}(\boldsymbol{\rho}^m) - \mathbf{X}^m = 0$, where $\boldsymbol{\rho}^m, \mathbf{X}^m$ are the measured leg lengths and platform pose parameters while \mathbf{F} represents the direct kinematics. For each calibration pose we get 6 constraint equations, and it is therefore necessary to have at least 22 calibration poses to identify the full model, and 7 for the reduced model. The constraint equations are solved with a Newton-Raphson like scheme using the pseudo-inverse of the system. Masory and Bai(22) agree that the optimal number of calibration

poses is approximately 10; a larger number of poses does not change the accuracy of the calibration.

There are major drawbacks in this approach. It is necessary to assume that the direct kinematics will always provide a real solution, and that it is able to select, among all possible real solutions, the one corresponding to the calibration poses (we have seen in the "Direct Kinematics" chapter that this is a complex issue). Given the noise on $\boldsymbol{\rho}^m$, and using the direct kinematics with various values for the geometrical parameters, it is unclear if there will always be at least one real solution: strictly speaking the algorithm should be adapted to deal with complex solutions, probably a difficult task.

10.3.3. CALIBRATION WITH INVERSE KINEMATICS

Wampler (601) has suggested a unified approach, the *implicit loop formulation*, that may be used both for serial and closed chains. For parallel robots, this approach uses the inverse kinematics. The constraint equations are $\mathbf{G}(\mathbf{X}^m, \boldsymbol{\rho}^m) = 0$, where \mathbf{G} is the inverse kinematics equations. For Masy's reduced model, the observation matrix can be calculated analytically, and its rank is maximal for $N \geq 7$ calibration poses; all parameters are observable (308). The minimal number of calibration poses is identical to that for the DK method. The constraint equations are solved using a statistical approach, assuming zero mean Gaussian noise on the measurements and on the parameters, an assumption that is not really appropriate for parallel robots. Zhuang (662) instead suggests determining \mathcal{P} that minimizes the sum of the squares of the constraint equations. The constraint equations are linearized in term of the unknown parameters, and consequently the changes in the parameters are directly obtained by solving a linear system whose matrix, called the *identification jacobian*, is an approximation to the observation matrix. It is block diagonal, hence the error parameters of each leg can be solved independently. A similar approach was suggested by Oliviers (455); he solves the resulting linear system by using a singular value decomposition. He then studies the influence of the measurement noise by computing the error between the measured pose and the pose calculated for the calibrated robot; it appears that the orientation part of the pose is much more sensitive to the noise than the translation part.

A drawback of minimizing the sum of the squares of the constraint equations has been identified by Daney (124). Daney proposes an interval analysis based approach that determines ranges for the parameters that are guaranteed to include the real parameter values. He then minimizes the sum of the squares of the constraint equations by using a least-square procedure, and notices that for some parameters the resulting values lie outside the certified range provided by the interval method means that some of the

constraint equations are not satisfied, even if the measurement noise is taken into account. In another paper Daney (125) shows on experimental data that there was no solution of the constraint equations. He then investigates the minimal value of an ϵ_i such that there is a solution to $|G_i(\mathbf{X}^m) - \rho_i^m| = \epsilon_i$ for all calibration poses. These quantities allow one to quantify both the quality of the measurement and the errors in the modeling. The problem of fitting the parameters to the constraint equations and pose measurements has also been addressed by Iurascu (278); he proposed a metric to solve the calibration problem. He then finds a multidimensional surface in the parameter space that is a best fit, in the sense of the metric. Even so this best fit may still not satisfy all the constraint equations.

For the 6-*UPS* robot, Wampler identifies 54 parameters: 42 parameters from Masory's reduced model and two parameters $\kappa_i, \theta_i^{\text{off}}$ for each sensor, such that the change in measured leg length is $\Delta\rho_i = \kappa_i\alpha_i + \theta_i^{\text{off}}$, where α_i is the sensor reading. Wampler selects 20 calibration poses, 13 of them being close to the boundary of the workspace. Zhuang (662) uses the Masory's reduced model and 12 calibration poses.

The implicit loop formulation has many advantages over the DK calibration method. Tanaka (572)* has shown that it allows a larger error in the initial guess of the kinematic parameters; for 7 calibrations poses the allowed error is 23%, and only 3.5% for the DK method. This method allows us to avoid the calculation of the direct kinematics (and is therefore faster than the DK method) and the management of the multiple solutions of the direct kinematics. This method has also been validated on robots with less than 6 d.o.f. by Vischer (596)*. In that case this author gives a rule of thumb for the accuracy of the external measurement system: this accuracy should be at least ten times better than the expected gain in the location of the joints (to get this location with an error of 1/10 mm we need to use sensors whose accuracy is better than 1/100 mm).

10.3.4. CALIBRATION WITH CONSTANT LEG LENGTHS

Zhuang (659) proposes another method, later modified by Geng (181). This method relies on the measurement of the position and of the orientation of the platform in various poses in which certain link lengths are kept constant. The initial set of constraint equation for leg i is $G_i(\mathbf{X}^m) - \rho_i^m + l_i = 0$, where ρ_i^m is the sensor measurement, and l_i the leg length offset. Consider a set k of calibration poses where the link i length is kept constant: subtracting the above equation obtained for one calibration pose from the equations obtained for the other calibration poses we get a set of $k - 1$ constraint equations in which the unknowns l_i no longer appear. As Zhuang considers Masory's the reduced model, there are 6 unknowns in the con-

straint equations for leg i and hence only 7 measurements are necessary to obtain a system of 6 equations in these 6 unknowns. When solved, the seventh measurement is used to calculate all the l_i . Geng also shows that carefully choosing the measurement configurations allows us to use only 13 measurements to calibrate the 6 links. The simulation results show that it is better to take measurement configurations in which only 3 links possess fixed lengths; this allows for a larger change in the moving platform configurations.

10.3.5. CALIBRATION WITH OTHER GEOMETRICAL ELEMENTS

Huang (244) proposes using two types of measurements for calibration :

- *flatness*: theoretically the robot performs a motion in a fixed $x-y$ plane, and the distance variations along the z axis are measured by a dial
- *straightness and squareness*: theoretically the robot performs a motion along the x axis, and a dial measures the motion variation in this direction. Similar motion may be performed along the y axis and the dial may be used to evaluate the orthogonality of the motion

The orientation at one calibration pose is measured also. The purpose of such a method is to allow the use of low cost measurement devices. However Huang mentions the use of 154 measurements, thus leading to a lengthy calibration process.

Takeda (571) proposes using a double-ball-bar (DBB) measuring device. The robot performs circular paths, and the deviation from circularity is measured by the DBB. He proposes an algorithm to determine which circular paths should be used for the calibration. Experimental results show that the circularity error on a test path was greatly decreased after calibration, although not all kinematic parameters are observable.

Chai (83) defines a straight line trajectory and uses a laser beam to measure the deviation of the end-effector with respect to this line. Such a measurement does not allow a full calibration of the robot, but Chai's purpose is just to correct the leg length measurement to minimize the deviation from the straight line trajectory. Experimental tests on a commercial robot seem to show that there was a leg length offset. This offset was identified, and after correction the deviation on the test trajectory was reduced by a factor of 7. However this should be confirmed by further tests on the whole workspace of the robot.

The vision-based approach proposed by Renaud (506) combines measurements on the end-effector poses with a specific calibration board and observation of the leg directions. This approach was validated on the 4 d.o.f. I_4 robot with 40 calibration poses.

10.3.6. UNICITY OF THE SOLUTION

The problem of the unicity of the solution of the constraint equations has been addressed by Innocenti (274) in a paper in which he suggests two calibration procedures. His first method uses 7 calibration poses. The link lengths then give 42 equations in the 36 unknowns that are the coordinates of A_i, B_i . For a given calibration pose all 6 equations giving the leg length as functions of the pose parameters include the term $x_c^2 + y_c^2 + z^2$. Innocenti suggests subtracting one leg length equation from each of the remaining 5 in order to get rid of this term. Hence he ends up with an algebraic system of 36 equations in 36 unknowns. By elimination, Innocenti shows that this system may be reduced to a 20 degree polynomial equation in one unknown (this result is closely related to the problem of finding an SS chain supporting a rigid body that has the same length for 7 poses of this body (365)). Hence there is no guarantee that the solution of the calibration problem is unique, even in the absence of noise.

In a second method Innocenti includes the offset of the link lengths as unknowns, and works on the previous principle, although with 8 calibration poses. He again shows that solving the obtained system is equivalent to solving a 20 degree polynomial equation in one unknown: a numerical minimization method such as that of Geng converges towards one of the 20 solutions, although there is no guarantee that this solution is effectively better than the initial estimate. Note also that it is unclear if adding calibration poses may always lead to a unique solution of the calibration problem.

10.3.7. OBSERVABILITY

Measuring all pose parameters at the calibration poses may be difficult and tedious. It has been suggested to measure only the location of the platform and not its orientation. Unfortunately, in that case, not all parameters may be identified (572)* with the constraint equations. For example, in Masory's reduced model, the observation matrix has rank 39, i.e. 3 parameters cannot be identified (308). Daney (121) suggests another method, the elimination of the orientation parameters in the constraint equations, so that the constraint equations are orientation free, and only the location of the platform has to be measured. Simulation results seem to show that the method is promising. Alternatively, the orientation may be measured only at some calibration poses; it has been shown in fact that measuring the orientation at one calibration pose is sufficient (572)*.

Oliviers (455) obtains the following results if a linearized version of the constraint equations is used:

- if one of the coordinates of C is not measured while all the orientation parameters are measured, then a constant offset error will appear in the direction of the unmeasured coordinates
- if the location of C is not measured while all the orientation parameters are measured, then the linearized system becomes singular
- if one orientation angle is not measured, then a constant orientation error will occur

If only 2 orientation angles are measured (for example using 2 orthogonal inclinometers (37)) the rank of the observation matrix for Masory's reduced model (to which is added as unknown the angle between the inclinometers) is 36 i.e. 7 parameters cannot be identified (308).

10.4. Auto-calibration

There are two possible approaches to *auto-calibration*:

- the robot has more internal sensors than strictly necessary for control
- a redundant instrumented passive leg is added to the robot

Using redundant internal sensors is the method suggested by Nahvi (438) for a 4 d.o.f. robot; by Zhuang (661) which uses 6 additional sensors, one for each of the U joint of the legs for 6- UPS robots; and by Yang for a three-legged robot (633). As no independent reference frame is used, the base and mobile frame may be chosen in such way that A_1, B_1 are the origins of the frames, A_2, B_2 lie on the x axes of the frames and A_3, B_3 have a zero z coordinate. Hence Masory's reduced model has only 30 independent parameters, and Zhuang shows that the rank of its observation matrix is 30 i.e. maximal. If only one additional sensor is used, the rank of the observation matrix is still 30, but the condition number of the observation matrix is about 5 times the one with 6 additional sensors (308). It is sometimes mentioned that enabling only a relative calibration is a drawback of the auto-calibration approach, but in most application cases all motions are specified relative to the base frame. Furthermore we have seen in the chapter "Direct kinematics", that adding sensors allows us to simplify, and speed up the direct kinematics problem, and hence auto-calibration is very promising.

Chiu (96) calibrates a 4 d.o.f. robot that has the structure of a 6- UPS robot constrained by a seventh leg. This extensible leg is connected to the base by a U joint, is fixed on the platform, and its extension is measured by a linear sensor. However the method is restricted to specific architectures, and the seventh leg may drastically reduce the workspace of the robot by interfering with the other legs.

10.5. Calibration with mechanical constraints

In this method, mechanical constraints are imposed on the robot during the calibration process through a locking device. With these constraints, some geometrical parameters will remain constant during the calibration, and the constraint equations will describe the invariance of these parameters.

Khalil (307) suggests clamping one or more links of a Gough platform, so that its direction remains the same for a set of calibration poses. The constraint equations indicate that this direction remains constant for all poses; these equations are functions only of the geometrical parameters of the robot and the link lengths. This method has the advantage that there is no need to have extra sensors, but solving the system so obtained, especially taking into account the errors in measurements of the lengths of the links is a difficult problem (119; 120). For Masory's reduced model, if one U or S joint of a leg is locked, the rank of the observation matrix is 29 i.e. 13 parameters are not observable. If during the calibration different joints are locked, the rank of the observation matrix becomes 30 (308).

Rauf (497) proposes using a locking device that allows only rotation of the platform around a fixed point (or alternatively around a point that can also translate along a given direction). Constraint equations may be obtained by writing that the coordinates of the rotation center are fixed, but this implies the use of the direct kinematics. Alternatively, the inverse kinematics may be used: 3 legs are used actively to apply a given orientation to the platform, the 3 other legs are passive and provide the measurements. Passive and active legs must be swapped during the process. Rauf notes that the best calibration results are obtained with large ranges of orientation angles. Observability of this method has still to be assessed.

This method is less expensive than external calibration, but more difficult to use than auto-calibration. As it does not allow one to use the full workspace of the robot, it may be thought that it will be less accurate than external and auto calibration. Its main drawback is that the actuators should be able to move passively according to the load to which they are submitted.

10.6. Determination of the calibration poses

For external and constrained calibrations it is necessary to choose the calibration poses. In many papers, random poses are selected, but it was observed that the quality of the calibration varied significantly according to the set of selected poses. Determining the best calibration poses was addressed by Nahvi (439) and Lintott (368) (for the *Delta* robot), who decided to find the poses of the end-effector so that a conditioning index for the observation matrix $J_{\mathcal{P}}$ is maximized. Various indices are defined, based on the

singular values of the observation matrix. According to Daney (126), the best index is O_1 , the square root of the determinant of $J_p^T J_p$, although the ratio between the square of the minimal singular value and the largest one (the *noise amplification index* (439)) may be of interest. The maximization of the index leads to poses which are on the edge of the articular workspace, and near singular configurations (368), which is a problem. Convergence toward singularity has to be expected; singular poses have maximal sensitivity of the platform motion to changes in the leg lengths.

Daney (122) proposes a method that may fail to find the calibration poses that maximize the conditioning index, but ensures that the calibration poses satisfy constraints, such as being in the workspace or not being singular. He notices that all calculated calibration poses are close to the articular workspace boundary (as intuitively suggested in (395)). In a following paper (126) he proposes the use of heuristics that may avoid the problem of local maxima of the conditioning index, and that mixes 2 different indices. He still observes that the calibration poses lie on the articular workspace, and that they are radially symmetric, as was the calibrated robot. These works were validated through experimental data.

10.7. Exercises

Exercise 10.1: Suggest a calibration method for the robot *Hexa*, based on Geng's method. What is the minimum number of measurements needed to calibrate one chain?

Problem 10.1: Given the measurements accuracy, is it possible to determine the poses of a 6-*UPS* robot so that the best calibration results are obtained?

Problem 10.2: Define quality indices for calibration methods that include measurement noises, and may be used either in simulation or with experimental data

Problem 10.3: Determine general rules to identify the elements of the geometry of a 6-*UPS* robot that play the most influential role on the positioning errors of the robot

Problem 10.4: Given a calibration method, an associated observation matrix, and bounds on the measurement errors, find the calibration poses that minimize the errors on the kinematic parameters, while ensuring that the calibration poses are not singular, and lie in the workspace of the robot

Problem 10.5: Assuming perfect joint and pose measurements, determine the minimal number of calibration poses for a given parallel robot so that the constraint equations will lead to a unique solution of the calibration problem

Design

This chapter will deal with the problem of designing parallel robots. We first have to choose the mechanical architecture, a problem that has been treated in the Chapter *Synthesis and Architectures*. Then we must determine the dimensions of the robot so that it complies as closely as possible with the performance needed for the task at hand. Indeed a direct result of the previous chapters is that the performance of a parallel robot is so dependent on its dimensions that a customization of the robot is absolutely necessary. We will describe the main approaches for solving this problem, even if only partially.

11.1. Introduction

The importance of design may be illustrated by a quote from Paul Sheldon, the designer of the Variax machine-tool: "*The Variax, which is now over 10 years old, still stands as an existence proof of PKM potential. For instance, it is 3 to 6 times stiffer than a typical good conventional machining center. But the many PKMs erroneously conceived and poorly executed since then have proven inferior to conventional approaches and have deterred the advance of the art...This sort of thing certainly does not instill confidence in the minds of potential customers, or encourage researchers to explore the technology.*" Fortunately parallel robots have shown their utility in various applications as we have already mentioned, and numerous examples show that a careful design optimization lead to large improvements over the initial design (42).

In order to design a mechanism, we first need to choose a mechanical architecture; this may be obtained either by a synthesis starting from the constraints on the task (mainly the number of d.o.f. required by the task), or by using an *a priori* solution. Secondly, the chosen mechanical architecture must be modeled, and the model then used to make a geometric synthesis, i.e. to determine the physical and geometrical characteristics of the mechanism that are the most appropriate for the task.

A first approach to geometric synthesis is trial and error, which consists in manually modifying the geometry of the mechanism, then evaluating the performance of the new mechanism after each modification, with

the help of a simulation software, until a mechanism is obtained that is deemed satisfactory. This approach relies on an efficient simulation system. Usual mechanical simulation software are not really efficient for the analysis of closed-loop mechanisms, and a general simulation system for parallel robot is difficult to design because of the diversity of the possible mechanical architectures. There are only a few generic simulation systems that have been proposed: kinematic simulation based on bond-graphs (651), or the *Map* dynamic simulator (299) but new projects are going on (104; 167) such as the *Synthetica* software (557). Some believe that the task of developing an efficient generic simulation software is so huge (and urgent) that it will require a collaborative world-wide effort. Even for a specific architecture there are very few simulation systems¹: one for planar robots (524), one for 3 d.o.f. spherical robots (*Smaps* (196)) and the RT4PM software for various manipulators (34). An on-line web service that allows workspace, error and conditioning analysis is also available[□] *AWE*. In any case, the number of parameters necessary to define the geometry of a parallel robot makes it difficult to use trial and error. Indeed we have already seen in the previous chapters that geometric parameters such as joint layout, link lengths, passive joint motion abilities, etc ... have a very important influence on the performance of the resulting mechanism. The number of design parameters does not allow to use a brute force approach where each design parameter is sampled, and the robot performances tested for each combination.

Instead of using the trial and error approach, we may try to identify the influence of the design parameters on the performances. For example Ji (288) proposed an approach for studying the influence of the layout of the joint centers of a 6-*UPS* robot on the workspace. But this influence on one performance is already complex, and if more than one performance is considered, the coupling effect becomes difficult to manage.

Hence a more systematic design methodology has to be used. A general approach is called *optimal design*: a procedure is used to determine the geometry of the mechanism such that the performances requirements are satisfied at best. This is a complex issue for the following reasons;

- *the number of design parameters is large*: we have already seen in the "Calibration" chapter that for a 6-*UPS* we need 36 kinematic parameters to define the geometry of the robot. We will need 12 more to define the minimum and maximum leg lengths just to perform workspace, accuracy, static and velocity analysis.
- *performance analysis is already complex*: we have already seen that performance analysis of a given robot is a complex problem for which

¹A software for visualizing various parallel planar and spatial robots is available via anonymous ftp, directory `coprin/Visu`

there are only partial solutions

The starting point of an optimal design approach is a list of specifications that indicates which performances are required for the robot according to the task (see an example for a machine-tool in (614)). In practice there are always multiple performance requirements to be satisfied. They can be categorized as follows:

- *imperative*: these performance requirements must be satisfied for any design solution
- *optimal*: a performance index is associated to the requirement and a maximal value of this index is required
- *primary*: although these performance requirements are specified in the specifications, their values may be modified to some extent if no design solution is found
- *secondary*: these requirements may not appear in the specifications list, but may be used to choose between design solutions that satisfy imperative, optimal and primary requirements

Usually we will attach some indices to each requirement that will allow us to determine to what extent the requirement is satisfied or violated. We will assume that these indices measure how much the requirements is violated, and increase with the violation of the requirement. With these indices we will see that a pair of performance requirements may also be classified according to the sign of their rates of change (RC) when the design parameters are modified:

- *antagonistic*: the sign of the RC for all performances indices for one requirement is always the opposite of the sign of the RC for all performance indices of the other requirement
- *related*: the RC between the two requirements are related, but the product of their sign may change according to the modified design parameters and/or their values
- *independent*: the RC of the performance indices are not related

The concept of "optimal design" applies only if there is at least one optimal performance requirements in the list of specifications. If there is one optimal requirement there will usually be one design solution. For more than one optimal requirements there usually will be no solution except if the optimal requirements are independent.

In our experience there is usually no optimal requirement for practical applications, as the list of specifications includes only imperative and primary requirements. In that case we have an *appropriate design* problem, and there may be multiple design solutions.

11.2. Reducing the number of design parameters

Clearly the first task of a designer is to try to reduce the number of design parameters. For example, it will be quite difficult to manage the 132 parameters identified by Masory (386) that define the basic geometry of an 6-*UPS* robot, and are not even sufficient to manage a workspace requirements (the minimal and maximal leg lengths should be added).

There are no simple guidelines for reducing the number of design parameters. If we consider a 6-*UPS* robot, and if the task is symmetrical, we may assume that the attachments points A_i, B_i are coplanar and lie on circles of radius R_1, r_1 , and that they are symmetrically disposed, with a separation angle α for 2 adjacent attachment points A_i on the base, and β for 2 adjacent attachment points B_i on the platform (figure 11.1). Hence we end up with 4 design parameters for the geometry of the robot. We may assume that the minimal and maximal leg lengths are identical for all legs; this will add 2 more design parameters. Note that this simplified layout

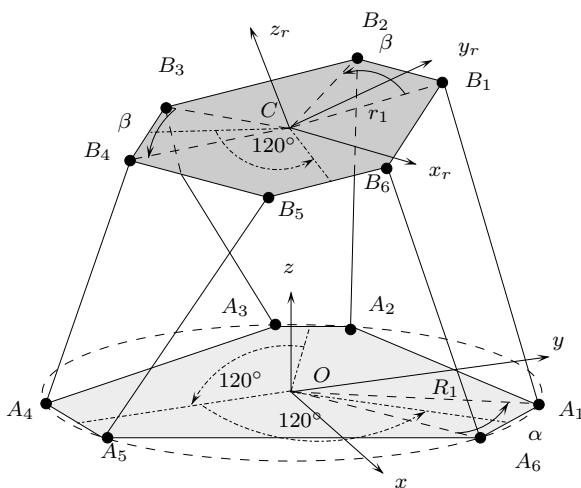


Figure 11.1. Simplified modeling of a 6-*UPS* robot

assumes planar base and platform, although non planar layout may be of interest, as mentioned by Bryfogle (63).

Note that this reduced model still allows us to deal with other design parameters that have an influence on specific performances. For example, it is good policy to offset the axis of the U, S joints to limit the influence of their mechanical limits on the workspace. It is generally considered that a good compromise for the directions of these joints corresponding to a zero angular motion are the directions of the legs when the actuators are

at mid-stroke. Note that it has never been proved however that this is the optimal solution.

We will now investigate how design solution(s) may be determined.

11.3. The atlas approach

The idea of the atlas approach is first to reduce the number of design parameters to 2 or 3, so that performance indices may be graphically represented as atlases. The designer then uses these atlases to choose the design parameters.

An example of such an approach is proposed by Bhattacharya (38) for a 6-*UPS* robot. He uses a reduced set of 3 design parameters: r_1/R_1 , ρ_{max}/R_1 , and β (the angle α is assumed to be fixed). The underlying assumption when normalizing by R_1 is that the performance indices depend only on the normalized parameters, an assumption that has to be verified (see exercise 11.1). Bhattacharya then calculates the average value of stiffness related indices over the workspace by using a discretisation method, and draws curves that show the value of the various criteria as functions of the design parameters; this allows him to choose the best geometry for the robot. The atlas approach has also been used by Clavel (100) for the dimensioning of the *Delta* robot. For the 3-*UPU* robot Badescu plots the workspace volume, the average of the inverse of the condition number and the *GCI* (21); Hong (238) defines global torque, force and velocity manipulability measures, and plots them as function of 2 design parameters. Liu (373)* plots the distribution of the shape of the workspace of planar and spatial robot in the design parameter space, while Williams (621) and Ceccarelli (77) propose atlases that illustrate the influence of the design parameters on the workspace size. Masuda (387) plots various manipulability measures as functions of the design parameters in order to choose the best design of a 6-*PUS* robot.

Clearly the atlas approach is very limited, and may be used only for a very small set of design parameters.

11.4. The cost function approach

In the literature, the most used design methodology for parallel robot is the *cost function approach*. This classical approach in mechanism theory (156) may be summarized as follows:

1. combine the m performance requirements indices $\mathcal{I}_1, \dots, \mathcal{I}_m$ in a weighted sum $\mathcal{C} = \sum_{j=1}^{j=m} w_j \mathcal{I}_j$ called the *cost function*
2. find the set of design parameters \mathcal{P} that minimizes the cost function by using a numerical procedure

A synthesis of the optimization methods that are used in mechanism theory may be found in (156). This approach is a particular instance of a classical design methodology that relies on the optimization of a function to determine the design solutions, but it does not have the property of other functions, whose purpose is to decrease the sensitivity of the design to variations in the design parameters (which is the purpose of other design methods such as Taguchi's).

This approach presents numerous disadvantages²:

- if only one criterion is used in the cost-function, the results can be misleading: thus Gosselin showed that the SSM with maximal workspace volume for a given stroke of the actuator has similar base and moving platform (see figure 7.17) and is therefore singular. Singularity consideration is an example of a *hidden criterion* that does not necessarily appear in the specifications list provided by the user, although the designer must always consider it.
- the approach is able to obtain the Pareto set³ only when the set is convex
- determining numerical weights to describe the designer preferences is difficult and tedious, and the results strongly depend on the weights
- the numerical optimization procedure will make an extensive use of the evaluation of the cost function. Hence the calculation of the performance indices must be very fast. But we have seen in the previous chapters that for most indices the *exact* calculation of the indices was expensive. Two approaches are used to deal with this problem:
 - performances indices are computed using a discretisation method: this induces discontinuity in the calculation of the indices, and this may be a problem for the optimization procedure. Furthermore, there is no guarantee on the optimality of the result
 - performances indices are computed only at poses that are considered significant with respect to the performance. This is probably more effective than the previous approach, but a careful check of the design solution has to be made
- the approach does not ensure that imperative requirements are satisfied. A classical way to manage this problem is to impose imperative requirements as penalty constraints to the optimization problem, but this complicates the procedure

²for an extensive study of these drawbacks see Das, I. and Dennis, J.E., "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problem", Structural Optimization, Vol 14, 1997, pp 63-69

³a point x^0 is called a Pareto solution if there is no other feasible point x such that $f_i(x) \leq f_i(x^0), i = 1, \dots, m$ with strict inequality for at least one i , see (92)

- it is necessary to constrain the value of the design parameters so that unrealistic values are not obtained. This complicates the optimization problem
- usually only one design solution is obtained (or a limited number if the optimization procedure is run with different initial guesses). There is no guarantee that the actual physical robot based on the design solution, but differing from it because of manufacturing tolerances, will still be optimal or satisfy the specifications. It is therefore necessary to investigate the effects of small changes on the design solution, a process often referred to as *post optimality analysis*. Furthermore, we believe that not all design constraints may be mastered by the designer (for instance economic constraints, such as privileged relation with a supplier, may influence the choice of the hardware)

In spite of all these drawbacks, the cost function approach is the method that is favored in the robotics literature. We will mention here a few characteristic examples; other examples may be found in the references Web page. Numerous papers consider only one performance index, the condition number of the inverse jacobian matrix (or the global conditioning index, *GCI*, i.e. the average value of the condition number over the workspace, see section 5.4.2.4) (196)*, (356; 380; 479; 552), sometimes with the additional constraint that a prescribed workspace should be obtained (516; 545). Related to this approach is the search for an *isotropic* robot (i.e. a robot for which there is at least one pose for which the condition number of the inverse jacobian matrix is 1). For example Zanganeh (649) determined the constraint on the layout of the joint centers of a SSM so that it becomes isotropic in a given pose: he then shows that, in general, it is not possible to design an isotropic SSM. Various authors have addressed this topic (11; 27; 73; 160; 584; 619). In our experience we have never encountered an application in which only one performance index has to be considered. Furthermore, we have seen in the "Velocity" chapter that the validity of the condition number to qualify the accuracy of the robot was doubtful, and that the *GCI* was difficult to calculate. As for the isotropy, considering that a robot is optimal because it is isotropic in one pose sounds strange (what will happen in the other poses of the workspace?).

Other one dimensional cost functions have been considered: Kim optimizes the average stiffness of a 3 d.o.f. robot over its workspace (316)*, Chakarov (85) suggests an optimization approach to determine the necessary stiffness in the links for a specified stiffness at the end-effector level for different poses, while Carretero (72) minimizes the unwanted d.o.f. of a 3 d.o.f. robot. Salcudean (517) has studied the influence of the radii of the platforms of a SSM on the joint forces necessary to obtain a given acceleration of the platform. Stoughton (555) considers the overall shape of the

workspace of a parallel micro-robot and tries to get it as close as possible to a sphere. Han (213) suggests determining the dimensions of the 4-bar mechanisms that actuate his robot so that the workspace is maximal, and ensure the lowest mass of the moving element.

There are other examples for which the cost function has only one term, but the optimization is performed under the constraint that include other performances: Ottaviano optimizes a criterion related to the overall size of the robot, while constraining the orientation workspace to be close to a prescribed one (457); Lou's (379)* objective is to have a given workspace included in the robot workspace, while the condition number in the prescribed workspace is low. This objective is converted into an optimization problem with LMI constraints. Han (213)* minimizes the weight of the legs of his micro-robot subject to constraints on their displacement, axial and buckling stresses.

The cost function may also have different terms related to the same performance criterion, such as mean value and standard deviation of the element of the stiffness matrix over the workspace (654)*.

Cost functions with mixed terms have also been considered. Khatib (310) introduces a cost function to quantify the dynamic performances of his 3-d.o.f. robot *Artisan* (figure 2.17). This function possesses several terms that evaluate the inertial behavior and the isotropy of the acceleration performances. He uses a numerical procedure to find the parameters that optimize the average of the cost function, as calculated over a finite number of poses. Stock (553) uses a cost function mixing manipulability and workspace requirement for a *Delta* robot. Stamper (550) uses both the workspace volume and a global conditioning index to optimize the design of a 3-d.o.f. translatory robot. Arsenault (16) combines a workspace requirement, the *GCI* and a binary index that indicates the presence of singularity in the workspace. Miller (420) proposes, for a 3 d.o.f. translational robot, a cost function that combines the *GCI*, and the ratio (workspace volume)/(volume of the robot's bounding box).

One way to avoid the problems related to mixing terms in the cost function is to consider only one performance index which already includes other primary requirements. For example Monsarrat (428) proposes finding the design parameters of a 6 d.o.f. robot that maximizes the volume of its singularity-free 3D constant orientation workspace.

As the optimization problem is often quite difficult, the use of genetic algorithms is sometimes mentioned, either to minimize the cost function (54), (558)*, (654) or to generate new design solutions that are then checked with respect to simplified design requirements (357).

Note that other cost-functions may be defined. In the *Compromise Programming* approach (91) the *utopia* point is defined as that with coordi-

nates are the minimal values of all the performance indices, assuming that a lower index provides a better design. The cost function is then defined as a weighted distance between the utopia point and the performance indices. In the *Physical Programming* approach (92) there is no need to define weights. Classes of constraints on the performance index are defined, and for each class a degree of desirability, from highly desirable to unacceptable, is defined for different ranges for the index value. A cost function taking into account the desirability may then be defined, imperative requirements being considered as constraints for the optimization. Although these approaches will most probably give better results than the weighted sum cost function, they share its drawbacks: it is difficult to define the index, it is difficult to calculate it, it is difficult to manage the various design requirements, and the method may not be robust with respect to tolerances.

