A Mathematical Introduction to

ROBOTIC MANIPULATION

Richard M. Murray
Zexiang Li
S. Shankar Sastry

CRC Press
Taylor & Francis Group

A Mathematical Introduction to
ROBOTIC MANIPULATION

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

A Mathematical Introduction to ROBOTIC MANIPULATION

Richard M. Murray
California Institute of Technology

Zexiang Li
Hong Kong University of Science and Technology

S. Shankar Sastry
University of California, Berkeley
and Harvard University

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

First issued in hardback 2017

© 1994 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

ISBN-13: 978-0-8493-7981-9 (pbk)
ISBN-13: 978-1-138-44016-6 (hbk)

Library of Congress Card Number 93-33167

The figure on the front cover is courtesy of John Hauser at the University of Colorado, Boulder.

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

The Open Access version of this book, available at www.taylorfrancis.com, has been made available under a Creative Commons Attribution (CC-BY) 4.0 International license.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Murray, Richard M.
A mathematical introduction to robotic mani pulation / Richard M. Murray, Zexiang Li, S. Shankar Sastry.
p. cm.
Includes bibliographical references and index.
ISBN 0-8493-7981-4
1. Robotics. I. Li, Zexiang, 1961-. II. Sastry, S. Shankar. III. Title.
TJ211.M87 1993
629.8′.923—dc20 93-33167

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

DOI: 10.1201/9781315136370

For Product Safety Concerns and Information please contact our EU representative:
GPSR@taylorandfrancis.com
Taylor & Francis Verlag GmbH,
Kaufingerstraße 24,
80331 München, Germany.

To RuthAnne (RMM)

To Jianghua (ZXL)

In memory of my father (SSS)

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Contents

Contents vii

Preface xiii

Acknowledgements xvii

Chapter 1 Introduction 1
1 Brief History 1
2 Multifingered Hands and Dextrous Manipulation 8
3 Outline of the Book 13
3.1 Manipulation using single robots 14
3.2 Coordinated manipulation using multifingered robot hands 15
3.3 Nonholonomic behavior in robotic systems 16
4 Bibliography 18

Chapter 2 Rigid Body Motion 19
1 Rigid Body Transformations 20
2 Rotational Motion in $\mathbb{R}^3$ 22
2.1 Properties of rotation matrices 23
2.2 Exponential coordinates for rotation 27
2.3 Other representations 31
3 Rigid Motion in $\mathbb{R}^3$ 34
3.1 Homogeneous representation 36
3.2 Exponential coordinates for rigid motion and twists 39
3.3 Screws: a geometric description of twists 45
4 Velocity of a Rigid Body 51
4.1 Rotational velocity 51
4.2 Rigid body velocity 54
4.3 Velocity of a screw motion 58
4.4 Coordinate transformations 59
5 Wrenches and Reciprocal Screws 61
5.1 Wrenches 61

5.2 Screw coordinates for a wrench 65
5.3 Reciprocal screws 66
6 Summary 70
7 Bibliography 72
8 Exercises 73

Chapter 3 Manipulator Kinematics 81
1 Introduction 81
2 Forward Kinematics 83
2.1 Problem statement 83
2.2 The product of exponentials formula 85
2.3 Parameterization of manipulators via twists 91
2.4 Manipulator workspace 95
3 Inverse Kinematics 97
3.1 A planar example 97
3.2 Paden-Kahan subproblems 99
3.3 Solving inverse kinematics using subproblems . . . 104
3.4 General solutions to inverse kinematics problems . 108
4 The Manipulator Jacobian 115
4.1 End-effector velocity 115
4.2 End-effector forces 121
4.3 Singularities 123
4.4 Manipulability 128
5 Redundant and Parallel Manipulators 129
5.1 Redundant manipulators 130
5.2 Parallel manipulators 132
5.3 Four-bar linkage 135
5.4 Stewart platform 139
6 Summary 144
7 Bibliography 146
8 Exercises 147

Chapter 4 Robot Dynamics and Control 155
1 Introduction 155
2 Lagrange's Equations 156
2.1 Basic formulation 157
2.2 Inertial properties of rigid bodies 160
2.3 Example: Dynamics of a two-link planar robot . . 164
2.4 Newton-Euler equations for a rigid body 165
3 Dynamics of Open-Chain Manipulators 168
3.1 The Lagrangian for an open-chain robot 168
3.2 Equations of motion for an open-chain manipulator 169
3.3 Robot dynamics and the product of exponentials formula 175
4 Lyapunov Stability Theory 179

4.1 Basic definitions 179
4.2 The direct method of Lyapunov 182
4.3 The indirect method of Lyapunov 184
4.4 Examples 185
4.5 Lasalle's invariance principle 188
5 Position Control and Trajectory Tracking 190
5.1 Problem description 190
5.2 Computed torque 191
5.3 PD control 193
5.4 Workspace control 196
6 Control of Constrained Manipulators 199
6.1 Dynamics of constrained systems 200
6.2 Control of constrained manipulators 202
6.3 Example: A planar manipulator moving in a slot . 203
7 Summary 206
8 Bibliography 207
9 Exercises 208

Chapter 5 Multifingered Hand Kinematics 211
1 Introduction to Grasping 211
2 Grasp Statics 214
2.1 Contact models 214
2.2 The grasp map 218
3 Force-Closure 223
3.1 Formal definition 223
3.2 Constructive force-closure conditions 224
4 Grasp Planning 229
4.1 Bounds on number of required contacts 229
4.2 Constructing force-closure grasps 232
5 Grasp Constraints 234
5.1 Finger kinematics 234
5.2 Properties of a multifingered grasp 237
5.3 Example: Two SCARA fingers grasping a box . . 240
6 Rolling Contact Kinematics 243
6.1 Surface models 243
6.2 Contact kinematics 248
6.3 Grasp kinematics with rolling 253
7 Summary 256
8 Bibliography 257
9 Exercises 259

Chapter 6 Hand Dynamics and Control **265**
1 Lagrange's Equations with Constraints 265
1.1 Pfaffian constraints 266
1.2 Lagrange multipliers 269
1.3 Lagrange-d'Alembert formulation 271
1.4 The nature of nonholonomic constraints 274
2 Robot Hand Dynamics 276
2.1 Derivation and properties 276
2.2 Internal forces 279
2.3 Other robot systems 281
3 Redundant and Nonmanipulable Robot Systems 285
3.1 Dynamics of redundant manipulators 286
3.2 Nonmanipulable grasps 290
3.3 Example: Two-fingered SCARA grasp 291
4 Kinematics and Statics of Tendon Actuation 293
4.1 Inelastic tendons 294
4.2 Elastic tendons 296
4.3 Analysis and control of tendon-driven fingers . . . 298
5 Control of Robot Hands 300
5.1 Extending controllers 300
5.2 Hierarchical control structures 302
6 Summary 311
7 Bibliography 313
8 Exercises 314

Chapter 7 Nonholonomic Behavior in Robotic Systems **317**
1 Introduction 317
2 Controllability and Frobenius' Theorem 321
2.1 Vector fields and flows 322
2.2 Lie brackets and Frobenius' theorem 323
2.3 Nonlinear Controllability 329
3 Examples of Nonholonomic Systems 332
4 Structure of Nonholonomic Systems 339
4.1 Classification of nonholonomic distributions 340
4.2 Examples of nonholonomic systems, continued . . 341
4.3 Philip Hall basis 344
5 Summary 346
6 Bibliography 347
7 Exercises 349

Chapter 8 Nonholonomic Motion Planning **355**
1 Introduction 355
2 Steering Model Control Systems Using Sinusoids 358
2.1 First-order controllable systems: Brockett's system 358
2.2 Second-order controllable systems 362

2.3 Higher-order systems: chained form systems 363
3 General Methods for Steering 366
3.1 Fourier techniques 367
3.2 Conversion to chained form 369
3.3 Optimal steering of nonholonomic systems 371
3.4 Steering with piecewise constant inputs 375
4 Dynamic Finger Repositioning 382
4.1 Problem description 382
4.2 Steering using sinusoids 383
4.3 Geometric phase algorithm 384
5 Summary 389
6 Bibliography 390
7 Exercises 391

Chapter 9 Future Prospects **395**
1 Robots in Hazardous Environments 396
2 Medical Applications for Multifingered Hands 398
3 Robots on a Small Scale: Microrobotics 399

Appendix A Lie Groups and Robot Kinematics **403**
1 Differentiable Manifolds 403
1.1 Manifolds and maps 403
1.2 Tangent spaces and tangent maps 404
1.3 Cotangent spaces and cotangent maps 405
1.4 Vector fields 406
1.5 Differential forms 408
2 Lie Groups 408
2.1 Definition and examples 408
2.2 The Lie algebra associated with a Lie group 410
2.3 The exponential map 412
2.4 Canonical coordinates on a Lie group 414
2.5 Actions of Lie groups 415
3 The Geometry of the Euclidean Group 416
3.1 Basic properties 416
3.2 Metric properties of $SE(3)$ 422
3.3 Volume forms on $SE(3)$ 430

Appendix B A Mathematica Package for Screw Calculus **435**

Bibliography **441**

Index **449**

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Preface

In the last two decades, there has been a tremendous surge of activity in robotics, both in terms of research and in terms of capturing the imagination of the general public as to its seemingly endless and diverse possibilities. This period has been accompanied by a technological maturation of robots as well, from the simple pick and place and painting and welding robots, to more sophisticated assembly robots for inserting integrated circuit chips onto printed circuit boards, to mobile carts for parts handling and delivery. Several areas of robotic automation have now become "standard" on the factory floor and, as of the writing of this book, the field is on the verge of a new explosion to areas of growth involving hazardous environments, minimally invasive surgery, and micro electro-mechanical mechanisms.

Concurrent with the growth in robotics in the last two decades has been the development of courses at most major research universities on various aspects of robotics. These courses are taught at both the undergraduate and graduate levels in computer science, electrical and mechanical engineering, and mathematics departments, with different emphases depending on the background of the students. A number of excellent textbooks have grown out of these courses, covering various topics in kinematics, dynamics, control, sensing, and planning for robot manipulators.

Given the state of maturity of the subject and the vast diversity of students who study this material, we felt the need for a book which presents a slightly more abstract (mathematical) formulation of the kinematics, dynamics, and control of robot manipulators. The current book is an attempt to provide this formulation not just for a single robot but also for multifingered robot hands, involving multiple cooperating robots. It grew from our efforts to teach a course to a hybrid audience of electrical engineers who did not know much about mechanisms, computer scientists who did not know about control theory, mechanical engineers who were suspicious of involved explanations of the kinematics and dynamics of garden variety open kinematic chains, and mathematicians who were curious, but did not have the time to build up lengthy prerequisites before beginning a study of robotics.

It is our premise that abstraction saves time in the long run, in return for an initial investment of effort and patience in learning some mathematics. The selection of topics—from kinematics and dynamics of single robots, to grasping and manipulation of objects by multifingered robot hands, to nonholonomic motion planning—represents an evolution from the more basic concepts to the frontiers of the research in the field. It represents what we have used in several versions of the course which have been taught between 1990 and 1993 at the University of California, Berkeley, the Courant Institute of Mathematical Sciences of New York University, the California Institute of Technology, and the Hong Kong University of Science and Technology (HKUST). We have also presented parts of this material in short courses at the Università di Roma, the Center for Artificial Intelligence and Robotics, Bangalore, India, and the National Taiwan University, Taipei, Taiwan.

The material collected here is suitable for advanced courses in robotics consisting of seniors or first- and second-year graduate students. At a senior level, we cover Chapters 1–4 in a twelve week period, augmenting the course with some discussion of technological and planning issues, as well as a laboratory. The laboratory consists of experiments involving on-line path planning and control of a few industrial robots, and the use of a simulation environment for off-line programming of robots. In courses stressing kinematic issues, we often replace material from Chapter 4 (Robot Dynamics) with selected topics from Chapter 5 (Multifingered Hand Kinematics). We have also covered Chapters 5–8 in a ten week period at the graduate level, in a course augmented with other advanced topics in manipulation or mobile robots.

The prerequisites that we assume are a good course in linear algebra at the undergraduate level and some familiarity with signals and systems. A course on control at the undergraduate level is helpful, but not strictly necessary for following the material. Some amount of mathematical maturity is also desirable, although the student who can master the concepts in Chapter 2 should have no difficulty with the remainder of the book.

We have provided a fair number of exercises after Chapters 2–8 to help students understand some new material and review their understanding of the chapter. A toolkit of programs written in Mathematica for solving the problems of Chapters 2 and 3 (and to some extent Chapter 5) have been developed and are described in Appendix B. We have studiously avoided numerical exercises in this book: when we have taught the course, we have adapted numerical exercises from measurements of robots or other "real" systems available in the laboratories. These vary from one time to the next and add an element of topicality to the course.

The one large topic in robotic manipulation that we have not covered in this book is the question of motion planning and collision avoidance

for robots. In our classroom presentations we have always covered some aspects of motion planning for robots for the sake of completeness. For graduate classes, we can recommend the recent book of Latombe on motion planning as a supplement in this regard. Another omission from this book is sensing for robotics. In order to do justice to this material in our respective schools, we have always had computer vision, tactile sensing, and other related topics, such as signal processing, covered in separate courses.

The contents of our book have been chosen from the point of view that they will remain foundational over the next several years in the face of many new technological innovations and new vistas in robotics. We have tried to give a snapshot of some of these vistas in Chapter 9. In reading this book, we hope that the reader will feel the same excitement that we do about the technological and social prospects for the field of robotics and the elegance of the underlying theory.

Richard Murray
Zexiang Li
Shankar Sastry

Berkeley, August 1993

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Acknowledgments

It is a great pleasure to acknowledge the people who have collaborated with one or more of us in the research contained in this book. A great deal of the material in Chapters 2 and 3 is based on the Ph.D. dissertation of Bradley Paden, now at the University of California, Santa Barbara. The research on multifingered robot hands, on which Chapters 5 and 6 are founded, was done in collaboration with Ping Hsu, now at San Jose State University; Arlene Cole, now at AT&T Bell Laboratories; John Hauser, now at the University of Colorado, Boulder; Curtis Deno, now at Intermedics, Inc. in Houston; and Kristofer Pister, now at the University of California, Los Angeles. In the area of nonholonomic motion planning, we have enjoyed collaborating with Jean-Paul Laumond of LAAS in Toulouse, France; Paul Jacobs, now at Qualcomm, Inc. in San Diego; Greg Walsh, Dawn Tilbury, and Linda Bushnell at the University of California, Berkeley; Richard Montgomery of the University of California, Santa Cruz; Leonid Gurvits of Siemens Research, Princeton; and Chris Fernandez at New York University.

The heart of the approach in Chapters 2 and 3 of this book is a derivation of robot kinematics using the product of exponentials formalism introduced by Roger Brockett of Harvard University. For this and manifold other contributions by him and his students to the topics in kinematics, rolling contact, and nonholonomic control, it is our pleasure to acknowledge his enthusiasm and encouragement by example. In a broader sense, the stamp of the approach that he has pioneered in nonlinear control theory is present throughout this book.