11.5. The exact synthesis approach

It may happen that, after reduction of the number of parameters, all appropriate design solutions may be found analytically; we will call this case the *exact synthesis case*. For example, Chablat (81) is able to determine the dimensioning of the 3 d.o.f. translational Orthoglide (figure 2.8) so that its workspace includes a prescribed workspace, and such that the eigenvalues of $J_k^{-T} J_k^{-1}$, where J_k^{-1} is the inverse kinematic jacobian, lie within a given range for all poses in the prescribed workspace. Huang (242) was able to determine analytically the actuator stroke of a 6-*PUS* robot so that its workspace includes a prescribed translational workspace with a minimum reachable yaw angle. Exact synthesis may also be obtained if it is assumed that the specifications are to be satisfied only at a limited number of poses. For example, Simaan (541) determines what should be the design parameters to obtain a given stiffness matrix at a given pose, while Jafari (280) proposes a method for designing a 6-*UPS* robot so that $J_{fk}^{-T} J_{fk}^{-1}$ is diagonal at a given pose, the purpose being to obtain given maximal translational and angular velocities at this pose.

We will now describe more precisely an exact synthesis method for important requirements: workspace and velocity.

11.5.1. WORKSPACE SYNTHESIS

Workspace requirement is always present in a specification list, and is often seen as an imperative requirement, or a least a primary one. In recent years, many papers have addressed the issue of designing a robot for a given workspace. The proposed approaches may be classified as follows:

- *precision poses approach*: a set of poses is specified, and the design parameters are determined so that the workspace of the robot includes these poses. This approach has been used for 3-*RPR* planar robots (434; 536), 3 d.o.f. spherical robots (343), for the *Delta* robot and a spherical wrist (331)*, or with a general purpose application in mind (470). In some relatively simple cases it is possible to determine analytically all the values of the design parameters such that the precision poses are included in the corresponding workspace. In that case, another primary requirement index, as the *GCI* (331), is considered, and an optimization procedure is used to find its best value over the region determined in the workspace step. However, in most cases, only a limited number of precision poses may be specified, and finding the design parameters is a difficult task (315). McCarthy and co-workers have developed a general strategy for that task, and included it in the simulation software *Synthetica* (557), while emphasizing the specificities of mechanism design within the more general framework of optimal design (392)
- *trajectory approach*: a set of trajectories that must be included in the workspace is specified, and the design algorithm must find the corresponding design parameters. In (406) we presented such an algorithm that was designed for 6-*UPS* robots, but may be extended to other types of parallel robots. The design parameters are only the radii R_1, r_1 of the base and of the platform, so that we can graphically represent the result in the R_1-r_1 plane. It is assumed that the orientation of the end-effector is constant for each trajectory, but may change between 2 trajectories. The constraints are that the leg lengths should lie within a given range $[\rho_{min}, \rho_{max}]$. The main theoretical results are the following:
 - for a given trajectory the leg lengths will be less than or equal to ρ_{max} if the design parameters R_1, r_1 belong to the intersection of 2 ellipses $E(0), E(1)$ in the R_1-r_1 plane, that can be computed from the start and goal point of the trajectory
 - for a given trajectory, the leg lengths will be greater than or equal to ρ_{min} if the design parameters R_1, r_1 do not belong to any member of a one-dimensional set of ellipsis $E_s(\lambda)$, where λ is the parameter of the set and has a value in $[0,1]$

In (406) we show how to compute the domains $S_1 = E(0) \cap E(1)$, $S_2 = \cup E_s(\lambda) \forall \lambda \in [0, 1]$, and $S_3 = S_1 - S_2$. The domain S_3 then represents all possible values of the design parameters: a robot designed with R_1, r_1 in S_3 will have a workspace that includes the trajectory. Figure 11.2 shows an example of such a calculation, obtained in about 50 ms on a DELL D400, 1.2 Ghz. For a set of trajectories, the intersection of the S_3 leads to the domain in the R_1, r_1 plane that defines **all**

possible values of R_1, r_1 for which the corresponding robot will have a workspace that includes all the desired trajectories. This algorithm may be extended to include the mechanical limits of the joints in A, B , if we adopt the constraint model as described in the "Workspace" chapter. This algorithm may be extended to deal with other types of parallel robots.

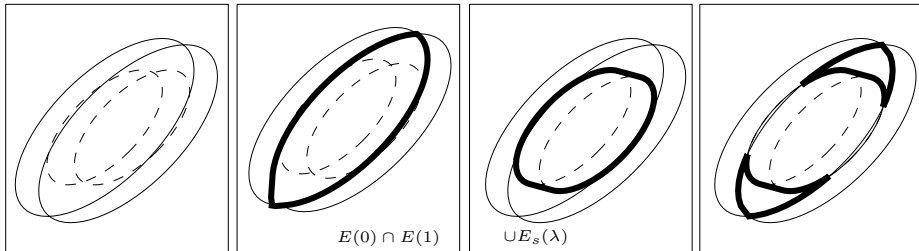


Figure 11.2. All the drawings are shown in the R_1, r_1 plane. For a given trajectory, that must be included in the workspace, we are able to calculate 2 ellipses $E(0), E(1)$ such that if (R_1, r_1) belongs to $E(0) \cap E(1)$, then for all poses on the trajectory $\rho \leq \rho_{max}$. We are also able to calculate a one-dimensional set of ellipses $E_s(\lambda)$ such that if (R_1, r_1) does not belong to $\cup E_s(\lambda)$, then for all poses on the trajectory $\rho \geq \rho_{min}$. On the left, the thin lines represent the ellipses $E(0), E(1)$, and the dotted lines show two of the ellipses $E_s(\lambda)$. We calculate the intersection S_1 of $E(0), E(1)$, then the union S_2 of the ellipses $E_s(\lambda)$ (in bold). Lastly, we subtract the union S_2 of the E_s from S_1 : the resulting domain gives all robots whose workspace will include the trajectory

11.5.2. VELOCITY SYNTHESIS

Constraints on the end-effector velocity may also be part of the specifications list. In (405) we have addressed the following problem for a 6- UPS robot whose design parameters are R_1, r_1 : given a bounds on the joint velocities determine the region \mathcal{Z} of the plane R_1, r_1 such that the center C of the moving platform may reach a fixed velocity in a given direction, whatever is its location on a set of trajectories. We showed that various conics in the R_1, r_1 plane where playing a role by splitting the plane into various regions, and that \mathcal{Z} was obtained by performing union and intersection operations on these regions. Here again the algorithm provides **all** possible values of R_1, r_1 for which the specification will be satisfied. Figure 11.3 presents one example with the conics that play a role, and the resulting \mathcal{Z} . The computation time is about 250 ms on a DELL D400, 1.2 Ghz.

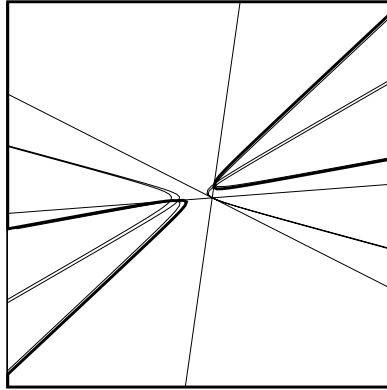


Figure 11.3. In the R_1, r_1 plane we may calculate a region \mathcal{Z} (in bold) such that the corresponding robots will be able to reach a given end-effector velocity for bounded joint velocities, whatever the pose on a given trajectory. This calculation involves union and intersection of regions that are bounded by conics, some of them being shown in thin line on the figure.

11.6. The parameter space approach

As seen in the introduction in most practical design problems we have to find *appropriate* design solutions and not *optimal* ones. The objectives of the parameter space approach are as follows:

- to propose not one design solution but a set of design solutions
- to guarantee that all design solutions satisfy the imperative requirements
- to guarantee that all design solutions are robust with respect to manufacturing tolerances i.e. the design parameter values of the real robot may differ from the theoretical design solutions by bounded tolerances, but the real robot will still satisfy the imperative requirements
- to offer various design solutions with different compromises on the secondary requirements

11.6.1. THE PARAMETER SPACE

The key-point of this approach is the concept of the *parameter space*. Each dimension of this space represents a design parameter, and consequently a point in this space represents a unique geometry for the robot. Searching for an optimal or appropriate robot therefore consists in finding the location of the points in the parameter space, such that the specifications list is best satisfied. Although in theory the parameter space is unbounded, practical considerations on each design parameter generally restricts its value

to some range. Hence the search for an optimal robot will have to be done only within a bounded domain of the parameter space, which we will call the *search space*. In the previous section we presented an example of a parameter space, the R_1, r_1 plane, for the 6-*UPS* robot. In this section we will present examples that show that we can deal with a larger number of design parameters.

11.6.2. PRINCIPLE OF THE METHOD

In the previous section we have seen that it was possible in some cases to design algorithms that calculate **all** the possible values of the design parameters, so that the corresponding robots satisfy, at least partially, one specific requirements. In terms of parameter space, this calculation amounts to determining a domain of the parameter space which includes all geometries satisfying a given requirement: such domain will be called the *allowed region* for the requirement. The design of algorithms for calculating the allowed regions is central in the parameter space approach.

The steps of the method are as follows:

1. define the parameter space
2. compute the allowed region for each requirement in the specification list
3. compute the intersection of all allowed regions: any robot geometry represented by a point in this intersection satisfies all specifications
4. determine a set of appropriate robots by sampling the intersection so that all various compromises for the secondary requirements are presented in the set, and verify that they satisfy all the primary requirements

All these steps of this theoretical method will be detailed in the following sections, and mostly illustrated on 6-*UPS* and 6-*PUS* robots.

11.6.3. CALCULATION OF THE ALLOWED REGIONS

As mentioned previously, the calculation of the allowed regions is the central point of the method. Ideally an allowed region algorithm should be able to solve the following problem: find **all** possible design parameters \mathcal{P} in the search domain such that some given relations $\mathbf{F}(\mathcal{P}, \mathbf{X})$ are satisfied for all poses in a given workspace \mathcal{W} . The relation \mathbf{F} , called the *requirement constraint*, will involve all the performance indices that have been mentioned in the previous chapters. As we have already seen the evaluation of these performance indices for a given robot, introducing additional unknowns with the design parameters will complicate the task. On the other hand practi-

cal and theoretical considerations will play an important role in getting a tractable problem:

- \mathcal{H}_1 , *completeness of the result*: it is not necessary to determine the allowed region *exactly*. Indeed, as the design parameters will be submitted to tolerances, point on the border of the allowed region cannot be chosen as nominal value for the design parameters as the physical instance may be *outside* of the allowed region
- \mathcal{H}_2 , *completeness of the verification of the requirement constraint*: for a requirement constraint involving the verification of an inequality $\mathbf{F}(\mathcal{P}, \mathbf{X}) \leq 0$ it is not necessary to calculate the values of \mathbf{F} *exactly* but just to ensure that its maximal value is indeed negative. In previous chapters we have already presented algorithms that are able to provide approximations to the maximal values for \mathbf{F} with arbitrary accuracy. These algorithms have a computation time that changes according to the accuracy, and may thus be very fast in a verification framework
- \mathcal{H}_3 , *relaxation of the workspace constraints*: for simplifying the calculation of the allowed region it is possible to assume that the workspace is reduced to a set of characteristic elements such as poses, or segments between two poses. This impose only an additional verification step, after completing the design process, in which the design solution performances are checked with respect to the specification list over the whole workspace. We will see that if an algorithm for the calculation of the allowed region for a relaxed version of the workspace has been designed for a specific requirement, then it is possible in general to produce a variant of this algorithm for the verification of the requirement over the whole workspace for a set of design solutions
- \mathcal{H}_4 , *distributed implementation*: design is in general computer intensive. But computer science now offers powerful tools for the distribution of heavy calculations over a network of computers. Hence design algorithms that allow for distributed calculation should be favored

Another necessary feature of the calculation is that the description of the resulting allowed region should be convenient for later intersection with other allowed regions.

Clearly, we should seek as much as possible analytic descriptions of the allowed regions (for example as presented in section 11.5.1) but such descriptions are difficult to obtain for relatively large number of design parameters or for complicated specification.

In our opinion, interval analysis is once again an appropriate tool for computing the allowed region, provided the number of parameters is relatively small, as it allows us to design a generic algorithm for computing allowed regions. For that we use the algorithms \mathcal{C} based on interval analy-

sis presented in the previous chapters that, given ranges for the unknowns, determine 3 different states for a property in \mathbf{F} : verified, violated or cannot be asserted, i.e. the overestimation of interval analysis does not allow to determine if the property is satisfied or violated.

The principle of the generic algorithm is quite simple and uses a branch-and-bound principle: a *box* is a set of ranges for the design parameters, the algorithm processes a list \mathcal{L} of boxes indexed by the integer i . At the start, the algorithm \mathcal{L} has only one box, the search domain for $i = 1$. A box will be valid only if the width of the range for design parameter s_j is larger than ϵ_j , $j \in [1, m]$.

1. verify if the properties \mathbf{F} are satisfied by the box i of \mathcal{L} , using any checking algorithm \mathcal{C}
 - if yes, store the box as an allowed region
 - if no, $i = i + 1$ and return to 1
2. if one of the properties in \mathbf{F} cannot be asserted, check the width of each range in the box
 - if all widths are lower than ϵ_j , then $i = i + 1$ and go to step 1
 - otherwise select the design parameter that has the largest range in the box, split the box into 2 boxes according to this parameter and store them in \mathcal{L} , then $i = i + 1$ and return to 1

The algorithm will stop when i is larger than the number of boxes in \mathcal{L} , i.e. all boxes have been processed.

Such an algorithm will in general satisfy property \mathcal{H}_1 . Indeed the algorithm provides an approximation to the allowed region as a set of boxes, the boxes getting smaller near the boundary of the real allowed region. We will usually set ϵ_j to be twice the tolerance on the design parameter j . In the result we will get boxes with range $[a_j, b_j]$ for the design parameter j , and we may choose as nominal value for this parameter any value in $[a_j + \epsilon_j, b_j - \epsilon_j]$ so that we can ensure that the real value for this parameters is indeed included in $[a_j, b_j]$.

As for property \mathcal{H}_2 , all will depend on the checking algorithm \mathcal{C} . Some of the algorithms presented in the previous chapters are able to verify that a property satisfies a constraint without computing the exact value of this property.

For property \mathcal{H}_3 , we have not mentioned in the algorithm description what was the domain for \mathbf{X} . There are various possibilities:

- the domain for \mathbf{X} is \mathcal{W} : the checking algorithm is able to manage the whole workspace, and the result we will get is the allowed region for the current specification; the algorithm will be computer intensive
- the domain for \mathbf{X} is a subdomain of \mathcal{W} , which may be only a set of poses or a collection of small domains around specific poses: the real

allowed region for the specification is a subset of the region; it will be necessary to verify that the design solutions satisfy the specification over the whole workspace. The algorithm will be much more faster than in the previous case

An interesting point however is that the verification algorithm that is necessary in the second case may be directly deduced from the design algorithm. Indeed, for the design algorithm, we start with large ranges for the design parameters \mathcal{P} and small ranges for \mathbf{X} . On the other hand, for the verification algorithm we have small ranges for \mathcal{P} (the one resulting from the design algorithm) and large ranges for \mathbf{X} (to cover \mathcal{W}). Hence the verification algorithm is simply obtained from the design algorithm by exchanging the role of \mathcal{P} and \mathbf{X} : the boxes will be set of ranges for \mathbf{X} and the bisection process operates on the pose parameters. As for property \mathcal{H}_4 , it is an intrinsic feature of branch-and-bound algorithm.

There is also an additional advantage of the presented algorithm. The calculation of the intersection of the allowed regions for various specifications may be done easily, using two possible approaches:

- the result of the algorithms is a set of boxes and computing the intersection of two such sets is easy. But before calculating this intersection it may be necessary to regroup boxes into a larger boxes before calculating the intersection. Indeed otherwise the intersection calculation may lead to boxes whose widths will be smaller than ϵ_j and are therefore not robust with respect to tolerances
- alternatively we may use an incremental approach. Assume that it is necessary to calculate the allowed regions for a set of n specification $\{\mathcal{S}_1, \dots, \mathcal{S}_n\}$. A possibility for calculating the allowed region for specification \mathcal{S}_k , $k > 1$ is to initialize the list \mathcal{L} not with the search domain but with the list of boxes obtained when calculating the allowed region for the specification \mathcal{S}_{k-1} . Hence we start with the calculation of the allowed region for \mathcal{S}_1 with the full search domain. Then the result is used for the calculation for \mathcal{S}_2 : the resulting boxes will be the design parameters values such that both $\mathcal{S}_1, \mathcal{S}_2$ are verified. Consequently there is no need to compute the intersection.

We have implemented such algorithms for 6- \underline{UPS} , 6- \underline{PUS} robots with the following set of design parameters:

- 6- \underline{UPS} : R_1, r_1, α, β , the lowest leg length and the stroke (6 design parameters)
- 6- \underline{PUS} : R_1, r_1, α, β , the fixed length of the leg and the stroke (6 design parameters)

We have started by designing such algorithms for the workspace specification for 6- \underline{UPS} robots (414) and 6- \underline{PUS} robots (216).

More recently we have developed a generic algorithm that may deal with various requirements such as accuracy, velocity and statics. We assume that we have to deal with the following problem \mathcal{G} : being given an interval matrix \mathbf{A} and an interval vector \mathbf{b} we want to show that the components x_i of the solution \mathbf{x} of the linear interval system

$$\mathbf{Ax} = \mathbf{b} \quad (11.1)$$

verifies $|x_i| \leq \epsilon_i$, where ϵ_i is a given threshold, for any instance of the matrix \mathbf{A} in the interval set \mathbf{A} and of the vector in \mathbf{b} . To illustrate the interest of this problem for our design problem, consider the relation $\Delta\Theta = \mathbf{J}_{\text{fk}}^{-1}\Delta\mathbf{X}$, which characterizes the relation between the sensor errors and the positioning errors of the platform. The vector \mathbf{b} of our problem will be $\Delta\Theta$, which is clearly a known interval vector. The matrix \mathbf{A} will be $\mathbf{J}_{\text{fk}}^{-1}$ which is an interval matrix, if the design parameters are ranges. Hence if we are able to solve problem \mathcal{G} , we will be able to compute the allowed region for an accuracy specification: find the design parameters so that, for bounded sensor errors, the positioning accuracy of the robot is better than given thresholds at specific poses of the workspace.

Dealing with the linear interval system (11.1) is a classical problem in interval analysis, but our design problem has a special feature. The algorithms developed for interval analysis assume that all elements of \mathbf{A} are independent i.e. each element may have any value within its range. As shown for the accuracy problem this is not the case in our design problem: as the elements of $\mathbf{J}_{\text{fk}}^{-1}$ are not independent. Hence we have designed a linear interval system solving algorithm that takes into account the dependence between the elements of \mathbf{A} , and therefore improves the bounds on the x_i (216; 418). This algorithm allows one to compute the allowed region for all specifications that involve a linear system, such as accuracy, velocity and static analysis.

Still, not all specifications may be managed by our algorithms. For example, we are not yet able to design an algorithm that compute the allowed region for a stiffness specification. But we will see in the next section that this not a problem as long as we have a verification algorithm that allows us to check if a design solution (or more exactly a family of design solutions as tolerances will be taken into account) satisfies a specification.

11.6.4. SEARCH FOR APPROPRIATE ROBOTS

In the previous steps we have determined the intersection of the allowed regions, possibly for all or some specifications, and hence we have determined a domain \mathcal{D} that includes all appropriate robots. Indeed the real domain may be different because:

- some allowed regions may have been computed with relaxed versions of the constraints
- allowed region may have not been computed for some specifications

Anyhow, a designer cannot propose an infinite set of design solutions, and therefore it is necessary to select a set of design solutions. For that purpose the domain \mathcal{D} will be sampled at regular intervals, each node of the sampling representing an unique robot geometry.

The design solution obtained for a node will be checked with respect to the specifications. Indeed the node may have been obtained for a relaxed version of the constraints. Whenever possible this verification will be performed by assigning a range for the design parameters, whose width will be the corresponding tolerances; if a node is validated as design a solution then the real robot obtained for the node, with the stated tolerances, will also satisfy the specifications.

Secondary requirements are also calculated at each node. After verifying all nodes, we retain a primary set of design solutions, satisfying the specifications. Then, among this set we select as design solutions the one providing the most different compromises for the secondary requirements. For example, if the stiffnesses k_x, k_y are secondary requirements, we will select as design solutions the one with the largest k_x , the one with the largest k_y and the one having a mean value for k_x, k_y . Finding the nodes with these extremal compromises is yet an open problem, and the regular sampling of the \mathcal{D} may be not the best solution but there is no known algorithm with a better strategy.

11.6.5. DESIGN EXAMPLES

We have used the parameter space approach for numerous design studies:

- *fine positioning devices*: we have studied positioning devices for the European Synchrotron Radiation Facility (ESRF) based in Grenoble, having a load between 250 and 2500 kg, and an absolute accuracy better than $1\mu\text{m}$ in a relatively small workspace; 40 of such devices are currently in use^{□ ESRF}. Another device is currently under construction at the Institut Laue Langevin (ILL) for the SALSA project^{□ ILL}
- *machine-tool*: the CMW-300 milling machine has been designed using this approach^{□ CMW}
- *space telescope*: a space active deployable telescope
- *measurement device*: a wire robot for measuring the displacement of automotive components

11.6.6. ADVANTAGES AND DRAWBACK

The clear advantages of the parameter space approach is to provide robust solutions (a robot constructed with stated tolerances from the theoretical solution will satisfy the specifications) that satisfy the imperative and primary requirements. Furthermore, the approach provides an infinite set of design solutions: this gives flexibility to manage constraints that were not known at the start of the design process.

There are drawbacks: up to now to we can manage relatively few design parameters (but this is not really a problem for parallel robots); the calculation of the allowed regions is difficult and computer intensive. The approach has also difficulties in managing optimal requirements, but we believe that is not in general a problem for parallel robots: optimal solution for one design requirement will usually lead to other unacceptable performances. Another difficulty is to manage a failure in the design which is obtained when the intersection of the allowed regions is empty. The size of each allowed region may indicate which requirement is too strict, but we still do not know which requirement(s) must be relaxed, and by how much.

11.7. Other design issues

In the previous sections we have focused on design parameters that are related to the *geometry* of the robots. But there may be other types of design parameters:

- *dynamics*: surprisingly, although fast parallel robots have been developed, few works have addressed the design issue for the dynamics, apart from the work of Khatib (310) and Di Gregorio (136), both for 3 d.o.f. robots. Nagai (437) uses an optimization procedure to maximize the accelerations of a wire robot under velocity and accuracy constraints
- *thermal*: thermal effects may play a role in the positioning accuracy of fine positioning robots. If this problem is an issue, a possible correction method is based on the use of temperature sensors. Sellgren (530) addresses this issue for a *Tricept* robot with the following design problems:
 - find the location of the sensors that are suitable for controlling positioning errors caused by thermal effects. Sellgren shows that the temperature sensor should be mounted along the screw of the linear actuator, which is not easy
 - is it possible to compensate for leg length variations by combining the measurements of temperature sensor with the other sensors data: no final answer has been given to this question

- *reliability*: fault tolerant parallel manipulators have been addressed by Notash (451), in view of determining a layout of the joints and sensors that is the least sensitive to failure.
- *control*: in the *design for control* approach, the purpose is to determine the design of a system to simplify its control. This is an approach that may be used for parallel robots (458). For example, it may be thought that an appropriate design may help to simplify the dynamic modeling

11.8. Exercises

Exercise 11.1: Show under what assumption the diagonal elements of the stiffness matrix of a 6-*UPS* robot depend only of the ratio r_1/R_1

Exercise 11.2: Show how to calculate the union of the ellipses $E_s(\lambda)$ (see page 310) when λ lies in the interval $[0,1]$. Indication: consider the location of the points M defined by $\mathbf{MO} = R_1\mathbf{u} + r_1\mathbf{v}$.

Exercise 11.3: Show that the centers of the ellipses $E(\lambda)$ (see page 310) lie on a line as λ varies, and that the angle between their principal axes and the x axis is always $\pi/4$ (assuming that the vectors \mathbf{u}, \mathbf{v} are not identical).

Exercise 11.4: Show that the algorithm for the determination of the allowed region based on the workspace constraints described at page 310 may be extended to 3-*RPR*, 3-*RRR* planar robots

Problem 11.1: Design a general parallel robot simulator that takes as input, for example, a DH description of the legs, and automatically performs kinematic, velocity and static analyses (inverse and direct kinematics, velocities and statics)

Problem 11.2: Find the principal direction of the U, S joints of a 6-*UPS* robot so that the extremal joint motions over a given workspace are minimal

Problem 11.3: For the parameter space approach, determine a sampling algorithm for the intersection of the allowed regions that will determine the robots presenting the most various values for the secondary requirements

Problem 11.4: Determine, for a given robot, which performance requirements may be independently optimized

Appendix: system solution

We give a brief, intuitive summary of classical system solution methods, most of which can be used only for *algebraic systems*, i.e. for polynomial equations. Fortunately, in robotics, many of the problems (but not all of them) may be formulated in terms of polynomial equations, sometimes at the expense of a higher complexity. The purpose of these methods is to determine *all* solutions of the system.

12.1. Homotopy

This method, also called *continuation*, is based on the following steps: suppose S is the given system of algebraic equations which are to be solved; we take a new system S_1 , "similar" in size to S , with known solutions. The coefficients of S_1 are slightly modified by successive steps, so that they tend towards those of S . At each step, the solutions for the modified system are obtained from the previous solutions by an iterative numerical method, usually a Newton-Raphson scheme. Different solution branches are therefore followed, some of which will lead to infinite solutions and are discarded. One remarkable characteristic of algebraic systems is that the number of branches leading to infinite solutions is in general constant if the coefficients are sufficiently general.

For the direct kinematics, it is therefore sufficient to take randomly selected geometries in order to determine the total number of complex and real solutions. Raghavan (492) thus randomly chooses 11 geometries of a 6-*UPS* robot to show that the direct kinematics problem admits up to 40 complex solutions.

The main drawback of homotopy is that to get all solutions of the system, S_1 should have at least the same number of possible complex solutions as the final system, which is usually very large. Consequently, the number of followed branches will be large (960 in the algorithm of Raghavan); this will have a strong influence on the computation time. Mu (432) reduces this number by computing the 40 real and complex solutions of a special 6-*UPS* robot whose solution may be easily found, and then incrementally modify the geometry of this robot toward the current geometry. Only 40 branches have to be followed, the computation time is still large. A nice

overview of continuation methods may be found in (546).

12.2. Elimination

Consider an algebraic system of n equations with n unknowns x_1, \dots, x_n ; each equation of the system is the sum of *monomials* (e.g. $x_2^3 x_3, x_4 x_2, 1$). The first step of elimination is to *hide* an unknown (e.g. x_1) and to consider that the system has only the unknowns x_2, \dots, x_n . Each monomial of the system is considered as a new unknown y_i , so that the system of n equations has now as unknowns the $m > n$ monomials y_i . Multiplying the initial set of equations by a monomial M leads to a system of $2n$ equations in m_1 unknown monomials. The multiplication by M of the monomials in the initial set of equations may introduce new monomials (and hence $m_1 > m$), but some of these products may already be present in the initial set (and hence $m_1 < 2m$). The multiplication by a monomial is repeated for the $2n$ equations, and for the resulting systems, until we get a square system of K equations in K unknown monomials y_i . This system may be written as a linear system:

$$A(x_1)\mathbf{y} = \mathbf{0},$$

where \mathbf{y} is a vector constituted of all monomials including the constant monomial 1 (hence \mathbf{y} cannot be the zero vector). This system admits a solution if and only if $|A(x_1)| = 0$, this equation being an univariate polynomial in x_1 . After solving this polynomial it is possible to backtrack i.e. to determine all the x_i unknowns for each root in x_1 .

The drawbacks of the elimination approach are:

- for systems of 2 equations in 2 unknowns, the matrix A is obtained as the Sylvester matrix, and its determinant is the *resultant* of the system. For more equations there are various ways to produce the matrix A , that will lead to different polynomials with different degrees.
- the univariate polynomial may lead to *spurious* roots, i.e. values of x_1 that do not lead to a solution of the system
- it is usually difficult to calculate the determinant of the matrix A in analytical form (a rule of thumb is that the analytical form cannot be obtained as soon as the size of A is larger than 5). Even if getting the analytical form is possible, round-off errors may affect the coefficients of the polynomial. Instead of using floating points it is better to convert any numerical value in A to rational. But the rationals that will be present in the determinant may have very large integers as numerator and denominator and this will require using a software allowing multiple precision
- to avoid this drawback it has been proposed to compute numerically the determinant for random values of x_1 . For each of these values, the

determinant is a linear function of the coefficients of the polynomial. Hence for a polynomial of degree m it is sufficient to compute the determinant at $m + 1$ random values for x_1 to get a system of $m + 1$ linear equations in the $m + 1$ coefficients of the polynomial, which can be solved numerically to get the coefficients. Unfortunately it can be shown that the linear system is extremely ill-conditioned and consequently that this approach is very sensitive to round-off errors, and is thus far from being robust numerically. Wampler (603) proposes another approach that amounts solving a generalized eigenvalue problem, for which robust algorithms exist.

A nice overview of resultant, discriminant and elimination methods may be found in (178).

12.3. Gröbner basis

The Gröbner basis gives us a method for writing a system of algebraic equations $\mathbf{f}(x_1, \dots, x_n) = \mathbf{0}$ in the unknowns x_1, \dots, x_n with finitely many solutions into a system that has the same roots and which is in *triangular form* $g_n(x_n) = 0, g_{n-1}(x_{n-1}, x_n) = 0, \dots, g_1(x_1, \dots, x_n) = 0$, called a Gröbner basis. The principle may be understood on a simple example. Consider 2 intersecting circles whose centers lie on the x axis. The intersection of these circles can be computed by solving a system of 2 second order equations in the coordinates of the intersection points. But there are many other algebraic varieties that have the same intersection points as the two circles, among which there is a vertical line $x - b = 0$, this later equation being the $g_n(x_n) = 0$ of the Gröbner basis. Calculation of a Gröbner basis is not easy (see a nice introduction to this theory in (112)) but Faugère has supplied the very efficient package *FGB* for that purpose, and Rouillier uses it for providing the fastest known algorithm for solving the direct kinematics of 6-*UPS* robots (514).

The first drawback of the Gröbner basis is that the calculation time of the Gröbner basis is heavily dependent upon the size of the system (i.e. the number of equations and their degree). Its second drawback is that its calculation with real numbers is numerically unstable: to avoid this problem, the initial floating point coefficients of the initial system should be converted to rationals so that the Gröbner calculations are done over the integers. But the coefficients of the Gröbner basis may then become huge and the solution will require appropriate software to be manageable.

Appendix: interval analysis

Interval analysis is a mathematical tool that is appropriate for solving many problems related to robotics. Many of the results presented in this book have been obtained through this method, convenient for optimization, uncertainties in the modeling, system solving, ... We present a short introduction to this tool, outlining general algorithm principles: more details may be found in (214; 281; 430; 445) and in the ALIAS home page

www.inria-sop.fr/coprin/logiciel/ALIAS/ALIAS.html.

13.1. Introduction

The *interval* $X = [\underline{x}, \bar{x}]$ is defined as the set of real numbers y such that $\underline{x} \leq y \leq \bar{x}$. The *width* of an interval is defined as $\bar{x} - \underline{x}$, and the mid-point of the interval is $(\bar{x} + \underline{x})/2$.

An *interval vector* \mathbf{X} is a set of intervals; it is also called a *box*. The mid-point of a box is the vector whose components are those of the mid-point of its interval. Let $f(x = \{x_1, x_2, \dots, x_n\})$ be a function in n unknowns, and a box $\mathcal{B} = \{X_1, \dots, X_n\}$ with n intervals, one for each unknown: an *interval evaluation* $F(\mathcal{B})$ of f for the box is an interval $[\underline{F}, \bar{F}]$ such that $\underline{F} \leq f(x_1, \dots, x_n) \leq \bar{F}$ for any value of the x_i such that $x_i \in X_i$. In other words, \underline{F}, \bar{F} are lower and upper bounds for the value of $f(x)$ for any values of x in the box.

There are many ways to implement an interval evaluation of a function but the simplest one is the *natural evaluation*, in which each mathematical operator is substituted by an interval equivalent. For example let f be $f(x) = x^2 - 2x + 1$ with x in $[4, 5]$. We have

$$f([4, 5]) = [4, 5]^2 - 2[4, 5] + 1 = [16, 25] - [8, 10] + [1, 1] = [7, 18]$$

Note that the bounds are not exact: the upper (lower) bound may be larger (lower) than the real maximum (minimum) of the function. But this over-estimation will decrease with the width of the parameter interval, and there are cases and methods that allow one to get exact bounds.

Interval analysis may be used for almost all classical mathematical functions, and are not restricted to specific structures such as algebraic functions.

An important point is that such interval evaluation can be implemented on a computer in a guaranteed manner i.e. in a way that takes into account numerical round-off errors (for example we are currently using the C++ BIAS/Profil package¹ that is quite effective). It must be known that numerical errors occur much more frequently than may be thought. Consider for example the following function, proposed by Rump:

$$f(x, y) = 333.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y}$$

that has to be evaluated at $x = 77617, y = 33096$. Various tools may be used to perform this calculation: table 13.1 gives the result. We no-

Matlab	Scilab	C (double)	Maple (10 Digits)	Maple (20 digits)
-1.1806×10^{21}	-1.1806×10^{21}	1.1726039	0.1×10^{28}	-1×10^{17}

TABLE 13.1. Evaluated value of f at $x = 77617, y = 33096$

tice immediately a large discrepancy in both the sign and value of the evaluation. If we use an interval evaluation we get as interval evaluation $[-0.56610^{23}, 0.55510^{23}]$ while the exact value is about $-0.8273960599 \dots$

13.2. Function properties and interval evaluation

The following property of f can be deduced from its interval evaluations $[\underline{F}, \overline{F}]$ for a box \mathcal{B} :

1. if $0 \notin [\underline{F}, \overline{F}]$: there are no values of the unknowns in \mathcal{B} that may make $f = 0$
2. if $\overline{F} < 0$ ($\underline{F} > 0$), then for any values of the unknowns in \mathcal{B} we have $f < 0$ ($f > 0$)
3. let A be a known constant; if $\overline{F} < A$ ($\underline{F} > A$), then for any values of the unknowns in \mathcal{B} we have $f < A$ ($f > A$)

13.3. Generic interval-based algorithm

Most of the algorithms that are based on interval analysis have a generic structure that we will now describe. We assume that we are dealing with a set of n unknowns $\{x_1, \dots, x_n\}$ and that we are solving a problem for which the unknowns are constrained to lie within a box \mathcal{B}_1 . During the algorithm

¹<http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html>

we will process in sequence the m boxes of a list \mathcal{L} and the index i will be used to indicate the number of the box that is currently processed.

A fundamental step in the algorithm is the *bisection* step. At this stage we are dealing with a box $\mathcal{B}_d = \{\underline{x}_1^d, \overline{x}_1^d, \dots, \underline{x}_n^d, \overline{x}_n^d\}$. Using various heuristics we will choose one of the unknowns x_j and create two new ranges I_1, I_2 for this variable as $I_1 = [\underline{x}_j^d, (\overline{x}_j^d + \underline{x}_j^d)/2]$, $I_2 = [(\overline{x}_j^d + \underline{x}_j^d)/2, \overline{x}_j^d]$. We extract from \mathcal{B}_d two new boxes $\mathcal{B}_d^{I_1}, \mathcal{B}_d^{I_2}$ that have the same ranges for the unknowns, except for the variable x_j , which will have as range either I_1 or I_2 . These boxes will be stored in \mathcal{L} , and the number of boxes in this list will be updated.

A key point in the algorithm is the operator \mathcal{S} that takes as input a box and returns -1 if there is no solution of the problem in the box, 1 if the box is a solution, and 0 otherwise.

An optional element is the filter operator \mathcal{F} that takes as input a box and returns -1 if there is no solution in the box. It may alternatively return a box whose width is lower than the input box, after having determined that the removed part of the input box cannot contain a solution.

With these elements, a typical interval analysis based algorithm proceeds the following steps:

1. $i = 1$, $\mathcal{L} = \{\mathcal{B}_1\}$, $m = 1$
2. if $i > m$ then EXIT
3. if $\mathcal{F}(\mathcal{B}_i) = -1$ then $i = i + 1$, go to 2, else $\mathcal{B}_i = \mathcal{F}(\mathcal{B}_i)$
4. compute $k = \mathcal{S}(\mathcal{B}_i)$
 - (a) if $k = -1$ then $i = i + 1$, go to 2
 - (b) if $k = 1$ then \mathcal{B}_i is a solution, $i = i + 1$, go to 2
5. bisect \mathcal{B}_i , $\mathcal{L} = \mathcal{L} \cup \{\mathcal{B}_i^{I_1}, \mathcal{B}_i^{I_2}\}$, $m = m + 2$, $i = i + 1$, go to 2

Such an algorithm will always terminate, as the size of the box always decreases after a bisection. Provided that the boxes resulting from the bisection are put at the top of the list, there is usually no problem of memory storage. The worst case complexity is exponential, but quite often the experimental complexity is more tractable.

It must be noted that such an algorithm is appropriate for a distributed implementation (and this will be valid for most of the interval analysis based algorithms). Indeed the processing of a given box in the list does not, in general, depends on the other boxes of the list. Hence a master computer may manage the list, and send a box to a slave computer that will execute the algorithm with its own boxes list, but will perform only a few bisections. Then the slave will return the remaining boxes in its list to the master and will be ready to process another box. Such a scheme may be easily implemented with classical workstations, by using a message

passing mechanism such as PVM². The computation time decrease will be in general somewhat smaller than the number of slaves, due to the overhead of the data transmission between the master and the slaves. But in some cases the gain may be larger than the number of slaves: for example for global optimization (see next section) or when only one solution has to be found.

Note that we have assumed that the initial domain was a box \mathcal{B}_1 : this is not restrictive, and other bounded domain shapes may be treated as long it is possible to determine a bounding box of the domain, and there is a procedure to test whether a given box lies inside or outside the domain. This test will be used before step 3 of the generic algorithm. If the test indicates that the current box is outside the domain, then the box is discarded and the next box is processed. If the test is not able to determine if the box is fully inside or outside the domain then two strategies are possible:

- bisect the box
- proceed with the box. If $\mathcal{S}(\mathcal{B}_i)$ is equal to -1 the box will be eliminated but if it is equal to 1 the box will be bisected

13.4. General purpose applications

The generic algorithms allow us to solve several robotics problems.

13.4.1. SYSTEM SOLVING

Let $\mathbf{f}(x = \{x_1, x_2, \dots, x_n\}) = \mathbf{0}$ be a system of $m \geq n$ equations in n unknowns, and assume that we are looking for the solutions of this system in a given box \mathcal{B}_d . The operator \mathcal{S} is based on the interval evaluation $\mathbf{F}(\mathcal{B}_d)$ of the equations: if the interval evaluation of one equation does not include 0 the operator will return -1. In its simplest implementation the operator will return 1 (i.e. the current box is considered a solution of the system) if the width of the box is lower than a given, small, threshold, but more sophisticated implementation obtains all certified solutions.

Note that this algorithm may also be used when the system includes inequalities.

13.4.2. GLOBAL OPTIMIZATION

To solve a global optimization problem (may be constrained) i.e. to find the maximum of a given function f up to an arbitrary accuracy ϵ we will maintain an estimated value of the maximum f^M whose value is initialized by computing f for an arbitrary value of the unknowns.

²<http://www.netlib.org/pvm3/index.html>

The operator \mathcal{S} computes the interval evaluation of f for the current box and returns -1 if $\bar{f} \leq f^M + \epsilon$ and 0 otherwise. This operator also computes f for the mid-point of the box and eventually updates f^M .

13.4.3. LINEAR SYSTEM SOLVING

An *interval linear system* is defined as:

$$\mathbf{A}\mathbf{X} = \mathbf{b}$$

where \mathbf{A} is a square interval matrix (i.e. a matrix whose elements are intervals) and \mathbf{b} is an interval vector. Two problems arise for interval linear systems:

1. *outer approximation*: determine a box that includes all the solutions in \mathbf{x} obtained for all real instances of \mathbf{A}, \mathbf{b} (i.e. when the elements of \mathbf{A}, \mathbf{b} have a fixed value that belong to the ranges)
2. *inner approximation*: determine a box for \mathbf{x} such that for all elements \mathbf{x}_r of the box there is a matrix \mathbf{A} in \mathbf{A} and a vector \mathbf{b}_r in \mathbf{b} such that $\mathbf{A}\mathbf{x}_r = \mathbf{b}_r$

These problems are classical in interval analysis, especially the outer approximation, and may be solved using interval adaptation of classical methods in linear algebra (e.g. Gaussian elimination). These methods however usually provides an over estimate of the outer approximation.

13.5. Robotics applications

13.5.1. WORKSPACE CALCULATION

The workspace \mathcal{W} of a parallel robot is usually defined as the set of poses X such that a set of inequalities $F(X) \leq 0$ is satisfied. It is also usually possible to determine a bounding box \mathcal{B}_d that includes \mathcal{W} . The generic algorithm may be adapted to compute an approximation to \mathcal{W} as a set of boxes for the components of X (see (411) for more details). Here the operator \mathcal{S} computes the interval evaluations of all the inequalities in F and returns -1 if the lower bound of one of these interval evaluations is positive, and 1 if the upper bound of all the evaluations is negative. In the later case, the whole box is included in the workspace. The operator will also return -1 if the box width is lower than a given threshold ϵ (in that case the box is called a *neglected box*)

The approximation to \mathcal{W} is constituted by the boxes in the file with a total volume of V_a . When neglecting a box, we may update the total volume V_n of the neglected boxes, and the volume of \mathcal{W} is less than or equal to $V_a + V_n$. Hence the ratio V_n/V_a is a good index for measuring the quality of the approximation.

Note that the quality of the approximation may be improved by an incremental process. We start with a relatively large value for ϵ and as soon as a box is neglected it is written in a file. After completing the run, we decrease the value of ϵ , and start again using the neglected boxes as initial elements in the boxes list.

13.5.2. SINGULARITY DETECTION

In the same way, we may deal with the problem of finding if at least one singularity exists within a given connected workspace \mathcal{W} as long as we are able either to calculate an analytical form for the determinant of the inverse Jacobian matrix, or to evaluate it for given intervals for the pose parameters. We need also a test $\mathcal{T}(\mathcal{B}_i)$ to determine if the box \mathcal{B}_i is fully inside, fully outside or partly inside \mathcal{W} .

We start by calculating the numerical value of the determinant $D(X_i)$ of the inverse Jacobian matrix at an arbitrary pose X_i chosen in \mathcal{W} , and look at its sign. Without loss of generality we may assume that $D(X_i)$ is strictly positive. If we are able to determine another pose X_k in \mathcal{W} such that $D(X_k)$ is strictly negative, then any path connecting X_i and X_k must include a singular pose X_s for which $D(X_s)$ is 0, and hence \mathcal{W} includes a singularity.

To determine if a singularity exists in \mathcal{W} we calculate a bounding box \mathcal{B}_0 of \mathcal{W} and use the generic algorithm with the operator \mathcal{S} that returns -1 if the box is outside \mathcal{W} or if the lower bound of the interval evaluation of D for the box is positive. If the upper bound of this interval evaluation is negative, then we have determined that a singularity exists in \mathcal{W} . Refinement of this algorithm may be found in (413).

13.6. Conclusion

We have outlined general principles for interval-analysis based algorithms that have been proved to be effective for robotics problems. But interval analysis should not be considered as a "black box", because it relies on a large combination of heuristics and numerical methods to be effective. Furthermore, the way the problem is formulated has to take into account that interval analysis will be used for the solution. For example, we may have many ways to formulate the problem that will result in systems that differ in their number of unknowns, and by the complexity of the equations. A classical practice is to try to obtain the system having the lowest possible number of unknowns but with interval analysis that may be not the best choice if the complexity of the final equations is such that their overestimation will always be large. The difficulty is then to find the right compromise between the number of unknowns and the complexity of the equations.

Conclusion

The characteristics of parallel robots are very different from those of serial manipulators, and involve problems which, although they may be similar to those encountered for serial structure, need a specific treatment. We saw that parallel robots are particularly appropriate for numerous applications. We believe that employing such mechanisms will become more and more common for many tasks:

- accurate manipulations, possibly implying heavy loads ;
- tasks implying significant velocities ;
- applications involving contacts between the robot and its environment;
- entertainment and machine-tools;
- micro-machines;

and this list is not exhaustive. Some of these markets are small, but involve high technology, and need a very careful design. Markets such as entertainment may be huge but require relatively low technology if the initial design is good. Machine-tool applications are promising, but not surprisingly, after only a few years of research we cannot expect them to have reached the same level of reliability as serial structures, which have been studied for many years.

It is not the cost of parallel robots that hinders their development; they are most often constituted of standard components. It is the complexity of their implantation, of their control and of their design that is a potential problem. Control is a key issue, especially for machine-tools, for which most controllers are basically the same as those used for classical linear machines, and therefore not appropriate for a non-linear machine. Our extensive simulations have shown that, for a well-designed parallel robot, 70% of the positioning errors may be attributed to the controller, and only 10% to the machine, the remaining errors coming from the CAD system. Hence a large effort should be devoted to developing a dedicated parallel robot controller that will make full use of the potentialities of the robot. Dynamic modeling should probably be included in the controller, whenever appropriate, although it is still difficult to figure out its benefit. The controller must also be able to deal with on-line calibration of the robot, and the robot

itself should be designed from the beginning with calibration in mind (e.g. it should include sensors in its passive joint).

CAD and simulation software must also be addressed, especially for the development of parallel robots having less than 6 d.o.f. Synthesizing machines with less than 6 d.o.f. is an exciting field of research, but we lack the simulation tools to perform a critical analysis of these new mechanisms, and to determine how useful they may be. Within the CAD system, a module for trajectory planning should be developed. But trajectory planning for closed-chains has not reached the same level as for serial chains; other related problems should be addressed, such as optimal part positioning or optimal use of eventually redundant d.o.f.

Much progress have been made in recently years on theoretical issues such as kinematics, singularity analysis and workspace calculation. There still remain many open problems, many of them having being listed in the previous chapters. Among the most important, we should mention the performance analysis issue (define appropriate performance characteristics of the robot and a means of calculating them exactly and robustly), and the optimal or appropriate design problem, for which we are lacking both theoretical background and software.

The study of parallel robots has become increasingly important during recent years, as shown both by the ever-increasing number of papers published on the subject, and by the various applications for which these mechanisms have been used. Some research has extended our knowledge about particular aspects of this type of mechanism, but numerous topics still remain open. The concept of a parallel robot has proved to be not entirely academic ; on the contrary, it is a system appropriate for the industrial world. We hope that this book will help in the understanding of the complex phenomena that are encountered in the design and use of parallel robots.

References

A bibliographic data base on parallel manipulators is available at:
http://www.inria.fr/coprin/equipe/merlet/merlet_eng.html

- [1] Adkins F.A. and Haug E.J. Operational envelope of a spatial Stewart platform. *ASME J. of Mechanical Design*, 119(2):330–332, June 1997.
- [2] Agrawal S.K. Workspace boundaries of in-parallel manipulator systems. *Int. J. of Robotics and Automation*, 7(2):94–99, 1992.
- [3] Ait-Ahmed M. *Contribution à la modélisation géométrique et dynamique des robots parallèles*. Ph.D. Thesis, Université Paul Sabatier, Toulouse, February, 2, 1993.
- [4] Albus J., Bostelman R., and Dagalakis N. The NIST ROBOCRANE. *J. of Robotic Systems*, 10(5):709–724, July 1993.
- [5] Alizade R.I. and Tagiyev N.R. A forward and reverse displacement analysis of a 6-dof in-parallel manipulator. *Mechanism and Machine Theory*, 29(1):115–124, January 1994.
- [6] Alizade R.I., Tagiyev N.R., and Duffy J. A forward and reverse displacement analysis of an in-parallel spherical manipulator. *Mechanism and Machine Theory*, 29(1):125–137, January 1994.
- [7] Alizade R.I. and Bayram C. Structural synthesis of parallel manipulators. *Mechanism and Machine Theory*, 39(8):857–870, August 2004.
- [8] Almonacid M. and others . Motion planning of climbing parallel robots. *IEEE Trans. on Robotics and Automation*, 19(3):485–489, June 2003.
- [9] Amirat M.Y., Pontnau J., and Artigue F. A three-dimensional measurement system for robot manipulators. *J. of Intelligent and Robotic Systems*, 9(3):291–299, 1994.
- [10] Angeles J. and Gosselin C. Détermination du degré de liberté des chaînes cinématiques simples et complexes. In *7th World Congress on Theory of Machines and Mechanisms*, pages 199–202, Seville, September, 17–22, 1987.
- [11] Angeles J. The robust design of parallel manipulators. In *1st Int. Colloquium, Collaborative Research Centre 562*, pages 9–30, Braunschweig, May, 29–30, 2002.
- [12] Angeles J. The qualitative synthesis of parallel manipulators. *ASME J. of Mechanical Design*, 126(4):617–624, July 2004.

- [13] Angeles J. Is there a characteristic length of a rigid-body displacement? In *Computational Kinematics*, Cassino, May, 4-6, 2005.
- [14] Arai T., Cleary K., and others . Design, analysis and construction of a prototype parallel link manipulator. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 205–212, Ibaraki, Japan, July, 3-6, 1990.
- [15] Arai T., Larssonneur R., and Jaya Y.M. Calibration and basic motion of a micro-hand module. In *Int. Conf. on Indus. Electronics, Control and Instrumentation (IECON)*, pages 1660–1665, Hawaii, November, 15-19, 1993.
- [16] Arsenault M. and Boudreau R. The synthesis of three-degree-of-freedom planar parallel mechanisms with revolute joints (3-RRR) for an optimal singularity-free workspace. *J. of Robotic Systems*, 21(5):259–274, 2004.
- [17] Artigue F., Amirat M.Y., and Pontnau J. Isoelastic behavior of parallel robots. *Robotica*, 7:323–325, 1989.
- [18] Arun V. and others . Determination of the workspace of the 3-dof double-octahedral variable-geometry-truss manipulator. In *22nd Biennial Mechanisms Conf.*, pages 493–500, Scottsdale, September, 13-16, 1992.
- [19] Asada H. and Granito C. Kinematic and static characterization of wrist joints and their optimal design. In *IEEE Int. Conf. on Robotics and Automation*, pages 244–250, St Louis, March, 25-28, 1985.
- [20] Austad A. Arm device, June, 4, 1987. IPN n° WO 87,03239.
- [21] Badescu M. and Mavroidis C. Workspace optimization of 3-legged UPU and UPS parallel platforms with joint constraints. *ASME J. of Mechanical Design*, 126(2):291–300, March 2004.
- [22] Bai S. and Teo M.Y. Kinematic calibration and pose measurement of a medical parallel manipulator by optical position sensors. *J. of Robotic Systems*, 20(4):201–209, 2003.
- [23] Baker J.E. An analysis of the Bricard linkages. *Mechanism and Machine Theory*, 15(4):267–286, 1980.
- [24] Bamberger H. and Shoham M. A new configuration of a six degrees-of-freedom parallel robot for mems fabrication. In *IEEE Int. Conf. on Robotics and Automation*, pages 4545–4550, New Orleans, April, 28-30, 2004.
- [25] Bande P. and others . Kinematics analyses of Dodekapod. *Mechanism and Machine Theory*, 40(6):740–756, June 2005.
- [26] Baron L. and Angeles J. The direct kinematics of parallel manipulators under joint-sensor redundancy. *IEEE Trans. on Robotics and Automation*, 16(1):12–19, February 2000.
- [27] Baron L., Wang X., and Cloutier G. The isotropic conditions of

- parallel manipulators of Delta topology. In *ARK*, pages 357–366, Caldes de Malavalla, June 29- July 2, 2002.
- [28] Baumann R., Maeder W., Glauser D., and Clavel R. The Pantoscope: a spherical remote-center-of-motion parallel manipulators for force reflection. In *IEEE Int. Conf. on Robotics and Automation*, pages 718–723, Albuquerque, April, 21-28, 1997.
- [29] Behi F. Kinematic analysis for a six-degree-of-freedom 3-PRPS parallel mechanism. *IEEE J. of Robotics and Automation*, 4(5):561–565, October 1988.
- [30] Behi F., Mehregany M., and Gabriel K.J. A microfabricated three-degree-of-freedom parallel mechanism. In *IEEE Micro Electro Mechanical Workshop*, pages 159–165, Napa Valley, February, 11-14, 1990.
- [31] Ben-Horin R. and Shoham M. Construction of a new type of a six-degrees-of-freedom parallel manipulator with three planarly actuated links. In *ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pages 96–DETC/MECH–1561, Irvine, August, 18-22, 1996.
- [32] Ben-Horin R., Shoham M., and Djerassi S. Kinematics, dynamics and construction of a planarly actuated parallel robot. *Robotics and Computer-Integrated Manufacturing*, 14(2):163–172, April 1998.
- [33] Ben-Horin R. and Shoham M. Singularity analysis of a class of parallel robots based on Grassmann-Cayley algebra. In *Computational Kinematics*, Cassino, May, 4-6, 2005.
- [34] Ben Sghaier A. and Romdhane L. A software package for parallel mechanisms modeling and simulation. In *Computational Kinematics*, Cassino, May, 4-6, 2005.
- [35] Berger K.T., Horta L.G., and Taleghani B.K. Static testing of an inflatable/rigidizable hexapod structure. In *45th AIAA Structures, Structural Dynamics and Material Conf.*, Palm-Spring, April, 19-22, 2004.
- [36] Bernier D., Castelain J-M., and Li X. A new parallel structure with six degree of freedom. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 8–12, Milan, August 30- September 2, 1995.
- [37] Besnard S. and Khalil W. Calibration of parallel robot using two inclinometers. In *IEEE Int. Conf. on Robotics and Automation*, pages 1758–1763, Detroit, May, 10-15, 1999.
- [38] Bhattacharya S., Hatwal H., and Ghosh A. On the optimum design of a Stewart platform type parallel manipulators. *Robotica*, 13(2):133–140, March - April, 1995.
- [39] Bhattacharya S., Hatwal H., and Ghosh A. An on-line estimation

- scheme for generalized Stewart platform type parallel manipulators. *Mechanism and Machine Theory*, 32(1):79–89, January 1997.
- [40] Bhattacharya S., Nenchev D.N., and Uchiyama M. A recursive formula for the inverse of the inertia matrix of a parallel manipulator. *Mechanism and Machine Theory*, 33(7):957–964, October 1998.
- [41] Bhattacharya S., Hatwal H., and Ghosh A. Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators. *Mechanism and Machine Theory*, 33(7):965–974, October 1998.
- [42] Bleicher F. Optimizing the three-axis machine-tool with parallel kinematic structure. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 883–894, Chemnitz, April, 23-25, 2002.
- [43] Bombin C., Ros L., and Thomas F. On the computation of the direct kinematics of parallel spherical mechanism using Bernstein polynomials. In *IEEE Int. Conf. on Robotics and Automation*, pages 3332–3337, Seoul, May, 21-26, 2001.
- [44] Bonev J., I.A. and Ryu. A new method for solving the direct kinematics of general 6-6 Stewart platforms using three linear extra sensors. *Mechanism and Machine Theory*, 35(3):423–436, March 2000.
- [45] Bonev I.A. and Ryu J. A geometrical method for computing the constant-orientation workspace of 6-Prss parallel manipulators. *Mechanism and Machine Theory*, 36(1):1–13, 2001.
- [46] Bonev I.A. and Ryu J. A new approach to orientation workspace analysis of 6 dof parallel manipulator. *Mechanism and Machine Theory*, 36(1):15–28, January 2001.
- [47] Bonev I.A. Delta parallel robot-the story of success. May, 6, 2001, <http://www.parallemic.org/Reviews/Review002.html>.
- [48] Bonev I.A. and Zlatanov D. The mystery of the singular SNU translational parallel robot. June, 12, 2001, www.parallemic.org/Reviews/Review004.html.
- [49] Bonev I.A. and others . A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors. *IEEE Trans. on Robotics and Automation*, 17(2):148–156, April 2001.
- [50] Bonev I.A. The true origins of parallel robots. January, 24, 2003, <http://www.parallemic.org/Reviews/Review007.html>.
- [51] Bonev I.A., Zlatanov D., and Gosselin C. Singularity analysis of 3 dof planar mechanisms via screw theory. *ASME J. of Mechanical Design*, 125(3):573–581, September 2003.
- [52] Borel E. Mémoire sur les déplacements à trajectoire sphériques. *Mémoire présentés par divers savants*, 33(1):1–128, 1908.
- [53] Bosscher P. and Ebert-Uphoff I. A novel mechanism for implementing

- multiple collocated spherical joints. In *IEEE Int. Conf. on Robotics and Automation*, pages 336–341, Taipei, September, 14–19, 2003.
- [54] Boudreau R. and Gosselin C.M. The synthesis of planar parallel manipulators with a genetic algorithm. *ASME J. of Mechanical Design*, 121(4):533–537, December 1999.
- [55] Brandt G. and others . A compact robot for image guided orthopedic surgery. In *First Joint Conf. of Computer Vision, Virtual Reality and Robotics (CRVMED)II and Medical Robotics and Computer Assisted Surgery (MRCAS)III*, Grenoble, March, 19–22, 1997.
- [56] Bricard R. Mémoire sur la théorie de l’octaèdre articulé. *Journal de Mathématiques pures et appliquées, Liouville*, tome 3:113–148, 1897.
- [57] Bricard R. Mémoire sur les déplacements à trajectoire sphériques. *Journal de l’École Polytechnique*, 11(2):1–96, 1906.
- [58] Brodsky V., Glozman D., and Shoham M. Double circular-triangular six-degree-of-freedom parallel robot. In *ARK*, pages 155–164, Strobl, June 29– July 4, 1998.
- [59] Brogardh T., Hanssen S., and Hovland G. Application-oriented development of parallel kinematic manipulators with large workspace. In *2nd Int. Colloquium, Collaborative Research Centre 562*, pages 153–170, Braunschweig, May, 10–11, 2005.
- [60] Bruyninckx H. The 321-hexa: a fully parallel manipulator with closed-form position and velocity kinematics. In *IEEE Int. Conf. on Robotics and Automation*, pages 2657–2662, Albuquerque, April, 21–28, 1997.
- [61] Bruyninckx H. Closed-form position kinematics of a $(3 - 1 - 1 - 1)^2$ fully parallel manipulator. *IEEE Trans. on Robotics and Automation*, 14(2):326–328, April 1998.
- [62] Bruyninckx H. and De Schutter J. Unified kinetostatics for serial, parallel and mobile robots. In *ARK*, pages 343–352, Strobl, June 29– July 4, 1998.
- [63] Bryfogle M.D., Nguyen C.C., Zhou Z-l., and Antrazi S.S. A methodology for geometry design of closed kinematic chain mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 2974–2979, Albuquerque, April, 21–28, 1997.
- [64] Bulca F., Angeles J., and Zsombor-Murray P.J. On the workspace determination of spherical serial and platform mechanisms. *Mechanism and Machine Theory*, 34(3):497–512, April 1999.
- [65] Burdet E., Honegger M., and Codourey A. Controllers with desired dynamic compensation and their implementation on a 6 dof parallel manipulator. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Takamatsu, October 30– November 5, 2000.
- [66] Byun Y.K. and Cho H-S. Analysis of a novel 6-dof,3-PPSP parallel manipulator. *Int. J. of Robotics Research*, 16(6):859–872, December

- 1997.
- [67] Callegari M. and Marzetti P. Kinematic characterization of the 3-PUU parallel robot. In *Proc. Intelligent Manipulation and Grasping, IMG'04*, pages 377–382, Genova, June 30–July 1, 2004.
 - [68] Canfield S.L. and Reinholtz C.F. Development of the carpal robotic wrist. In *ISER*, pages 360–371, Barcelone, June, 15-18, 1997.
 - [69] Cappel K.L. Motion simulator, January, 3, 1967. United States Patent n° 3,295,224 The Franklin Institute.
 - [70] Cappel K.L. Invention and development of the Synergistic motion system. January 1999,
<http://www.triz-journal.com/archives/1999/01/a>.
 - [71] Carretero J. A. and others . Kinematic analysis of a three-dof parallel mechanism for telescope applications. In *ASME Design Engineering Technical Conference*, pages DETC97/DAC–3981, Sacramento, September, 14-17, 1997.
 - [72] Carretero J. A. and others . Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. *ASME J. of Mechanical Design*, 122(1):17–24, March 2000.
 - [73] Carricato M. and Parenti-Castelli V. Singularity-free fully isotropic translational parallel mechanisms. *Int. J. of Robotics Research*, 21(2): 161–174, February 2002.
 - [74] Carricato M. and Parenti-Castelli V. A family of 3-DOF translational parallel manipulators. *ASME J. of Mechanical Design*, 125(2):302–307, June 2003.
 - [75] Cauchy A. Deuxième mémoire sur les polygones et les polyèdres. *Journal de l'École Polytechnique*, pages 87–98, May 1813.
 - [76] Ceccarelli M. A new 3 d.o.f. spatial parallel mechanism. *Mechanism and Machine Theory*, 32(8):896–902, 1997.
 - [77] Ceccarelli M. and Sorli M. The effects of design parameters on the workspace of a Turin parallel robot. *Int. J. of Robotics Research*, 17(8):886–902, August 1998.
 - [78] Ceccarelli M. and Carbone G. Numerical and experimental analysis of the stiffness performances of parallel manipulators. In *2nd Int. Colloquium, Collaborative Research Centre 562*, pages 21–36, Braunschweig, May, 10-11, 2005.
 - [79] Chablat D. and Wenger P. Moveability and collision analysis for fully-parallel manipulators. In *12th RoManSy*, pages 61–68, Paris, July, 6-9, 1998.
 - [80] Chablat D. *Domain d'unicité et parcourabilité pour les manipulateurs pleinement parallèles*. Ph.D. Thesis, Ecole Centrale, Nantes, November, 6, 1998.
 - [81] Chablat D. and Wenger P. Architecture optimization of a 3-dof

- translational parallel mechanism for machining applications, the Orthoglide. *IEEE Trans. on Robotics and Automation*, 19(3):403–410, June 2003.
- [82] Chablat D., Wenger P., Majou F., and Merlet J-P. An interval analysis based study for the design and the comparison of three-degrees-of-freedom parallel kinematic machine. *Int. J. of Robotics Research*, 23(6):615–624, 2004.
- [83] Chai K-S., Young K., and Tuersley I. A practical calibration process using partial information for a commercial Stewart platform. *Robotica*, 20(3):315–322, 2002.
- [84] Chakarov D. and Parushev P. Synthesis of parallel manipulator with linear drive modules. *Mechanism and Machine Theory*, 29(7):917–932, October 1994.
- [85] Chakarov D. Study of the passive compliance of parallel manipulators. *Mechanism and Machine Theory*, 34(3):373–389, April 1999.
- [86] Chang W-T., Lin C-C., and Lee J-J. Force transmissibility performance of parallel manipulators. *J. of Robotic Systems*, 20(11):659–670, 2003.
- [87] Charentus S. *Modélisation et commande d'un robot manipulateur redondant composé de plusieurs plate-formes*. Ph.D. Thesis, Université Paul Sabatier, Toulouse, April, 13, 1990.
- [88] Chen I-M. and others . The management of parallel-manipulator singularities using joint-coupling. In *IEEE Int. Conf. on Robotics and Automation*, pages 773–778, Taipei, September, 14-19, 2003.
- [89] Chen W-J. and others . A novel 4-dof parallel manipulator and its kinematic modelling. In *IEEE Int. Conf. on Robotics and Automation*, pages 3350–3355, Seoul, May, 23-25, 2001.
- [90] Chen Y., McInroy J.E., and Yi Y. Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance. *IEEE Trans. on Robotics and Automation*, 19(4):681–691, August 2003.
- [91] Chen W. and others . Quality utility- a compromise programming approach to robust design. *ASME J. of Mechanical Design*, 121(2):179–187, June 1999.
- [92] Chen W. and others . Exploration of the effectiveness of physical programming in robust design. *ASME J. of Mechanical Design*, 122(2):155–163, June 2000.
- [93] Chiacchio P., Pierrot F., Sciavicco L., and Siciliano B. Robust design of independent joint controllers with experimentation on a high-speed parallel robot. *IEEE Trans. on Industrial Electronics*, 40(4):393–403, August 1993.
- [94] Chirikjian G.S. A binary paradigm for robotic manipulators. In *IEEE*

- Int. Conf. on Robotics and Automation*, pages 3063–3070, San Diego, May, 8-13, 1994.
- [95] Chiu Y.J. and Perng M-H. Forward kinematics of a general fully parallel manipulator with auxiliary sensors. *Int. J. of Robotics Research*, 20(5):401–414, May 2001.
- [96] Chiu Y.J. and Perng M-H. Self-calibration of a general hexapod manipulator with enhanced precision in 5-dof motions. *Mechanism and Machine Theory*, 39(1):1–23, January 2004.
- [97] Choi K., Jiang S., and Li Z. Spatial stiffness realization with parallel springs using geometric parameters. *IEEE Trans. on Robotics and Automation*, 18(3):274–284, June 2002.
- [98] Chung G.B. and others . Design and analysis of a spatial 3-dof micromanipulator for tele-operation. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 337–342, Maui, Hawaii, October 29- November 3, 2001.
- [99] Clavel R. DELTA, a fast robot with parallel geometry. In *18th Int. Symp. on Industrial Robot*, pages 91–100, Lausanne, April, 26-28, 1988.
- [100] Clavel R. *Conception d'un robot parallèle rapide à 4 degrés de liberté*. Ph.D. Thesis, EPFL, Lausanne, 1991. n° 925.
- [101] Clavel R. and others . A new 5 dof parallel kinematics for production applications. In *Int. Symp. on Robotics*, Stockholm, October, 9-11, 2002.
- [102] Clavel R. and others . High precision parallel robots for micro-factory applications. In *2nd Int. Colloquium, Collaborative Research Centre 562*, pages 285–296, Braunschweig, May, 10-11, 2005.
- [103] Cleary K. and Brooks T. Kinematic analysis of a novel 6-dof parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 708–713, Atlanta, May, 2-6, 1993.
- [104] Cobet M. Designing PKMs: working volume, stiffness, frequencies. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 83–103, Chemnitz, April, 23-25, 2002.
- [105] Codourey A. and Burdet E. A body oriented method for finding a linear form of the dynamic equations of fully parallel robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 1612–1618, Albuquerque, April, 21-28, 1997.
- [106] Collins C.L. and Long G.L. On the duality of twist/wrench in serial and parallel chain robot manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 526–531, Nagoya, May, 25-27, 1995.
- [107] Collins C.L. and Long G.L. The singularity analysis of an in-parallel hand controller for force-reflected teleoperation. *IEEE Trans. on Robotics and Automation*, 11(5):661–669, October 1995.