We fondly remember the seminar given at Berkeley in 1983 by P. S. Krishnaprasad of the University of Maryland, where he attempted to convince us of the beauty of the product of exponentials formula, and the numerous stimulating conversations with him, Jerry Marsden of Berkeley, and Tony Bloch of Ohio State University on the many beautiful connections between classical mechanics and modern mathematics and control theory. Another such seminar which stimulated our interest was one on multifingered robot hands and cooperating robots given at Berkeley in 1987 by Yoshi Nakamura, now of the University of Tokyo. We have also

enjoyed discussing kinematics, optimal control, and redundant mechanisms with John Baillieul of Boston University; Jeff Kerr, now of Zebra Robotics; Mark Cutkosky of Stanford University and Robert Howe, now of Harvard University; Dan Koditscheck, now of the University of Michigan; Mark Spong of the University of Illinois at Urbana-Champaign; and Joel Burdick and Elon Rimon at the California Institute of Technology. Conversations with Hector Sussmann of Rutgers University and Gerardo Lafferiere of Portland State University on nonholonomic motion planning have been extremely stimulating as well.

Our colleagues have provided both emotional and technical support to us at various levels of development of this material: John Canny, Charles Desoer, David Dornfeld, Ronald Fearing, Roberto Horowitz, Jitendra Malik, and "Tomi" Tomizuka at Berkeley; Jaiwei Hong, Bud Mishra, Jack Schwartz, James Demmel, and Paul Wright at New York University; Joel Burdick and Pietro Perona at Caltech; Peter Cheung, Ruey-Wen Liu, and Matthew Yuen at HKUST; Robyn Owens at the University of West Australia; Georges Giralt at LAAS in Toulouse, France; Dorotheè Normand Cyrot at the LSS in Paris, France; Alberto Isidori, Marica Di Benedetto, Alessandro De Luca, and 'Nando' Nicoló at the Università di Roma; Sanjoy Mitter and Anita Flynn at MIT; Antonio Bicchi at the Università di Pisa; M. Vidyasagar at the Center for Artificial Intelligence and Robotics in Bangalore, India; Li-Chen Fu of the National Taiwan University, Taipei, Taiwan; and T.-J. Tarn of Washington University. Finally, we are grateful to Mark Spong, Kevin Dowling, and Dalila Argez for their help with the photographs.

Our research has been generously supported by the National Science Foundation under grant numbers DMC 84-51129, IRI 90-14490, and IRI 90-03986, nurtured especially by Howard Moraff, the Army Research Office under grant DAAL88-K-0372 monitored by Jagdish Chandra, IBM, the AT&T Foundation, the GE Foundation, and HKUST under grant DAG 92/93 EG23. Our home institutions at UC Berkeley, the California Institute of Technology, and the Hong Kong University of Science and Technology have been exemplarily supportive of our efforts, providing the resources to help us to grow programs where there were none. We owe a special debt of gratitude in this regard to Karl Pister, Dean of Engineering at Berkeley until 1990.

The manuscript was classroom tested in various versions by James Clark at Harvard, John Canny, Curtis Deno and Matthew Berkemeier at Berkeley, and Joel Burdick at Caltech, in addition to the three of us. Their comments have been invaluable to us in revising the early drafts. We appreciate the detailed and thoughtful reviews by Greg Chirikjian of Johns Hopkins, and Michael McCarthy and Frank Park of the University of California, Irvine.

In addition, many students suffering early versions of this course have

contributed to debugging the text. They include L. Bushnell, N. Getz, J.-P. Tennant, D. Tilbury, G. Walsh, and J. Wendlandt at Berkeley; R. Behnken, S. Kelly, A. Lewis, S. Sur, and M. van Nieuwstadt at Caltech; and A. Lee and J. Au of the Hong Kong University of Science and Technology. Sudipto Sur at Caltech helped develop a Mathematica package for screw calculus which forms the heart of the software described in Appendix B. We are ultimately indebted to these and the unnamed others for the inspiration to write this book.

Finally, on a personal note, we would like to thank our families for their support and encouragement during this endeavor.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 1

Introduction

DOI: 10.1201/9781315136370-1
This chapter has been made available under a CC-BY 4.0 license.

In the last twenty years, our conception and use of robots has evolved from the stuff of science fiction films to the reality of computer-controlled electromechanical devices integrated into a wide variety of industrial environments. It is routine to see robot manipulators being used for welding and painting car bodies on assembly lines, stuffing printed circuit boards with IC components, inspecting and repairing structures in nuclear, undersea, and underground environments, and even picking oranges and harvesting grapes in agriculture. Although few of these manipulators are anthropomorphic, our fascination with humanoid machines has not dulled, and people still envision robots as evolving into electromechanical replicas of ourselves. While we are not likely to see this type of robot in the near future, it is fair to say that we have made a great deal of progress in introducing simple robots with crude end-effectors into a wide variety of circumstances. Further, it is important to recognize that our impatience with the pace of robotics research and our expectations of what robots can and cannot do is in large part due to our lack of appreciation of the incredible power and subtlety of our own biological motor control systems.

1 Brief History

The word *robot* was introduced in 1921 by the Czech playwright Karel Capek in his satirical play *R. U. R.* (Rossum's Universal Robots), where he depicted robots as machines which resembled people but worked tirelessly. In the play, the robots eventually turn against their creators and annihilate the human race. This play spawned a great deal of further science fiction literature and film which have contributed to our perceptions of robots as being human-like, endowed with intelligence and even personality. Thus, it is no surprise that present-day robots appear primitive

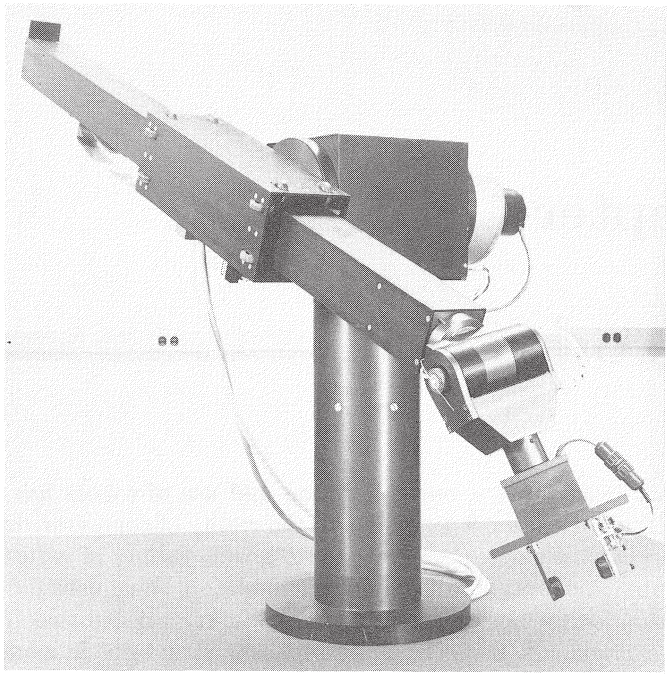


Figure 1.1: The Stanford manipulator. (Courtesy of the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign)

when compared with the expectations created by the entertainment industry. To give the reader a flavor of the development of modern robotics, we will give a much abbreviated history of the field, derived from the accounts by Fu, Gonzalez, and Lee [35] and Spong and Vidyasagar [111]. We describe this roughly by decade, starting from the fifties and continuing up to the eighties.

The early work leading up to today's robots began after World War II in the development of remotely controlled mechanical manipulators, developed at Argonne and Oak Ridge National Laboratories for handling radioactive material. These early mechanisms were of the master-slave type, consisting of a master manipulator guided by the user through a series of moves which were then duplicated by the slave unit. The slave unit was coupled to the master through a series of mechanical linkages. These linkages were eventually replaced by either electric or hydraulic powered coupling in "teleoperators," as these machines are called, made by General Electric and General Mills. Force feedback to keep the slave manipulator from crushing glass containers was also added to the teleoperators in 1949.

In parallel with the development of the teleoperators was the devel-

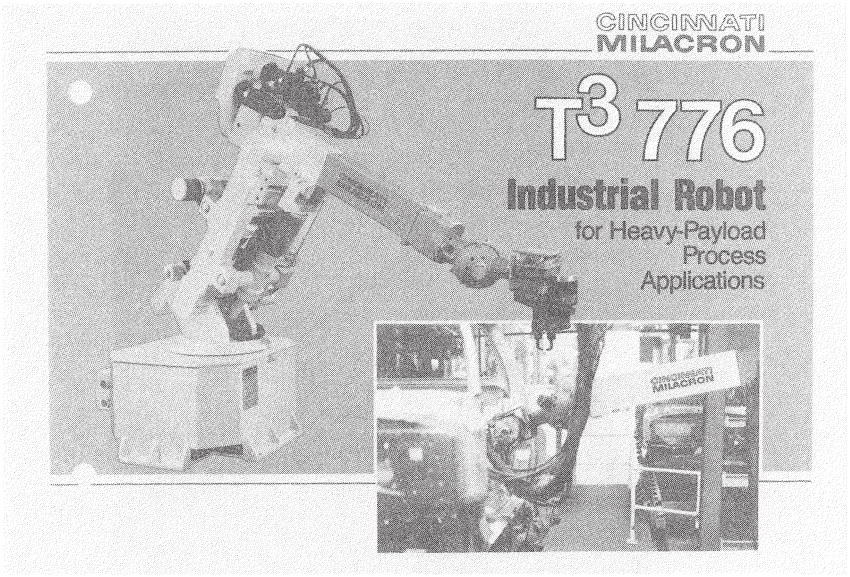


Figure 1.2: The Cincinnati Milacron T^3 (The Tomorrow Tool) robot. (Courtesy of Cincinnati Milacron)

opment of Computer Numerically Controlled (CNC) machine tools for accurate milling of low-volume, high-performance aircraft parts. The first robots, developed by George Devol in 1954, replaced the master manipulator of the teleoperator with the programmability of the CNC machine tool controller. He called the device a "programmed articulated transfer device." The patent rights were bought by a Columbia University student, Joseph Engelberger, who later founded a company called Unimation in Connecticut in 1956. Unimation installed its first robot in a General Motors plant in 1961. The key innovation here was the "programmability" of the machine: it could be retooled and reprogrammed at relatively low cost so as to enable it to perform a wide variety of tasks. The mechanical construction of the Unimation robot arm represented a departure from conventional machine design in that it used an open kinematic chain: that is to say, it had a cantilevered beam structure with many degrees of freedom. This enabled the robot to access a large workspace relative to the space occupied by the robot itself, but it created a number of problems for the design since it is difficult to accurately control the end point of a cantilevered arm and also to regulate its stiffness. Moreover, errors at the base of the kinematic chain tended to get amplified further out in the chain. To alleviate this problem, hydraulic actuators capable of both high power and generally high precision were

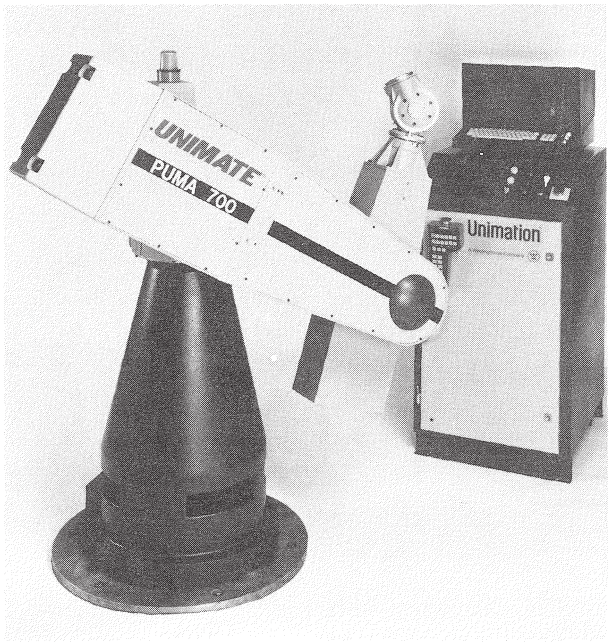


Figure 1.3: The Unimation PUMA (Programmable Universal Manipulator for Assembly). (Courtesy of Stäubli Unimation, Inc.)

used for the joint actuators.

The flexibility of the newly introduced robots was quickly seen to be enhanced through sensory feedback. To this end, Ernst in 1962 developed a robot with force sensing which enabled it to stack blocks. To our knowledge, this system was the first that involved a robot interacting with an unstructured environment and led to the creation of the Project MAC (Man And Computer) at MIT. Tomovic and Boni developed a pressure sensor for the robot which enabled it to squeeze on a grasped object and then develop one of two different grasp patterns. At about the same time, a binary robot vision system which enabled the robot to respond to obstacles in its environment was developed by McCarthy and colleagues in 1963. Many other kinematic models for robot arms, such as the Stanford manipulator, the Boston arm, the AMF (American Machine and Foundry) arm, and the Edinburgh arm, were also introduced around this time. Another novel robot of the period was a walking robot developed by General Electric for the Army in 1969. Robots that responded to voice commands and stacked randomly scattered blocks were developed at Stanford and other places. Robots made their appearance in Japan through Kawasaki's acquisition of a license from Unimation in 1968.

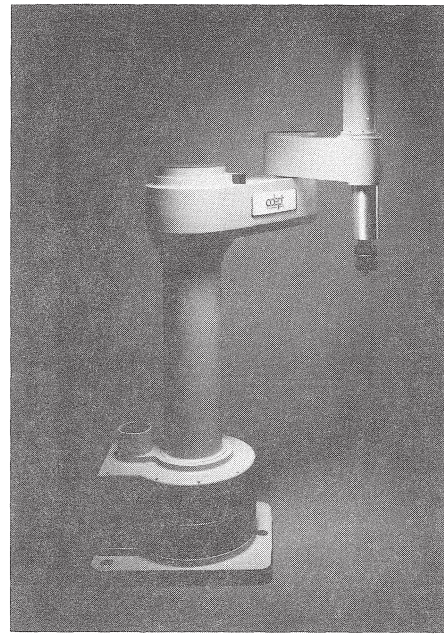


Figure 1.4: The AdeptOne robot. (Courtesy of Adept Technology, Inc.)

In 1973, the first language for programming robot motion, called WAVE, was developed at Stanford to enable commanding a robot with high-level commands. About the same time, in 1974, the machine tool manufacturer Cincinnati Milacron, Inc. introduced its first computer-controlled manipulator, called The Tomorrow Tool (T^3), which could lift a 100 pound load as well as track moving objects on an assembly line. Later in the seventies, Paul and Bolles showed how a Stanford arm could assemble water pumps, and Will and Grossman endowed a robot with touch and force sensors to assemble a twenty part typewriter. At roughly the same time, a group at the Draper Laboratories put together a Remote Center Compliance (RCC) device for part insertion in assembly.

In 1978, Unimation introduced a robot named the Programmable Universal Machine for Assembly (PUMA), based on a General Motors study. Bejczy at Jet Propulsion Laboratory began a program of teleoperation for space-based manipulators in the mid-seventies. In 1979, the SCARA (Selective Compliant Articulated Robot for Assembly) was introduced in Japan and then in the United States.