- [108] Collins C.L. Forward kinematics of planar parallel manipulators in the Clifford algebra of P^2 . *Mechanism and Machine Theory*, 37(8):799–813, August 2002.
- [109] Comin F. Six degree-of-freedom scanning supports and manipulators based on parallel robots. *Review of Scientific Instruments*, 66(2):1665–1667, February 1995.
- [110] Company O., Marquet F., and Pierrot F. A new high speed 4-dof parallel robot. Synthesis and modeling issues. *IEEE Trans. on Robotics and Automation*, 19(3):411–420, June 2003.
- [111] Cortés J. and Siméon T. Probabilistic motion planning for parallel mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 4354–4359, Taipei, September, 14-19, 2003.
- [112] Cox D., Little J., and O’Shead D. *Ideals, Varieties, and Algorithms*. Springer, 1992.
- [113] Culpepper M.L. and Chen S-C. Design of precision manipulator using binary actuation and differential compliant mechanisms. In *ASPE 18th Annual Meeting*, Portland, October, 26-31, 2003.
- [114] Culpepper M.L. and Anderson G. Design of a low-cost nano-manipulator which utilizes a monolithic, spatial compliant mechanism. *Journal of Precision Engineering*, 28(4):469–482, October 2004.
- [115] Culpepper M.L., Kartik M.V., and DiBiasio C. Design of integrated mechanisms and exact constraint fixtures for micron-level repeatability and accuracy. *Journal of Precision Engineering*, 29(1):65–80, January 2005.
- [116] Dafaoui M., Amirat Y., Pontnau J., and Francois C. Analysis and design of a six-dof parallel manipulator, modeling, singular configurations and workspace. *IEEE Trans. on Robotics and Automation*, 14(1):78–92, February 1998.
- [117] Dandurand A. The rigidity of compound spatial grid. *Structural Topology* 10, pages 43–55, 1984.
- [118] Danescu G. *Une méthode algébrique de synthèse et conception de mécanismes articulés*. Ph.D. Thesis, Université de Franche-Comté, Besançon, June, 22, 1995.
- [119] Daney D. Mobility constraints on the legs of a parallel robot to improve the kinematic calibration. In *New machine concepts for handling and manufacturing devices on the basis of parallel structures*, pages 187–200, Braunschweig, November, 10-11, 1998.
- [120] Daney D. Self calibration of Gough platform using leg mobility constraints. In *10th World Congress on the Theory of Machines and Mechanisms*, pages 104–109, Oulu, June, 20-24, 1999.
- [121] Daney D. and Emiriz I.Z. Variable elimination for reliable parallel robot calibration. In F.C. Park C.C. Iurascu, editor, *Computational*

- Kinematics*, pages 133–144. EJCK, May, 20-22, 2001.
- [122] Daney D. Optimal measurement configurations for Gough platform calibration. In *IEEE Int. Conf. on Robotics and Automation*, pages 147–152, Washington, May, 11-15, 2002.
- [123] Daney D. Kinematic calibration of the Gough platform. *Robotica*, 21(6):677–690, December 2003.
- [124] Daney D., Papegay Y., and Neumaier A. Interval methods for certification of the kinematic calibration of parallel robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 1913–1918, New Orleans, April, 28-30, 2004.
- [125] Daney D., Andreff N., and Papegay Y. Interval method for calibration of parallel robots: a vision-based experimentation. In *Computational Kinematics*, Cassino, May, 4-6, 2005.
- [126] Daney D., Papegay Y., and Madeline B. Choosing measurement poses for robot calibration with the local convergence method and Tabu search. *Int. J. of Robotics Research*, 24(6):501–518, June 2005.
- [127] Daniali H.R.M., Zsombor-Murray P.J., and Angeles J. The kinematics of a 3 d.o.f. planar and spherical double-triangular parallel manipulators. In J. Angeles P. Kovacs, G. Hommel, editor, *Computational Kinematics*, pages 153–164. Kluwer, 1993.
- [128] Dasgupta B. and Mruthyunjaya T.S. Singularity-free path planning for the Stewart platform manipulator. *Mechanism and Machine Theory*, 33(6):711–725, August 1998.
- [129] Dasgupta B. and Choudhury P. A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 34(6):801–824, August 1999.
- [130] Dash A.K. and others. Workspace generation and planning singularity-free path for parallel manipulators. *Mechanism and Machine Theory*, 40(7):778–805, July 2005.
- [131] Dedieu J-P and Norton G.H. Stewart varieties: a direct algebraic method for Stewart platforms. *SigSam*, 24(4):42–59, October 1990.
- [132] Dietmaier P. The Stewart-Gough platform of general geometry can have 40 real postures. In *ARK*, pages 7–16, Strobl, June 29- July 4, 1998.
- [133] Di Gregorio R. A new parallel wrist using only revolute pairs: the 3-RUU wrist. *Robotica*, 19(3):305–309, May 2001.
- [134] Di Gregorio R. and Parenti-Castelli V. Kinematics of a six-dof fixation device for long-bone fracture reduction. *J. of Robotic Systems*, 18(12):715–722, 2001.
- [135] Di Gregorio R. and Parenti-Castelli V. Mobility analysis of the 3-UPU parallel mechanism assembled for a pure translational motion.

- ASME J. of Mechanical Design*, 124(2):259–264, June 2002.
- [136] Di Gregorio R. Dynamic performance indices for 3-dof parallel manipulators. In *ARK*, pages 11–20, Caldes de Malavalla, June 29- July 2, 2002.
- [137] Di Gregorio R. A new family of spherical parallel manipulators. *Robotica*, 20(4):353–358, July 2002.
- [138] Di Gregorio R. and Zanforlin R. Workspace analytic determination of two similar translational parallel manipulators. *Robotica*, 21(5):555–566, October 2003.
- [139] Di Gregorio R. Kinematics of the 3-RSR wrist. *IEEE Trans. on Robotics*, 20(4):750–754, August 2004.
- [140] Do W.Q.D. and Yang D.C.H. Inverse dynamic analysis and simulation of a platform type of robot. *J. of Robotic Systems*, 5(3):209–227, 1988.
- [141] Dohner J.L. Active chatter suppression in an octahedral hexapod milling machine. *Proc. of the SPIE*, 2721:316–325, 1996.
- [142] Downing D.M., Samuel A.E., and Hunt K.H. Identification of the special configurations of the octahedral manipulators using the pure condition. *Int. J. of Robotics Research*, 21(2):147–159, February 2002.
- [143] Dubowsky S. and others . The design and implementation of a laboratory test bed for space robotics: the VES mod. II. In *ASME Design Automation Conf.*, pages 99–108, Minneapolis, September, 11-14, 1994.
- [144] Duffy J. *Statics and Kinematics with Applications to Robotics*. Cambridge University Press, New-York, 1996.
- [145] Dunlop G.R. and Jones T.P. Gravity counter balancing of a parallel robot for antenna aiming. In *6th ISRAM*, pages 153–158, Montpellier, May, 28-30, 1996.
- [146] Dunlop G.R. and Jones T.P. Position analysis of a 3-dof parallel manipulator. *Mechanism and Machine Theory*, 32(8):903–920, November 1997.
- [147] Dunlop G.R. and Jones T.P. Position analysis of a two DOF parallel mechanism- the Canterbury tracker. *Mechanism and Machine Theory*, 34(4):599–614, May 1999.
- [148] Du Plessis L.J. and Snyman J.A. Design and optimum operation of a reconfigurable planar Gough-Stewart machining platform. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 729–749, Chemnitz, April, 23-25, 2002.
- [149] Dutré S., Bruyninckx H., and De Schutter J. The analytical jacobian and its derivative for a parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 2961–2966, Albuquerque, April,

- 21-28, 1997.
- [150] Earl C.F. and Rooney J. Some kinematics structures for robot manipulator designs. *J. of Mechanisms, Transmissions and Automation in Design*, 105(1):15–22, March 1983.
- [151] Ebert-Uphoff I. and Chirikjian G.S. Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities. In *IEEE Int. Conf. on Robotics and Automation*, pages 139–145, Minneapolis, April, 24-26, 1996.
- [152] Ebert-Uphoff I. and Gosselin C.M. Kinematic study of a new type of spatial parallel platform mechanism. In *ASME Design Engineering Technical Conferences*, Atlanta, September, 13-16, 1998.
- [153] Ebert-Uphoff I., Gosselin C.M., and Laliberté T. Static balancing of spatial parallel platform-revisited. *ASME J. of Mechanical Design*, 122(1):43–51, March 2000.
- [154] Ebert-Uphoff I., Lee J-K., and Lipkin H. Characteristic tetrahedron of wrench singularities for parallel manipulators with three legs. *Proc. Instn Mech Engrs, Part C: J. Mechanical Engineering Science*, 216(1):81–93, January 2002.
- [155] El-Khaswneh B. and Ferreira P.M. Computation of stiffness and stiffness bounds for parallel manipulators. *Int. J. of Machine Tools & Manufacture*, 39(2):321–342, February 1989.
- [156] Erdman A.G. *Modern Kinematics*. Wiley, New-York, 1993.
- [157] Everett L.J. Forward calibration of closed-loop jointed manipulators. *Int. J. of Robotics Research*, 8(4):85–91, August 1989.
- [158] Fang Y. and Tsai L-W. Structure synthesis of a class of 4-dof and 5-dof parallel manipulators with identical limb structures. *Int. J. of Robotics Research*, 21(9):799–810, September 2002.
- [159] Fang Y. and Tsai L-W. Structure synthesis of a class of 3-DOF rotational parallel manipulators. *IEEE Trans. on Robotics and Automation*, 20(1):117–121, February 2004.
- [160] Fattah A. and Hasan Ghasemi A.M. Isotropic design of spatial parallel manipulators. *Int. J. of Robotics Research*, 21(9):811–824, September 2002.
- [161] Faugère J.C. and Lazard D. The combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, 30(6):765–776, August 1995.
- [162] Faulring E.L., Colgate J.E., and Peshkin M.A. A high performance 6-dof haptic Cobot. In *IEEE Int. Conf. on Robotics and Automation*, pages 1980–1985, New Orleans, April, 28-30, 2004.
- [163] Fichter E.F. and McDowell E.D. A novel design for a robot arm. In *Proc. Int. Computer Technical Conf.*, pages 250–255, San Francisco, 1980.

- [164] Fichter E.F. A Stewart platform based manipulator: general theory and practical construction. *Int. J. of Robotics Research*, 5(2):157–181, Summer 1986.
- [165] Fioretti A. Implementation-oriented kinematics analysis of a 6 dof parallel robotic platform. In *4th IFAC Symp. on Robot Control, Syroco*, pages 43–50, Capri, September, 19-21, 1994.
- [166] Firmani F. and Podhorodeski R.P. Force unconstrained poses for a redundantly-actuated planar parallel manipulator. *Mechanism and Machine Theory*, 39(5):459–476, May 2004.
- [167] Franke H.J. and others . Knowledge based development environment. In *2nd Int. Colloquium, Collaborative Research Centre 562*, pages 221–236, Braunschweig, May, 10-11, 2005.
- [168] French C.W. and others . Multi-axial subassembly testing (Mast) system: description and capabilities. In *13th World Conf. on Earthquake Engineering*, page No. 2146, Vancouver, August, 1-6, 2004.
- [169] Freudenstein F. On the variety of motion generated by mechanisms. *Transaction of the ASME*, pages 156–160, February 1962.
- [170] Fried G. and others . A 3-D sensor for parallel robot calibration. A parameter perturbation analysis. In *ARK*, pages 451–460, Portoroz-Bernadin, June, 22-26, 1996.
- [171] Frisoli A. and others . Synthesis by screw algebra of translating in-parallel actuated mechanisms. In *ARK*, Piran, June, 25-29, 2000.
- [172] Fujimoto K. and others . Derivation and analysis of equations of motion for a 6 d.o.f. direct drive wrist joint. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 779–784, Osaka, November, 3-5, 1991.
- [173] Funabashi H. and Takeda Y. Determination of singular points and their vicinity in parallel manipulators based on the transmission index. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1977–1981, Milan, August 30- September 2, 1995.
- [174] Gallardo J. and others . Dynamics of parallel manipulators by means of screw theory. *Mechanism and Machine Theory*, 38(11):1113–1131, November 2003.
- [175] Gao F., Liu X-J., and Chen X. The relation ships between the shapes of the workspaces and the link lengths of 3-DOF symmetrical planar parallel manipulators. *Mechanism and Machine Theory*, 36(2):205–220, February 2001.
- [176] Gao F. and others . New kinematic structures for 2-,3-,4- and 5-dof parallel manipulator designs. *Mechanism and Machine Theory*, 37(11):1395–1411, November 2002.
- [177] Geike T. and McPhee J. Inverse dynamics analysis of parallel manipulators with full mobility. *Mechanism and Machine Theory*, 38(6):549–

- 562, June 2003.
- [178] Gelfand I.M., Kapranov M.M., and Zelevinsky A.V. *Discriminants, resultants, and multidimensional determinants*. Birkhäuser, 1994.
 - [179] Geng Z. and Haynes L.S. On the dynamic model and kinematic analysis of a class of Stewart platforms. *Robotics and Autonomous Systems*, 9(4):237–254, 1992.
 - [180] Geng Z. and Haynes L.S. Six-degree-of-freedom active vibration isolation using a Stewart platform mechanism. *J. of Robotic Systems*, 10(5):725–744, July 1993.
 - [181] Geng Z. and Haynes L.S. An effective kinematics calibration method for Stewart platform. In *ISRAM*, pages 87–92, Hawaiï, August, 15-17, 1994.
 - [182] Geng Z. and Haynes L.S. A 3-2-1 kinematic configuration of a Stewart platform and its application to six degree of freedom pose measurements. *Robotics & Computer-Integrated Manufacturing*, 11(1):23–34, March 1994.
 - [183] Ghorbel F., Chet el at O., and Longchamp R. A reduced model for constrained rigid bodies with application to parallel robots. In *4th IFAC Symp. on Robot Control, Syroco*, pages 57–62, Capri, September, 19-21, 1994.
 - [184] Girone M. and others . A Stewart platform-based system for ankle telerehabilitation. *Autonomous Robots*, 10(2):203–212, March 2001.
 - [185] Gogu G. Mobility criterion and overconstraints of parallel manipulators. In *Computational Kinematics*, Cassino, May, 4-6, 2005.
 - [186] Gosselin C. *Kinematic analysis optimization and programming of parallel robotic manipulators*. Ph.D. Thesis, McGill University, Montr eal, June, 15, 1988.
 - [187] Gosselin C. Stiffness mapping for parallel manipulators. *IEEE Trans. on Robotics and Automation*, 6(3):377–382, June 1990.
 - [188] Gosselin C. Determination of the workspace of 6-dof parallel manipulators. *ASME J. of Mechanical Design*, 112(3):331–336, September 1990.
 - [189] Gosselin C. and Angeles J. Singularity analysis of closed-loop kinematic chains. *IEEE Trans. on Robotics and Automation*, 6(3):281–290, June 1990.
 - [190] Gosselin C. and Angeles J. Kinematic inversion of parallel manipulators in the presence of incompletely specified tasks. *ASME J. of Mechanical Design*, 112(4):494–500, December 1990.
 - [191] Gosselin C. Dexterity indices for planar and spatial robotic manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 650–655, Cincinnati, May, 13-18, 1990.
 - [192] Gosselin C., Sefrioui J., and Richard M.J. Solution polynomiale au

- problème de la cinématique directe des manipulateurs parallèles plans à 3 degrés de liberté. *Mechanism and Machine Theory*, 27(2):107–119, March 1992.
- [193] Gosselin C., Lavoie E., and Toutant P. An efficient algorithm for the graphical representation of the three-dimensional workspace of parallel manipulators. In *22nd Biennial Mechanisms Conf.*, pages 323–328, Scottsdale, September, 13–16, 1992.
- [194] Gosselin C. and Merlet J-P. On the direct kinematics of planar parallel manipulators: special architectures and number of solutions. *Mechanism and Machine Theory*, 29(8):1083–1097, November 1994.
- [195] Gosselin C., Sefrioui J., and Richard M.J. On the direct kinematics of spherical three-degree-of-freedom parallel manipulators with a coplanar platform. *ASME J. of Mechanical Design*, 116(2):587–593, June 1994.
- [196] Gosselin C., Perreault L., and Vaillancourt C. Simulation and computer-aided kinematic design of three-degree-of-freedom spherical parallel manipulators. *J. of Robotic Systems*, 12(12):857–860, 1995.
- [197] Gosselin C., Lemieux S., and Merlet J-P. A new architecture of planar three-degree-of-freedom parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 3738–3743, Minneapolis, April, 24–26, 1996.
- [198] Gosselin C. Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators. *ASME J. of Dynamic Systems, Measurement and Control*, 118(1):22–28, March 1996.
- [199] Gosselin C. and Wang J. Singularity loci of planar parallel manipulators with revolute actuators. *Robotics and Autonomous Systems*, 21(4):377–398, October 1997.
- [200] Gosselin C. and St-Pierre E. Development and experimentation of a fast 3-dof orienting device. *Int. J. of Robotics Research*, 16(15):619–630, October 1997.
- [201] Gosselin C.M. and Wang J. Static balancing of spatial six-degree-of-freedom parallel mechanisms with revolute actuators. *J. of Robotic Systems*, 17(3):159–170, 2000.
- [202] Goswami A. and Peshkin M.A. Mechanically implementable accommodation matrices for passive force control. *Int. J. of Robotics Research*, 18(7), 1999.
- [203] Gough V.E. Contribution to discussion of papers on research in automobile stability, control and tyre performance, 1956–1957. Proc. Auto Div. Inst. Mech. Eng.
- [204] Gough V.E. and Whitehall S.G. Universal tire test machine. In *Proceedings 9th Int. Technical Congress F.I.S.I.T.A.*, volume 117, pages

- 117–135, May 1962.
- [205] Grace K.W. and others . A six degree of freedom micromanipulator for ophthalmic surgery. In *IEEE Int. Conf. on Robotics and Automation*, pages 630–635, Atlanta, May, 2-6, 1993.
- [206] Griffis M. and Duffy J. A forward displacement analysis of a class of Stewart platform. *J. of Robotic Systems*, 6(6):703–720, 1989.
- [207] Griffis M., Crane C., and Duffy J. A smart kinestatic interactive platform. In *ARK*, pages 459–464, Ljubljana, July, 4-6, 1994.
- [208] Guglielmetti P. *Model-Based control of fast parallel robots: a global approach in operational space*. Ph.D. Thesis, EPFL, Lausanne, March, 24, 1994.
- [209] Guozhen W. Forward displacement analysis of a class of the 6-6 Stewart platforms. In *22nd Biennial Mechanisms Conf.*, volume DE-45, pages 113–117, Scottsdale, September, 13-16, 1992.
- [210] Gwinnett J.E. Amusement device, January, 20, 1931. United States Patent n° 1,789,680.
- [211] Hafez M., Lichter M.D., and Dubowsky S. Optimized binary modular reconfigurable robotic devices. *IEEE Trans. on Mechatronics*, 8(1):152–162, March 2003.
- [212] Han C. and others . Kinematic sensitivity analysis of the 3-UPU parallel manipulator. *Mechanism and Machine Theory*, 37(8):787–798, August 2002.
- [213] Han C-S., Hudgens J.C., Tesar D., and Traver A.E. Modeling, synthesis, analysis and design of high resolution micromanipulator to enhance robot accuracy. In *IEEE Int. Conf. on Intelligent Robot and Systems (IROS)*, pages 1153–1162, Osaka, November, 3-5, 1991.
- [214] Hansen E. *Global optimization using interval analysis*. Marcel Dekker, 1992.
- [215] Hao F. and McCarthy J.M. Conditions for line-based singularities in spatial platform manipulators. *J. of Robotic Systems*, 15(1):43–55, 1998.
- [216] Hao F. and Merlet J-P. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and Machine Theory*, 40(2):151–171, February 2005.
- [217] Hara A. and Sugimoto K. Synthesis of parallel micromanipulators. *J. of Mechanisms, Transmissions and Automation in Design*, 111(1):34–39, March 1989.
- [218] Harib K. and Srinivasan K. Kinematic and dynamic analysis of Stewart platform-based machine tool structures. *Robotica*, 21(5):541–554, October 2003.
- [219] Harris D.M.J. Parallel-linkage robot coordinate transformation through screw theory. In *9th World Congress on the Theory of Machines and*

- Mechanisms*, pages 1565–1568, Milan, August 30- September 2, 1995.
- [220] Hashimoto M. and Imamura Y. Design and characteristics of a parallel link compliant wrist. In *IEEE Int. Conf. on Robotics and Automation*, pages 2457–2462, San Diego, May, 8-13, 1994.
- [221] Haugh E.J., Adkins F.A., and Luh C.M. Operational envelopes for working bodies of mechanisms and manipulators. *ASME J. of Mechanical Design*, 120(1):84–91, March 1998.
- [222] Hay A.M. and Snyman J.A. The optimal synthesis of parallel manipulators for desired workspace. In *ARK*, pages 337–346, Caldes de Malavalla, June 29- July 2, 2002.
- [223] Hayes M.J.D. and Zsombor-Murray P.J. Inverse kinematics of a planar manipulator with holonomic higher pairs. In *ARK*, pages 59–68, Strobl, June 29- July 4, 1998.
- [224] Heerah I. and others . Architecture selection and singularity analysis of a three-degree-of-freedom planar parallel manipulator. *J. of Robotic Systems*, 37(4):355–374, August 2003.
- [225] Hertz R.B. and Hughes P.C. Kinematic analysis of a general double-tripod parallel manipulator. *Mechanism and Machine Theory*, 33(6): 683–696, August 1998.
- [226] Hervé J.M. Analyse structurelle des mécanismes par groupe de déplacements. *Mechanism and Machine Theory*, 13(4):437–450, 1978.
- [227] Hervé J-M. and Sparacino F. Star, a new concept in robotics. In *ARK*, pages 176–183, Ferrare, September, 7-9, 1992.
- [228] Hervé J.M. Group mathematics and parallel link mechanisms. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 2079–2082, Milan, August 30- September 2, 1995.
- [229] Hervé J.M. and Karouia M. The novel 3-RUU wrist with no idle pair. In *Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Québec, October, 3-4, 2002.
- [230] Hervé J.M. Parallel mechanisms with pseudo-planar motion generators. In J. Lenarčič C. Galletti, editor, *ARK*, pages 431–440. Kluwer, 2004.
- [231] Hesselbach J. and Kerle H. Structurally adapted kinematic algorithms for parallel robots up to six degrees of freedom (dof). In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1935–1939, Milan, August 30- September 2, 1995.
- [232] Hesselbach J. and others . Connecting assembly modes for workspace enlargement. In *ARK*, pages 347–356, Caldes de Malavalla, June 29- July 2, 2002.
- [233] Hesselbach J. and others . A generic formulation of the dynamics of plane parallel robots for real-time applications. In *RAAD*, Cassino,

- May, 7-10, 2003.
- [234] Hesselbach J. and others . Parallel robot specific control functionalities. In *2nd Int. Colloquium, Collaborative Research Centre 562*, pages 93–108, Braunschweig, May, 10-11, 2005.
- [235] Higuchi T., Ming A., and Jiang-Yu J. Application of multi-dimensional wire crane in construction. In *5th Int. Symp. on Robotics in Construction*, pages 661–668, Tokyo, June, 6-8, 1988.
- [236] Hiller M. and others . Design, analysis and realization of tendon-based parallel manipulators. *Mechanism and Machine Theory*, 40(4):429–445, April 2005.
- [237] Honegger M., Codourey A., and Burdet E. Adaptive control of the Hexaglide, a 6 dof parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 543–548, Albuquerque, April, 21-28, 1997.
- [238] Hong K-S. Kinematic optimal design of a new parallel-type rolling mill: paramill. *Advanced Robotics*, 17(9):837–862, 2003.
- [239] Huang C., Hung W-H., and Kao I. New conservative stiffness mapping for the Stewart-Gough platform. In *IEEE Int. Conf. on Robotics and Automation*, pages 823–828, Washington, May, 11-15, 2002.
- [240] Huang J.M., S. anmd Schimmels. The bounds and realization of spatial stiffnesses achieved with simple springs connected in parallel. *IEEE Trans. on Robotics and Automation*, 14(3):466–474, June 1998.
- [241] Huang T. and others . Determination of closed form solution to the 2-D orientation workspace of Gough-Stewart parallel manipulators. *IEEE Trans. on Robotics and Automation*, 15(6):1121–1125, December 1999.
- [242] Huang T., Jiang B., and Whitehouse D.J. Determination of the carriage stroke of 6-PSS parallel manipulators having the specific orientation capability in a prescribed workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 2382–2385, San Francisco, April, 24-28, 2000.
- [243] Huang T., Zhao X., and Whitehouse D.J. Stiffness estimation of a Tripod-based parallel kinematic machine. *IEEE Trans. on Robotics and Automation*, 18(1):50–58, February 2002.
- [244] Huang T. and others . A general and novel approach for parameter identification of 6-dof parallel kinematic machines. *Mechanism and Machine Theory*, 40(2):219–239, February 2005.
- [245] Huang Z., Tao W.S., and Fang Y.F. Study on the kinematics characteristics of 3 DOF in-parallel actuated platform mechanisms. *Mechanism and Machine Theory*, 31(8):999–1007, November 1996.
- [246] Huang Z. and Li Q.C. General methodology for type synthesis of symmetrical lower-mobility parallel manipulators and several novel

- manipulators. *Int. J. of Robotics Research*, 21(2):131–145, February 2002.
- [247] Huang Z. and Li Q.C. Type synthesis of symmetrical lower mobility parallel mechanisms using the constraint synthesis method. *Int. J. of Robotics Research*, 22(1):59–79, January 2003.
- [248] Hunt K.H. *Kinematic geometry of mechanisms*. Clarendon Press, Oxford, 1978.
- [249] Hunt K.H. Geometry of robotics devices. *Mechanical Engineering Transactions*, 7(4):213–220, 1982.
- [250] Hunt K.H. Structural kinematics of in parallel actuated robot arms. *J. of Mechanisms, Transmissions and Automation in Design*, 105(4): 705–712, March 1983.
- [251] Hunt K.H. and Primrose E.J.F. Assembly configurations of some in-parallel actuated manipulators. *Mechanism and Machine Theory*, 28(1):31–42, January 1993.
- [252] Hunt K.H. and McAree P.R. The octahedral manipulator: geometry and mobility. *Int. J. of Robotics Research*, 17(8):868–885, 1998.
- [253] Husain M. and Waldron K.J. Position kinematics of a mixed mechanism. In *22nd Biennial Mechanisms Conf.*, volume DE-45, pages 41–48, Scottsdale, September, 13-16, 1992.
- [254] Husain M. and Waldron K.J. Direct position kinematics of the 3-1-1-1 Stewart platform. *ASME J. of Mechanical Design*, 116(4):1102–1108, December 1994.
- [255] Husty M.L. On the workspace of planar three-legged platforms. In *World Automation Congress*, volume 3, pages 339–344, Montpellier, May, 28-30, 1996.
- [256] Husty M.L. An algorithm for solving the direct kinematic of Stewart-Gough-type platforms. *Mechanism and Machine Theory*, 31(4):365–380, May 1996.
- [257] Husty M.L. and Karger A. Self-motions of Griffis-Duffy type parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 7–12, San Francisco, April, 24-28, 2000.
- [258] Husty M.L. and Karger A. Architecture singular parallel manipulators and their self-motions. In *ARK*, pages 355–364, Piran, June, 25-29, 2000.
- [259] Husty M.L. and Eberharter J. Kinematic analysis of the Hexapod telescope. In F.C. Park C.C. Iurascu, editor, *Computational Kinematics*, pages 269–278. EJCK, May, 20-22, 2001.
- [260] Husty M., Mielczarek S., and Hiller M. Redundant spatial Stewart-Gough platform with a maximal forward kinematic solution set. In *ARK*, pages 147–154, Caldes de Malavalla, June 29- July 2, 2002.
- [261] Ider S.K. Inverse dynamics of parallel manipulators in the presence

- of drive singularities. *Mechanism and Machine Theory*, 40(1):33–34, January 2005.
- [262] Idle M.K. and others . Use of a zero-gravity suspension system for testing a vibration isolation system. In *17th Aerospace testing Seminar*, pages 79–84, Manhattan Beach, October, 14-16, 1997.
- [263] Innocenti C. and Parenti-Castelli V. Direct position analysis of the Stewart platform mechanism. *Mechanism and Machine Theory*, 25(6): 611–621, 1990.
- [264] Innocenti C. and Parenti-Castelli V. Direct kinematics of the 6-4 fully parallel manipulator with position and orientation uncoupled. In *European Robotics and Intelligent Systems Conf.*, Corfou, June, 23-28, 1991.
- [265] Innocenti C. and Parenti-Castelli V. Direct kinematics of the reverse Stewart platform mechanism. In *3rd IFAC/IFIP/IMACS Symp. on Robot Control, Syroco*, pages 75–80, Vienne, September, 16-18, 1991.
- [266] Innocenti C. and Parenti-Castelli V. Forward kinematics of the general 6-6 Stewart fully-parallel mechanism: an exhaustive numerical approach via a mono-dimensional search algorithm. In *22nd Biennial Mechanisms Conf.*, volume DE-45, pages 545–552, Scottsdale, September, 13-16, 1992.
- [267] Innocenti C. and Parenti-Castelli V. Analytical form solution of the direct kinematics of a 4-4 fully in-parallel actuated six degree-of-freedom mechanism. In *9th RoManSy*, pages 41–50, Udine, 1992.
- [268] Innocenti C. and Parenti-Castelli V. Echelon form solution of direct kinematics for the general fully-parallel spherical wrist. *Mechanism and Machine Theory*, 28(4):553–561, July 1993.
- [269] Innocenti C. and Parenti-Castelli V. Direct kinematics in analytical form of a general 5-4 fully-parallel manipulators. In J. Angeles P. Kovacs, G. Hommel, editor, *Computational Kinematics*, pages 141–152. Kluwer, 1993.
- [270] Innocenti C. and Parenti-Castelli V. Closed-form direct position analysis of a 5-5 parallel mechanism. *ASME J. of Mechanical Design*, 115(3):515–521, September 1993.
- [271] Innocenti C. Analytical determination of the intersection of two coupler-point curves generated by two four-bar linkages. In J. Angeles P. Kovacs, G. Hommel, editor, *Computational Kinematics*, pages 251–262. Kluwer, 1993.
- [272] Innocenti C. and Parenti-Castelli V. Symbolic-form forward kinematics of a 5-4 fully-parallel manipulators. In Lenarčič J. and Ravani B., editors, *ARK*, pages 429–438, Ljubljana, July, 4-6, 1994. Springer-Verlag.
- [273] Innocenti C. Direct kinematics in analytical form of the 6-4 fully

- parallel mechanism. *ASME J. of Mechanical Design*, 117(1):89–95, March 1995.
- [274] Innocenti C. Algorithms for kinematic calibration of fully-parallel manipulators. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 241–250. Kluwer, 1995.
- [275] Innocenti C. and Parenti-Castelli V. Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators. *ASME J. of Mechanical Design*, 120(1):73–79, March 1998.
- [276] Innocenti C. and Parenti-Castelli V. Closed-form determination of the location of a rigid body by seven in-parallel linear transducers. *ASME J. of Mechanical Design*, 120(2):293–298, June 1998.
- [277] Inoue H., Tsusaka Y., and Fukuizumi T. Parallel manipulator. In *Proc. 3rd ISRR*, pages 321–327, Gouvieux, France, October, 7-11, 1985.
- [278] Iurascu C.C. and Park F.C. Geometric algorithm for kinematic calibration of robots containing closed loops. *ASME J. of Mechanical Design*, 125(1):23–32, March 2003.
- [279] Iyun O., D.P. Borschnek, and Ellis R.E. Computer-assisted correction of bone-deformities using a 6-dof parallel spatial mechanism. In *MICCAI*, pages 232–240, Tokyo, November, 16-18, 2002.
- [280] Jafari F. and McInroy J.E. Orthogonal Gough-Stewart platforms for micromanipulation. *IEEE Trans. on Robotics and Automation*, 19(4):595–603, August 2003.
- [281] Jaulin L., Kieffer M., Didrit O., and Walter E. *Applied Interval Analysis*. Springer-Verlag, 2001.
- [282] Jean M. and Gosselin C. Static balancing of planar parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 3732–3737, Minneapolis, April, 24-26, 1996.
- [283] Jelenkovic L. and Budin L. Error analysis of a Stewart platform based manipulators. In *Int. Conf. on Intelligent Engineering Systems (INES)*, Opatija, May, 26-28, 2002.
- [284] Jeong J.I. and others . Kinematic calibration for redundantly actuated parallel mechanisms. *ASME J. of Mechanical Design*, 126(2):307–318, March 2004.
- [285] Jeong J.W., Kim S.H., and Kwak Y.K. Kinematics and workspace analysis of a parallel wire mechanism for measuring a robot pose. *Mechanism and Machine Theory*, 34(6):825–841, August 1999.
- [286] Ji P. and Wu H.T. A fast solution to identity placement parameters for modular platform manipulators. *J. of Robotic Systems*, 17(5):251–253, 2000.
- [287] Ji Z. Dynamic decomposition for Stewart platforms. *ASME J. of Mechanical Design*, 116(1):67–69, March 1994.

- [288] Ji Z. Analysis of design parameters in platform manipulators. *ASME J. of Mechanical Design*, 118(4):526–531, December 1996.
- [289] Ji Z. and Li Z. Identification of placement parameters for modular platform manipulators. *J. of Robotic Systems*, 16(4):227–236, 1999.
- [290] Jin Q. and Yang T-L. Theory for topology synthesis of parallel manipulators and its application to three-dimension-translation parallel manipulators. *ASME J. of Mechanical Design*, 126(1):625–639, January 2004.
- [291] Jin Q. and Yang T-L. Synthesis and analysis of a group of 3-degree-of-freedom partially decoupled parallel manipulators. *ASME J. of Mechanical Design*, 126(2):301–306, March 2004.
- [292] Jo D.Y. and Haug E.J. Workspace analysis of closed loop mechanisms with unilateral constraints. In *ASME Design Automation Conf.*, pages 53–60, Montréal, September, 17-20, 1989.
- [293] Joshi S.A. and Tsai L-W. Jacobian analysis of limited-dof parallel manipulators. *ASME J. of Mechanical Design*, 124(2):254–258, June 2002.
- [294] Joshi S.A. and Tsai L-W. A comparison study of two 3-DOF parallel manipulators: one with three and the other with four supporting legs. *IEEE Trans. on Robotics and Automation*, 19(2):200–209, April 2003.
- [295] Jui C.K.K. and Sun Q. Path trackability and verification for parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 4336–4341, Taipei, September, 14-19, 2003.
- [296] Kang B. and Mills J.K. Dynamic modeling of structurally flexible planar parallel manipulator. *Robotica*, 20(3):329–339, May 2002.
- [297] Kang B., Yeung B., and Mills J.K. Two-time scale controller design for a high speed planar parallel manipulator with structural flexibility. *Robotica*, 20(5):519–528, September 2002.
- [298] Kang B.H. and others . Analysis and design of parallel mechanisms with flexure joints. In *IEEE Int. Conf. on Robotics and Automation*, pages 4097–4102, New Orleans, April, 28-30, 2004.
- [299] Kang H.J. and Freeman R.A. An interactive software package (MAP) for the dynamic modeling and simulation of parallel robotic systems including redundancy. In *ASME Int. Computer in Engineering Conf.*, pages 117–123, Boston, September, 5-9, 1990.
- [300] Karger A. Architecture singular planar parallel manipulators. *Mechanism and Machine Theory*, 38(11):1149–1164, November 2003.
- [301] Karouia M. and Hervè J.M. A three-dof tripod for generating spherical motion. In *ARK*, pages 395–402, Piran, June, 25-29, 2000.
- [302] Karouia M. and Hervè J.M. A family of novel orientational 3-dof parallel robots. In *14th RoManSy*, pages 359–368, Udine, July, 1-4,

- 2002.
- [303] Karouia M. and Hervè J.M. Asymmetrical 3-dof spherical parallel mechanisms. *European Journal of Mechanics A/Solids*, 24(1):47–57, - February 2005.
 - [304] Kassner D.J. Kinematics analysis of a planar three-degree-of-freedom platform-type robot manipulator. Master's thesis, Purdue University, Purdue, December 1990.
 - [305] Kawamura S. and others . Development of an ultrahigh speed robot FALCON using wire drive system. In *IEEE Int. Conf. on Robotics and Automation*, pages 215–220, Nagoya, May, 25-27, 1995.
 - [306] Kawamura S. and others . High-speed manipulation by using parallel wire-driven robots. *Robotica*, 18(1):13–21, January 2000.
 - [307] Khalil W. and Besnard S. Self calibration of Stewart-Gough parallel robot without extra sensors. *IEEE Trans. on Robotics and Automation*, 15(6):1116–1121, December 1999.
 - [308] Khalil W. and Besnard S. Identifiable parameters for the geometric calibration of parallel robots. *Archive of Control Sciences*, 11(3-4):263–277, 2001.
 - [309] Khalil W. and Guegan S. Inverse and direct dynamic modeling of Gough-Stewart robots. *IEEE Trans. on Robotics*, 20(4):755–761, August 2004.
 - [310] Khatib O. and Bowling A. Optimization of the inertial and acceleration characteristics of manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2883–2889, Minneapolis, April, 24-26, 1996.
 - [311] Kim D.I., Ching W.K., and Youm Y. Geometrical approach for the workspace of 6-dof parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2986–2991, Albuquerque, April, 21-28, 1997.
 - [312] Kim D., W. Chung, and Youm Y. Analytic jacobian of in-parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2376–2381, San Francisco, April, 24-28, 2000.
 - [313] Kim D.H, Kang J-Y., and Lee K-I. Robust nonlinear observer for forward kinematics solution of a Stewart platform: an experimental verification. *Robotica*, 18(6):601–610, November 2000.
 - [314] Kim H.S. and Choi Y.J. Forward/inverse force transmission capability analyses of fully parallel manipulators. *IEEE Trans. on Robotics and Automation*, 17(4):526–531, August 2001.
 - [315] Kim H.S. and Tsai L-W. Kinematic synthesis of a spatial 3-RPS parallel manipulator. *ASME J. of Mechanical Design*, 125(1):92–97, March 2003.
 - [316] Kim H.S. and Tsai L-W. Design optimization of Cartesian parallel manipulator. *ASME J. of Mechanical Design*, 125(1):43–51, March

- 2003.
- [317] Kim J. and others . Eclipse II: a new parallel mechanism enabling continuous 360-degree spinning plus three-axis translational motions. *IEEE Trans. on Robotics and Automation*, 18(3):367–373, June 2002.
 - [318] Kim S-G. and Ryu J. New dimensionally homogeneous jacobian matrix formulation by three end-effector points for optimal design of parallel manipulators. *IEEE Trans. on Robotics and Automation*, 19(4):731–736, Aout June 2003.
 - [319] Kim W-K., Yi B.J., and Cho W. RCC characteristics of planar/spherical three degree-of-freedom parallel mechanism with joint compliance. *ASME J. of Mechanical Design*, 122(1):10–16, March 2000.
 - [320] Kleinfinger J.F. *Modélisation dynamique de robots à chaînes cinématiques simple arborescente ou fermée en vue de leur commande*. Ph.D. Thesis, Université de Nantes, Nantes, May, 16, 1986.
 - [321] Knapczyk J. and Dzierzek S. Elastokinematic analysis of the 6-5 in-parallel mechanism with translational springs supporting the platform. In *12th RoManSy*, pages 88–94, Paris, July, 6-9, 1998.
 - [322] Koevermans W.P. and others . Design and performance of the four d.o.f. motion system of the NLR research flight simulator. In *AGARD Conf. Proc. No 198, Flight Simulation*, pages 17–1/17–11, La Haye, October, 20-23, 1975.
 - [323] Kohli D., Lee S-H, Tsai K-Y, and Sandor G.N. Manipulator configurations based on Rotary-Linear (R-L) actuators and their direct and inverse kinematics. *J. of Mechanisms, Transmissions and Automation in Design*, 110:397–404, December 1988.
 - [324] Koliskor A. Sh. The l-coordinate approach to the industrial robot design. In *V IFAC/IFIP/IMACS/IFORS Symposium*, pages 108–115, Suzdal, URSS, April, 22-25, 1986.
 - [325] Kong X-W. and Yang T-L. Generation and forward displacement analyses of two new classes of analytic 6 SPS parallel robot. In *ASME Design Automation Conf.*, pages 293–300, Minneapolis, September, 11-14, 1994.
 - [326] Kong X. and Gosselin C.M. Kinematics and singularity analysis of a novel type of 3-CRR 3-dof translational parallel manipulator. *Int. J. of Robotics Research*, 21(9):791–798, September 2002.
 - [327] Kong X. and Gosselin C.M. Type synthesis of three-degree-of-freedom spherical parallel manipulators. *Int. J. of Robotics Research*, 23(3): 237–245, March 2004.
 - [328] Kong X. and Gosselin C.M. Type synthesis of 3 d.of. translational parallel manipulators based on screw theory. *ASME J. of Mechanical Design*, 126(1):83–92, January 2004.
 - [329] Kong X. and Gosselin C.M. Type synthesis of 3T1R 4-dof parallel

- manipulators based on screw theory. *IEEE Trans. on Robotics and Automation*, 20(2):181–190, April 2004.
- [330] Koseki Y. and others . Kinematic analysis of translational 3-dof micro-parallel mechanism using matrix method. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Takamatsu, Japan, October 30- November 5, 2000.
- [331] Kosinska A., Galicki M., and Kedzior K. Designing and optimization of parameters of Delta-4 parallel manipulator for a given workspace. *J. of Robotic Systems*, 20(9):539–548, 2003.
- [332] Kozak K. and others . Locally linearized dynamic analysis of parallel manipulators and application of input shaping to reduce vibrations. *ASME J. of Mechanical Design*, 126(1):156–168, January 2004.
- [333] Krut S. and others . A high-speed parallel robot for Scara motion. In *IEEE Int. Conf. on Robotics and Automation*, pages 4109–4115, New Orleans, April, 28-30, 2004.
- [334] Krut S., Company O., and Pierrot F. Velocity performance indices for parallel mechanisms with actuation redundancy. *Robotica*, 22(2):129–139, March 2004.
- [335] Krut S., Company O., and Pierrot F. Force performance indexes for parallel mechanisms with actuation redundancy, especially for parallel wire-driven manipulators. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Sendai, September 28- October 2, 2004.
- [336] Kumar V. Characterization of workspaces of parallel manipulators. *ASME J. of Mechanical Design*, 114(3):368–375, September 1992.
- [337] Kumar V. Instantaneous kinematics of parallel-chain robotic mechanisms. *ASME J. of Mechanical Design*, 114(3):349–358, September 1992.
- [338] Lafourcade P. *Contribution à l'étude de manipulateurs parallèles à câbles*. Ph.D. Thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, December, 9, 2004.
- [339] Lallemand J-P., Goudali A., and Zeghloul S. The 6-dof 2-Delta parallel robot. *Robotica*, 15(4):407–416, July - August , 1997.
- [340] Lambert M. Polyarticulated retractile mechanism, March, 24, 1987. United States Patent n° 4,651,589, Société Becart S.A.
- [341] Lande M.A. and David R.J.P. Articulation for manipulator arm, November, 17, 1981. United States Patent n° 4,300,362, Association des Ouvriers en Instruments de Précision, Paris.
- [342] Landsberger S.E. and Sheridan T.B. A minimal, minimal linkage: the tension-compression parallel link manipulator. In *IMACS/SICE Int. Symp. on Robotics, Mechatronics, and Manufacturing Systems*, pages 493–500, Kobe, September, 16-20, 1992.
- [343] Larochelle P.M. Design of 3-dof spherical robotic mechanisms. In *9th*

- World Congress on the Theory of Machines and Mechanisms*, pages 1826–1830, Milan, August 30- September 2, 1995.
- [344] Laumond J.P. Obstacle growing in a non-polygonal world. *Information processing letters*, 25(1):41–50, 1987.
- [345] Lazard D. Stewart platform and Gröbner basis. In *ARK*, pages 136–142, Ferrare, September, 7-9, 1992.
- [346] Lazard D. On the representation of rigid-body motions and its application to generalized platform manipulators. In J. Angeles P. Kovacs, G. Hommel, editor, *Computational Kinematics*, pages 175–182. Kluwer, 1993.
- [347] Lazard D. and Merlet J-P. The (true) Stewart platform has 12 configurations. In *IEEE Int. Conf. on Robotics and Automation*, pages 2160–2165, San Diego, May, 8-13, 1994.
- [348] Lebesgue H. Octaèdre articulé de Bricard. *L’enseignement mathématique*, (13):150–160, 1967.
- [349] Leblond M. and Gosselin C.M. Static balancing of spatial and planar parallel manipulators with prismatic actuators. In *ASME Design Engineering Technical Conferences*, Atlanta, September, 13-16, 1998.
- [350] Lebret G., Liu K., and Lewis F. Dynamic analysis and control of a Stewart platform manipulator. *J. of Robotic Systems*, 10(5):629–655, July 1993.
- [351] Lee D.S. and Chirikjian G.S. A combinatorial approach to trajectory planning for binary manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2749–2754, Minneapolis, April, 24-26, 1996.
- [352] Lee J., Duffy J., and Keler M. The optimum quality index for the stability of in-parallel planar platform devices. *ASME J. of Mechanical Design*, 121(1):15–20, March 1999.
- [353] Lee J.D. and Geng Z. A dynamic model of a flexible Stewart platform. *Computers & Structures*, 48(3):367–374, August, 3, 1993.
- [354] Lee S.H., Yi B-J., and Kim S.H. Modeling and analysis on the internal impact of a Stewart platform used for spacecraft docking. *Advanced Robotics*, 15(7):763–777, 2001.
- [355] Lee K-M. and Shah D.K. Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE J. of Robotics and Automation*, 4(3):354–360, June 1988.
- [356] Leguay-Durand S. and Reboulet C. Optimal design of a redundant spherical parallel manipulator. *Robotica*, 15(4):399–405, July - August, 1997.
- [357] Lemay J. and Notash L. Configuration engine for architecture planning of modular parallel robot. *Mechanism and Machine Theory*, 39(1):101–117, January 2004.
- [358] Lenarčič J. and Stanišić M.M. A humanoid shoulder complex and the

- humeral pointing kinematics. *IEEE Trans. on Robotics and Automation*, 19(3):499–506, June 2003.
- [359] Lerbet J. *Mécanique des systèmes de solides rigides comportant des boucles fermées*. Ph.D. Thesis, Paris VI, Paris, June, 19, 1987.
- [360] Leroy N. and others . Dynamic modeling of a parallel robot. Application to a surgical simulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 4330–4335, Taipei, September, 14-19, 2003.
- [361] Li Q-C. and Huang Z. Mobility analysis of a novel 3-5R parallel mechanism family. *ASME J. of Mechanical Design*, 126(1):79–82, January 2004.
- [362] Li T. and Payandeh S. Design of spherical parallel mechanisms for application to laparoscopic surgery. *Robotica*, 20(2):133–138, March 2002.
- [363] Li Y. and Bone G.M. Are parallel manipulators more energy efficient ? In *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, Banff, August 29- September 1, 2001.
- [364] Li D. and Salcudean T. Modeling, simulation and control of hydraulic Stewart platform. In *IEEE Int. Conf. on Robotics and Automation*, pages 3360–3366, Albuquerque, April, 21-28, 1997.
- [365] Liao Q. and McCarthy J. M. On the seven position synthesis of a 5-SS platform linkage. *ASME J. of Mechanical Design*, 123(1):74–79, March 2001.
- [366] Lin W., Duffy J., and Griffis M. Forward displacement analysis of the 4-4 Stewart platform. *ASME J. of Mechanical Design*, 114(3):444–450, September 1992.
- [367] Lin W., Crane III C.D., and Duffy J. Closed-form forward displacement analysis of the 4-5 in-parallel platforms. *ASME J. of Mechanical Design*, 116(1):47–53, March 1994.
- [368] Lintott A.B. and Dunlop G.R. Parallel topology robot calibration. *Robotica*, 15(4):395–398, July - August , 1997.
- [369] Liu G., Lou Y., and Li Z. Singularities of parallel manipulators: a geometric treatment. *IEEE Trans. on Robotics and Automation*, 19(4):579–594, Aout August 2003.
- [370] Liu G.F. and others . Analysis and control of redundant parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 3748–3754, Seoul, May, 23-25, 2001.
- [371] Liu G.F., Wu X.Z., and Li Z.X. Inertial equivalence principle and adaptive control of redundant parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 835–840, Washington, May, 11-15, 2002.
- [372] Liu M-J., Li C-X., and C-N. Li. Dynamics analysis of the Gough-Stewart platform manipulator. *IEEE Trans. on Robotics and Au-*

- tomation*, 16(1):94–98, February 2000.
- [373] Liu X-J., Wang J., and Gao F. Performance atlases of the workspace for planar 3-dof parallel manipulators. *Robotica*, 18(5):563–568, September 2000.
- [374] Liu X-J. and Wang J. Some new parallel mechanisms containing the planar four-bar parallelogram. *Int. J. of Robotics Research*, 22(9):717–732, September 2003.
- [375] Liu K., Fitzgerald M.K., and Lewis F. Kinematic analysis of a Stewart platform manipulator. *IEEE Trans. on Industrial Electronics*, 40(2):282–293, April 1993.
- [376] Liu K., Lewis F., Lebret G., and Taylor D. The singularities and dynamics of a Stewart platform manipulator. *J. of Intelligent and Robotic Systems*, 8:287–308, 1993.
- [377] Lösch S. Inverse force analysis of the general planar parallel manipulator. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1831–1835, Milan, August 30- September 2, 1995.
- [378] Lösch S. Parallel redundant manipulator based on open and closed normal Assur chains. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 251–260. Kluwer, 1995.
- [379] Lou Y. and others . Optimal design of a parallel machine based on multiple criteria. In *IEEE Int. Conf. on Robotics and Automation*, pages 3230–3235, Barcelona, April, 19-22, 2005.
- [380] Ma O. and Angeles J. Optimum architecture design of platform manipulator. In *ICAR*, pages 1131–1135, Pise, June, 19-22, 1991.
- [381] McCallion H. and Pham D.T. The analysis of a six degrees of freedom work station for mechanized assembly. In *Proc. 5th World Congress on Theory of Machines and Mechanisms*, pages 611–616, Montréal, July 1979.
- [382] Maeda K. and others. An analysis of passive impedance of 6-dof direct-drive wrist joint. In *IMACS/SICE Int. Symp. on Robotics, Mechatronics, and Manufacturing Systems*, pages 433–438, Kobe, September, 16-20, 1992.
- [383] Marquet F. and others . Enhancing parallel robots accuracy with redundant sensors. In *IEEE Int. Conf. on Robotics and Automation*, pages 4114–4119, Washington, May, 11-15, 2002.
- [384] J.M.R. Martínez and Ravani B. On mobility analysis of linkages using group theory. *ASME J. of Mechanical Design*, 125(1):70–80, March 2003.
- [385] Masory O. and Wang J. Workspace evaluation of Stewart platforms. *Advanced Robotics*, 9(4):443–461, 1995.
- [386] Masory O., Wang J., and Zhuang H. Kinematic modeling and calibration of a Stewart platform. *Advanced Robotics*, 11(5):519–539,

- 1997.
- [387] Masuda T. and others . Mechanism configuration evaluation of a linear-actuated parallel mechanism using manipulability. In *IEEE Int. Conf. on Robotics and Automation*, pages 489–495, Washington, May, 11-15, 2002.
 - [388] Matone R. and Roth B. In-parallel manipulators: a framework on how to model actuation scheme and a study of their effects on singular postures. *ASME J. of Mechanical Design*, 121(1):2–8, March 1999.
 - [389] Maurine P. and Dombre E. A calibration procedure for the parallel robot Delta 4. In *IEEE Int. Conf. on Robotics and Automation*, pages 975–980, Minneapolis, April, 24-26, 1996.
 - [390] Mavroidis C. Completely specified displacements of a rigid body and their application in the direct kinematics of in-parallel mechanisms. *ASME J. of Mechanical Design*, 121(4):485–491, December 1999.
 - [391] Mayer St-Onge B. and Gosselin C.M. Singularity analysis and representation of the general Gough-Stewart platform. *Int. J. of Robotics Research*, 19(3):271–288, March 2000.
 - [392] McCarthy J.M. Mechanism synthesis theory and the design of robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 55–60, San Francisco, April, 24-28, 2000.
 - [393] McInroy J. E. and Hamann J.C. Design and control of flexure jointed hexapods parallel manipulator. *IEEE Trans. on Robotics and Automation*, 16(4):372–381, August 2000.
 - [394] McInroy J. E. Modeling and design of flexure jointed Stewart platforms for control purposes. *IEEE/ASME Trans. on Mechatronics*, 7(1):95–99, March 2002.
 - [395] Meng G., Tiemin L., and Wensheng Y. Calibration method and experiment of Stewart platform using a laser tracker. In *Int. Conf on Systems, Man and Cybernetics*, pages 2797–2802, The Hague, October, 10-13, 2003.
 - [396] Merkle R.C. A new family of six degree of freedom positional devices. 1994, <http://nano.xerox.com/nanotech/6dof.html>.
 - [397] Merlet J-P. Singular configurations of parallel manipulators and Grassmann geometry. *Int. J. of Robotics Research*, 8(5):45–56, October 1989.
 - [398] Merlet J-P. and Gosselin C. Nouvelle architecture pour un manipulateur parallèle à 6 degrés de liberté. *Mechanism and Machine Theory*, 26(1):77–90, 1991.
 - [399] Merlet J-P. Direct kinematics and assembly modes of parallel manipulators. *Int. J. of Robotics Research*, 11(2):150–162, April 1992.
 - [400] Merlet J-P. Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data. In *IEEE Int. Conf. on*

- Robotics and Automation*, pages 200–204, Atlanta, May, 2-7, 1993.
- [401] Merlet J-P. Trajectory verification in the workspace for parallel manipulators. *Int. J. of Robotics Research*, 13(4):326–333, August 1994.
- [402] Merlet J-P. Détermination de l'espace de travail d'un robot parallèle pour une orientation constante. *Mechanism and Machine Theory*, 29(8):1099–1113, November 1994.
- [403] Merlet J-P. Determination of the orientation workspace of parallel manipulators. *Journal of Intelligent and Robotic Systems*, 13(1):143–160, 1995.
- [404] Merlet J-P. Direct kinematics of planar parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 3744–3749, Minneapolis, April, 24-26, 1996.
- [405] Merlet J-P. Articular velocities of parallel manipulators, Part II: Finding all the robots with fixed extremal articular velocity for performing a fixed cartesian velocity over a whole workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 3262–3267, Albuquerque, April, 21-28, 1997.
- [406] Merlet J-P. Designing a parallel manipulator for a specific workspace. *Int. J. of Robotics Research*, 16(4):545–556, August 1997.
- [407] Merlet J-P. Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et singularités dans un espace de travail en translation. Research Report 3243, INRIA, September 1997.
- [408] Merlet J-P., Gosselin C., and Mouly N. Workspaces of planar parallel manipulators. *Mechanism and Machine Theory*, 33(1/2):7–20, January 1998.
- [409] Merlet J-P. Efficient computation of the extremum of the articular velocities of a parallel manipulator in a translation workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 1976–1981, Louvain, May, 18-20, 1998.
- [410] Merlet J-P. Efficient estimation of the extremal articular forces of a parallel manipulator in a translation workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 1982–1987, Louvain, May, 18-20, 1998.
- [411] Merlet J-P. Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries. *Int. J. of Robotics Research*, 18(9):902–916, October 1999.
- [412] Merlet J-P., Perng M-W., and Daney D. Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator. In *ARK*, pages 315–322, Piran, June, 25-29, 2000.
- [413] Merlet J-P. and Daney D. A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot.

- In F.C. Park C.C. Iurascu, editor, *Computational Kinematics*, pages 167–176. EJCK, Seoul, May, 20–22, 2001.
- [414] Merlet J-P. An improved design algorithm based on interval analysis for parallel manipulator with specified workspace. In *IEEE Int. Conf. on Robotics and Automation*, Seoul, May, 23–25, 2001.
- [415] Merlet J-P. A parser for the interval evaluation of analytical functions and its applications to engineering problems. *J. Symbolic Computation*, 31(4):475–486, 2001.
- [416] Merlet J-P. Optimal design for the micro robot MIPS. In *IEEE Int. Conf. on Robotics and Automation*, Washington, May, 11–15, 2002.
- [417] Merlet J-P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int. J. of Robotics Research*, 23(3):221–236, 2004.
- [418] Merlet J-P. and Daney D. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *IEEE Int. Conf. on Robotics and Automation*, Barcelona, April, 19–22, 2005.
- [419] Miller K. and Clavel R. The Lagrange-based model of Delta-4 robot dynamics. *Robotersysteme*, 8(1):49–54, 1992.
- [420] Miller K. Optimal design and modeling of spatial parallel manipulators. *Int. J. of Robotics Research*, 23(2):127–140, February 2004.
- [421] Ming A. and Higuchi T. Study on multiple degree of freedom positioning mechanisms using wires, Part 1, Concept, Design and Control. *Int. J. Japan Soc. Prec. Eng.*, 28(2):131–138, June 1994.
- [422] Ming A. and Higuchi T. Study on multiple degree of freedom positioning mechanisms using wires, Part 2, Development of a planar completely restrained positioning mechanism. *Int. J. Japan Soc. Prec. Eng.*, 28(3):235–242, September 1994.
- [423] Minsky M. Manipulator design vignettes. Research Report 267, MIT AI Lab., 1972.
- [424] Miura K. and Furuya H. Variable geometry truss and its application to deployable truss and space crane arms. In *35th Congress of the Int. Astronautical Federation*, pages 1–9, Lausanne, October, 7–13, 1984.
- [425] Mohamed M.G. and Duffy J. A direct determination of the instantaneous kinematics of fully parallel robot manipulators. *J. of Mechanisms, Transmissions and Automation in Design*, 107(2):226–229, June 1985.
- [426] Monckton S.P. and Chrystall K. Design and development of an automated footwear testing system. In *IEEE Int. Conf. on Robotics and Automation*, pages 3684–3689, Washington, May, 11–15, 2002.
- [427] Monsarrat B. and Gosselin C.M. Singularity analysis of a three-leg

- six-degree-of-freedom parallel platform mechanism based on Grassman line geometry. *Int. J. of Robotics Research*, 20(4):312–328, April 2001.
- [428] Monsarrat B. and Gosselin C.M. Workspace analysis and optimal design of a 3-leg 6-DOF parallel platform mechanism. *IEEE Trans. on Robotics and Automation*, 19(6):954–966, December 2003.
- [429] Moon Y-M. and Kota S. Design of compliant parallel kinematic machine. In *ASME 27th Biennial Mechanisms and Robotics Conf.*, pages 1–7, Montréal, September 29- October 2, 2002.
- [430] Moore R.E. *Methods and Applications of Interval Analysis*. SIAM Studies in Applied Mathematics, 1979.
- [431] Mourrain B. The 40 generic positions of a parallel robot. In Bronstein M., editor, *ISSAC'93*, ACM press, pages 173–182, Kiev (Ukraine), July 1993.
- [432] Mu Z. and Kazerounian K. A real parameter continuation method for complete solution of forward position analysis of the general Stewart. *ASME J. of Mechanical Design*, 124(2):236–244, June 2002.
- [433] Mukherjee S. and Murlidhar S. Massively parallel binary manipulators. *ASME J. of Mechanical Design*, 123(1):68–73, March 2001.
- [434] Murray A.P., Pierrot F., Dauchez P., and McCarthy J.M. A planar quaternion approach to the kinematic synthesis of a parallel manipulator. *Robotica*, 15(4):361–365, July - August , 1997.
- [435] Murray A.P. and Pierrot F. N-position synthesis of parallel planar RPR platforms. In *ARK*, pages 69–78, Strobl, June 29- July 4, 1998.
- [436] Murthy V. and Waldron K.J. Position kinematics of the generalized lobster arm and its series-parallel dual. *ASME J. of Mechanical Design*, 114(3):406–413, September 1992.
- [437] Nagai K. and others . Development of parallel manipulator "NINJA" with ultra-high-acceleration. In *IEEE Int. Conf. on Robotics and Automation*, pages 3678–3685, Taipei, September, 14-19, 2003.
- [438] Nahvi A., Hollerbach J.M., and Hayward V. Calibration of a parallel robot using multiple kinematics closed loops. In *IEEE Int. Conf. on Robotics and Automation*, pages 407–412, San Diego, May, 8-13, 1994.
- [439] Nahvi A. and Hollerbach J.M. The noise amplification index for optimal pose selection in robot calibration. In *IEEE Int. Conf. on Robotics and Automation*, pages 647–654, Minneapolis, April, 24-26, 1996.
- [440] Nair P. On the forward kinematics of parallel manipulators. *Int. J. of Robotics Research*, 13(2):171–188, April 1994.
- [441] Nakamura Y. and Ghodoussi M. Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators.