As applications of industrial robots grew, different kinds of robots with attendant differences in their actuation methods were developed. For light-duty applications, electrically powered robots were used both

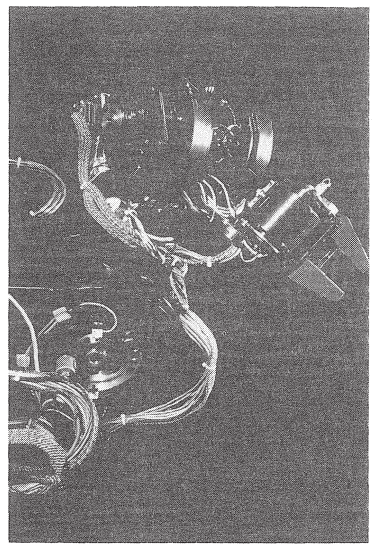


Figure 1.5: The CMU DD Arm I. (Courtesy of M.J. Dowling)

for reasons of relative inexpensiveness and cleanliness. The difficulty with electric motors is that they produce their maximum power at relatively high speeds. Consequently, the motors need to be geared down for use. This gear reduction introduces friction, backlash, and expense to the design of the motors. Consequently, the search was on to find a way of driving the robot's joints directly without the need to gear down its electric motors. In response to this need, a direct drive robot was developed at Carnegie Mellon by Asada in 1981.

In the 1980s, many efforts were made to improve the performance of industrial robots in fine manipulation tasks: active methods using feedback control to improve positioning accuracy and program compliance, and passive methods involving a mechanical redesign of the arm. It is fair to say, however, that the eighties were not a period of great innovation in terms of building new types of robots. The major part of the research was dedicated to an understanding of algorithms for control, trajectory planning, and sensor aggregation of robots. Among the first active control methods developed were efficient recursive Lagrangian and computational schemes for computing the gravity and Coriolis force terms in the dynamics of robots. In parallel with this, there was an effort in exactly linearizing the dynamics of a multi-joint robot by state feedback, using a technique referred to as computed torque. This approach, while computationally demanding, had the advantage of giving precise bounds on the transient performance of the manipulator. It involved exact cancellation, which in practice had to be done either approximately or adaptively. There were may other projects on developing position/force

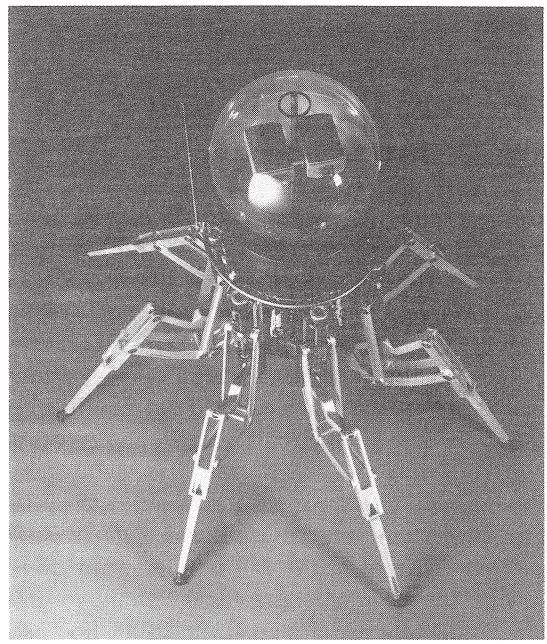


Figure 1.6: The Odex I six-legged walking robot. (Photo courtesy of Odetics, Inc.)

control strategies for robots in contact with the environment, referred to as hybrid or compliant control. In the search for more accurately controllable robot manipulators, robot links were getting to be lighter and to have harmonic drives, rather than gear trains in their joints. This made for flexible joints and arms, which in turn necessitated the development of new control algorithms for flexible link and flexible joint robots.

The trend in the nineties has been towards robots that are modifiable for different assembly operations. One such robot is called Robotworld, manufactured by Automatix, which features several four degree of freedom modules suspended on air bearings from the stator of a Sawyer effect motor. By attaching different end-effectors to the ends of the modules, the modules can be modified for the assembly task at hand. In the context of robots working in hazardous environments, great strides have been made in the development of mobile robots for planetary exploration, hazardous waste disposal, and agriculture. In addition to the extensive programs in reconfigurable robots and robots for hazardous environments, we feel that the field of robotics is primed today for some large technological advances incorporating developments in sensor and actuator technology at the milli- and micro-scales as well as advances

in computing and control. We defer a discussion of these prospects for robotics to Chapter 9.

2 Multifingered Hands and Dextrous Manipulation

The vast majority of robots in operation today consist of six joints which are either rotary (articulated) or sliding (prismatic), with a simple "end-effector" for interacting with the workpieces. The applications range from pick and place operations, to moving cameras and other inspection equipment, to performing delicate assembly tasks involving mating parts. This is certainly nowhere near as fancy as the stuff of early science fiction, but is useful in such diverse arenas such as welding, painting, transportation of materials, assembly of printed circuit boards, and repair and inspection in hazardous environments.

The term *hand* or *end-effector* is used to describe the interface between the manipulator (arm) and the environment, out of anthropomorphic intent. The vast majority of hands are simple: grippers (either two- or three-jaw), pincers, tongs, or in some cases remote compliance devices. Most of these end-effectors are designed on an ad hoc basis to perform specific tasks with specific parts. For example, they may have suction cups for lifting glass which are not suitable for machined parts, or jaws operated by compressed air for holding metallic parts but not suitable for handling fragile plastic parts. Further, a difficulty that is commonly encountered in applications is the clumsiness of a six degree of freedom robot equipped only with these simple hands. The clumsiness manifests itself in:

1. *A lack of dexterity.* Simple grippers enable the robot to hold parts securely but they cannot manipulate the grasped object.

2. *A limited number of possible grasps* resulting in the need to change end-effectors frequently for different tasks.

3. *Large motions of the arm* are sometimes needed for even small motions of the end-effector. Since the motors of the robot arm are progressively larger away from the end-effector, the motion of the earliest motors is slow and inefficient.

4. *A lack of fine force control* which limits assembly tasks to the most rudimentary ones.

A *multifingered* or articulated hand offers some solutions to the problem of endowing a robot with dexterity and versatility. The ability of a multifingered hand to reconfigure itself for performing a variety of different grasps reduces the need for changing end-effectors. The large number

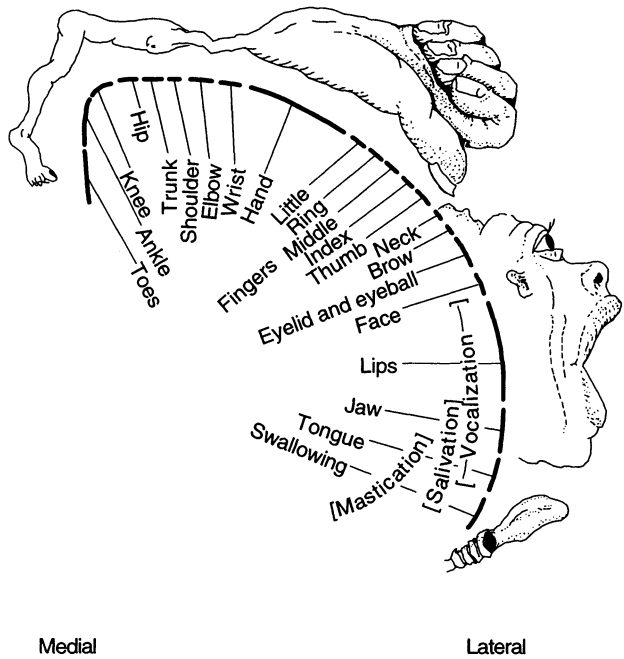


Figure 1.7: Homunculus diagram of the motor cortex. (Reprinted, by permission, from Kandel, Schwartz, and Jessel, *Principles of Neural Science*, Third Edition [Appleton and Lange, Norwalk, CT, 1991]. Adapted from Penfield and Rasmussen, *The Cerebral Cortex of Man: A Clinical Study of Localization of Function* [Macmillan, 1950])

of lightweight actuators associated with the degrees of freedom of the hand allows for fast, precise, and energy-efficient motions of the object held in the hand. Fine motion force-control at a high bandwidth is also facilitated for similar reasons. Indeed, multifingered hands are a truly anthropomorphically motivated concept for dextrous manipulation: we use our arms to position our hands in a given region of space and then use our wrists and fingers to interact in a detailed and intricate way with the environment. We preform our fingers into grasps which pinch, encircle, or immobilize objects, changing grasps as a consequence of these actions. One measure of the intelligence of a member of the mammalian family is the fraction of its motor cortex dedicated to the control of its hands. This fraction is discerned by painstaking mapping of the body on the motor cortex by neurophysiologists, yielding a homunculus of the kind shown in Figure 1.7. For humans, the largest fraction (30–40%) of the motor cortex is dedicated to the control of the hands, as compared with 20–30% for most monkeys and under 10% for dogs and cats.

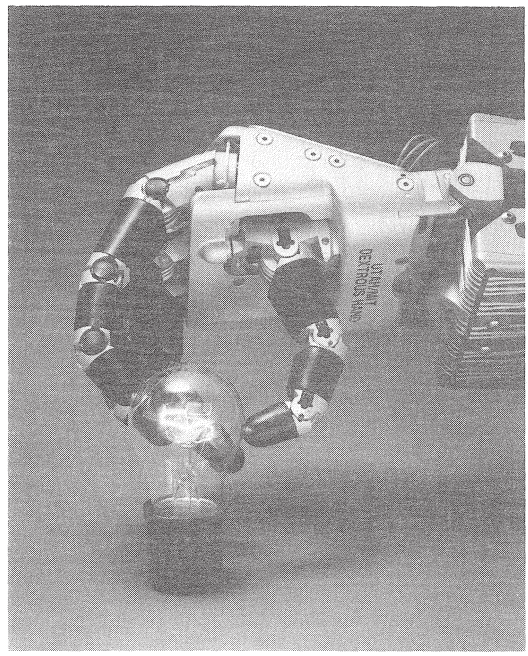


Figure 1.8: The Utah/MIT hand. (Photo courtesy of Sarcos, Inc.)

From a historical point of view, the first uses of multifingered hands were in prosthetic devices to replace lost limbs. Childress [18] refers to a device from 1509 made for a knight, von Berlichingen, who had lost his hand in battle at an early age. This spring-loaded device was useful in battle but was unfortunately not handy enough for everyday functions. After the Berlichingen hand, numerous other hand designs have been made from 1509 to the current time. Several of these hands are still available today; some are passive (using springs), others are body-powered (using arm flexion control or shrug control). Some of the hands had the facility for voluntary closure and others involuntary closure. Childress classifies the hands into the following four types:

1. *Cosmetic.* These hands have no real movement and cannot be activated, but they can be used for pushing or as an opposition element for the other hand.

2. *Passive.* These hands need the manual manipulation of the other (non-prosthetic) hand to adjust the grasping of the hand. These were the earliest hands, including the Berlichingen hand.

3. *Body powered.* These hands use motions of the body to activate the

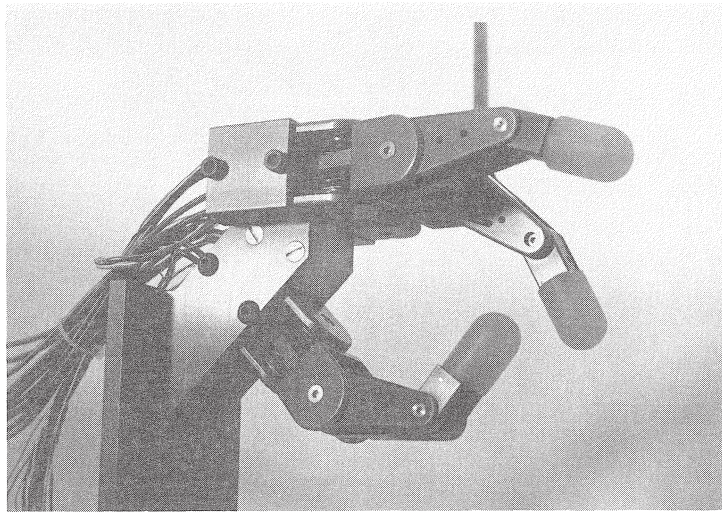


Figure 1.9: The Salisbury Hand, designed by Kenneth Salisbury. (Photo courtesy of David Lampe, MIT)

hand. Two of the most common schemes involve pulling a cable when the arm is moved forward (arm-flexion control) or pulling the cable when the shoulders are rounded (shrug control). Indeed, one frequently observes these hands operated by an amputee using shrugs and other such motions of her upper arm joints.

4. *Externally powered.* These hands obtain their energy from a storage source such as a battery or compressed gas. These are yet to displace the body-powered hands in prostheses.

Powered hand mechanisms came into vogue after 1920, but the greatest usage of these devices has been only since the 1960s. The Belgrade hand, developed by Tomović and Boni [114], was originally developed for Yugoslav veterans who had lost their arms to typhus. Other hands were invented as limb replacements for "thalidomide babies." There has been a succession of myoelectrically controlled devices for prostheses culminating in some advanced devices at the University of Utah [45], developed mainly for grasping objects. While these devices are quite remarkable mechanisms, it is fair to say that their dexterity arises from the vision-guided feedback training of the wearer, rather than any feedback mechanisms inherent in the device per se.

As in the historical evolution of robots, teleoperation in hazardous or hard to access environments—such as nuclear, underwater, space, mining, and, recently, surgical environments—has provided a large impetus for

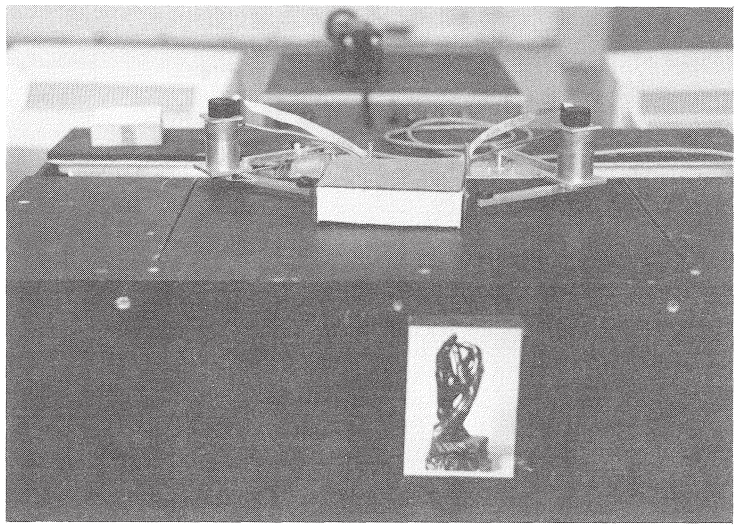


Figure 1.10: Styx, a two-fingered planar hand built at UC Berkeley in 1988.

the development of dextrous multifingered hands. These devices enable the operator to perform simple manipulations with her hands in a remote environment and have the commands be relayed to a remote multifingered manipulator. In the instance of surgery, the remote manipulator is a surgical device located inside the body of the patient.

There have been many attempts to devise multifingered hands for research use which are somewhere between teleoperation, prosthesis, and dextrous end-effectors. These hands truly represent our dual point of view in terms of jumping back and forth from an anthropomorphic point of view (mimicking our own hands) to the point of view of intelligent end-effectors (for endowing our robots with greater dexterity). Some examples of research on multifingered hands can be found in the work of Skinner [107], Okada [85], and Hanafusa and Asada [39]. The Okada hand was a three-fingered cable-driven hand which accomplished tasks such as attaching a nut to a bolt. Hanafusa and Asada's hand has three elastic fingers driven by a single motor with three claws for stably grasping several oddly shaped objects.

Later multifingered hands include the Salisbury Hand (also known as the Stanford/JPL hand) [70], the Utah/MIT hand [45], the NYU hand [24], and the research hand Styx [77]. The Salisbury hand is a three-fingered hand; each finger has three degrees of freedom and the joints are all cable driven. The placement of the fingers consists of one finger (the thumb) opposing the other two. The Utah/MIT hand has

four fingers (three fingers and a thumb) in a very anthropomorphic configuration; each finger has four degrees of freedom and the hand is cable driven. The difference in actuation between the Salisbury Hand and the Utah/MIT hand is in how the cables (tendons) are driven: the first uses electric motors and the second pneumatic pistons. The NYU hand is a non-anthropomorphic planar hand with four fingers moving in a plane, driven by stepper motors. Styx was a two-fingered hand with each finger having two joints, all direct driven. Like the NYU hand, Styx was used as a test bed for performing control experiments on multifingered hands.

At the current time, several kinds of multifingered hands at different scales—down to millimeters and even micrometers—are either being developed or put in use. Some of them are classified merely as custom or semi-custom end-effectors. A recent multifingered hand developed in Pisa is used for picking oranges in Sicily, another developed in Japan is used to play a piano! One of the key stumbling blocks to the development of lightweight hands has been lightweight high-torque motors. In this regard, muscle-like actuators, inch-worm motors, and other novel actuator technologies have been proposed and are currently being investigated. One future application of multifingered robot hands which relies on these technologies is in minimally invasive surgery. This application is further discussed in Chapter 9.

3 Outline of the Book

This book is organized into eight chapters in addition to this one. Most chapters contain a summary section followed by a set of exercises. We have deliberately not included numerical exercises in this book. In teaching this material, we have chosen numbers for our exercises based on some robot or other physical situation in the laboratory. We feel this adds greater realism to the numbers.

Chapter 2 is an introduction to rigid body motion. In this chapter, we present a geometric view to understanding translational and rotational motion of a rigid body. While this is one of the most ubiquitous topics encountered in textbooks on mechanics and robotics, it is also perhaps one of the most frequently misunderstood. The simple fact is that the careful description and understanding of rigid body motion is subtle. The point of view in this chapter is classical, but the mathematics modern. After defining rigid body rotation, we introduce the use of the exponential map to represent and coordinatize rotations (Euler's theorem), and then generalize to general rigid motions. In so doing, we introduce the notion of screws and twists, and describe their relationship with homogeneous transformations. With this background, we begin the study of infinitesimal rigid motions and introduce twists for representing rigid body velocities. The dual of the theory of twists is covered in a section

on wrenches, which represent generalized forces. The chapter concludes with a discussion of reciprocal screws. In classroom teaching, we have found it important to cover the material of Chapter 2 at a leisurely pace to let students get a feel for the subtlety of understanding rigid body motion.

The theory of screws has been around since the turn of the century, and Chasles' theorem and its dual, Poinsot's theorem, are even more classical. However, the treatment of the material in this chapter easily extends to other more abstract formulations which are also useful in thinking about problems of manipulation. These are covered in Appendix A.

The rest of the material in the book may be subdivided into three parts: an introduction to manipulation for single robots, coordinated manipulation using a multifingered robot hand, and nonholonomic motion planning. We will discuss the outline of each part in some detail.

3.1 Manipulation using single robots

Chapter 3 contains the description of manipulator kinematics for a single robot. This is the description of the position and orientation of the end-effector or gripper in terms of the angles of the joints of the robot. The form of the manipulator kinematics is a natural outgrowth of the exponential coordinatization for rigid body motion of Chapter 2. We prove that the kinematics of open-link manipulators can be represented as a product of exponentials. This formalism, first pointed out by Brockett [12], is elegant and combines within it a great deal of the analytical sophistication of Chapter 2. Our treatment of kinematics is something of a deviation from most other textbooks, which prefer a Denavit-Hartenberg formulation of kinematics. The payoff for the product of exponentials formalism is shown in this chapter in the context of an elegant formulation of a set of canonical problems for solving the inverse kinematics problem: the problem of determining the joint angles given the position and orientation of the end-effector or gripper of the robot. These problems, first formulated by Paden and Kahan [86], enable a precise determination of all of the multiple inverse kinematic solutions for a large number of industrial robots. The extension of this approach to the inverse kinematics of more general robots actually needs some recent techniques from classical algebraic geometry, which we discuss briefly.

Another payoff of using the product of exponentials formula for kinematics is the ease of differentiating the kinematics to obtain the manipulator Jacobian. The columns of the manipulator Jacobian have the interpretation of being the twist axes of the manipulator. As a consequence, it is easy to geometrically characterize and describe the singularities of the manipulator. The product of exponentials formula is also used for deriving the kinematics of robots with one or more closed kinematic chains,

such as a Stewart platform or a four-bar planar linkage.

Chapter 4 is a derivation of the dynamics and control of single robots. We start with a review of the Lagrangian equations of motion for a system of rigid bodies. We also specialize these equations to derive the Newton-Euler equations of motion of a rigid body. As in Chapter 2, this material is classical but is covered in a modern mathematical geometric framework. Using once again the product of exponentials formula, we derive the Lagrangian of an open-chain manipulator and show how the geometric structure of the kinematics reflects into the form of the Lagrangian of the manipulator.

Finally, we review the basics of Lyapunov theory to provide some machinery for proving the stability of the control schemes that we will present in this book. We use this to prove the stability of two classes of control laws for single manipulators: the computed torque control law and the so-called PD (for proportional + derivative) control law for single manipulators.

3.2 Coordinated manipulation using multifingered robot hands

Chapter 5 is an introduction to the kinematics of grasping. Beginning with a review of models of contact types, we introduce the notion of a grasp map, which expresses the relationship between the forces applied by the fingers contacting the object and their effect at the center of mass of the object. We characterize what are referred to as stable grasps or force-closure grasps. These are grasps which immobilize an object robustly. Using this characterization, we discuss how to construct force-closure grasps using an appropriate positioning of the fingers on the surface of the object.

The first half of the chapter deals with issues of force exerted on the object by the fingers. The second half deals with the dual issue of how the movements of the grasped object reflect the movements of the fingers. This involves the interplay between the qualities of the grasp and the kinematics of the fingers (which are robots in their own right) grasping the object. A definition dual to that of force-closure, called manipulability, is defined and characterized. Finally, we discuss the rolling of fingertips on the surface of an object. This is an important way of repositioning fingers on the surface of an object so as to improve a grasp and may be necessitated by the task to be performed using the multifingered hand.

Chapter 6 is a derivation of the dynamics and control for multifingered robot hands. The derivation of the kinematic equations for a multifingered hand is an exercise in writing equations for robotic systems with constraints, namely the constraints imposed by the grasp. We develop the necessary mathematical machinery for writing the Lagrangian equations

for systems with so-called Pfaffian constraints. There is a preliminary discussion of why these Pfaffian or velocity constraints cannot be simplified to constraints on the configuration variables of the system alone. Indeed, this is the topic of Chapters 7 and 8. We use our formalism to write the equations of motion for a multifingered hand system. We show connections between the form of these equations and the dynamical equations for a single robot. The dynamical equations are particularly simple when the grasps are nonredundant and manipulable. In the instance that the grasps are either redundant or nonmanipulable, some substantial changes need to be made to their dynamics. Using the form of dynamical equations for the multifingered hand system, we propose two separate sets of control laws which are reminiscent of those of the single robot, namely the computed torque control law and the PD control law, and prove their performance.

A large number of multifingered hands, including those involved in the study of our own musculo-skeletal system, are driven not by motors but by networks of tendons. In this case, the equations of motion need to be modified to take into account this mechanism of force generation at the joints of the fingers. This chapter develops the dynamics of tendon-driven robot hands.

Another important topic to be considered in the control of systems of many degrees of freedom, such as the multifingered robot hand, is the question of the hierarchical organization of the control. The computed torque and PD control law both suffer from the drawback of being computationally expensive. One could conceive that a system with hundreds of degrees of freedom, such as the mammalian musculo-skeletal system, has a hierarchical organization with coarse control at the cortical level and progressively finer control at the spinal and muscular level. This hierarchical organization is key to organizing a fan-out of commands from the higher to the lower levels of the hierarchy and is accompanied by a fan-in of sensor data from the muscular to the cortical level. We have tried to answer the question of how one might try to develop an environment of controllers for a multifingered robotic system so as to take into account this sort of hierarchical organization by way of a sample multi-robot control programming paradigm that we have developed here.

3.3 Nonholonomic behavior in robotic systems

In Chapter 6, we run into the question of how to deal with the presence of Pfaffian constraints when writing the dynamical equations of a multifingered robot hand. In that chapter, we show how to incorporate the constraints into the Lagrangian equations. However, one question that is left unanswered in that chapter is the question of trajectory planning for the system with nonholonomic constraints. In the instance of a multifingered hand grasping an object, we give control laws for getting the

grasped object to follow a specified position and orientation. However, if the fingertips are free to roll on the surface of the object, it is not explicitly possible for us to control the locations to which they roll using only the tools of Chapter 6. In particular, we are not able to give control strategies for moving the fingers from one contact location to another. Motivated by this observation, we begin a study of nonholonomic behavior in robotic systems in Chapter 7.

Nonholonomic behavior can arise from two different sources: bodies rolling without slipping on top of each other, or conservation of angular momentum during the motion. In this chapter, we expand our horizons beyond multifingered robot hands and give yet other examples of nonholonomic behavior in robotic systems arising from rolling: car parking, mobile robots, space robots, and a hopping robot in the flight phase. We discuss methods for classifying these systems, understanding when they are partially nonholonomic (or nonintegrable) and when they are holonomic (or integrable). These methods are drawn from some rudimentary notions of differential geometry and nonlinear control theory (controllability) which we develop in this chapter. The connection between nonholonomy of Pfaffian systems and controllability is one of duality, as is explained in this chapter.

Chapter 8 contains an introduction to some methods of motion planning for systems with nonholonomic constraints. This is the study of trajectory planning for nonholonomic systems consistent with the constraints on the system. This is a very rapidly growing area of research in robotics and control. We start with an overview of existing techniques and then we specialize to some methods of trajectory planning. We begin with the role of sinusoids in generating Lie bracket motions in nonholonomic systems. This is motivated by some solutions to optimal control problems for a simple class of model systems. Starting from this class of model systems, we show how one can generalize this class of model systems to a so-called chain form variety. We then discuss more general methods for steering nonholonomic systems using piecewise constant controls and also Ritz basis functions. We apply our methods to the examples presented in the previous chapter. We finally return to the question of dynamic finger repositioning on the surface of a grasped object and give a few different techniques for rolling fingers on the surface of a grasped object from one grasp to another.

Chapter 9 contains a description of some of the growth areas in robotics from a technological point of view. From a research and an analytical point of view, we hope that the book in itself will convince the reader of the many unexplored areas of our understanding of robotic manipulation.

4 Bibliography

It is a tribute to the vitality of the field that so many textbooks and books on robotics have been written in the last fifteen years. It is impossible to do justice or indeed to list them all here. We just mention some that we are especially familiar with and apologize to those whom we omit to cite.

One of the earliest textbooks in robotics is by Paul [91], on the mathematics, programming, and control of robots. It was followed in quick succession by the books of Gorla and Renaud [36], Craig [21], and Fu, Gonzalez and Lee [35]. The first two concentrated on the mechanics, dynamics, and control of single robots, while the third also covered topics in vision, sensing, and intelligence in robots. The text by Spong and Vidyasagar [111] gives a leisurely discussion of the dynamics and control of robot manipulators. Also significant is the set of books by Coiffet [20], Asada and Slotine [2], and Koivo [53]. As this book goes to print, we are aware also of a soon to be completed new textbook by Siciliano and Sciavicco. An excellent perspective of the development of control schemes for robots is provided by the collection of papers edited by Spong, Lewis and Abdallah [110].

The preceding were books relevant to single robots. The first monograph on multifingered robot hands was that of Mason and Salisbury [70], which covered some details of the formulation of grasping and substantial details of the design and control of the Salisbury three-fingered hand. Other books in the area since then have included the monographs by Cutkosky [22] and by Nakamura [80], and the collection of papers edited by Iberall and Venkataraman [43].

There are a large number of collections of edited papers on robotics. Some recent ones containing several interesting papers are those edited by Brockett [13], based on the contents of a short course of the American Mathematics Society in 1990; and a collection of papers on all aspects of manipulation edited Spong, Lewis, and Abdallah [110]; and a recent collection of papers on nonholonomic motion planning edited by Li and Canny [62], based on the contents of a short course at the 1991 IEEE International Conference on Robotics and Automation.

Not included in this brief bibliographical survey are books on computer vision or mobile robots which also have witnessed a flourish of activity.

Chapter 2

Rigid Body Motion

DOI: 10.1201/9781315136370-2
This chapter has been made available under a CC-BY 4.0 license.

A rigid motion of an object is a motion which preserves distance between points. The study of robot kinematics, dynamics, and control has at its heart the study of the motion of rigid objects. In this chapter, we provide a description of rigid body motion using the tools of linear algebra and screw theory.

The elements of screw theory can be traced to the work of Chasles and Poinsot in the early 1800s. Chasles proved that a rigid body can be moved from any one position to any other by a movement consisting of rotation about a straight line followed by translation parallel to that line. This motion is what we refer to in this book as a *screw motion.* The infinitesimal version of a screw motion is called a *twist* and it provides a description of the instantaneous velocity of a rigid body in terms of its linear and angular components. Screws and twists play a central role in our formulation of the kinematics of robot mechanisms.

The second major result upon which screw theory is founded concerns the representation of forces acting on rigid bodies. Poinsot is credited with the discovery that any system of forces acting on a rigid body can be replaced by a single force applied along a line, combined with a torque about that same line. Such a force is referred to as a *wrench.* Wrenches are dual to twists, so that many of the theorems which apply to twists can be extended to wrenches.

Using the theorems of Chasles and Poinsot as a starting point, Sir Robert S. Ball developed a complete theory of screws which he published in 1900 [6]. In this chapter, we present a more modern treatment of the theory of screws based on linear algebra and matrix groups. The fundamental tools are the use of homogeneous coordinates to represent rigid motions and the matrix exponential, which maps a twist into the corresponding screw motion. In order to keep the mathematical prerequisites to a minimum, we build up this theory assuming only a good knowledge of basic linear algebra. A more abstract version, using the tools of matrix

Lie groups and Lie algebras, can be found in Appendix A.

There are two main advantages to using screws, twists, and wrenches for describing rigid body kinematics. The first is that they allow a global description of rigid body motion which does not suffer from singularities due to the use of local coordinates. Such singularities are inevitable when one chooses to represent rotation via Euler angles, for example. The second advantage is that screw theory provides a very geometric description of rigid motion which greatly simplifies the analysis of mechanisms. We will make extensive use of the geometry of screws throughout the book, and particularly in the next chapter when we study the kinematics and singularities of mechanisms.

1 Rigid Body Transformations

The motion of a particle moving in Euclidean space is described by giving the location of the particle at each instant of time, relative to an inertial Cartesian coordinate frame. Specifically, we choose a set of three orthonormal axes and specify the particle's location using the triple $(x, y, z) \in \mathbb{R}^3$, where each coordinate gives the projection of the particle's location onto the corresponding axis. A trajectory of the particle is represented by the parameterized curve $p(t) = (x(t), y(t), z(t)) \in \mathbb{R}^3$.