- IEEE Trans. on Robotics and Automation*, 5(3):294–302, June 1989.
- [442] Nanua P. and Waldron K.J. Direct kinematic solution of a Stewart platform. *IEEE Trans. on Robotics and Automation*, 6(4):438–444, August 1991.
- [443] Nenchev D.N. and Uchiyama M. Singularity-consistent path planning and control of parallel robot motion through instantaneous-self-motion type. In *IEEE Int. Conf. on Robotics and Automation*, pages 1864–1870, Minneapolis, April, 24–26, 1996.
- [444] Neugebauer R. and others . Hybrid struts with smart piezo actuators for high dynamic parallel kinematics. In *1st Int. Colloquium, Collaborative Research Centre 562*, pages 131–140, Braunschweig, May, 29–30, 2002.
- [445] Neumaier A. *Interval methods for systems of equations*. Cambridge University Press, 1990.
- [446] Neumann K.E. Robot, March, 22, 1988. United States Patent n° 4,732,525, Neos Product HB Norrtalje Suède.
- [447] Nguyen C.C. and Pooran F.J. Dynamic analysis of a 6 d.o.f. CKCM robot end-effector for dual-arm telerobot systems. *Robotics and Autonomous Systems*, 5(4):377–394, 1989.
- [448] Nguyen C.C. and others . Adaptive control of a Stewart platform-based manipulator. *J. of Robotic Systems*, 10(5):657–687, July 1993.
- [449] Niaritsiry F-T., Fazenda N., and Clavel R. Study of the source of inaccuracy of a 3 dof flexure hinge-based parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 4091–4096, New Orleans, April, 28–30, 2004.
- [450] Notash L. and Podhorodeski R.P. Forward displacement analysis of uncertainty configurations of parallel manipulators with a redundant branch. *J. of Robotic Systems*, 13(9):587–601, September 1996.
- [451] Notash L. and Huang L. On the design of fault tolerant parallel manipulators. *Mechanism and Machine Theory*, 38(1):85–101, January 2003.
- [452] O’Brien J.F. and Wen J.T. Kinematic control of parallel robots in the presence of unstable singularities. In *IEEE Int. Conf. on Robotics and Automation*, pages 3154–3159, Seoul, May, 23–25, 2001.
- [453] Oh S-R. and others . Dynamic modeling and robust controller design of a two-stage parallel cable robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 3678–3683, New Orleans, April, 28–30, 2004.
- [454] Ojala P., Arai T., and Tanikawa T. Kinematic analysis and motion control of a redundant micro manipulator. In *2nd Japan-France Congress on Mechatronics*, pages 473–476, Takamatsu, November, 1–3, 1994.
- [455] Oliviers M.P. and Mayer J.R.R. Global kinematic calibration of a

- Stewart platform. *ASME DSC*, 57(1):129–136, 1995.
- [456] Orin D.E. and Oh S.Y. Control of force distribution in robotic mechanisms containing closed kinematic chains. *J. of Dyn. Syst. Meas. and Control*, 102:134–141, June 1981.
- [457] Ottaviano E. and Ceccarelli M. Optimal design of CAPAMAN (Cassino parallel manipulator) with a specific orientation workspace. *Robotica*, 20(2):159–166, March 2002.
- [458] Ouyang P.R., Zhang W.J., and Wu F.X. Nonlinear PD control for trajectory tracking with consideration of the design for control methodology. In *IEEE Int. Conf. on Robotics and Automation*, pages 4126–4131, Washington, May, 11-15, 2002.
- [459] Parenti-Castelli V. and Innocenti C. Direct displacement analysis for some classes of spatial parallel mechanisms. In *8th RoManSy*, pages 123–130, Cracow, July, 2-6, 1990.
- [460] Parenti-Castelli V. and Innocenti C. Forward displacement analysis of parallel mechanisms: closed-form solution of PRR-3S and PPR-3S structures. *ASME J. of Mechanical Design*, 114(1):68–73, March 1992.
- [461] Parenti-Castelli V. and Di Gregorio R. Determination of the actual configuration of the general Stewart platform using only one additional sensor. *ASME J. of Mechanical Design*, 121(1):21–25, March 1999.
- [462] Parenti-Castelli V. and Di Gregorio R. 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*, 122(3):294–298, September 2000.
- [463] Parenti-Castelli V. and Di Gregorio R. Influence of manufacturing errors on the kinematic performance of the 3-UPU parallel mechanism. In *2nd Chemnitzer Parallelkinematik Seminar*, pages 85–99, Chemnitz, April, 12-13, 2000.
- [464] Parenti-Castelli V. and Di Gregorio R. Real-time actual pose determination of the general fully parallel spherical wrist, using only one extra sensor. *J. of Robotic Systems*, 18(12):723–729, 2001.
- [465] Parenti-Castelli V. and Venanzi S. On the joint clearance effects in serial and parallel manipulators. In *Workshop: Fundamental issues and future directions for parallel mechanisms and manipulators*, pages 215–223, Québec, October, 3-4, 2002.
- [466] Parenti-Castelli V. and others . On the modeling of passive motion of the human knee joint by means of equivalent planar and spatial parallel mechanisms. *Autonomous Robots*, 16(2):219–232, 2004.
- [467] Park M.K. and Kim J.W. Kinematic manipulability of closed chains. In *ARK*, pages 99–108, Portoroz-Bernadin, June, 22-26, 1996.

- [468] Patel A.J. and Ehmman K.F. Volumetric error analysis of a Stewart platform based machine tool. *Annals of the CIRP*, 46/1/1997:287–290, 1997.
- [469] Pennock G.R. and Kassner D.J. Kinematic analysis of a planar eight-bar linkage: application to a platform-type robot. In *ASME Proc. of the 21th Biennial Mechanisms Conf.*, pages 37–43, Chicago, September, 16-19, 1990.
- [470] Perez A. and McCarthy J.M. Dual quaternion synthesis of constrained robotic systems. *ASME J. of Mechanical Design*, 126(3):425–435, May 2004.
- [471] Pernette E. and others . Design of parallel robots in microrobotics. *Robotica*, 15(4):417–420, July - August , 1997.
- [472] Perng M-H. and Hsiao L. Inverse kinematics solutions for a fully parallel robot with singularity robustness. *Int. J. of Robotics Research*, 18(6):575–583, June 1999.
- [473] Pernkopf F. *Workspace analysis of Stewart-Gough platforms*. Ph.D. Thesis, Bau fakultät, University of Innsbruck, September, 11, 2003.
- [474] Pham H.H. and Chen I-M. Optimal synthesis for workspace and manipulability of parallel flexure mechanism. In *11th World Congress on Theory of Machines and Mechanisms*, Tianjin, April, 1-4, 2004.
- [475] Pierrot F. *Robots Pleinement Parallèles Légers : Conception Modélisation et Commande*. Ph.D. Thesis, Université Montpellier II, Montpellier, April, 24, 1991.
- [476] Pierrot F., Dauchez P., and Fournier A. Fast parallel robots. *Journal of Robotic Systems*, 8(6):829–840, December 1991.
- [477] Pierrot F. and Chiacchio P. Evaluation of velocity capabilities for redundant parallel robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 774–779, Albuquerque, April, 21-28, 1997.
- [478] Pierrot F. Parallel mechanisms and redundancy. In *1st Int. Colloquium, Collaborative Research Centre 562*, pages 261–277, Braunschweig, May, 29-30, 2002.
- [479] Pittens K.H. and Podhorodeski R.P. A family of Stewart platforms with optimal dexterity. *J. of Robotic Systems*, 10(4):463–479, June 1993.
- [480] Podhorodeski R. Three branch hybrid-chain manipulators. In *ARK*, pages 150–155, Ferrare, September, 7-9, 1992.
- [481] Pollard W.L.V. Position controlling apparatus, June, 16, 1942. United States Patent n° 2,286,571.
- [482] Portman V.T., Sandler B-Z, and Zahavi E. Rigid 6x6 parallel platform for precision 3D micromanipulation: theory and design application. *IEEE Trans. on Robotics and Automation*, 16(6):629–643, December 2000.

- [483] Pott A., Franitz D., and Hiller M. Orientation workspace verification for parallel kinematic machines with constant legs length. In *Mechatronics and Robotics Conf.*, Aachen, September, 13-15, 2004.
- [484] Pott A. and Hiller M. A new approach to error analysis in parallel kinematic structures. In *ARK*, Sestri-Levante, June 28- July 1, 2004.
- [485] Pottmann H., Peternell M., and Ravani B. Approximation in line space. Applications in robot kinematics. In *ARK*, pages 403–412, Strobl, June 29- July 4, 1998.
- [486] Pottmann H., Peternell M., and Ravani B. An introduction to line geometry with applications. *Computer-aided design*, 31(1):3–16, 1999.
- [487] Powell I.L. The kinematic analysis and simulation of the parallel topology manipulator. *The Marconi Review*, XLV(226):121–138, Third Quarter 1982.
- [488] Pritschow G., Eppler C., and Lehner W-D. Highly dynamic drives for parallel kinematic machines with constant arm length. In *1st Int. Colloquium, Collaborative Research Centre 562*, pages 199–211, Braunschweig, May, 29-30, 2002.
- [489] Pritschow G., Eppler C., and Garber T. Influence of the dynamic stiffness on the accuracy of PKM. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 313–333, Chemnitz, April, 23-25, 2002.
- [490] S. Pugazhenti and others . Optimal trajectory planning for a hexapod machine-tool. *Proc. Instn Mech Engrs, Part C: J. Mechanical Engineering Science*, 216(12):1247–1257, December 2002.
- [491] Pusey J. and others . Design and workspace analysis of a 6-6 cable-suspended parallel robot. *Mechanism and Machine Theory*, 139(7):761–778, July 2004.
- [492] Raghavan M. The Stewart platform of general geometry has 40 configurations. *ASME J. of Mechanical Design*, 115(2):277–282, 1993.
- [493] Ramachandran S. and others . A finite element approach to the design and dynamic analysis of platform type manipulators. *Finite elements in Analysis and Design*, 10(4):335–350, 1992.
- [494] Ramrakhani D.S. and Lesieutre G.A. Aircraft structure morphing using tendon actuated compliant cellular trusses. In *45th AIAA Structures, Structural & Materials Conf.*, Palm Springs, April, 19-22, 2004.
- [495] Ranganath R. and others . A force-torque sensor based on a Stewart platform in a near-singular configuration. *Mechanism and Machine Theory*, 39(9):971–998, September 2004.
- [496] Rao A.C. Parallelism in planar kinematic chains (manipulators). *Mechanism and Machine Theory*, 39(10):1111–1122, October 2004.
- [497] Rauf A. and Ryu J. Fully autonomous calibration of parallel manipulators by imposing position constraint. In *IEEE Int. Conf. on*

- Robotics and Automation*, pages 2389–2394, Seoul, May, 23–25, 2001.
- [498] Rebman J. Object manipulator, August, 23, 1988. United States Patent n° 4,765,795 Lord Corporation, Eric, Pa.
- [499] Reboulet C. and Robert A. Hybrid control of a manipulator with an active compliant wrist. In *3rd ISRR*, pages 76–80, Gouvieux, France, October, 7–11, 1985.
- [500] Reboulet C. and Berthomieu T. Dynamic model of a six degree of freedom parallel manipulator. In *ICAR*, pages 1153–1157, Pise, June, 19–22, 1991.
- [501] Reboulet C., Lambert C., and Nombraïl N. A parallel redundant manipulator: SPEED-R-MAN and its control. In *ISRAM*, pages 285–291, Santa-Fe, November, 11–13, 1992.
- [502] Reboulet C. and Pigeyre R. Hybrid control of a 6 d.o.f. in parallel actuated micro-manipulator mounted on a SCARA robot. *Int. J. of Robotics and Automation*, 7(1):10–14, 1992.
- [503] Reboulet C. Parallel-structure manipulator device for displacing and orienting an object in a cylindrical workspace, July, 23, 1996. United States Patent n° 5,539,291 ONERA.
- [504] Rees Jones J. Cross-coordinate control of a robot manipulator. In *Int. Workshop on Nuclear Robotics Technologies and Applications: Present and Future*, pages 1–10, University of Lancaster, -Juillet June, 26–1, 1979.
- [505] Reinholtz C.F. and Gokhale D. Design and analysis of variable geometry truss robots. In *9th Annual Conf. on Applied Mechanisms*, pages 1–5, Oklahoma State University, 1987.
- [506] Renaud P., Andreff N., Pierrot F., and Martinet P. Combining end-effector and legs observation for kinematic calibration of parallel mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 4116–4121, New Orleans, April, 28–30, 2004.
- [507] Rico J.M. and Ravani B. Group theory can explain the mobility of paradoxical linkages. In *ARK*, pages 245–254, Caldes de Malavalla, June 29– July 2, 2002.
- [508] Riechel A.T. and Ebert-Uphoff I. Force-feasible workspace analysis for underconstrained point-mass cable robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 4956–4962, New Orleans, April, 28–30, 2004.
- [509] Roberts R.G. Minimal realization of an arbitrary spatial stiffness matrix with a parallel connection of simple and complex springs. *IEEE Trans. on Robotics and Automation*, 16(5):603–608, October 2000.
- [510] Rojeski P. J. *A systems analysis approach to landing gear design*. Ph.D. Thesis, Cornell University, May 1972.
- [511] Ronga F. and Vust T. Stewart platforms without computer? In *Conf.*

- Real Analytic and Algebraic Geometry*, pages 197–212, Trento, 1992.
- [512] Rooney J. and Earl C.F. Manipulator postures and kinematics assembly configurations. In *6th World Congress on Theory of Machines and Mechanisms*, pages 1014–1020, New Dehli, 1983.
- [513] Roth B. The kinematics of motion through finitely separated positions. *ASME J. of Applied Mechanics*, pages 591–597, September 1967.
- [514] Rouillier F. Real roots counting for some robotics problems. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 73–82. Kluwer, 1995.
- [515] Ryu R.J. and others . Eclipse: an overactuated parallel mechanism for rapid machining. In *12th RoManSy*, pages 79–86, Paris, July, 6-9, 1998.
- [516] Ryu J. and Cha J. Volumetric error analysis and architecture optimization for accuracy of HexaSlide type parallel manipulators. *Mechanism and Machine Theory*, 38(3):227–240, March 2003.
- [517] Salcudean S.E. and others . A six degree-of-freedom, hydraulic, one person motion simulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 2437–2443, San Diego, May, 8-13, 1994.
- [518] Sarkissyan Y.L. and Parikyan T.F. Manipulator, 1990. Russian Patent n° 1585144.
- [519] Sarkissyan Y.L. and Parikyan T.F. Direct position problem for Stewart platform and multiple points of 5(SS) linkage coupler curve. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1614–1618, Milan, August 30- September 2, 1995.
- [520] Sayapin S.N. Application of parallel kinematics machines for active vibration isolation and pointing of high-precision large deployable space structure (HLDSS). In *3rd Chemnitzer Parallelkinematik Seminar*, pages 957–962, Chemnitz, April, 23-25, 2002.
- [521] Schmid H.A. Spreadbands drive parallel robots. *Industrial robot*, 28(4):320–327, 2001.
- [522] Schoenflies A. Beweis eines Satzes über Bewegungsgruppen. *Nachrichten von der Königl. Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen*, (15):495–501, 1886.
- [523] Schönherr M. Vorrichtung zur messung des Kräfte und momente ruhender and bewegte objkete, June, 9, 1990. German Patent DE 4018558C2.
- [524] Sefrioui J. and Gosselin C. Singularity analysis and representation of planar parallel manipulators. *Robotics and Autonomous Systems*, 10(4):209–224, 1992.
- [525] Sefrioui J. and Gosselin C. Étude et représentation des lieux de singularités des manipulateurs parallèles sphériques à trois degrés de lib-

- erté avec actionneurs prismatiques. *Mechanism and Machine Theory*, 29(4):559–579, May 1994.
- [526] Sefrioui J. and Gosselin C.M. On the quadratic nature of the singularity curves of planar three-degree-of-freedom parallel manipulators. *Mechanism and Machine Theory*, 30(4):533–551, May 1995.
- [527] Seguchi Y., Tanaka M., and others . Dynamic analysis of a truss-type flexible robot arm. *JSME Int. J.*, 33(2):183–190, 1990.
- [528] Seguchi Y., Tanaka M., and others . Criteria-oriented configuration control of adaptive structure and its modular neural network representation. In *First Joint USA/Japan Conf. on adaptive structure*, pages 402–421, Maui, Hawaii, November, 13-15, 1990.
- [529] Selig J.M. and Ding X. Theory of vibrations in Stewart platforms. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2190–2195, Maui, Hawaii, October 29- November 3, 2001.
- [530] Sellgren U. Modeling of mechanical interfaces in a systems context. In *Int. ANSYS Conf.*, Pittsburgh, April 2002.
- [531] Sen S., Dasgupta B., and Mallik A.K. Variational approach for singularity-path planning of parallel manipulators. *Mechanism and Machine Theory*, 38(11):1165–1183, November 2003.
- [532] Shaw D. and Chen Y-S. Cutting path generation of the Stewart platform-based milling machine using an end-mill. *Int. J. Prod. Res.*, 39(7):1367–1383, 2001.
- [533] Sheldon P.C. and others . Metrology instrument arm system, February, 6, 1996. United States Patent n° 5,489,168 Giddings & Lewis.
- [534] Shelef G. Six degree of freedom micromanipulator, April, 11, 1989. United States Patent n° 4,819,496 Air Force Washington.
- [535] Shiang W-J., Cannon D., and Gorman J. Optimal force distribution applied to a robotic crane with flexible cables. In *IEEE Int. Conf. on Robotics and Automation*, pages 1948–1954, San Francisco, April, 24-28, 2000.
- [536] Shirkhodaie A.H. and Soni A.H. Forward and inverse synthesis for a robot with three degree-of-freedom. In *19th Summer Computer Simulation Conf.*, pages 851–856, Montréal, July, 27-30, 1987.
- [537] Shoham M. and others . Bone-mounted miniature robot for surgical procedures: concept and clinical applications. *IEEE Trans. on Robotics and Automation*, 19(5):893–901, October 2003.
- [538] Shoham M. Twisting wire actuator. *ASME J. of Mechanical Design*, 127(3):441–445, May 2005.
- [539] Siciliano B. The Tricept robot: inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm. *Robotica*, 17(4): 437–445, July 1999.
- [540] Simaan N. and others . Design considerations of new six degrees-of-

- freedom parallel robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 1327–1333, Louvain, May, 18-20, 1998.
- [541] Simaan N. and Shoham M. Stiffness synthesis of a variable geometry six-degree-of-freedom double planar parallel robot. *Int. J. of Robotics Research*, 22(9):757–775, September 2003.
- [542] Simaan N. and Shoham M. Geometric interpretation of the derivatives of parallel robots' jacobian matrix with application to stiffness control. *ASME J. of Mechanical Design*, 125(1):33–42, March 2003.
- [543] Sincarsin W.G. and Hughes P.C. Trussarm : candidate geometries. Research Report 28-611/0401, Dynacon Enterprises Ltd., 1987.
- [544] Snyman J.A., Du Plessis L.J., and Duffy J. An optimisation approach to the determination of the boundaries of manipulator workspaces. *ASME J. of Mechanical Design*, 122(4):447–456, December 2000.
- [545] Snyman J.A. and Hay A.M. Optimal synthesis for a continuous prescribed dexterity interval of a 3-dof parallel planar manipulator for different prescribed output workspaces. In *Computational Kinematics*, Cassino, May, 4-6, 2005.
- [546] Sommese A.J., Verschelde J., and Wampler C.W. Advances in polynomial continuation for solving problems in kinematics. *ASME J. of Mechanical Design*, 126(2):262–268, March 2004.
- [547] Sorli M. and others . Mechanics of Turin parallel robot. *Mechanism and Machine Theory*, 32(1):51–77, January 1997.
- [548] Staffetti E., Bruyninckx H., and De Schutter J. On the invariance of manipulability indices. In *ARK*, pages 57–66, Caldes de Malavalla, June 29- July 2, 2002.
- [549] Staffetti E. Kinestatic analysis of robot manipulators using the Grassmann-Cayley algebra. *IEEE Trans. on Robotics and Automation*, 20(2):200–210, April 2004.
- [550] Stamper R.C., Tsai C-W., and Walsh G.C. Optimization of a three dof translational platform for well-conditioned workspace. In *IEEE Int. Conf. on Robotics and Automation*, pages 3250–3255, Albuquerque, April, 21-28, 1997.
- [551] Stewart D. A platform with 6 degrees of freedom. *Proc. of the Institution of mechanical engineers*, 180(Part 1, 15):371–386, 1965.
- [552] Stocco L. and Salcudean T. Optimal kinematic design of a haptic pen. *IEEE/ASME Trans. on Mechatronics*, 6(3):210–220, September 2001.
- [553] Stock M. and Miller K. Optimal kinematic design of spatial parallel manipulators: application to linear Delta robot. *ASME J. of Mechanical Design*, 125(2):292–301, June 2003.
- [554] Stoughton R. and Arai T. Optimal sensor placement for forward kinematics evaluation of a 6-dof parallel link manipulator. In *IEEE*

- Int. Conf. on Intelligent Robots and Systems, (IROS)*, pages 785–790, Osaka, November, 3–5, 1991.
- [555] Stoughton R. and Arai T. A modified Stewart platform manipulator with improved dexterity. *IEEE Trans. on Robotics and Automation*, 9(2):166–173, April 1993.
- [556] Su H.-J., Dietmaier P., and J.M. McCarthy. Trajectory planning for constrained parallel manipulators. *ASME J. of Mechanical Design*, 125(4):709–716, December 2003.
- [557] Su H.-J. and J.M. McCarthy. Dimensioning a constrained parallel robot to reach a set of task positions. In *IEEE Int. Conf. on Robotics and Automation*, pages 4037–4041, Barcelona, April, 19–22, 2005.
- [558] Su X.S. and others . Singularity analysis of fine-tuning Stewart platform for large radio telescope using genetic algorithm. *Mechatronics*, 13(5):413–425, June 2003.
- [559] Sugar T.G. and Kumar V. Design and control of a compliant parallel manipulator. *ASME J. of Mechanical Design*, 124(4):676–683, December 2002.
- [560] Sugimoto K., Duffy J., and Hunt K.H. Special configurations of spatial mechanisms and robot arms. *Mechanism and Machine Theory*, 17(2):119–132, 1982.
- [561] Sugimoto K. Computational scheme for dynamic analysis of parallel manipulators. *J. of Mechanisms, Transmissions and Automation in Design*, 111(1):29–33, March 1989.
- [562] Sutter T.R. and others . Structural characterization of a first generation articulated truss joint for space crane application. Research Report TM 4371, NASA Research Center, Langley, June 1992.
- [563] Svinin M.M., Ueda K., and Uchiyama M. On the stability conditions for a class of parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2386–2391, San Francisco, April, 24–28, 2000.
- [564] Svinin M., Hosoe S., and Uchiyama M. On the stiffness and stability of Gough-Stewart platforms. In *IEEE Int. Conf. on Robotics and Automation*, pages 3268–3273, Seoul, May, 23–25, 2001.
- [565] Tadokoro S. Control of parallel mechanisms. *Advanced Robotics*, 8(6):559–571, December 1994.
- [566] Tadokoro S. and others . A portable parallel manipulator for search and rescue at large-scale urban earthquakes and an identification algorithm for the installation in unstructured environments. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1222–1227, Kyongju, October, 17–21, 1999.
- [567] Tadokoro S. and others . A motion base with 6-dof by parallel cable drive architecture. *IEEE/ASME Trans. on Mechatronics*, 7(2):115–123, June 2002.

- [568] Tahmasebi F. and Tsai L.-W. Closed form direct kinematics solution of a new parallel minimanipulator. *ASME J. of Mechanical Design*, 116(4):1141–1147, December 1994.
- [569] Tahmasebi F. and Tsai L.-W. Workspace and singularity analysis of a novel six-dof parallel minimanipulator. *J. of Applied Mechanisms and Robotics*, 1(2):31–40, March 1994.
- [570] Takanobu H. and others . Mouth opening and closing training with 6-dof parallel robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 1384–1389, San Francisco, April, 24-28, 2000.
- [571] Takeda Y., Shen G., and Funabashi H. A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series. In *ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*, Montréal, September 29- October 2, 2002.
- [572] Tanaka W. and others . Calibration method by simplified measurement for parallel mechanism. In *11th ICAR*, pages 1781–1786, Coimbra, June 30- July 3, 2003.
- [573] Tancredi L. and Teillaud M. Application de la géométrie synthétique au problème de modélisation géométrique directe des robots parallèles. *Mechanism and Machine Theory*, 34(2):255–269, February 1999.
- [574] Tanev T.K. Forward displacement analysis of a three legged four-degree-of-freedom parallel manipulator. In *ARK*, pages 147–154, Strobl, June 29- July 4, 1998.
- [575] Tapia R.A. The Kantorovitch theorem for Newton’s method. *American Mathematic Monthly*, 78(1.ea):389–392, 1971.
- [576] Thomas F. and others . Uncertainty model and singularities of 3-2-1 wire-based tracking systems. In *ARK*, pages 107–116, Caldes de Malavalla, June 29- July 2, 2002.
- [577] Thompson C.J. and Campbell P.D. Tendon suspended platform robot, December, 17, 1996. United States Patent n° 5,585,707, McDonnell Douglas Corporation.
- [578] Thornton G.S. The GEC Tetrabot-a new serial-parallel assembly robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 437–439, Philadelphia, April, 24-29, 1988.
- [579] Tischler C.R. and Samuel A.E. Predicting the slop of in-series/parallel manipulators caused by joint clearances. In *ARK*, pages 227–236, Strobl, June 29- July 4, 1998.
- [580] Tischler C.R., Andrew S., and Hunt K.H. Selecting multi-freedom multi-loop kinematic chain to suit a given task. *Mechanism and Machine Theory*, 36(8):925–938, August 2001.
- [581] Tönshoff K., Grendel H., and Kaak R. A hybrid manipulator for laser

- machining. In *First European-American Forum on Parallel Kinematic Machines*, Milan, August 31- September 1, 1998.
- [582] Tönshoff K. and others . Modelling of error effects on the new hybrid kinematic DUMBO structure. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 639–653, Chemnitz, April, 23-25, 2002.
- [583] Trinkle J.C. and R.J. Milgram. Complete path planning for closed kinematic chains with spherical joints. *Int. J. of Robotics Research*, 21(9):773–789, September 2002.
- [584] Tsai K-Y. and Huang K.D. The design of isotropic 6-DOF parallel manipulators using isotropy generators. *Mechanism and Machine Theory*, 38(11):1199–1214, November 2003.
- [585] Tsai L-W. and Tahmasebi F. Synthesis and analysis of a new class of six-degree-of-freedom parallel minimanipulators. *J. of Robotic Systems*, 10(5):561–580, July 1993.
- [586] Tsai L-W. Kinematics of a three-dof platform with three extensible limbs. In *ARK*, pages 401–410, Portoroz-Bernadin, June, 22-26, 1996.
- [587] Tsai L-W. The jacobian analysis of a parallel manipulator using reciprocal screws. In *ARK*, pages 327–336, Strobl, June 29- July 4, 1998.
- [588] Tsai L-W. Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work. *ASME J. of Mechanical Design*, 122(1):3–9, March 2000.
- [589] Tsai L-W. and Joshi S. Kinematics and optimization of a spatial 3-UPU parallel manipulator. *ASME J. of Mechanical Design*, 112(4):439–446, December 2000.
- [590] Tsai L-W. and Joshi S. Kinematic analysis of 3-dof position mechanisms for use in hybrid kinematic machines. *ASME J. of Mechanical Design*, 124(2):245–253, June 2002.
- [591] Uchiyama M. A 6 d.o.f. parallel robot HEXA. *Advanced Robotics*, 8(6):601, December 1994.
- [592] Veblen O. and Young J.W. *Projective geometry*. The Athenaeum Press, 1910.
- [593] Verhoeven R. *Analysis of the workspace of tendon-based Stewart platforms*. Ph.D. Thesis, University of Duisburg-Essen, Duisburg, 2004.
- [594] Vertecky R., Dunlop G.R., and Parenti-Castelli V. An accurate algorithm for the real-time solution of the direct kinematics of 6-3 Stewart platform manipulators. In *ARK*, pages 369–378, Caldes de Malavalla, June 29- July 2, 2002.
- [595] Vischer P. *Improving the accuracy of parallel robots*. Ph.D. Thesis, EPFL, Lausanne, 1996.
- [596] Vischer P. and Clavel R. Argos: a novel 3-dof parallel wrist mecha-

- nism. *Int. J. of Robotics Research*, 19(1):5–11, January 2000.
- [597] Voglewede P.A. and Ebert-Uphoff I. Measuring "closeness" to singularities for parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 4539–4544, New Orleans, April, 28-30, 2004.
- [598] Voglewede P.A. and Ebert-Uphoff I. Application of workspace generation techniques to determine the unconstrained motion of parallel manipulators. *ASME J. of Mechanical Design*, 126(2):283–290, March 2004.
- [599] Waldron K.J., Raghavan M., and Roth B. Kinematics of a hybrid series-parallel manipulation system. *J. of Mechanisms, Transmissions and Automation in Design*, 111(2):211–221, June 1989.
- [600] Waldron K.J. and Hunt K.H. Series-parallel dualities in actively coordinated mechanisms. *Int. J. of Robotics Research*, 10(2):473–480, April 1991.
- [601] Wampler C.W., Hollerbach J.M., and Arai T. An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Trans. on Robotics and Automation*, 11(5):710–724, October 1995.
- [602] Wampler C.W. Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mechanism and Machine Theory*, 31(3):331–337, April 1996.
- [603] Wampler C.W. Displacement analysis of spherical mechanism having three or fewer loops. *ASME J. of Mechanical Design*, 126(1):93–100, January 2004.
- [604] Wang L.C.T. and Chen C.C. On the dynamic analysis of a general parallel robotic manipulators. *Int. J. of Robotics and Automation*, 9(2):81–87, 1994.
- [605] Wang S. and others . Kinematics and force analysis of a 6 d.o.f. parallel mechanism with elastic joints. In *ARK*, pages 87–96, Strobl, June 29- July 4, 1998.
- [606] Wang S.M. and Ehmann K.F. Error model and accuracy analysis of a six-dof Stewart platform. *ASME Journal of Manufacturing Science and Engineering*, 124(2):286–295, May 2002.
- [607] Wang Z. and others . A study on workspace, boundary workspace analysis and workpiece positioning for parallel machine tools. *Mechanism and Machine Theory*, 36(6):605–622, June 2001.
- [608] Wang J. and Masory O. On the accuracy of a Stewart platform-part I: The effect of manufacturing tolerances. In *IEEE Int. Conf. on Robotics and Automation*, pages 114–120, Atlanta, May, 2-6, 1993.
- [609] Wang J. and Gosselin C.M. A new approach for the dynamic analysis of parallel manipulators. *Multibody System Dynamics*, 2(3):317–334, September 1998.