In robotics, we are frequently interested not in the motion of individual particles, but in the collective motion of a set of particles, such as the link of a robot manipulator. To this end, we loosely define a perfectly rigid body as a completely "undistortable" body. More formally, a *rigid body* is a collection of particles such that the distance between any two particles remains fixed, regardless of any motions of the body or forces exerted on the body. Thus, if p and q are any two points on a rigid body then, as the body moves, p and q must satisfy

$$\|p(t) - q(t)\| = \|p(0) - q(0)\| = \text{constant}.$$

A *rigid motion* of an object is a continous movement of the particles in the object such that the distance between any two particles remains fixed at all times. The net movement of a rigid body from one location to another via a rigid motion is called a *rigid displacement*. In general, a rigid displacement may consist of both translation and rotation of the object.

Given an object described as a subset O of $\mathbb{R}^3$, a rigid motion of an object is represented by a continuous family of mappings $g(t) : O \to \mathbb{R}^3$ which describe how individual points in the body move as a function of time, relative to some fixed Cartesian coordinate frame. That is, if we move an object along a continuous path, $g(t)$ maps the initial coordinates of a point on the body to the coordinates of that same point at time t. A rigid displacement is represented by a single mapping $g : O \to \mathbb{R}^3$ which

maps the coordinates of points in the rigid body from their initial to final configurations.

Given two points $p, q \in O$, the *vector* $v \in \mathbb{R}^3$ connecting p to q is defined to be the directed line segment going from p to q. In coordinates this is given by $v = q - p$ with $p, q \in \mathbb{R}^3$. Though both points and vectors are represented by 3-tuples of numbers, they are conceptually quite different. A vector has a direction and a magnitude. (By the magnitude of a vector, we will mean its Euclidean norm, i.e., $\sqrt{v_1^2 + v_2^2 + v_3^2}$.) It is, however, not attached to the body, since there may be other pairs of points on the body, for instance r and s with $q - p = s - r$, for which the same vector v also connects r to s. A vector is sometimes called a *free vector* to indicate that it can be positioned anywhere in space without changing its meaning.

The action of a rigid transformation on points induces an action on vectors in a natural way. If we let $g : O \to \mathbb{R}^3$ represent a rigid displacement, then vectors transform according to

$$g_*(v) = g(q) - g(p).$$

Note that the right-hand side is the difference of two points and is hence also a vector.

Since distances between points on a rigid body are not altered by rigid motions, a necessary condition for a mapping $g : O \to \mathbb{R}^3$ to describe a rigid motion is that distances be preserved by the mapping. However, this condition is not sufficient since it allows internal reflections, which are not physically realizable. That is, a mapping might preserve distance but not preserve orientation. For example, the mapping $(x, y, z) \mapsto (x, y, -z)$ preserves distances but reflects points in the body about the xy plane. To eliminate this possibility, we require that the cross product between vectors in the body also be preserved. We will collect these requirements to define a *rigid body transformation* as a mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$ which represents a rigid motion:

Definition 2.1. Rigid body transformation
A mapping $g : \mathbb{R}^3 \to \mathbb{R}^3$ is a *rigid body transformation* if it satisfies the following properties:

1. Length is preserved: $\|g(p) - g(q)\| = \|p - q\|$ for all points $p, q \in \mathbb{R}^3$.
2. The cross product is preserved: $g_*(v \times w) = g_*(v) \times g_*(w)$ for all vectors $v, w \in \mathbb{R}^3$.

There are some interesting consequences of this definition. The first is that the inner product is preserved by rigid body transformations. One way to show this is to use the *polarization identity*,

$$v_1^T v_2 = \frac{1}{4}(\|v_1 + v_2\|^2 - \|v_1 - v_2\|^2),$$

and the fact that

$$\|v_1 + v_2\| = \|g_*(v_1) + g_*(v_2)\| \qquad \|v_1 - v_2\| = \|g_*(v_1) - g_*(v_2)\|$$

to conclude that for any two vectors v_1, v_2,

$$v_1^T v_2 = g_*(v_1)^T g_*(v_2).$$

In particular, orthogonal vectors are transformed to orthogonal vectors. Coupled with the fact that rigid body transformations also preserve the cross product (property 2 of the definition above), we see that rigid body transformations take orthonormal coordinate frames to orthonormal coordinate frames.

The fact that the distance between points and cross product between vectors is fixed does not mean that it is inadmissible for particles in a rigid body to move relative to each other, but rather that they can rotate but not translate with respect to each other. Thus, to keep track of the motion of a rigid body, we need to keep track of the motion of any one particle on the rigid body and the rotation of the body about this point. In order to do this, we represent the *configuration* of a rigid body by attaching a Cartesian coordinate frame to some point on the rigid body and keeping track of the motion of this body coordinate frame relative to a fixed frame. The motion of the individual particles in the body can then be retrieved from the motion of the body frame and the motion of the point of attachment of the frame to the body. We shall require that all coordinate frames be *right-handed*: given three orthonormal vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^3$ which define a coordinate frame, they must satisfy $\mathbf{z} = \mathbf{x} \times \mathbf{y}$.

Since a rigid body transformation $g : \mathbb{R}^3 \to \mathbb{R}^3$ preserves the cross product, right-handed coordinate frames are transformed to right-handed coordinate frames. The action of a rigid transformation g on the body frame describes how the body frame rotates as a consequence of the rigid motion. More precisely, if we describe the configuration of a rigid body by the right-handed frame given by the vectors v_1, v_2, v_3 attached to a point p, then the configuration of the rigid body after the rigid body transformation g is given by the right-handed frame of vectors $g_*(v_1), g_*(v_2), g_*(v_3)$ attached to the point $g(p)$.

The remainder of this chapter is devoted to establishing more detailed properties, characterizations, and representations of rigid body transformations and providing the necessary mathematical preliminaries used in the remainder of the book.

2 Rotational Motion in $\mathbb{R}^3$

We begin the study of rigid body motion by considering, at the outset, only the rotational motion of an object. We describe the orientation of

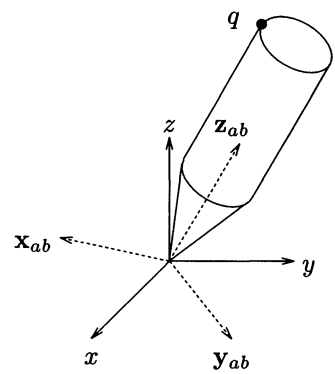


Figure 2.1: Rotation of a rigid object about a point. The dotted coordinate frame is attached to the rotating rigid body.

the body by giving the relative orientation between a coordinate frame attached to the body and a fixed or inertial coordinate frame. From now on, all coordinate frames will be right-handed unless stated otherwise. Let A be the inertial frame, B the body frame, and $\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab} \in \mathbb{R}^3$ the coordinates of the principal axes of B relative to A (see Figure 2.1). Stacking these coordinate vectors next to each other, we define a 3×3 matrix:

$$R_{ab} = \begin{bmatrix} \mathbf{x}_{ab} & \mathbf{y}_{ab} & \mathbf{z}_{ab} \end{bmatrix}.$$

We call a matrix constructed in this manner a *rotation matrix*: every rotation of the object relative to the ground corresponds to a matrix of this form.

2.1 Properties of rotation matrices

A rotation matrix has two key properties that follow from its construction. Let $R \in \mathbb{R}^{3\times 3}$ be a rotation matrix and $r_1, r_2, r_3 \in \mathbb{R}^3$ be its columns. Since the columns of R are mutually orthonormal, it follows that

$$r_i^T r_j = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j. \end{cases}$$

As conditions on the matrix R, these properties can be written as

$$RR^T = R^T R = I. \tag{2.1}$$

From this it follows that

$$\det R = \pm 1.$$

To determine the sign of the determinant of R, we recall from linear algebra that

$$\det R = r_1^T (r_2 \times r_3).$$

Since the coordinate frame is right-handed, we have that $r_2 \times r_3 = r_1$ so that $\det R = r_1^T r_1 = 1$. Thus, coordinate frames corresponding to right-handed frames are represented by orthogonal matrices with determinant 1. The set of all 3×3 matrices which satisfy these two properties is denoted $SO(3)$. The notation SO abbreviates *special orthogonal.* Special refers to the fact that $\det R = +1$ rather than ± 1.

More generally, we may define the space of rotation matrices in $\mathbb{R}^{n \times n}$ by

$$SO(n) = \{R \in \mathbb{R}^{n \times n} : RR^T = I, \det R = +1\}. \tag{2.2}$$

We will be primarily interested in $n = 3$, although the $n = 2$ case (planar rotations) will also prove useful and is explored in the exercises.

$SO(3) \subset \mathbb{R}^{3 \times 3}$ is a *group* under the operation of matrix multiplication. A set G together with a binary operation $\circ$ defined on elements of G is called a group if it satisfies the following axioms:

1. *Closure:* If $g_1, g_2 \in G$, then $g_1 \circ g_2 \in G$.
2. *Identity:* There exists an identity element, e, such that $g \circ e = e \circ g = g$ for every $g \in G$.
3. *Inverse:* For each $g \in G$, there exists a (unique) inverse, $g^{-1} \in G$, such that $g \circ g^{-1} = g^{-1} \circ g = e$.
4. *Associativity:* If $g_1, g_2, g_3 \in G$, then $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.

In the instance of $SO(3)$, note that

1. If $R_1, R_2 \in SO(3)$, then $R_1 R_2 \in SO(3)$ since

$$R_1 R_2 (R_1 R_2)^T = R_1 R_2 R_2^T R_1^T = R_1 R_1^T = I$$
$$\det(R_1 R_2) = \det(R_1) \det(R_2) = +1.$$

2. The identity matrix is the identity element.
3. By equation (2.1) it follows that the inverse of $R \in SO(3)$ is $R^T \in SO(3)$.
4. The associativity of the group operation follows from the associativity of matrix multiplication; that is, $(R_1 R_2) R_3 = R_1 (R_2 R_3)$.

Thus, $SO(3)$ is a group using the identity matrix I as the identity element and matrix multiplication as the group operation. We refer to $SO(3)$ as the *rotation group* of $\mathbb{R}^3$.

Every configuration of a rigid body that is free to rotate relative to a fixed frame can be identified with a unique $R \in SO(3)$. Under this identification, the rotation group $SO(3)$ is referred to as the *configuration*

space of the system and a trajectory of the system is a curve $R(t) \in SO(3)$ for $t \in [0, T]$. More generally, we shall call a set Q a configuration space for a system if every element $x \in Q$ corresponds to a valid configuration of the system and each configuration of the system can be identified with a unique element of Q.

A rotation matrix $R \in SO(3)$ also serves as a transformation, taking coordinates of a point from one frame to another. Consider the point q shown in Figure 2.1. Let $q_b = (x_b, y_b, z_b)$ be the coordinates of q relative to frame B. The coordinates of q relative to frame A can be computed as follows: since $x_b, y_b, z_b \in \mathbb{R}$ are projections of q onto the coordinate axes of B, which, in turn, have coordinates $\mathbf{x}_{ab}, \mathbf{y}_{ab}, \mathbf{z}_{ab} \in \mathbb{R}^3$ with respect to A, the coordinates of q relative to frame A are given by

$$q_a = \mathbf{x}_{ab} x_b + \mathbf{y}_{ab} y_b + \mathbf{z}_{ab} z_b.$$

This can be rewritten as

$$q_a = \begin{bmatrix} \mathbf{x}_{ab} & \mathbf{y}_{ab} & \mathbf{z}_{ab} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = R_{ab} q_b.$$

In other words, R_{ab}, when considered as a map from $\mathbb{R}^3$ to $\mathbb{R}^3$, rotates the coordinates of a point from frame B to frame A.

The action of a rotation matrix on a point can be used to define the action of the rotation matrix on a vector. Let v_b be a vector in the frame B defined as $v_b = q_b - p_b$. Then,

$$R_{ab}(v_b) := R_{ab} q_b - R_{ab} p_b = q_a - p_a = v_a.$$

Since matrix multiplication is linear, it may be verified that if $v_b = q_b - p_b = s_b - r_b$ then we still have that

$$R_{ab} s_b - R_{ab} r_b = R_{ab} q_b - R_{ab} p_b = v_a$$

and hence the action of R_{ab} on a vector is well defined.

Rotation matrices can be combined to form new rotation matrices using matrix multiplication. If a frame C has orientation R_{bc} relative to a frame B, and B has orientation R_{ab} relative to another frame A, then the orientation of C relative to A is given by

$$R_{ac} = R_{ab} R_{bc}. \tag{2.3}$$

R_{ac}, when considered as a map from $\mathbb{R}^3$ to $\mathbb{R}^3$, rotates the coordinates of a point from frame C to frame A by first rotating from C to B and then from B to A. Equation (2.3) is the *composition rule* for rotations.

A rotation matrix represents a rigid body transformation in the sense of the definition of the previous section. This is to say, it preserves

distance and orientation. We prove this using some algebraic properties of the cross product operation between two vectors. Recall that the cross product between two vectors $a, b \in \mathbb{R}^3$ is defined as

$$a \times b = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}.$$

Since the cross product by a is a linear operator, $b \mapsto a \times b$ may be represented using a matrix. Defining

$$(a)^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \tag{2.4}$$

we can write

$$a \times b = (a)^\wedge b. \tag{2.5}$$

We will often use the notation $\widehat{a}$ as a replacement for $(a)^\wedge$.

Lemma 2.1. *Given $R \in SO(3)$ and $v, w \in \mathbb{R}^3$, the following properties hold:*

$$R(v \times w) = (Rv) \times (Rw) \tag{2.6}$$

$$R(w)^\wedge R^T = (Rw)^\wedge. \tag{2.7}$$

The first property in the lemma asserts that rotation by the matrix R commutes with the cross product operation; that is, the rotation of the cross product of two vectors is the cross product of the rotation of each of the vectors by R. The second property has an interpretation in terms of rotation of an instantaneous axis of rotation, which will become clear shortly. For now, we will merely use it as an algebraic fact. The proof of the lemma is by calculation.

Proposition 2.2. Rotations are rigid body transformations
A rotation $R \in SO(3)$ is a rigid body transformation; that is,

1. *R preserves distance: $\|Rq - Rp\| = \|q - p\|$ for all $q, p \in \mathbb{R}^3$.*
2. *R preserves orientation: $R(v \times w) = Rv \times Rw$ for all $v, w \in \mathbb{R}^3$.*

Proof. Property 1 can be verified by direct calculation:

$$\begin{aligned} \|Rq - Rp\|^2 &= (R(q-p))^T (R(q-p)) = (q-p)^T R^T R (q-p) \\ &= (q-p)^T (q-p) = \|q-p\|^2. \end{aligned}$$

Property 2 follows from equation (2.6). □

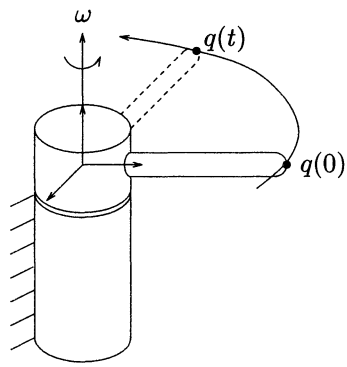


Figure 2.2: Tip point trajectory generated by rotation about the ω-axis.