- [610] Wang J. and Gosselin C.M. Static balancing of spatial three-degree-of-freedom parallel mechanisms. *Mechanism and Machine Theory*, 34(3):437–452, April 1999.
- [611] Wang J. and Gosselin C.M. Singularity loci of a special class of spherical 3-dof parallel mechanisms with prismatic actuators. *ASME J. of Mechanical Design*, 126(2):319–326, March 2004.
- [612] Wang J. and Gosselin C.M. Kinematic analysis and design of kinematically redundant parallel mechanisms. *ASME J. of Mechanical Design*, 126(1):109–118, January 2004.
- [613] Wapler M. and others . A Stewart platform for precision surgery. *Trans. of the Institute of Measurement and Control*, 25(4):329–334, 2003.
- [614] Weck M. and Giesler M. Task oriented multi-objective-optimization of parallel kinematics for machine-tools. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 187–211, Chemnitz, April, 23-25, 2002.
- [615] Weikert S. and Knapp W. Application of the grid-bar device on the Hexaglide. In *3rd Chemnitzer Parallelkinematik Seminar*, pages 295–310, Chemnitz, April, 23-25, 2002.
- [616] Wendlandt J.M. and Sastry S.S. Design and control of a simplified Stewart platform for endoscopy. In *33rd Conf. on Decision and Control*, pages 357–362, Lake Buena Vista, December, 14-16, 1994.
- [617] Wenger P. and Chablat D. Workspace and assembly modes in fully parallel manipulators: a descriptive study. In *ARK*, pages 117–126, Strobl, June 29- July 4, 1998.
- [618] Wenger P. and Chablat D. Kinematic analysis of a new parallel machine-tool: the Orthoglide. In *ARK*, pages 305–314, Piran, June, 25-29, 2000.
- [619] Wenger P. and Chablat D. Design of a three-axis isotropic parallel manipulator for machining applications: the Orthoglide. In *Workshop: Fundamental issues and future directions for parallel mechanisms and manipulators*, pages 16–23, Québec, October, 3-4, 2002.
- [620] Whitney D.E. What is the RCC and what it can do. In *9^{eme} ISIR*, pages 135–147, Washington, 13-15 Mars 1979.
- [621] Williams II R.L. and Hexter E.R. Maximizing kinematic motion for a 3-dof VGT module. *ASME J. of Mechanical Design*, 120(2):333–336, June 1998.
- [622] Williams II R.L., Albus J., and Bostelman R. 3D cable-based cartesian metrology system. *J. of Robotic Systems*, 21(5):237–257, 2004.
- [623] Wohlhart K. Degrees of shakiness. *Mechanism and Machine Theory*, 34(7):1103–1126, October 1999.
- [624] Wohlhart K. Architectural shakiness or architectural mobility of platforms. In *ARK*, pages 365–374, Piran, June, 25-29, 2000.

- [625] Wohlhart K. Mobile 6-SPS parallel manipulators. *J. of Robotic Systems*, 20(8):509–516, 2003.
- [626] Wolf A. and Shoham M. Investigation of parallel manipulators using linear complex approximation. *ASME J. of Mechanical Design*, 125(3):564–572, September 2003.
- [627] Wu H. and others . Design of parallel intersector weld/cut robot for machining processes in ITER vacuum vessel. *Fusion Engineering and Design*, 69(1):327–331, 2003.
- [628] Xi F. Dynamic balancing of hexapods for high-speed applications. *Robotica*, 17(3):335–342, May 1999.
- [629] Xi F. and others . A comparison study on tripod units for machine tools. In *3rd Chemnitz Parallelkinematik Seminar*, pages 923–939, Chemnitz, April, 23-25, 2002.
- [630] Xie D. and Anamato N.M. A kinematics-based probabilistic roadmap method for high dof closed chain systems. In *IEEE Int. Conf. on Robotics and Automation*, New Orleans, April, 28-30, 2004.
- [631] Yakey J.H. and others . Randomized path planning for linkages with closed kinematic chains. *IEEE Trans. on Robotics and Automation*, 17(6):951–958, December 2001.
- [632] Yang G. and others . Kinematic design of modular reconfigurable in-parallel robots. *Autonomous Robots*, 10(2):83–89, January 2001.
- [633] Yang G. and others . Self-calibration of three-legged modular reconfigurable parallel robots based on leg-end distance errors. *Robotica*, 19(2):187–198, March 2001.
- [634] Yang G., Chen I-M., and Angeles J. Singularity analysis of three-legged parallel robots based on passive joint velocities. *IEEE Trans. on Robotics and Automation*, 17(4):413–422, August 2001.
- [635] Yang J. and Geng Z.J. Closed form forward kinematics solution to a class of hexapod robots. *IEEE Trans. on Robotics and Automation*, 14(7):503–508, June 1998.
- [636] Yi B-J. and others . Design and experiment of a 3-DOF parallel micromechanism utilizing flexure hinges. *IEEE Trans. on Robotics and Automation*, 19(4):604–612, August 2003.
- [637] Yin J.P. and Liang C.G. The forward displacement analysis of a kind of special platform manipulator. *Mechanism and Machine Theory*, 29(1):1–9, January 1994.
- [638] Yiu Y.K. and others . On the dynamics of parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 3766–3771, Seoul, May, 23-25, 2001.
- [639] Yoon W-K. and others . Stiffness analysis and design of a compact modified Delta parallel mechanism. *Robotica*, 22(5):463–475, September 2004.

- [640] Yoshikawa T. Manipulability of robotic mechanisms. *Int. J. of Robotics Research*, 4(2):3–9, 1985.
- [641] Yu X. and others . Measuring data based non-linear error modeling for parallel machine-tool. In *IEEE Int. Conf. on Robotics and Automation*, pages 3535–3540, Seoul, May, 23-25, 2001.
- [642] Yufeng L. and Tingli Y. Structure types and characteristics of six degree-of-freedom closed-chain manipulators with all actuators on base. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1795–1799, Milan, August 30- September 2, 1995.
- [643] Zabalza I. and others . Tri-Scott. a new kinematic structure for a 6-dof decoupled parallel manipulator. In *Workshop: Fundamental issues and future directions for parallel mechanisms and manipulators*, pages 12–15, Québec, October, 3-4, 2002.
- [644] Zabalza I. and others . A variant of a 6-RKS Hunt-type parallel manipulator to easily use insensitivity position configurations. In *ARK*, pages 291–300, Caldes de Malavalla, June 29- July 2, 2002.
- [645] Zamanov V.B and Sotirov Z.M. Structures and kinematics of parallel topology manipulating systems. In *Proc. Int. Symp. on Design and Synthesis*, pages 453–458, Tokyo, July, 11-13, 1984.
- [646] Zamanov V.B and Sotirov Z.M. A contribution to the serial and parallel manipulator duality. In *8th World Congress on the Theory of Machine and Mechanisms*, pages 517–520, Prague, August, 26-31, 1991.
- [647] Zamanov V.B and Sotirov Z.M. Parallel manipulators in robotics. In *IMACS/SICE Int. Symp. on Robotics, Mechatronics, and Manufacturing Systems*, pages 409–418, Kobe, September, 16-20, 1992.
- [648] Zanganeh K.E. and Angeles J. Instantaneous kinematics and design of a novel redundant parallel manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 3043–3048, San Diego, May, 8-13, 1994.
- [649] Zanganeh K.E. and Angeles J. Kinematic isotropy and the optimum design of parallel manipulators. *Int. J. of Robotics Research*, 16(2):185–197, April 1997.
- [650] Zanganeh K.E., Sinatra R., and Angeles J. Kinematics and dynamics of a six-degree-of-freedom parallel manipulator with revolute legs. *Robotica*, 15(4):385–394, July - August , 1997.
- [651] Zeid A.A., Overholt J.L., and Beck R.R. Modeling of multibody systems for control using general purpose simulation languages. *Simulation*, 67(1):7–19, January 1994.
- [652] Zhang D. and Gosselin C.M. Kinetostatic modeling of N-DOF parallel mechanisms with a passive constraining leg and prismatic actuators. *ASME J. of Mechanical Design*, 123(3):375–384, September 2001.

- [653] Zhang D. and Gosselin C.M. Parallel kinematic machine design with kinetostatic model. *Robotica*, 20(4):429–438, July 2002.
- [654] Zhang D. and others . Optimum design of parallel kinematic tool-heads with genetic algorithm. *Robotica*, 22(1):77–84, January 2004.
- [655] Zhang M.D. and Song S.M. Study of three-degree-of-freedom parallel platforms for reactional compensation. In *ISRAM*, pages 373–378, Hawaiï, August, 14-18, 1994.
- [656] Zhang C-D. and Song S.M. Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions. In *IEEE Int. Conf. on Robotics and Automation*, pages 2676–2681, Sacramento, April, 11-14, 1991.
- [657] Zhang C-D. and Song S.M. A efficient method for inverse dynamics of manipulators based on the virtual work principle. *J. of Robotic Systems*, 10(5):605–627, July 1993.
- [658] Zhou K. and others . Singularity loci research on high speed travelling type of double four-rod spatial parallel mechanism. *Mechanism and Machine Theory*, 38(3):195–221, Mars July 2003.
- [659] Zhuang H. and Roth Z.S. Method for kinematic calibration of Stewart platforms. *J. of Robotic Systems*, 10(3):391–405, 1993.
- [660] Zhuang H. and Wang Y. A coordinate measuring machine with parallel mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 3256–3261, Albuquerque, April, 21-28, 1997.
- [661] Zhuang H. Self calibration of parallel mechanisms with a case study on Stewart platform. *IEEE Trans. on Robotics and Automation*, 13(3):387–397, June 1997.
- [662] Zhuang H., Yan J., and Masory O. Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. *J. of Robotic Systems*, 15(7):395–405, 1998.
- [663] Zlatanov D., Fenton R.G., and Benhabib B. A unifying framework for classification and interpretation of mechanism singularities. *ASME J. of Mechanical Design*, 117(4):566–572, December 1995.
- [664] Zlatanov D. and Gosselin C.M. A family of new parallel architectures with four degrees of freedom. In F.C. Park C.C. Iurascu, editor, *Computational Kinematics*, pages 57–66. EJCK, May, 20-22, 2001.
- [665] Zlatanov D., Bonev I.A., and Gosselin C.M. Constraint singularities as configuration space singularities. September, 6, 2001, <http://www.parallemic.org/Reviews/Review008.html>.
- [666] Zlatanov D., Bonev I.A., and Gosselin C.M. Constraint singularities. July, 5, 2001, www.parallemic.org/Reviews/Review005.html.
- [667] Zlatanov D., Bonev I.A., and Gosselin C.M. Constraint singularities as configuration space singularities. In *ARK*, Caldes de Malavalla, June 29- July 2, 2002.

- [668] Zobel P.B., Di Stefano P., and Raparelli T. The design of a 3 dof parallel robot with pneumatic drives. In *27th ISIR*, pages 707–710, Milan, October, 6-8, 1996.
- [669] Zoppi M., Bruzzone L.E., and Molfino R.M. A novel 5-dof inter-connected-chains PKM for manufacturing revolute surfaces. In *4th Chemnitzer Parallelkinematik Seminar*, Chemnitz, April, 20-21, 2004.

Index

A

A^* , 253
AACTS, 68
ABB, 2, 31, 34
absolute conic, 107
acceleration
 active wrist, 174
 angular, 173
 cartesian, 173
 joint, 173
 maximal, 173
 6-PUS, 174
 6-UPS, 174
accommodation matrix, 274
accuracy, 163, 175–177
 absolute, 3
 indices, 165
 parallel robot, 171
 repeatability, 3
accuracy point, 250
active stiffness, 267
active wrist INRIA, 51, 93
actuator, 30
 double linear, 57
 electrical, 49
 hydraulic, 66, 75
 linear, 5
 magnetostrictive, 75
 piezo-electric, 86
 pneumatic, 49, 274
 rotary-linear, 57
 twisting, 68
Adept, 2
ADS, 71
agile eye, 36
 A_i , xvii
AI Group, 92
Airbus, 78, 91
Alio, 86
antenna, 71, 275
application, 40, 49, 50, 70–93
Appolo, 71

appropriate design, 303
architectural singularity, 181
architecture, 19, 27–62
area of workspace, 231
Argos, 37
articulated octahedron, 93
Artisan, 40
aspect, 210, 218
assembly mode, 105
Assur, 21
August, 68
auto-calibration, 290, 298

B

backlash, 4, 175
balancing, 40, 274–275
 dynamic, 279
ball-screw, 49
bandwidth, 74, 267, 277
barycentric coordinates, 139
base inertial parameters, 278
Bennet, 15
Bezout, 107, 127
 B_i , xvii
binary robot, 64
Boeing, 71
bond-graphs, 302
bone, 77
Borel, 5, 208
box, 325
BRAID, 65
Bricard, 5, 208
building, 68
Burmester, 128

C

CAE, 10
calibration, 289–300
 auto, 290, 298
 constrained, 290
 external, 290
 Hexa robot, 300

- CaPaMan, 43
 Cappel, 10
 car painting, 32
 Cardan, 15
 Caren, 77, 79
 carpal wrist, 40
 Cauchy, 5
 Cayley, 113
 cell, 250–252
 centrifugal, 278
 CERT-DERA, 49
 chain
 generic, 111
 PUS, 51
 RRP_aR, 31
 RRPS, 34
 RUS, 52
 spherical, 36
 UPS, 48
 characteristic length, 166
 characteristic tetrahedron, 192
 Charlotte, 71
 Chasles, 208
 Cinaxe, 93
 circularity, 107
 4-bar mechanism, 106
 RSSR, 114
 CKCM, 71
 CLDK algorithm, 145
 clearance, 3, 171, 175, 191, 209
 closed-loop, 5
 closure equation, 162
 CMW, 84
 CNRS, 84
 Comau, 34, 82
 complex, 188
 general, 188
 non singular, 188
 singular, 188
 compliance, 11, 259, 267
 active, 12, 269
 matrix, 267
 passive, 12, 269
 computation time
 direct kinematics, 143
 dynamics, 286
 jacobian, 163
 computational geometry, 216
 condition number, **165**, 205, 209, 307
 conditioning, 163–171, 204–206
 congruence, 188
 degenerate, 188
 elliptic, 188
 hyperbolic, 188
 parabolic, 188
 conjugacy, 23
 connection degree, 4
 constrained calibration, 290
 constraint singularity, **181**, 191
 container, 68
 continuation, 145, 218, 321
 control
 direct kinematics, 140
 error, 163
 force-feedback, 269
 instability, 270
 PID, 277
 position, 95
 singularity, 181, 209
 stiffness, 274
 velocity, 105, 155
 convexity
 and singularity, 185
 direct kinematics, 121
 cooperation, 12
 coordinates
 generalized, xvii, 1
 homogeneous planar, 107
 joint, xvii, 12
 Study, 95
 Copra, 91
 Coriolis, 8, 278
 cost function, 305
 counterweight, 275
 coupler, 106
 coupler curve, 106, 128
 intersection, 224
 crane, 68, 289
 Crigos, 75
 CSA, 74
 CSEM, 71
 cylindrical motion, 43
- D**

 DARTS, 68
 DBB, 293, 296
 decoupled robot, 51, 59–62, 124
 definition, 12
 degree of connection, 4
 degree of freedom, 1

- in singularity, 201
- Delta, 31
 - acceleration, 173
 - applications, 88
 - calibration, 300
 - Cube, 32
 - design, 305
 - direct kinematics, 112
 - inverse dynamics, 288
 - linear, 32
 - singularity, 211
- DeltaLab, 93
- Demaurex, 31
- design, 301–320
 - appropriate, 303
 - examples, 318
 - 4-bar mechanism, 308
 - joint velocity, 311
 - methodology, 313
 - optimal, 302
 - planar robot, 309
 - spherical robot, 309
- determinant
 - inverse jacobian, 211
 - inverse jacobian SSM, 163
- dexterity, 163
 - index, 165
- dextrous workspace, **214**, 222, 240
 - controllably, 214
- direct dynamics, 277, 278
- direct kinematics, 105–149
 - 4-4 robot, 126
 - 5-3 robot, 150
 - 5-4 robot, 124
 - 5-5 robot, 124
 - 6-3 robot, 124
 - 6-4 robot, 123
 - 6-5 robot, 123
 - 6-UPS robot, 128
 - 7-7 robot, 133
 - 8-8 robot, 133
 - 9-9 robot, 132
 - accuracy, 136, 148
 - active wrist, 121
 - computation time, 143
 - convergence, 140
 - Delta, 112
 - Gröbner basis, 129
 - Hexa, 151
 - INRIA wrist, 120
 - interval analysis, 129
 - iterative methods, 136
 - MSSM, 121
 - noise, 144
 - numerical methods, 136
 - parallel implementation, 139
 - polynomial form, 108
 - PPP-3S robot, 122
 - PPR-3S robot, 122
 - PRR-3S robot, 122
 - real-time, 140
 - rotation wrist, 130
 - 6-PUS robot, 121
 - SSM, 123, 127
 - Stewart platform, 121
 - TSSM, 114
 - W0 robot, 150
 - with extra sensors, 145
 - Zhang, 127
- direct tolerance, 26
- direct-drive, 3
- dish, 71
- Disney, 92
- distance
 - between links, 237
 - Lagrange identity, 121
 - safety, 238
 - to a singularity, 204–206
- divergence, 231
- DNAT, 65
- Dockwelder, 40
- double tripod, 55
- dual quaternions, 128
- duality, 119
- Dymo, 16, 38
- dynamic balancing, 279
- dynamics, 277–288
 - computation time, 286
 - direct, 278
 - examples, 286
 - Hamilton, 278
 - index, 279
 - inverse, 278
 - Lagrange, 278
 - Newton-Euler, 278
 - parameters identification, 277
 - virtual work, 278
- Dynamil, 86

E

earthquake, 8, 93
 EasyTeach, 68
 Eclipse, 59
 8-8 robot, 133
 electrical actuator, 49
 elimination, 128, 322
 ellipsoid
 flexibility, 274
 force, 261
 manipulability, 163
 resistivity, 263
 end-effector, 1
 endoscope, 77
 Energen, 75
 energy, 7
 EPFL, 31
 Epson, 2
 equilibrium, 181, 275, 282
 equivalent mechanism, 113
 TSSM, 113
 ESRF, 86, 318
 Euler angles, xvii
 EX 800, 93
 exact synthesis, 309
 external calibration, 290
 extremum
 joint coordinates, 103
 joint force, 260
 joint velocities, 173
 stiffness, 273
 twist, 172
 wrench, 260

F

\mathcal{F} , xviii
 F-200i, 88
 failure, 62, 320
 Falcon, 68
 Fanuc, 2, 68, 88
 FCS, 91
 FGB, 323
 FIKP, 153
 5-5 robot, 124
 5-4 robot, 124
 flexibility ellipsoid, 274
 flexible
 joint, 30, 50, 54
 link, 44, 260

FlexPicker, 31
 flexure hinge, 29, 30, 32
 flight simulator, 8, 44, 77–78
 Flight-Avionics, 92
 Flying Carpet, 68
 food, 90
 force,
 see **joint or wrench**
 force ellipsoid, 261
 force sensor, 266
 force transmission index, 265
 force-feedback, 269
 4-bar mechanism, 42, 54, 106
 circularity, 106
 coupler curve, 106
 design, 308
 4-4 robot, 126
 Freudenstein, 21
 friction, 4, 171
 ftp, xv
 F206, 66, 86
 full inverse jacobian, 155

G

Gadfly, 50
 Gauss, 231
 GCI index, 170, 307
 GEC, 34
 generalized coordinates, xvii
 generalized polygon, 235
 geometry
 computational, 216
 Grassmann, 185
 Georg V, 34
 Giddings & Levis, 80
 global conditioning index, 169
 Goddard, 71
 Goldberg, 15
 Googol, 28
 Gough, 5
 Grübler, 14
 graph
 theory, 21
 visibility, 254
 Grassmann, 186
 Grassmann geometry, 185–189
 congruence, 188
 line, 187
 linear complex, 188
 plane, 187

point, 187
 Grassmann-Cayley, 156
 gravity, 71, 176
 Gröbner basis, 127, 129, 323
 group
 motion, 21
 theory, 23
 translations, 22

H

Hagenbuch, 75
 Half, 43
 Hana, 43
 haptic device, 91
 HCCM, 90
 Hephaist Seiko, 28, 30, 86
 Hexa, 53
 calibration, 300
 direct kinematics, 151
 singularity, 212
 workspace, 257
 Hexabot, 88
 Hexaglide, 51
 Hexamove, 75
 hexapod, 48
 Hexel, 88
 H4, 44
 hinge, 29, 30
 HITA-STT, 44
 homogeneous coordinates, 107
 homotopy, 128, 321
 horse riding, 79
 HP1, 34
 HR 4, 86
 Hughes Stx, 58
 Hunt, 192
 hydraulic actuator, 66, 75, 78
 control, 78
 hyperboloid, 187

I

identification
 dynamics, 277
 identification jacobian, 294
 I4, 44
 calibration, 296
 IIKP, 153
 Ilizarov, 77
 ILL, 88

imaginary circle, 107
 imaginary circular points, 107
 implicit loop formulation, 294
 INA, 30
 inclusive orientation workspace, 214,
 226
 increased instantaneous mobility, 180
 index
 conditioning, 163
 dexterity, 165
 force transmission, 265
 global conditioning, 169
 manipulability, 165
 mobility, 14
 parallelism, 93
 singularity, 204
 workspace, 245
 inertia, 4, 279
 infinitesimal motion, 180, 202-204
 inflation, 142
 INRIA active wrist, 51
 instantaneous rotation axis, 202
 interference between links, 220, 237
 internal sensor, 3
 intersection
 coupler curve, 224
 joint, 22
 links, 220, 237
 interval analysis, 129, 171, 207, 263,
 314, 325-330
 inverse dynamics, 277, 278
 inverse jacobian, 153
 determinant, 163
 Euler angles, 155
 full, 155
 kinematic, 156
 overall, 155
 6-PUS, 161
 6-UPS, 160
 3-PUS, 159
 3-UPU, 158
 inverse kinematics, 95-102
 6-PUS, 100
 6-RUS, 101
 6-UPS, 99
 3-UPU, 98, 104
 inverse tolerance, 26
 IRA, 202
 IRB 340, 31
 IRB940, 34

Iron cross, 8
 ISIS, 75
 iso-stiffness, 270
 curve, 270
 surface, 273
 isotropic pose, 169
 isotropic robot, 169
 isotropy, 163
 ITER, 47

J

jacobian, 162
 analytic formulation, 163
 identification, 294
 inverse, 153
 iterative, 171
 kinematic, 162
 number of terms, 163
 of a SSM, 162
 practical computation, 163
 joint
 Cardan, 15
 composition, 22
 deformable, 30
 direction influence, 236
 flexible, 30, 50, 54
 layout, 244, 302
 mechanical limits, 220, 233, 305
 modeling, 234
 parallelogram, 59
 prismatic, 1
 revolute, 1
 S , 29, 61, 234
 double, 50
 multiple, 30
 skew axis, 30
 triple, 51, 61
 U , 15
 joint acceleration, 173
 joint coordinates, xvii, 12, 95
 calculation, 95
 extremum, 103
 joint force, 259
 extremum, 260, 261
 extremum in a pose, 261
 in a singularity, 211
 iterative scheme, 260
 \leftarrow wrench, 259
 \rightarrow wrench, 259
 joint velocities, 171

\rightarrow twist, 171
 in design, 311
 \leftarrow twist, 156, 171

joystick, 91

K

Kantorovitch, 141
 kinematic branches, 182
 kinematic chain
 closed-loop, 5
 open-loop, 5
 simple, 5
 kinematic mapping, 95
 kinematics,
 see **inverse kinematics**,
 see **direct kinematics**
 kinematics polyhedron, 164
 kinetostatic indices, 265
 Kuka, 2

L

LAAS, 115
 Lagrange, 121, 280
 multiplier, 280
 landing gear, 71
 Lebesgue, 5
 left hand
 INRIA, 50
 stiffness, 270
 workspace, 230
 Lie group, 21
 Limbro, 55
 Linapod, 32, 51
 line
 Plücker vector, 185
 skew, 187
 linear actuator, 7
 linear complex, 188
 linear Delta, 32
 link
 beam model, 268
 distance, 237
 elastic model, 267
 flexible, 44, 260
 inertia, 279
 interference, 220, 237
 LIRMM, 52, 62
 LM, 71
 LME, 86

Logabex, 63
 lumped model, 260
 lunar module, 71

M

Mach21, 86
 machine-tool, 80–86
 CMW 380, 84
 Comau Urane SX, 82
 Cross Hüller Genius 500, 82
 DiGiHex, 83
 DR Mader, 84
 DS Technology Sprint Z3, 82
 Eclipse, 59
 Greif, 84
 Hexaglide, 51
 Index V100, 82
 Ingersoll, 84
 Kovosvit Mas Trijoint 900H, 82
 Krauseco/Mauser HS500, 82
 Metrom P800
 P2000, 83
 Mikromat 6X, 84
 Multicraft 560, 84
 Octahedral Hexapod, 84
 Okuma PM-600, 84
 Orthoglide, 33
 Reichenbacher Pegasus, 84
 Savelovo Hexamech-1, 84
 SMT Tricept, 34
 Starrag-Heckert SKM 400, 82
 Triax, 83
 Variax, 80
 workspace, 242
 manipulability ellipsoid, 163
 manipulability index, **165**, 205, 308
 manufacturing tolerance, 26, 175
 Map, 302
 MARS, 75
 MAST, 7
 mastication, 77
 matrix
 inertia, 281
 inverse jacobian, 153, 171
 jacobian, 162
 norm, 165
 rotation, xvii
 stiffness, 267
 maximal workspace, **214**, 223, 226,
 240

Mazor, 75
 MBARS, 76
 measuring machine, 68, 90
 mechanical limits, 220, 233
 mechanism
 Bennet, 15
 equivalent, 113
 4-bar, 106
 Goldberg, 15
 redundant, 12
 medical, 40, 75–77, 256
 MEL, 236
 MEMS, 66
 M-850, 86
 M-840, 86
 Micos, 86
 micro-positioning, 66
 micro-robot, 4, 50, 56
 dynamics, 279
 joints, 30
 workspace, 218
 MicroMega, 74
 microscope, 75
 milling machine, 10, 80
 minimal kinematic set, 154
 Mips, 40, 77
 mobility, 43
 formula, 14
 index, 14
 modular robots, 50
 calibration, 289
 Moog, 92
 Motek, 77, 79
 motion
 infinitesimal, 180
 parasitic, 20
 Schönflies, 22, 44
 motion group, 21
 generator, 22
 motion planning, 249–257
 movie theater, 93
 MPE, 170
 MSSM, **93**
 direct kinematics, 114, 121
 dynamics, 279
 singularity, 191
 munition loader, 93

N

Nabla 6, 51, 61

direct kinematics, 151
 NADS, 78
 NAF3, 28, 86
 nanopod, 133
 NAOS, 71
 NASA, 36, 43, 71
 natural length, 166
 Neos Robotics, 34
 Newton method, 136–141
 Ninja, 58
 NIST, 68
 noise amplification index, 300
 nominal load

- parallel robot, 7
- serial robot, 2

 notation, xvii

O

observability, 291
 observation matrix, 291
 octahedra, 43
 octopod, 75
 off-shore, 10
 Omega, 91
 Ω , xvii
 Omni-Wrist, 39
 open-loop, 5
 operating point, 153, 213, 259
 optics, 86, 87
 optimal design, 302
 orientation

- interval, 226, 227
- representation, xvii, 129, 214
- workspace, 214, 239

 Orion, 40
 Orthoglide, 33, 208
 overall inverse jacobian, 155
 overconstrained, 14, 39

P

packaging, 90, 91
 palm tree, 93
 Pantoscope, 37
 paradoxical, 15, 24
 parallelism index, 13
 parallelogram, 31, 57, 59, 145, 275
 parasitic motion, 20, 26
 Paros, 86
 part positioning, 256

partitioning, 210
 passive stiffness, 267
 path planning, 209, 249–257
 Persival, 79
 PH1, 74
 PHEX1, 74
 ϕ , xvii
 Phoenix, 91
 Physik Instrumente, 86
 pick-and-place, 11, 50, 90, 277
 piezo-electric actuator, 50
 pitch, 202, 203
 Plücker vector, 185

- normalized, 185

 planar robot, 27

- 3D workspace, 251
- architecture, 27
- balancing, 275
- design, 309, 320
- determinant inverse jacobian, 211
- dextrous workspace, 222
- direct kinematics, 105
- generic chain, 111
- inverse jacobian, 157
- inverse kinematics, 97
- maximal workspace, 223
- mobility, 14
- motion planning, 251
- redundant, 29
- singularity, 189
- stiffness matrix, 267
- 3-PPR, 103
- 3-PRP, 104
- 3-PRR, 28, 257
- 3-RPP, 103
- 3-RPR, 28, 189, 219, 221–223
- 3-RRP, 103
- 3-RRR, 28, 190, 257

 pneumatic actuator, 49, 274

- compliance, 274

 pointing, 36
 Pollard, 11
 polygon

- generalized, 235

 polyhedron

- kinematics, 163
- wrench, 263

 position control, 96
 positioning device, 86–88, 318
 post optimality analysis, 307

PPP-3S, 122
 PPR-3S, 122
 PRR-3S, 122
 PRRS, 22
 ψ , xvii
 PSSR, 122
 PUS, 51
 pyramid, 234

Q

quaternions, 129, 250
 dual, 128

R

R , xvii
 rank theorem, 182
 RCC, 269
 reachable workspace, **214**, 223, 226,
 240
 real-time, 140
 reciprocal screw, 157
 reciprocity, 24
 reconfigurable robot, 50
 reconstruction, 251
 reduced total orientation workspace,
 214
 redundancy, 12, 29, 33, 37, 55, 62,
 210, 298
 actuation, 62
 kinematic, 62
 measurement, 62
 regulus, 187
 complementary, 187
 rehabilitation, 77, 79
 reliability, 320
 repeatability, 3, 86
 resistivity ellipsoid, 263
 resultant, 116, 322
 Rexroth Hydrauline, 92
 roadmap, 250, 254
 Robea, 84
 Robocrane, 68
 robot
 binary, 64
 decoupled,
 see **decoupled robot**
 fast parallel, 52
 5 d.o.f., 45–47
 4 d.o.f., 43–44

fully parallel, 13
 general parallel, 12
 hybrid, 34
 isotropic, 169
 modular, 50
 movie theater, 93
 overconstrained, 14
 parallel, 13
 parallel fast, 31
 planar,
 see **planar robot**
 PRR-3S,
 see **active wrist**
 reconfigurable, 50
 redundant, 29, 33, 37, 55, 62
 serial, 1
 6 d.o.f., 48–62
 spherical,
 see **spherical robot**
 3 d.o.f., 27–29, 31–43
 wires,
 see **wire robot**