2.2 Exponential coordinates for rotation

A common motion encountered in robotics is the rotation of a body about a given axis by some amount. For example, we might wish to describe the rotation of the link of a robot about a fixed axis, as shown in Figure 2.2. Let $\omega \in \mathbb{R}^3$ be a unit vector which specifies the direction of rotation and let $\theta \in \mathbb{R}$ be the angle of rotation in radians. Since every rotation of the object corresponds to some $R \in SO(3)$, we would like to write R as a function of ω and θ.

To motivate our derivation, consider the velocity of a point q attached to the rotating body. If we rotate the body at *constant* unit velocity about the axis ω, the velocity of the point, $\dot{q}$, may be written as

$$\dot{q}(t) = \omega \times q(t) = \widehat{\omega} q(t). \tag{2.8}$$

This is a time-invariant linear differential equation which may be integrated to give

$$q(t) = e^{\widehat{\omega} t} q(0),$$

where $q(0)$ is the initial ($t = 0$) position of the point and $e^{\widehat{\omega} t}$ is the matrix exponential

$$e^{\widehat{\omega} t} = I + \widehat{\omega} t + \frac{(\widehat{\omega} t)^2}{2!} + \frac{(\widehat{\omega} t)^3}{3!} + \cdots$$

It follows that if we rotate about the axis ω at unit velocity for θ units of time, then the net rotation is given by

$$R(\omega, \theta) = e^{\widehat{\omega} \theta}. \tag{2.9}$$

From its definition, it is easy to see that the matrix $\widehat{\omega}$ is a skew-symmetric matrix, i.e., it satisfies $\widehat{\omega}^T = -\widehat{\omega}$. The vector space of all 3×3

skew matrices is denoted $so(3)$ and more generally the space of $n \times n$ skew-symmetric matrices is

$$so(n) = \{S \in \mathbb{R}^{n \times n} : S^T = -S\}. \tag{2.10}$$

(The reason for the notation $so(n)$ will become clear shortly.) As with $SO(n)$, the cases we are interested in are $n = 2$ and $n = 3$. We concentrate on $n = 3$ here and explore $n = 2$ in the exercises.

The set $so(3) \subset \mathbb{R}^{3\times 3}$ is a vector space over the reals. Thus, the sum of two elements of $so(3)$ is an element of $so(3)$ and the scalar multiple of any element of $so(3)$ is an element of $so(3)$. Furthermore, we can identify $so(3)$ with $\mathbb{R}^3$ using the relationship (2.4) and the fact $(v + w)^\wedge = \widehat{v} + \widehat{w}$.

It will be convenient to represent a skew-symmetric matrix as the product of a *unit* skew-symmetric matrix and a real number. Given a matrix $\widehat{\omega} \in so(3)$, $\|\omega\| = 1$, and a real number $\theta \in \mathbb{R}$, we write the exponential of $\widehat{\omega}\theta$ as

$$\exp(\widehat{\omega}\theta) = e^{\widehat{\omega}\theta} = I + \theta\widehat{\omega} + \frac{\theta^2}{2!}\widehat{\omega}^2 + \frac{\theta^3}{3!}\widehat{\omega}^3 + \dots \tag{2.11}$$

Equation (2.11) is an infinite series and, hence, not useful from a computational standpoint. To obtain a closed-form expression for $\exp(\widehat{\omega}\theta)$, we make use of the following formulas for powers of $\widehat{a}$, which are verified by direct calculation.

Lemma 2.3. *Given $\widehat{a} \in so(3)$, the following relations hold:*

$$\widehat{a}^2 = aa^T - \|a\|^2 I \tag{2.12}$$

$$\widehat{a}^3 = -\|a\|^2 \widehat{a} \tag{2.13}$$

and higher powers of $\widehat{a}$ can be calculated recursively.

Utilizing this lemma with $a = \omega\theta$, $\|\omega\| = 1$, equation (2.11) becomes

$$e^{\widehat{\omega}\theta} = I + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots\right)\widehat{\omega} + \left(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} - \cdots\right)\widehat{\omega}^2$$

and hence

$$\boxed{e^{\widehat{\omega}\theta} = I + \widehat{\omega}\sin\theta + \widehat{\omega}^2(1 - \cos\theta)} \tag{2.14}$$

This formula, commonly referred to as *Rodrigues' formula*, gives an efficient method for computing $\exp(\widehat{\omega}\theta)$. When $\|\omega\| \neq 1$, it may be verified (see Exercise 12) that

$$e^{\widehat{\omega}\theta} = I + \frac{\widehat{\omega}}{\|\omega\|}\sin(\|\omega\|\theta) + \frac{\widehat{\omega}^2}{\|\omega\|^2}\big(1 - \cos(\|\omega\|\theta)\big).$$

We now verify that $\exp(\widehat{\omega}\theta)$ is indeed a rotation matrix.

Proposition 2.4. Exponentials of skew matrices are orthogonal *Given a skew-symmetric matrix $\widehat{\omega} \in so(3)$ and $\theta \in \mathbb{R}$,*

$$e^{\widehat{\omega}\theta} \in SO(3).$$

Proof. Defining $R := \exp(\widehat{\omega}\theta)$, we must verify that $R^T R = I$ and $\det R = +1$. To verify the first property, we have the following chain of equalities, which can be checked using equation (2.14),

$$\left[e^{\widehat{\omega}\theta}\right]^{-1} = e^{-\widehat{\omega}\theta} = e^{\widehat{\omega}^T\theta} = \left[e^{\widehat{\omega}\theta}\right]^T.$$

Thus $R^{-1} = R^T$ and consequently $R^T R = I$ as desired. From this, it follows that $\det R = \pm 1$. Using the continuity of the determinant as a function of the entries of a matrix, combined with continuity of the exponential map and the fact that $\det \exp(0) = 1$, we conclude that $\det R = +1$. □

Proposition 2.4 asserts that the exponential map transforms skew-symmetric matrices into orthogonal matrices. Geometrically, the skew-symmetric matrix corresponds to an axis of rotation (via the mapping $\omega \mapsto \widehat{\omega}$) and the exponential map generates the rotation corresponding to rotation about the axis by a specified amount θ. This relationship between skew-symmetric matrices and orthogonal matrices explains, in part, the notation $so(3)$. We will now show that *every* rotation matrix can be represented as the matrix exponential of some skew-symmetric matrix; that is, the map $\exp : so(3) \to SO(3)$ is surjective (onto).

Proposition 2.5. The exponential map is surjective onto $SO(3)$ *Given $R \in SO(3)$, there exists $\omega \in \mathbb{R}^3$, $\|\omega\| = 1$ and $\theta \in \mathbb{R}$ such that $R = \exp(\widehat{\omega}\theta)$.*

Proof. The proof is constructive. We equate terms of R and $\exp(\widehat{\omega}\theta)$ and solve the corresponding equations. By way of notation, we have the rotation matrix R to be

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{2.15}$$

Defining $v_\theta = 1 - \cos\theta, c_\theta = \cos\theta$, and $s_\theta = \sin\theta$, write equation (2.14) as

$$\begin{aligned} e^{\widehat{\omega}\theta} &= I + \widehat{\omega}\sin\theta + \widehat{\omega}^2(1 - \cos\theta) \\ &= \begin{bmatrix} 1 - v_\theta(\omega_2^2 + \omega_3^2) & \omega_1\omega_2 v_\theta - \omega_3 s_\theta & \omega_1\omega_3 v_\theta + \omega_2 s_\theta \\ \omega_1\omega_2 v_\theta + \omega_3 s_\theta & 1 - v_\theta(\omega_1^2 + \omega_3^2) & \omega_2\omega_3 v_\theta - \omega_1 s_\theta \\ \omega_1\omega_3 v_\theta - \omega_2 s_\theta & \omega_2\omega_3 v_\theta + \omega_1 s_\theta & 1 - v_\theta(\omega_1^2 + \omega_2^2) \end{bmatrix} \\ &= \begin{bmatrix} \omega_1^2 v_\theta + c_\theta & \omega_1\omega_2 v_\theta - \omega_3 s_\theta & \omega_1\omega_3 v_\theta + \omega_2 s_\theta \\ \omega_1\omega_2 v_\theta + \omega_3 s_\theta & \omega_2^2 v_\theta + c_\theta & \omega_2\omega_3 v_\theta - \omega_1 s_\theta \\ \omega_1\omega_3 v_\theta - \omega_2 s_\theta & \omega_2\omega_3 v_\theta + \omega_1 s_\theta & \omega_3^2 v_\theta + c_\theta \end{bmatrix}. \end{aligned} \tag{2.16}$$

Equating (2.15) with (2.16), we see that

$$\text{trace}(R) = r_{11} + r_{22} + r_{33} = 1 + 2\cos\theta.$$

To verify that this equation has a solution, we recall that the trace of R is equal to the sum of its eigenvalues. Since R preserves lengths and $\det R = +1$, its eigenvalues have magnitude 1 and occur in complex conjugate pairs (see Exercise 3). It follows that $-1 \leq \text{trace}(R) \leq 3$ and hence we can set

$$\theta = \cos^{-1}\left(\frac{\text{trace}(R) - 1}{2}\right). \tag{2.17}$$

Note that there is an ambiguity in the value of θ, in the sense that $\theta \pm 2\pi n$ or $-\theta \pm 2\pi n$ could be chosen as well.

Now, equating the off-diagonal terms of R and $\exp(\widehat{\omega}\theta)$, we get

$$\begin{aligned} r_{32} - r_{23} &= 2\omega_1 s_\theta \\ r_{13} - r_{31} &= 2\omega_2 s_\theta \\ r_{21} - r_{12} &= 2\omega_3 s_\theta. \end{aligned}$$

If $\theta \neq 0$, we choose

$$\omega = \frac{1}{2s_\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \tag{2.18}$$

Note that if $2\pi - \theta$ had been chosen earlier in equation (2.17), the axis of rotation would have been $-\omega$. Indeed, the exponential map is a many-to-one map from $\mathbb{R}^3$ onto $SO(3)$. If $R = I$, then $\text{trace}(R) = 3$ and hence $\theta = 0$ and ω can be chosen arbitrarily. If $R \neq I$, the above construction shows that there are two distinct ω and $\theta \in [0, 2\pi)$ such that $R = \exp(\widehat{\omega}\theta)$. □

The components of the vector $\omega\theta \in \mathbb{R}^3$ given by equations (2.17) and (2.18) are called the *exponential coordinates* for R. Considering $\omega \in \mathbb{R}^3$ to be an axis of rotation with unit magnitude and $\theta \in \mathbb{R}$ to be an angle, Propositions 2.4 and 2.5 combine to give the following classic theorem.

Theorem 2.6 (Euler). *Any orientation $R \in SO(3)$ is equivalent to a rotation about a fixed axis $\omega \in \mathbb{R}^3$ through an angle $\theta \in [0, 2\pi)$.*

This method of representing a rotation is also known as the *equivalent axis representation.* We note from the preceding proof that this representation is *not* unique since choosing $\omega' = -\omega$ and $\theta' = 2\pi - \theta$ gives the same rotation as ω and θ. Furthermore, if we insist that ω have unit magnitude, then ω is arbitrary for $R = I$ (by choosing $\theta = 0$). The former problem is a consequence of the exponential map being many-to-one and

the latter is referred to as a *singularity* of the equivalent axis representation, alluding to the fact that one may lose smooth dependence of the equivalent axis as a function of the orientation R at $R = I$.

2.3 Other representations

The exponential coordinates are called the *canonical* coordinates of the rotation group. Other coordinates for the rotation group also exist and are briefly described below and in the exercises. We emphasize the connection of these other representations with the exponential coordinates presented above; more classical treatments of these representations can be found in standard kinematics texts.

Euler angles

One method of describing the orientation of a coordinate frame B relative to another coordinate frame A is as follows: start with frame B coincident with frame A. First, rotate the B frame about the z-axis of frame B (at this time coincident with frame A) by an angle α, then rotate about the (new) y-axis of frame B by an angle β, and then rotate about the (once again, new) z-axis of frame B by an angle γ. This yields a net orientation $R_{ab}(\alpha, \beta, \gamma)$ and the triple of angles (α, β, γ) is used to represent the rotation.

The angles (α, β, γ) are called the *ZYZ Euler angles*. Since all rotations are performed about the principal axes of the moving frame, we define the following elementary rotations about the x-, y-, and z-axes:

$$R_{\mathbf{x}}(\phi) := e^{\widehat{\mathbf{x}}\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix},$$

$$R_{\mathbf{y}}(\beta) := e^{\widehat{\mathbf{y}}\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},$$

and

$$R_{\mathbf{z}}(\alpha) := e^{\widehat{\mathbf{z}}\alpha} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

To derive the final orientation of frame B, it is easiest to derive the formula by viewing the rotation with B considered as the fixed frame, since then all rotations then occur around fixed axes. The appropriate sequence of rotations for the frame A, considering the B frame as fixed, is

$$R_{ba} = R_{\mathbf{z}}(-\gamma) R_{\mathbf{y}}(-\beta) R_{\mathbf{z}}(-\alpha).$$

Inverting this expression gives the rotation matrix of B relative to A:

$$\begin{aligned} R_{ab} &= R_{\mathbf{z}}(\alpha)R_{\mathbf{y}}(\beta)R_{\mathbf{z}}(\gamma) \\ &= \begin{bmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{bmatrix}. \end{aligned} \tag{2.19}$$

Here c_α, s_α are abbreviations for $\cos\alpha$ and $\sin\alpha$, respectively, and similarly for the other terms.

It is clear that any matrix of the form in equation (2.19) is an orthogonal matrix (since it is a composition of elementary rotations). As in the case of the exponential map, the converse question of whether the map from $(\alpha, \beta, \gamma) \to SO(3)$ is surjective is an important one. The answer to this question is affirmative: given a rotation $R \in SO(3)$, the Euler angles can be computed by solving equation (2.19) for α, β, and γ. For example, when $\sin\beta \neq 0$, the solutions are

$$\begin{aligned} \beta &= \text{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}) \\ \alpha &= \text{atan2}(r_{23}/s_\beta, r_{13}/s_\beta) \\ \gamma &= \text{atan2}(r_{32}/s_\beta, -r_{31}/s_\beta), \end{aligned} \tag{2.20}$$

where $\text{atan2}(y, x)$ computes $\tan^{-1}(y/x)$ but uses the sign of both x and y to determine the quadrant in which the resulting angle lies.

ZYZ Euler angles are an example of a *local* parameterization of $SO(3)$. As in the case of the equivalent axis representation, singularities in the parameterization (referring to the lack of existence of global, smooth solutions to the inverse problem of determining the Euler angles from the rotation) occur at $R = I$, the identity rotation. In particular, we note that (α, β, γ) of the form $(\alpha, 0, -\alpha)$ yields $R_{ab}(\alpha, 0, -\alpha) = I$. Thus, there are infinitely many representations of the identity rotation in the *ZYZ* Euler angles parameterization.

Other types of Euler angle parameterizations may be devised by using different ordered sets of rotation axes. Common choices include *ZYX* axes (Fick angles) and *YZX* axes (Helmholtz angles). The *ZYX* Euler angles are also referred to as the yaw, pitch, and roll angles, with R_{ab} defined by rotating about the x-axis in the body frame (roll), then the y-axis in the body frame (pitch), and finally the z-axis in the body frame (yaw). Both the *ZYX* and *YZX* Euler angle parameterizations have the advantage of not having a singularity at the identity orientation, $R = I$, though they do contain singularities at other, different, orientations. For example, in the instance of *ZYX* Euler angles, we have:

$$R_{ab} = R_{\mathbf{z}}(\psi)R_{\mathbf{y}}(\theta)R_{\mathbf{x}}(\phi) = e^{\widehat{\mathbf{z}}\psi}e^{\widehat{\mathbf{y}}\theta}e^{\widehat{\mathbf{x}}\phi},$$

which is singular when $\theta = -\pi/2$. It is a fundamental topological fact that singularities can never be eliminated in any 3-dimensional represen-

tation of $SO(3)$. This situation is similar to that of attempting to find a global coordinate chart on a sphere, which also fails.

Quaternions

Quaternions generalize complex numbers and can be used to represent rotations in much the same way as complex numbers on the unit circle can be used to represent planar rotations. Unlike Euler angles, quaternions give a global parameterization of $SO(3)$, at the cost of using four numbers instead of three to represent a rotation.

Formally, a quaternion is a vector quantity of the form

$$Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \qquad q_i \in \mathbb{R}, i = 0, \ldots, 3,$$

where q_0 is the *scalar* component of Q and $\vec{q} = (q_1, q_2, q_3)$ is the *vector* component. A convenient shorthand notation is $Q = (q_0, \vec{q})$ with $q_0 \in \mathbb{R}$, $\vec{q} \in \mathbb{R}^3$. The set of quaternions $\mathbb{Q}$ is a 4-dimensional vector space over the reals and forms a group with respect to quaternion multiplication, denoted "$\cdot$". Multiplication is distributive and associative, but *not* commutative; it satisfies the relations

$$a\mathbf{i} = \mathbf{i}a \qquad a\mathbf{j} = \mathbf{j}a \qquad a\mathbf{k} = \mathbf{k}a \qquad a \in \mathbb{R}$$
$$\mathbf{i}\cdot\mathbf{i} = \mathbf{j}\cdot\mathbf{j} = \mathbf{k}\cdot\mathbf{k} = \mathbf{i}\cdot\mathbf{j}\cdot\mathbf{k} = -1$$
$$\mathbf{i}\cdot\mathbf{j} = -\mathbf{j}\cdot\mathbf{i} = \mathbf{k} \qquad \mathbf{j}\cdot\mathbf{k} = -\mathbf{k}\cdot\mathbf{j} = \mathbf{i} \qquad \mathbf{k}\cdot\mathbf{i} = -\mathbf{i}\cdot\mathbf{k} = \mathbf{j}$$

The *conjugate* of a quaternion $Q = (q_0, \vec{q})$ is given by $Q^* = (q_0, -\vec{q})$ and the magnitude of a quaternion satisfies

$$\|Q\|^2 = Q \cdot Q^* = q_0^2 + q_1^2 + q_2^2 + q_3^2.$$

It is straightforward to verify that the inverse of a quaternion is $Q^{-1} = Q^*/\|Q\|^2$ and that $Q = (1, 0)$ is the identity element for quaternion multiplication.

The product between two quaternions has a simple form in terms of the inner and cross products between vectors in $\mathbb{R}^3$. Let $Q = (q_0, \vec{q})$ and $P = (p_0, \vec{p})$ be quaternions, where $q_0, p_0 \in \mathbb{R}$ are the scalar parts of Q and P and $\vec{q}, \vec{p}$ are the vector parts. It can be shown algebraically that the product of two quaternions satisfies:

$$Q \cdot P = (q_0 p_0 - \vec{q}\cdot\vec{p},\ q_0\vec{p} + p_0\vec{q} + \vec{q}\times\vec{p}).$$

In most applications, this formula eliminates the need to make direct use of the multiplicative relations given above.

The *unit quaternions* are the subset of all $Q \in \mathbb{Q}$ such that $\|Q\| = 1$. The unit quaternions also form a group with respect to quaternion

multiplication (Exercise 6). Given a rotation matrix $R = \exp(\widehat{\omega}\theta)$, we define the associated unit quaternion as

$$Q = \big(\cos(\theta/2), \omega \sin(\theta/2)\big),$$

where $\omega \in \mathbb{R}^3$ represents the unit axis of rotation and $\theta \in \mathbb{R}$ represents the angle of rotation. A detailed calculation shows that if Q_{ab} represents a rotation between frame A and frame B, and Q_{bc} represents a rotation between frames B and C, then the rotation between A and C is given by the quaternion

$$Q_{ac} = Q_{ab} \cdot Q_{bc}.$$

Thus, the group operation on unit quaternions directly corresponds to the group operation for rotations. Given a unit quaternion $Q = (q_0, \vec{q})$, we can extract the corresponding rotation by setting

$$\theta = 2\cos^{-1} q_0 \qquad \omega = \begin{cases} \dfrac{\vec{q}}{\sin(\theta/2)} & \text{if } \theta \neq 0, \\ 0 & \text{otherwise,} \end{cases}$$

and $R = \exp(\widehat{\omega}\theta)$.

Since the group structure for quaternions directly corresponds to that of rotations, quaternions provide an efficient representation for rotations which do not suffer from singularities. Their properties are explored more fully in the exercises.

3 Rigid Motion in $\mathbb{R}^3$

Recall from Section 1 that a rigid motion is one that preserves the distance between points and the angle between vectors. We represent rigid motions by using rigid body transformations to describe the instantaneous position and orientation of a body coordinate frame relative to an inertial frame. This representation relies on the fact that rigid body transformations map right-handed, orthonormal frames to right-handed, orthonormal frames, thus preserving distance and angles. In this book we refer to all transformations between coordinate frames as rigid body transformations (or just rigid transformations), whether or not a rigid body is explicitly present.

In general, rigid motions consist of rotation and translation. In the preceding section, we discussed representations of pure rotational motion. The procedure for representing pure translational motion is very simple: choose a (any) point in the body and keep track of the coordinates of the point relative to some known frame. This gives a curve $p(t) \in \mathbb{R}^3, t \in [0, T]$, for a trajectory of the entire rigid body.

The representation of general rigid body motion, involving both translation and rotation, is more involved. We describe the position and orientation of a coordinate frame B attached to the body relative to an

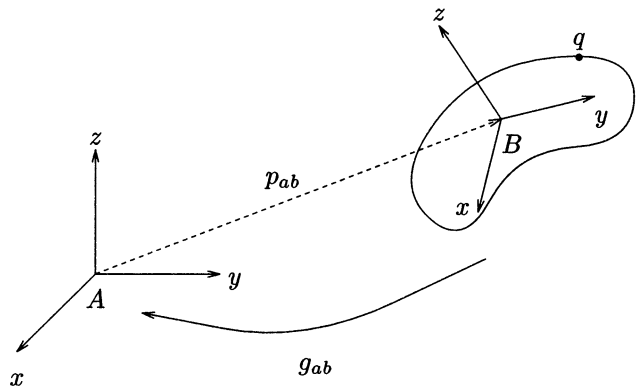


Figure 2.3: Coordinate frames for specifying rigid motions.

inertial frame A (see Figure 2.3). Let $p_{ab} \in \mathbb{R}^3$ be the position vector of the origin of frame B from the origin of frame A, and $R_{ab} \in SO(3)$ the orientation of frame B, relative to frame A. A configuration of the system consists of the pair (p_{ab}, R_{ab}), and the configuration space of the system is the product space of $\mathbb{R}^3$ with $SO(3)$, which shall be denoted as $SE(3)$ (for special Euclidean group):

$$SE(3) = \{(p, R) : p \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3). \quad (2.21)$$

We defer the proof of the fact that $SE(3)$ is a group to the next subsection. As in the case of $SO(3)$, there is a generalization to n dimensions,

$$SE(n) := \mathbb{R}^n \times SO(n).$$

Analogous to the rotational case, an element $(p, R) \in SE(3)$ serves as both a specification of the configuration of a rigid body and a transformation taking the coordinates of a point from one frame to another. More precisely, let $q_a, q_b \in \mathbb{R}^3$ be the coordinates of a point q relative to frames A and B, respectively. Given q_b, we can find q_a by a transformation of coordinates:

$$q_a = p_{ab} + R_{ab} q_b \quad (2.22)$$

where $g_{ab} = (p_{ab}, R_{ab}) \in SE(3)$ is the specification of the configuration of the B frame relative to the A frame. By an abuse of notation, we write $g(q)$ to denote the action of a rigid transformation on a point,

$$g(q) = p + Rq,$$

so that $q_a = g_{ab}(q_b)$.

The action of a rigid transformation $g = (p, R)$ on a vector $v = s - r$ is defined by the following formula:

$$g_*(v) := g(s) - g(r) = R(s - r) = Rv.$$

Thus, a vector is transformed by rotation.

3.1 Homogeneous representation

The transformation of points and vectors by rigid transformations has a simple representation in terms of matrices and vectors in $\mathbb{R}^4$. We begin by adopting some notation. We append 1 to the coordinates of a point to yield a vector in $\mathbb{R}^4$,

$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{bmatrix}.$$

These are called the *homogeneous coordinates* of the point q. Thus, the origin has the form

$$\bar{O} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Vectors, which are the difference of points, then have the form

$$\bar{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix}.$$

Note that the form of the vector is different from that of a point. The 0 and 1 in the fourth component of vectors and points, respectively, will remind us of the difference between points and vectors and enforce a few rules of syntax:

1. Sums and differences of vectors are vectors.
2. The sum of a vector and a point is a point.
3. The difference between two points is a vector.
4. The sum of two points is meaningless.

The transformation $q_a = g_{ab}(q_b)$ given in equation (2.22) is an *affine* transformation. Using the preceding notation for points, we may represent it in *linear* form by writing it as

$$\bar{q}_a = \begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix} =: \bar{g}_{ab}\bar{q}_b.$$

The 4×4 matrix $\bar{g}_{ab}$ is called the *homogeneous representation* of $g_{ab} \in SE(3)$. In general, if $g = (p, R) \in SE(3)$, then

$$\bar{g} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}. \tag{2.23}$$

The price to be paid for the convenience of having a homogeneous or linear representation of the rigid body motion is the increase in the dimension of the quantities involved from 3 to 4.

The last row of the matrix of equation (2.23) appears to be "extra baggage" as well. However, in the graphics literature, the number 1 is frequently replaced by a scalar constant which is either greater than 1 to represent dilation or less than 1 to represent contraction. Also, the row vector of zeros in the last row may be replaced by some other row vector to provide "perspective transformations." In both these instances, of course, the transformation represented by the augmented matrix no longer corresponds to a rigid displacement.

Rigid body transformations can be composed to form new rigid body transformations. Let $g_{bc} \in SE(3)$ be the configuration of a frame C relative to a frame B, and g_{ab} the configuration of frame B relative to another frame A. Then, using equation (2.23), the configuration of C relative to frame A is given by

$$\bar{g}_{ac} = \bar{g}_{ab}\,\bar{g}_{bc} = \begin{bmatrix} R_{ab}R_{bc} & R_{ab}p_{bc} + p_{ab} \\ 0 & 1 \end{bmatrix}. \tag{2.24}$$

Equation (2.24) defines the *composition rule* for rigid body transformations to be the standard matrix multiplication. Using the homogeneous representation, it may be verified that the set of rigid transformations is a group; that is:

1. If $g_1, g_2 \in SE(3)$, then $g_1 g_2 \in SE(3)$.
2. The 4×4 identity element, I, is in $SE(3)$.
3. If $g \in SE(3)$, then the inverse of $\bar{g}$ is determined by straightforward matrix inversion to be:

$$\bar{g}^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \in SE(3)$$

   so that $g^{-1} = (-R^T p, R^T)$.
4. The composition rule for rigid body transformations is associative.

Using the homogeneous representation for a vector $v = s - r$, we obtain the representation for a rigid body transformation of v by multiplying the homogeneous representations of v by the homogeneous representation of g,

$$\bar{g}_* \bar{v} = \bar{g}(\bar{s}) - \bar{g}(\bar{r}) = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix}.$$

Note that by defining the homogeneous representation of a vector to have a zero in the bottom row, we are able to once again use matrix multiplication to represent the action of a rigid transformation, this time on

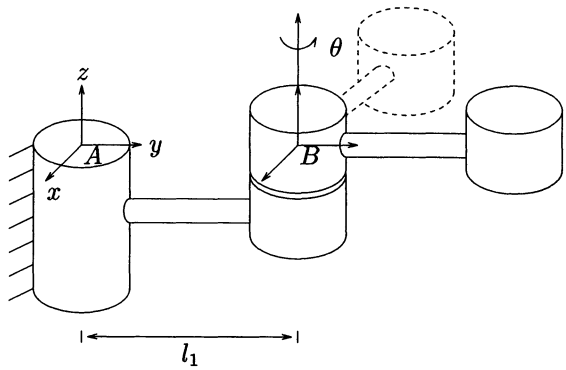


Figure 2.4: Rigid body motion generated by rotation about a fixed axis.

a vector instead of a point. For notational simplicity, in what follows we will confuse homogeneous representations and the abstract representation of points, vectors, and rigid body transformations. Thus, we will write gq and gv instead of $\bar{g}\bar{q}$ and $\bar{g}_*\bar{v}$.

The next proposition establishes that elements of $SE(3)$ are indeed rigid body transformations; namely, that they preserve angles between vectors and distances between points.

Proposition 2.7. Elements of $SE(3)$ represent rigid motions
Any $g \in SE(3)$ is a rigid body transformation:

1. *g preserves distance between points:*

$$\|gq - gp\| = \|q - p\| \quad \textit{for all points } q, p \in \mathbb{R}^3.$$

2. *g preserves orientation between vectors:*

$$g_*(v \times w) = g_*v \times g_*w \quad \textit{for all vectors } v, w \in \mathbb{R}^3.$$

Proof. The proofs follow directly from the corresponding proofs for rotation matrices:

$$\|gq_1 - gq_2\| = \|Rq_1 - Rq_2\| = \|q_1 - q_2\|$$
$$g_*v \times g_*w = Rv \times Rw = R(v \times w).$$

□

Example 2.1. Rotation about a line
Consider the motion of a rigid body rotated about a line in the z direction, through the point $(0, l_1, 0) \in \mathbb{R}^3$, as shown in Figure 2.4. If we let θ denote

the amount of rotation, then the orientation of coordinate frame B with respect to A is

$$R_{ab} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The coordinates for the origin of frame B are

$$p_{ab} = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix},$$

again relative to frame A. The homogeneous representation of the configuration of the rigid body is given by

$$g_{ab}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that when the angle $\theta = 0$, $g_{ab}(0)$ gives that the relative displacement between the two frames is a pure translation along the y-axis.

3.2 Exponential coordinates for rigid motion and twists

The notion of the exponential mapping introduced in Section 2 for $SO(3)$ can be generalized to the Euclidean group, $SE(3)$. We will make extensive use of this representation in the sequel since it allows an elegant, rigorous, and geometric treatment of spatial rigid body motion. We begin by presenting a pair of motivational examples and then present a formal set of definitions.