Robotool, 86

ρ_i , xvii

ρ_{max} , xvii

Romed, 76

ρ_{min} , xvii

rotation

 matrix, xvii
 parameters, xvii, 129
 quaternions, 129
 representation, 214

Rotobot, 53

RPRS, 22

RRPS, 22, 34

RRR-3S, 122

RRRS, 22

RSSP, 122

RSSR, 113

 circularity, 114

RUS, 52

S

Sacso, 69

safety distance, 238

SAGE III, 73

SALSA, 88, 318

sampling, 140, 170, 217, 286

Scara, 2

Schönflies, 22, 44

- screening, 91
- screw, 24, 43, 185, 203, 250
- search space, 313
- Segesta, 68
- Seiko, 2
- self-calibration, 290
- self-motion, 209
- sensor, 3
 - accuracy, 175
 - extra, 145
 - force, 266
 - internal, 3
 - layout, 146
- serial robot, 1
- Servos Simulation, 53
- 7-7 robot, 133
- sextic, 106, 108, 223, 224
- shake table, 7
- shakiness, 209
- Sheldon, 82, 301
- ship loading, 68
- Sikorsky, 10
- simulation, 301, 302
- simulator, 11, 77–79
 - flight, 8, 44, 77
- singularity, 179
 - and convexity, 185
 - architectural, 181, 208
 - aspect, 210
 - avoidance, 209
 - classification, 208
 - constraint, 181
 - control, 181, 209
 - definition, 182
 - degree of freedom, 201
 - distance to, 204
 - Hexa, 212
 - in a volume, 206
 - indices, 204
 - joint forces, 211
 - MSSM, 191
 - partitioning, 210
 - path planning, 209
 - permanent, 86, 208
 - planar robot, 189
 - redundant input, 180
 - redundant output, 181
 - redundant passive motion, 181
 - search for, 206
 - stratification, 208
 - structural, 208
- singularity-free
 - design, 306
 - region, 185, 218
 - trajectory, 135, 209
 - workspace, 206
- 6-5 robot, 123
- 6-4 robot, 123
- 6-PUS robot
 - acceleration, 174
 - accuracy, 175
 - direct kinematics, 121
 - dynamics, 285
 - INRIA, 51
 - inverse jacobian, 161
 - inverse kinematics, 100
 - stiffness, 268, 270
 - velocities, 161, 178
 - workspace, 257
- 6-RUS robot
 - inverse kinematics, 101
- 6-3 robot, 124
- 6-UPS robot, 48
 - acceleration, 174, 178
 - accuracy, 175
 - balancing, 275
 - determinant inverse jacobian, 185
 - direct kinematics, 128
 - dynamics, 278, 281
 - extremum of the stiffness, 273
 - inverse jacobian, 155, 160
 - inverse kinematics, 99
 - iso-stiffness, 272
 - jacobian, 162
 - joint forces, 261
 - maximal load, 275
 - singularity, 206
 - slender, 272
 - stiffness, 267
 - velocities, 160
 - workspace, 229, 231, 239
- skew line, 187
- SkyCam, 69
- Smaps, 302
- Smartee, 58
- Space, 49
- space shuttle, 40
- SpaceFab, 86
- spatial application, 71–73
- Speed-R-Man, 33

spherical robot, 36
 design, 309
 direct kinematics, 124
 inverse kinematics, 104
 redundant, 37
 Spine Assist, 75
 spreaddand, 30
 SSM, 93
 Star, 32, 37
 Star Tours, 92
 static, 259–266
 Stewart, 8
 Stewart platform, 8, 10
 direct kinematics, 121
 stick and slip, 56
 stiffness, 267–274
 active, 267
 control, 274
 extremum, 273
 map, 270
 matrix, 267
 passive, 267
 stiffness matrix, 267–268
 Study, 95
 Sturm, 111
 Surgiscope, 75
 SWF6, 86
 synchrotron, 86
 synthesis
 architectural, 19–26
 dimensional,
 see **design**
 Synthetica, 302, 310

T

TACOM, 78
 Taguchi, 306
 τ , xviii
 telescope, 71, 318
 θ , xvii
 Tetrabot, 34
 tetrahedron, 251
 thermal, 176, 319
 Thomson, 78
 3-CRR robot, 33
 3-PPR planar robot
 inverse kinematics, 103
 3-PRP planar robot
 inverse kinematics, 104
 3-PRR planar robot
 dextrous workspace, 257
 orientation workspace, 257
 workspace, 257
 3-PUS robot
 inverse jacobian, 159
 3-RPP planar robot
 inverse kinematics, 103
 3-RPR planar robot, 28
 design, 309
 dextrous workspace, 222
 direct kinematics, 108
 inverse jacobian, 157
 joint forces, 276
 maximal workspace, 223
 singularity, 189, 211
 stiffness matrix, 267
 workspace, 219, 221
 3-RRP planar robot
 inverse kinematics, 103
 3-RRR planar robot, 28
 dextrous workspace, 257
 maximal workspace, 257
 orientation workspace, 257
 singularity, 190
 velocities, 177
 workspace, 257
 3-RS, 113, 119
 3-UPU robot, 34
 accuracy, 176
 direct kinematics, 113
 inverse jacobian, 158
 inverse kinematics, 98
 optimal design, 305
 singularity, 190
 workspace, 228
 3-URU robot, 37
 tiling, 253
 tilt and torsion, 214
 TMBS, 79
 tolerance, 175, 293
 Toro, 50
 Toshiba, 2
 total orientation workspace, 214, 227
 training, 93
 trajectory
 interference between links, 248
 mechanical limits, 248
 planning,
 see **motion planning**
 singularity-free, 209

trajectory verification, 246–249, 258
 constant orientation, 246
 examples, 248
 line segment, 246
 translation workspace, 213
 translator, 31
 tree, 279
 Triax, 83
 Tricept, 34
 tricircular, 106
 tripod, 55
 truss, 63–66
 inverse kinematics, 104
 TSSM, 93
 direct kinematics, 114
 standard, 117
 with 16 assembly modes, 117
 Turin robot, 58
 Twice, 44
 twist, 153
 extremum, 172
 2-Delta, 61

U

uncertain configuration, 180
 unit sphere, 239
 Universal Rig, 5
 UPS, 48

V

Vaillant, 5, 208
 Variax, 80, 301
 velocity, 153
 velocity control, 105, 155
 velocity twist, 24
 Vertex, 71
 VES, 71
 VGT, 63
 vibration, 7, 68, 74–75, 86
 Virtogo, 92
 virtual reality, 69, 77
 virtual work, 278
 visibility graph, 254
 VISS, 74, 75
 VMS, 36
 volume of workspace, 231

W

W, xvii

waste, 68
 Web, xiv
 Weierstrass substitution, 102
 wire robot, 43, 68–69
 calibration, 289
 direct kinematics, 150
 singularity, 181
 stiffness, 267
 workspace, 218
 workspace, 213–244
 3-RPR planar robot, 219
 constant orientation, 219, 229
 controllably dextrous, 214
 cross-section area, 231
 dextrous, 214, 222, 240
 enlargement, 209
 examples, 230
 inclusive orientation, 214, 226
 index, 245
 interference between links, 220, 237
 maximal, 223, 226, 242
 orientation, 214, 221, 239
 reachable, 214, 223
 reduced total orientation, 214
 3D, 231, 251
 total orientation, 214, 227
 translation, 213
 types, 213
 volume, 231
 Wren, 5
 wrench, 24, 259
 extremum, 260, 265
 ← joint force, 259
 → joint force, 259
 wrench polyhedron, 263
 wrist
 active, 93
 carpal, 40
 INRIA active, 51, 120
 spherical, 36

X

X, xvii
 $\{X(\mathbf{w})\}$, 22

Z

zero-gravity, 71

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

Aims and Scope of the Series

The fundamental questions arising in mechanics are: *Why?*, *How?*, and *How much?* The aim of this series is to provide lucid accounts written by authoritative researchers giving vision and insight in answering these questions on the subject of mechanics as it relates to solids. The scope of the series covers the entire spectrum of solid mechanics. Thus it includes the foundation of mechanics; variational formulations; computational mechanics; statics, kinematics and dynamics of rigid and elastic bodies; vibrations of solids and structures; dynamical systems and chaos; the theories of elasticity, plasticity and viscoelasticity; composite materials; rods, beams, shells and membranes; structural control and stability; soils, rocks and geomechanics; fracture; tribology; experimental mechanics; biomechanics and machine design.

1. R.T. Haftka, Z. Gürdal and M.P. Kamat: *Elements of Structural Optimization*. 2nd rev.ed., 1990
ISBN 0-7923-0608-2
2. J.J. Kalker: *Three-Dimensional Elastic Bodies in Rolling Contact*. 1990 ISBN 0-7923-0712-7
3. P. Karasudhi: *Foundations of Solid Mechanics*. 1991 ISBN 0-7923-0772-0
4. *Not published*
5. *Not published*.
6. J.F. Doyle: *Static and Dynamic Analysis of Structures*. With an Emphasis on Mechanics and Computer Matrix Methods. 1991 ISBN 0-7923-1124-8; Pb 0-7923-1208-2
7. O.O. Ochoa and J.N. Reddy: *Finite Element Analysis of Composite Laminates*.
ISBN 0-7923-1125-6
8. M.H. Aliabadi and D.P. Rooke: *Numerical Fracture Mechanics*. ISBN 0-7923-1175-2
9. J. Angeles and C.S. López-Cajún: *Optimization of Cam Mechanisms*. 1991
ISBN 0-7923-1355-0
10. D.E. Grierson, A. Franchi and P. Riva (eds.): *Progress in Structural Engineering*. 1991
ISBN 0-7923-1396-8
11. R.T. Haftka and Z. Gürdal: *Elements of Structural Optimization*. 3rd rev. and exp. ed. 1992
ISBN 0-7923-1504-9; Pb 0-7923-1505-7
12. J.R. Barber: *Elasticity*. 1992 ISBN 0-7923-1609-6; Pb 0-7923-1610-X
13. H.S. Tzou and G.L. Anderson (eds.): *Intelligent Structural Systems*. 1992
ISBN 0-7923-1920-6
14. E.E. Gdoutos: *Fracture Mechanics*. An Introduction. 1993 ISBN 0-7923-1932-X
15. J.P. Ward: *Solid Mechanics*. An Introduction. 1992 ISBN 0-7923-1949-4
16. M. Farshad: *Design and Analysis of Shell Structures*. 1992 ISBN 0-7923-1950-8
17. H.S. Tzou and T. Fukuda (eds.): *Precision Sensors, Actuators and Systems*. 1992
ISBN 0-7923-2015-8
18. J.R. Vinson: *The Behavior of Shells Composed of Isotropic and Composite Materials*. 1993
ISBN 0-7923-2113-8
19. H.S. Tzou: *Piezoelectric Shells*. Distributed Sensing and Control of Continua. 1993
ISBN 0-7923-2186-3
20. W. Schiehlen (ed.): *Advanced Multibody System Dynamics*. Simulation and Software Tools. 1993
ISBN 0-7923-2192-8
21. C.-W. Lee: *Vibration Analysis of Rotors*. 1993 ISBN 0-7923-2300-9
22. D.R. Smith: *An Introduction to Continuum Mechanics*. 1993 ISBN 0-7923-2454-4
23. G.M.L. Gladwell: *Inverse Problems in Scattering*. An Introduction. 1993 ISBN 0-7923-2478-1

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

24. G. Prathap: *The Finite Element Method in Structural Mechanics*. 1993 ISBN 0-7923-2492-7
25. J. Herskovits (ed.): *Advances in Structural Optimization*. 1995 ISBN 0-7923-2510-9
26. M.A. González-Palacios and J. Angeles: *Cam Synthesis*. 1993 ISBN 0-7923-2536-2
27. W.S. Hall: *The Boundary Element Method*. 1993 ISBN 0-7923-2580-X
28. J. Angeles, G. Hommel and P. Kovács (eds.): *Computational Kinematics*. 1993
ISBN 0-7923-2585-0
29. A. Curnier: *Computational Methods in Solid Mechanics*. 1994 ISBN 0-7923-2761-6
30. D.A. Hills and D. Nowell: *Mechanics of Fretting Fatigue*. 1994 ISBN 0-7923-2866-3
31. B. Tabarrok and F.P.J. Rimrott: *Variational Methods and Complementary Formulations in Dynamics*. 1994 ISBN 0-7923-2923-6
32. E.H. Dowell (ed.), E.F. Crawley, H.C. Curtiss Jr., D.A. Peters, R. H. Scanlan and F. Sisto: *A Modern Course in Aeroelasticity*. Third Revised and Enlarged Edition. 1995
ISBN 0-7923-2788-8; Pb: 0-7923-2789-6
33. A. Preumont: *Random Vibration and Spectral Analysis*. 1994 ISBN 0-7923-3036-6
34. J.N. Reddy (ed.): *Mechanics of Composite Materials*. Selected works of Nicholas J. Pagano. 1994
ISBN 0-7923-3041-2
35. A.P.S. Selvadurai (ed.): *Mechanics of Poroelastic Media*. 1996 ISBN 0-7923-3329-2
36. Z. Mróz, D. Weichert, S. Dorosz (eds.): *Inelastic Behaviour of Structures under Variable Loads*. 1995
ISBN 0-7923-3397-7
37. R. Pyrz (ed.): *IUTAM Symposium on Microstructure-Property Interactions in Composite Materials*. Proceedings of the IUTAM Symposium held in Aalborg, Denmark. 1995
ISBN 0-7923-3427-2
38. M.I. Friswell and J.E. Mottershead: *Finite Element Model Updating in Structural Dynamics*. 1995
ISBN 0-7923-3431-0
39. D.F. Parker and A.H. England (eds.): *IUTAM Symposium on Anisotropy, Inhomogeneity and Nonlinearity in Solid Mechanics*. Proceedings of the IUTAM Symposium held in Nottingham, U.K. 1995
ISBN 0-7923-3594-5
40. J.-P. Merlet and B. Ravani (eds.): *Computational Kinematics '95*. 1995 ISBN 0-7923-3673-9
41. L.P. Lebedev, I.I. Vorovich and G.M.L. Gladwell: *Functional Analysis*. Applications in Mechanics and Inverse Problems. 1996
ISBN 0-7923-3849-9
42. J. Menčík: *Mechanics of Components with Treated or Coated Surfaces*. 1996
ISBN 0-7923-3700-X
43. D. Bestle and W. Schiehlen (eds.): *IUTAM Symposium on Optimization of Mechanical Systems*. Proceedings of the IUTAM Symposium held in Stuttgart, Germany. 1996
ISBN 0-7923-3830-8
44. D.A. Hills, P.A. Kelly, D.N. Dai and A.M. Korsunsky: *Solution of Crack Problems*. The Distributed Dislocation Technique. 1996
ISBN 0-7923-3848-0
45. V.A. Squire, R.J. Hosking, A.D. Kerr and P.J. Langhorne: *Moving Loads on Ice Plates*. 1996
ISBN 0-7923-3953-3
46. A. Pineau and A. Zaoui (eds.): *IUTAM Symposium on Micromechanics of Plasticity and Damage of Multiphase Materials*. Proceedings of the IUTAM Symposium held in Sèvres, Paris, France. 1996
ISBN 0-7923-4188-0
47. A. Naess and S. Krenk (eds.): *IUTAM Symposium on Advances in Nonlinear Stochastic Mechanics*. Proceedings of the IUTAM Symposium held in Trondheim, Norway. 1996
ISBN 0-7923-4193-7
48. D. Ieşan and A. Scalia: *Thermoelastic Deformations*. 1996
ISBN 0-7923-4230-5

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

49. J.R. Willis (ed.): *IUTAM Symposium on Nonlinear Analysis of Fracture*. Proceedings of the IUTAM Symposium held in Cambridge, U.K. 1997 ISBN 0-7923-4378-6
50. A. Preumont: *Vibration Control of Active Structures*. An Introduction. 1997 ISBN 0-7923-4392-1
51. G.P. Cherepanov: *Methods of Fracture Mechanics: Solid Matter Physics*. 1997 ISBN 0-7923-4408-1
52. D.H. van Campen (ed.): *IUTAM Symposium on Interaction between Dynamics and Control in Advanced Mechanical Systems*. Proceedings of the IUTAM Symposium held in Eindhoven, The Netherlands. 1997 ISBN 0-7923-4429-4
53. N.A. Fleck and A.C.F. Cocks (eds.): *IUTAM Symposium on Mechanics of Granular and Porous Materials*. Proceedings of the IUTAM Symposium held in Cambridge, U.K. 1997 ISBN 0-7923-4553-3
54. J. Roorda and N.K. Srivastava (eds.): *Trends in Structural Mechanics*. Theory, Practice, Education. 1997 ISBN 0-7923-4603-3
55. Yu.A. Mitropolskii and N. Van Dao: *Applied Asymptotic Methods in Nonlinear Oscillations*. 1997 ISBN 0-7923-4605-X
56. C. Guedes Soares (ed.): *Probabilistic Methods for Structural Design*. 1997 ISBN 0-7923-4670-X
57. D. François, A. Pineau and A. Zaoui: *Mechanical Behaviour of Materials*. Volume I: Elasticity and Plasticity. 1998 ISBN 0-7923-4894-X
58. D. François, A. Pineau and A. Zaoui: *Mechanical Behaviour of Materials*. Volume II: Viscoplasticity, Damage, Fracture and Contact Mechanics. 1998 ISBN 0-7923-4895-8
59. L.T. Tenek and J. Argyris: *Finite Element Analysis for Composite Structures*. 1998 ISBN 0-7923-4899-0
60. Y.A. Bahei-El-Din and G.J. Dvorak (eds.): *IUTAM Symposium on Transformation Problems in Composite and Active Materials*. Proceedings of the IUTAM Symposium held in Cairo, Egypt. 1998 ISBN 0-7923-5122-3
61. I.G. Goryacheva: *Contact Mechanics in Tribology*. 1998 ISBN 0-7923-5257-2
62. O.T. Bruhns and E. Stein (eds.): *IUTAM Symposium on Micro- and Macrostructural Aspects of Thermoplasticity*. Proceedings of the IUTAM Symposium held in Bochum, Germany. 1999 ISBN 0-7923-5265-3
63. F.C. Moon: *IUTAM Symposium on New Applications of Nonlinear and Chaotic Dynamics in Mechanics*. Proceedings of the IUTAM Symposium held in Ithaca, NY, USA. 1998 ISBN 0-7923-5276-9
64. R. Wang: *IUTAM Symposium on Rheology of Bodies with Defects*. Proceedings of the IUTAM Symposium held in Beijing, China. 1999 ISBN 0-7923-5297-1
65. Yu.I. Dimitrienko: *Thermomechanics of Composites under High Temperatures*. 1999 ISBN 0-7923-4899-0
66. P. Argoul, M. Frémond and Q.S. Nguyen (eds.): *IUTAM Symposium on Variations of Domains and Free-Boundary Problems in Solid Mechanics*. Proceedings of the IUTAM Symposium held in Paris, France. 1999 ISBN 0-7923-5450-8
67. F.J. Fahy and W.G. Price (eds.): *IUTAM Symposium on Statistical Energy Analysis*. Proceedings of the IUTAM Symposium held in Southampton, U.K. 1999 ISBN 0-7923-5457-5
68. H.A. Mang and F.G. Rammerstorfer (eds.): *IUTAM Symposium on Discretization Methods in Structural Mechanics*. Proceedings of the IUTAM Symposium held in Vienna, Austria. 1999 ISBN 0-7923-5591-1

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

69. P. Pedersen and M.P. Bendsøe (eds.): *IUTAM Symposium on Synthesis in Bio Solid Mechanics*. Proceedings of the IUTAM Symposium held in Copenhagen, Denmark. 1999
ISBN 0-7923-5615-2
70. S.K. Agrawal and B.C. Fabien: *Optimization of Dynamic Systems*. 1999
ISBN 0-7923-5681-0
71. A. Carpinteri: *Nonlinear Crack Models for Nonmetallic Materials*. 1999
ISBN 0-7923-5750-7
72. F. Pfeifer (ed.): *IUTAM Symposium on Unilateral Multibody Contacts*. Proceedings of the IUTAM Symposium held in Munich, Germany. 1999
ISBN 0-7923-6030-3
73. E. Lavendelis and M. Zakrzhevsky (eds.): *IUTAM/IFTToMM Symposium on Synthesis of Non-linear Dynamical Systems*. Proceedings of the IUTAM/IFTToMM Symposium held in Riga, Latvia. 2000
ISBN 0-7923-6106-7
74. J.-P. Merlet: *Parallel Robots*. 2000
ISBN 0-7923-6308-6
75. J.T. Pindera: *Techniques of Tomographic Isodyne Stress Analysis*. 2000
ISBN 0-7923-6388-4
76. G.A. Maugin, R. Drouot and F. Sidoroff (eds.): *Continuum Thermomechanics*. The Art and Science of Modelling Material Behaviour. 2000
ISBN 0-7923-6407-4
77. N. Van Dao and E.J. Kreuzer (eds.): *IUTAM Symposium on Recent Developments in Non-linear Oscillations of Mechanical Systems*. 2000
ISBN 0-7923-6470-8
78. S.D. Akbarov and A.N. Guz: *Mechanics of Curved Composites*. 2000
ISBN 0-7923-6477-5
79. M.B. Rubin: *Cosserat Theories: Shells, Rods and Points*. 2000
ISBN 0-7923-6489-9
80. S. Pellegrino and S.D. Guest (eds.): *IUTAM-IASS Symposium on Deployable Structures: Theory and Applications*. Proceedings of the IUTAM-IASS Symposium held in Cambridge, U.K., 6–9 September 1998. 2000
ISBN 0-7923-6516-X
81. A.D. Rosato and D.L. Blackmore (eds.): *IUTAM Symposium on Segregation in Granular Flows*. Proceedings of the IUTAM Symposium held in Cape May, NJ, U.S.A., June 5–10, 1999. 2000
ISBN 0-7923-6547-X
82. A. Lagarde (ed.): *IUTAM Symposium on Advanced Optical Methods and Applications in Solid Mechanics*. Proceedings of the IUTAM Symposium held in Futuroscope, Poitiers, France, August 31–September 4, 1998. 2000
ISBN 0-7923-6604-2
83. D. Weichert and G. Maier (eds.): *Inelastic Analysis of Structures under Variable Loads*. Theory and Engineering Applications. 2000
ISBN 0-7923-6645-X
84. T.-J. Chuang and J.W. Rudnicki (eds.): *Multiscale Deformation and Fracture in Materials and Structures*. The James R. Rice 60th Anniversary Volume. 2001
ISBN 0-7923-6718-9
85. S. Narayanan and R.N. Iyengar (eds.): *IUTAM Symposium on Nonlinearity and Stochastic Structural Dynamics*. Proceedings of the IUTAM Symposium held in Madras, Chennai, India, 4–8 January 1999
ISBN 0-7923-6733-2
86. S. Murakami and N. Ohno (eds.): *IUTAM Symposium on Creep in Structures*. Proceedings of the IUTAM Symposium held in Nagoya, Japan, 3–7 April 2000. 2001
ISBN 0-7923-6737-5
87. W. Ehlers (ed.): *IUTAM Symposium on Theoretical and Numerical Methods in Continuum Mechanics of Porous Materials*. Proceedings of the IUTAM Symposium held at the University of Stuttgart, Germany, September 5–10, 1999. 2001
ISBN 0-7923-6766-9
88. D. Durban, D. Givoli and J.G. Simmonds (eds.): *Advances in the Mechanis of Plates and Shells The Avinoam Libai Anniversary Volume*. 2001
ISBN 0-7923-6785-5
89. U. Gabbart and H.-S. Tzou (eds.): *IUTAM Symposium on Smart Structures and Structonic Systems*. Proceedings of the IUTAM Symposium held in Magdeburg, Germany, 26–29 September 2000. 2001
ISBN 0-7923-6968-8

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

90. Y. Ivanov, V. Cheshkov and M. Natova: *Polymer Composite Materials – Interface Phenomena & Processes*. 2001 ISBN 0-7923-7008-2
91. R.C. McPhedran, L.C. Botten and N.A. Nicorovici (eds.): *IUTAM Symposium on Mechanical and Electromagnetic Waves in Structured Media*. Proceedings of the IUTAM Symposium held in Sydney, NSW, Australia, 18-22 Januari 1999. 2001 ISBN 0-7923-7038-4
92. D.A. Sotiropoulos (ed.): *IUTAM Symposium on Mechanical Waves for Composite Structures Characterization*. Proceedings of the IUTAM Symposium held in Chania, Crete, Greece, June 14-17, 2000. 2001 ISBN 0-7923-7164-X
93. V.M. Alexandrov and D.A. Pozharskii: *Three-Dimensional Contact Problems*. 2001 ISBN 0-7923-7165-8
94. J.P. Dempsey and H.H. Shen (eds.): *IUTAM Symposium on Scaling Laws in Ice Mechanics and Ice Dynamics*. Proceedings of the IUTAM Symposium held in Fairbanks, Alaska, U.S.A., 13-16 June 2000. 2001 ISBN 1-4020-0171-1
95. U. Kirsch: *Design-Oriented Analysis of Structures. A Unified Approach*. 2002 ISBN 1-4020-0443-5
96. A. Preumont: *Vibration Control of Active Structures. An Introduction (2nd Edition)*. 2002 ISBN 1-4020-0496-6
97. B.L. Karihaloo (ed.): *IUTAM Symposium on Analytical and Computational Fracture Mechanics of Non-Homogeneous Materials*. Proceedings of the IUTAM Symposium held in Cardiff, U.K., 18-22 June 2001. 2002 ISBN 1-4020-0510-5
98. S.M. Han and H. Benaroya: *Nonlinear and Stochastic Dynamics of Compliant Offshore Structures*. 2002 ISBN 1-4020-0573-3
99. A.M. Linkov: *Boundary Integral Equations in Elasticity Theory*. 2002 ISBN 1-4020-0574-1
100. L.P. Lebedev, I.I. Vorovich and G.M.L. Gladwell: *Functional Analysis. Applications in Mechanics and Inverse Problems (2nd Edition)*. 2002 ISBN 1-4020-0667-5; Pb: 1-4020-0756-6
101. Q.P. Sun (ed.): *IUTAM Symposium on Mechanics of Martensitic Phase Transformation in Solids*. Proceedings of the IUTAM Symposium held in Hong Kong, China, 11-15 June 2001. 2002 ISBN 1-4020-0741-8
102. M.L. Munjal (ed.): *IUTAM Symposium on Designing for Quietness*. Proceedings of the IUTAM Symposium held in Bangkok, India, 12-14 December 2000. 2002 ISBN 1-4020-0765-5
103. J.A.C. Martins and M.D.P. Monteiro Marques (eds.): *Contact Mechanics*. Proceedings of the 3rd Contact Mechanics International Symposium, Praia da Consolação, Peniche, Portugal, 17-21 June 2001. 2002 ISBN 1-4020-0811-2
104. H.R. Drew and S. Pellegrino (eds.): *New Approaches to Structural Mechanics, Shells and Biological Structures*. 2002 ISBN 1-4020-0862-7
105. J.R. Vinson and R.L. Sierakowski: *The Behavior of Structures Composed of Composite Materials*. Second Edition. 2002 ISBN 1-4020-0904-6
106. Not yet published.
107. J.R. Barber: *Elasticity*. Second Edition. 2002 ISBN Hb 1-4020-0964-X; Pb 1-4020-0966-6
108. C. Miehe (ed.): *IUTAM Symposium on Computational Mechanics of Solid Materials at Large Strains*. Proceedings of the IUTAM Symposium held in Stuttgart, Germany, 20-24 August 2001. 2003 ISBN 1-4020-1170-9

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

109. P. Ståhle and K.G. Sundin (eds.): *IUTAM Symposium on Field Analyses for Determination of Material Parameters – Experimental and Numerical Aspects*. Proceedings of the IUTAM Symposium held in Abisko National Park, Kiruna, Sweden, July 31 – August 4, 2000. 2003
ISBN 1-4020-1283-7
110. N. Sri Namachchivaya and Y.K. Lin (eds.): *IUTAM Symposium on Nonlinear Stochastic Dynamics*. Proceedings of the IUTAM Symposium held in Monticello, IL, USA, 26 – 30 August, 2000. 2003
ISBN 1-4020-1471-6
111. H. Sobiechzky (ed.): *IUTAM Symposium Transsonicum IV*. Proceedings of the IUTAM Symposium held in Göttingen, Germany, 2–6 September 2002, 2003
ISBN 1-4020-1608-5
112. J.-C. Samin and P. Fiset: *Symbolic Modeling of Multibody Systems*. 2003
ISBN 1-4020-1629-8
113. A.B. Movchan (ed.): *IUTAM Symposium on Asymptotics, Singularities and Homogenisation in Problems of Mechanics*. Proceedings of the IUTAM Symposium held in Liverpool, United Kingdom, 8-11 July 2002. 2003
ISBN 1-4020-1780-4
114. S. Ahzi, M. Cherkaoui, M.A. Khaleel, H.M. Zbib, M.A. Zikry and B. LaMatina (eds.): *IUTAM Symposium on Multiscale Modeling and Characterization of Elastic-Inelastic Behavior of Engineering Materials*. Proceedings of the IUTAM Symposium held in Marrakech, Morocco, 20-25 October 2002. 2004
ISBN 1-4020-1861-4
115. H. Kitagawa and Y. Shibutani (eds.): *IUTAM Symposium on Mesoscopic Dynamics of Fracture Process and Materials Strength*. Proceedings of the IUTAM Symposium held in Osaka, Japan, 6-11 July 2003. Volume in celebration of Professor Kitagawa's retirement. 2004
ISBN 1-4020-2037-6
116. E.H. Dowell, R.L. Clark, D. Cox, H.C. Curtiss, Jr., K.C. Hall, D.A. Peters, R.H. Scanlan, E. Simiu, F. Sisto and D. Tang: *A Modern Course in Aeroelasticity*. 4th Edition, 2004
ISBN 1-4020-2039-2
117. T. Burczyński and A. Osyczka (eds.): *IUTAM Symposium on Evolutionary Methods in Mechanics*. Proceedings of the IUTAM Symposium held in Cracow, Poland, 24-27 September 2002. 2004
ISBN 1-4020-2266-2
118. D. Ieşan: *Thermoelastic Models of Continua*. 2004
ISBN 1-4020-2309-X
119. G.M.L. Gladwell: *Inverse Problems in Vibration*. Second Edition. 2004
ISBN 1-4020-2670-6
120. J.R. Vinson: *Plate and Panel Structures of Isotropic, Composite and Piezoelectric Materials, Including Sandwich Construction*. 2005
ISBN 1-4020-3110-6
121. *Forthcoming*
122. G. Rega and F. Vestroni (eds.): *IUTAM Symposium on Chaotic Dynamics and Control of Systems and Processes in Mechanics*. Proceedings of the IUTAM Symposium held in Rome, Italy, 8–13 June 2003. 2005
ISBN 1-4020-3267-6
123. E.E. Gdoutos: *Fracture Mechanics. An Introduction. 2nd edition*. 2005
ISBN 1-4020-3267-6
124. M.D. Gilchrist (ed.): *IUTAM Symposium on Impact Biomechanics from Fundamental Insights to Applications*. 2005
ISBN 1-4020-3795-3
125. J.M. Huyghe, P.A.C. Raats and S. C. Cowin (eds.): *IUTAM Symposium on Physicochemical and Electromechanical Interactions in Porous Media*. 2005
ISBN 1-4020-3864-X
126. H. Ding and W. Chen: *Elasticity of Transversely Isotropic Materials*. 2005
ISBN 1-4020-4033-4
127. W. Yang (ed): *IUTAM Symposium on Mechanics and Reliability of Actuating Materials*. Proceedings of the IUTAM Symposium held in Beijing, China, 1–3 September 2004. 2005
ISBN 1-4020-4131-6
128. J.-P. Merlet: *Parallel Robots*. 2006
ISBN 1-4020-4132-2

Mechanics

SOLID MECHANICS AND ITS APPLICATIONS

Series Editor: G.M.L. Gladwell

129. G.E.A. Meier and K.R. Sreenivasan (eds.): *IUTAM Symposium on One Hundred Years of Boundary Layer Research*. Proceedings of the IUTAM Symposium held at DLR-Göttingen, Germany, August 12–14, 2004. 2006 ISBN 1-4020-4149-7
130. H. Ulbrich and W. Günthner (eds.): *IUTAM Symposium on Vibration Control of Nonlinear Mechanisms and Structures*. 2005 ISBN 1-4020-4160-8
131. L. Librescu and O. Song: *Thin-Walled Composite Beams*. Theory and Application. 2006 ISBN 1-4020-3457-1
132. G. Ben-Dor, A. Dubinsky and T. Elperin: *Applied High-Speed Plate Penetration Dynamics*. 2006 ISBN 1-4020-3452-0
133. X. Markenscoff and A. Gupta (eds.): *Collected Works of J. D. Eshelby*. Mechanics and Defects and Heterogeneities. 2006 ISBN 1-4020-4416-X
134. R.W. Snidle and H.P. Evans (eds.): *IUTAM Symposium on Elastohydrodynamics and Micro-elastohydrodynamics*. Proceedings of the IUTAM Symposium held in Cardiff, UK, 1–3 September, 2004. 2006 ISBN 1-4020-4532-8