Consider the simple example of a one-link robot as shown in Figure 2.5a, where the axis of rotation is $\omega \in \mathbb{R}^3$, $\|\omega\| = 1$, and $q \in \mathbb{R}^3$ is a point on the axis. Assuming that the link rotates with unit velocity, then the velocity of the tip point, $p(t)$, is

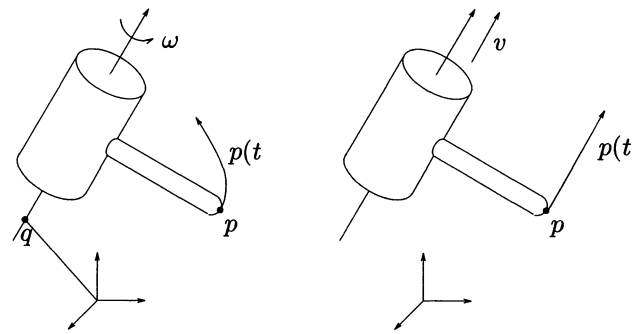
$$\dot{p}(t) = \omega \times (p(t) - q). \tag{2.25}$$

This equation can be conveniently converted into homogeneous coordinates by defining the 4×4 matrix $\widehat{\xi}$ to be

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}, \tag{2.26}$$

with $v = -\omega \times q$. Equation (2.25) can then be rewritten with an extra row appended to it as

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & -\omega \times q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \widehat{\xi} \begin{bmatrix} p \\ 1 \end{bmatrix} \quad \Longrightarrow \quad \dot{\bar{p}} = \widehat{\xi}\bar{p}.$$



(a) (b)

Figure 2.5: (a) A revolute joint and (b) a prismatic joint.

The solution of the differential equation is given by

$$\bar{p}(t) = e^{\widehat{\xi} t}\bar{p}(0),$$

where $e^{\widehat{\xi} t}$ is the matrix exponential of the 4×4 matrix $\widehat{\xi} t$, defined (as usual) by

$$e^{\widehat{\xi} t} = I + \widehat{\xi} t + \frac{(\widehat{\xi} t)^2}{2!} + \frac{(\widehat{\xi} t)^3}{3!} + \cdots$$

The scalar t is the total amount of rotation (since we are rotating with unit velocity). $\exp(\widehat{\xi} t)$ is a mapping from the initial location of a point to its location after rotating t radians.

In a similar manner, we can represent the transformation due to translational motion as the exponential of a 4×4 matrix. The velocity of a point attached to a prismatic joint moving with unit velocity (see Figure 2.5b) is

$$\dot{p}(t) = v. \tag{2.27}$$

Again, the solution of equation (2.27) can be written as $\exp(\widehat{\xi} t)\bar{p}(0)$, where t is the total amount of translation and

$$\widehat{\xi} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}. \tag{2.28}$$

The 4×4 matrix $\widehat{\xi}$ given in equations (2.26) and (2.28) is the generalization of the skew-symmetric matrix $\widehat{\omega} \in so(3)$. Analogous to the definition of $so(3)$, we define

$$se(3) := \{(v, \widehat{\omega}) : v \in \mathbb{R}^3, \widehat{\omega} \in so(3)\}. \tag{2.29}$$

In homogeneous coordinates, we write an element $\widehat{\xi} \in se(3)$ as

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}.$$

An element of $se(3)$ is referred to as a *twist*, or a (infinitesimal) generator of the Euclidean group. We define the $\vee$ (vee) operator to extract the 6-dimensional vector which parameterizes a twist,

$$\begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{2.30}$$

and call $\xi := (v, w)$ the *twist coordinates* of $\widehat{\xi}$. The inverse operator, $\wedge$ (wedge), forms a matrix in $se(3)$ out of a given vector in $\mathbb{R}^6$:

$$\begin{bmatrix} v \\ \omega \end{bmatrix}^{\wedge} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}. \tag{2.31}$$

Thus, $\xi \in \mathbb{R}^6$ represents the twist coordinates for the twist $\widehat{\xi} \in se(3)$; this parallels our notation for skew-symmetric matrices.

Proposition 2.8. Exponential map from $se(3)$ to $SE(3)$
Given $\widehat{\xi} \in se(3)$ and $\theta \in \mathbb{R}$, the exponential of $\widehat{\xi}\theta$ is an element of $SE(3)$, i.e.,

$$e^{\widehat{\xi}\theta} \in SE(3).$$

Proof. The proof is by explicit calculation. In the course of the proof, we will obtain a formula for $\exp(\widehat{\xi}\theta)$. Write $\widehat{\xi}$ as

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}.$$

Case 1 ($\omega = 0$). If $\omega = 0$, then a straightforward calculation shows that

$$\widehat{\xi}^2 = \widehat{\xi}^3 = \widehat{\xi}^4 = \cdots = 0$$

so that $\exp(\widehat{\xi}\theta) = I + \widehat{\xi}\theta$ and hence

$$\boxed{e^{\widehat{\xi}\theta} = \begin{bmatrix} I & v\theta \\ 0 & 1 \end{bmatrix} \qquad \omega = 0} \tag{2.32}$$

which is in $SE(3)$ as desired.

Case 2 ($\omega \neq 0$). Assume $\|\omega\| = 1$, by appropriate scaling of θ if necessary, and define a rigid transformation g by

$$g = \begin{bmatrix} I & \omega \times v \\ 0 & 1 \end{bmatrix}. \tag{2.33}$$

Now, using the calculation of Lemma 2.3, with $\|\omega\| = 1$, we have

$$\begin{aligned}\widehat{\xi}' &= g^{-1}\widehat{\xi}g \\ &= \begin{bmatrix} I & -\omega \times v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & \omega \times v \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \widehat{\omega} & \omega\omega^T v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & h\omega \\ 0 & 0 \end{bmatrix},\end{aligned} \tag{2.34}$$

where $h := \omega^T v$. Using the following identity (see Exercise 8),

$$e^{\widehat{\xi}\theta} = e^{g(\widehat{\xi}'\theta)g^{-1}} = ge^{\widehat{\xi}'\theta}g^{-1}, \tag{2.35}$$

it suffices to calculate $\exp(\widehat{\xi}'\theta)$. This simplifies the calculation since it may be verified (using $\widehat{\omega}\omega = \omega \times \omega = 0$) that

$$(\widehat{\xi}')^2 = \begin{bmatrix} \widehat{\omega}^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad (\widehat{\xi}')^3 = \begin{bmatrix} \widehat{\omega}^3 & 0 \\ 0 & 0 \end{bmatrix}, \quad \cdots$$

Hence,

$$e^{\widehat{\xi}'\theta} = \begin{bmatrix} e^{\widehat{\omega}\theta} & h\omega\theta \\ 0 & 1 \end{bmatrix},$$

and using equation (2.35) it follows that

$$\boxed{e^{\widehat{\xi}\theta} = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix} \qquad \omega \neq 0} \tag{2.36}$$

which is an element of $SE(3)$. □

The transformation $g = \exp(\widehat{\xi}\theta)$ is slightly different than the rigid transformations that we have encountered previously. We interpret it not as mapping points from one coordinate frame to another, but rather as mapping points from their initial coordinates, $p(0) \in \mathbb{R}^3$, to their coordinates after the rigid motion is applied:

$$p(\theta) = e^{\widehat{\xi}\theta}p(0).$$

In this equation, both $p(0)$ and $p(\theta)$ are specified with respect to a *single* reference frame. Similarly, if we let $g_{ab}(0)$ represent the initial configuration of a rigid body relative to a frame A, then the final configuration, still with respect to A, is given by

$$g_{ab}(\theta) = e^{\widehat{\xi}\theta}g_{ab}(0). \tag{2.37}$$

Thus, the exponential map for a twist gives the *relative* motion of a rigid body. This interpretation of the exponential of a twist as a mapping from

initial to final configurations will be especially important as we study the kinematics of robot mechanisms in the next chapter.

Our primary interest is to use the exponential map as a representation for rigid motion, and hence we must show that every rigid transformation can be written as the exponential of some twist. The following proposition asserts that this is always possible and gives a constructive procedure for finding the twist which generates a given rigid transformation.

Proposition 2.9. Surjectivity of the exponential map onto $SE(3)$ *Given $g \in SE(3)$, there exists $\widehat{\xi} \in se(3)$ and $\theta \in \mathbb{R}$ such that $g = \exp(\widehat{\xi}\theta)$.*

Proof. (Constructive). Let $g = (R, p)$ with $R \in SO(3)$, $p \in \mathbb{R}^3$. We ignore the trivial case $(R, p) = (I, 0)$ which is solved with $\theta = 0$ and arbitrary $\widehat{\xi}$.

Case 1 ($R = I$). If there is no rotational motion, set

$$\widehat{\xi} = \begin{bmatrix} 0 & \frac{p}{\|p\|} \\ 0 & 0 \end{bmatrix} \qquad \theta = \|p\|.$$

Equation (2.32) verifies that $\exp(\widehat{\xi}\theta) = (I, p) = g$.

Case 2 ($R \neq I$). To find $\xi = (v, \omega)$, we equate $\exp(\widehat{\xi}\theta)$ and g and solve for v, ω. Using equation (2.36):

$$e^{\widehat{\xi}\theta} = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}.$$

ω and θ are obtained by solving the rotation equation $\exp(\widehat{\omega}\theta) = R$, as in Proposition 2.5 of the previous section. This leaves the equation

$$(I - e^{\widehat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta = p, \tag{2.38}$$

which must be solved for v. It suffices to show that the matrix

$$A = (I - e^{\widehat{\omega}\theta})\widehat{\omega} + \omega\omega^T\theta$$

is nonsingular for all $\theta \in (0, 2\pi)$. This follows from the fact that the two matrices which comprise A have mutually orthogonal null spaces when $\theta \neq 0$ (and $R \neq I$). Hence, $Av = 0 \iff v = 0$. See Exercise 9 for more details. □

In light of Proposition 2.9, every rigid transformation g can be written as the exponential of some twist $\widehat{\xi}\theta \in se(3)$. We call the vector $\xi\theta \in \mathbb{R}^6$ the *exponential coordinates* for the rigid transformation g. Note that, as in the case of rotations, the mapping $\exp : se(3) \to SE(3)$ is many-to-one since the choice of ω and θ for solving the rotational component of the motion is not unique. This does not present great difficulties since for most applications we are given the twist as part of the problem and we wish to find the corresponding rigid motion.

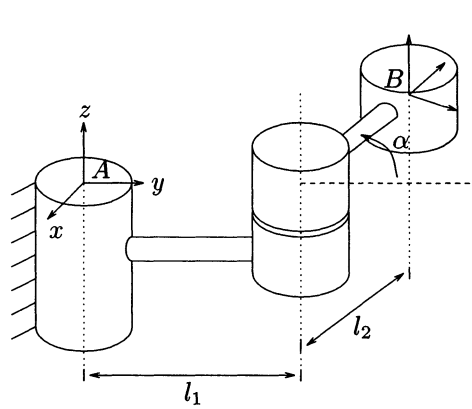


Figure 2.6: Rigid body displacement generated by rotation about a fixed axis.

Example 2.2. Twist coordinates for rotation about a line
Consider the rigid displacement generated by rotating about a fixed axis in space, as shown in Figure 2.6. The configuration of the B frame is given by

$$g_{ab} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & -l_2 \sin\alpha \\ \sin\alpha & \cos\alpha & 0 & l_1 + l_2 \cos\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We wish to calculate the twist coordinates corresponding to the configuration of the frame B relative to frame A.

To compute the twist which generates g_{ab}, we follow the proof of Proposition 2.9, assuming $\alpha \neq 0$ (so that $R \neq I$). The axis $\omega \in \mathbb{R}^3$ and angle $\theta \in \mathbb{R}$ which satisfy $\exp(\widehat{\omega}\theta) = R_{ab}$ are

$$\omega = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \theta = \alpha,$$

since we are rotating about the z-axis. To find v, we must solve

$$\left[\left(I - e^{\widehat{\omega\theta}}\right)\widehat{\omega} + \omega\omega^T\theta\right] v = p_{ab}.$$

Using the fact that $\theta = \alpha$ and expanding the left-hand side, this equation becomes

$$\begin{bmatrix} \sin\alpha & \cos\alpha - 1 & 0 \\ 1 - \cos\alpha & \sin\alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix} v = \begin{bmatrix} -l_2 \sin\alpha \\ l_1 + l_2 \cos\alpha \\ 0 \end{bmatrix}.$$

The solution is given by

$$v = \begin{bmatrix} \frac{\sin\alpha}{2(1-\cos\alpha)} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sin\alpha}{2(1-\cos\alpha)} & 0 \\ 0 & 0 & \frac{1}{\alpha} \end{bmatrix} \begin{bmatrix} -l_2\sin\alpha \\ l_1 + l_2\cos\alpha \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{l_1 - l_2}{2} \\ \frac{(l_1+l_2)\sin\alpha}{2(1-\cos\alpha)} \\ 0 \end{bmatrix}.$$

Thus, the twist coordinates for g_{ab} are

$$\xi = \begin{bmatrix} \frac{l_1-l_2}{2} \\ \frac{(l_1+l_2)\sin\alpha}{2(1-\cos\alpha)} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \theta = \alpha \neq 0.$$

This solution may be unexpected, considering that the motion was generated by a pure rotation about an axis. The reason for the complicated form of the solution is that we took the exponential coordinates of the *absolute* transformation between the B and A coordinate frames. Consider, instead, the exponential coordinates for the relative transformation

$$g(\alpha) = g_{ab}(\alpha)g_{ab}^{-1}(0),$$

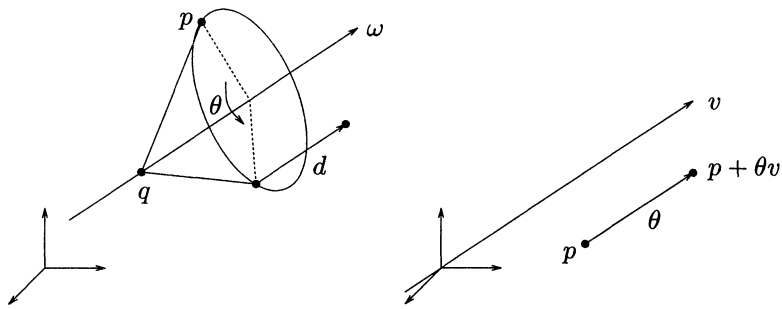
where $g_{ab}(0)$ is the transformation corresponding to $\alpha = 0$ (a pure translation). It can be verified that the exponential coordinates for the relative transformation $g(\alpha)$ are

$$\xi = \begin{bmatrix} l_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \theta = \alpha \neq 0.$$

3.3 Screws: a geometric description of twists

In this section, we explore some of the geometric attributes associated with a twist $\xi = (v, \omega)$. These attributes give additional insight into the use of twists to parameterize rigid body motions. We begin by defining a specific class of rigid body motions, called screw motions, and then show that a twist is naturally associated with a screw.

Consider a rigid body motion which consists of rotation about an axis in space through an angle of θ radians, followed by translation along the same axis by an amount d as shown in Figure 2.7a. We call such a motion a *screw motion*, since it is reminiscent of the motion of a screw, in so far as a screw rotates and translates about the same axis. To further encourage this analogy, we define the *pitch* of the screw to be the ratio of translation to rotation, $h := d/\theta$ (assuming $\theta \neq 0$). Thus, the net translational motion after rotating by θ radians is $h\theta$. We represent the axis as a directed line through a point; choosing $q \in \mathbb{R}^3$ to be a point on



(a) general screw (b) pure translation

Figure 2.7: Screw motions.

the axis and $\omega \in \mathbb{R}^3$ to be a *unit* vector specifying the direction, the axis is the set of points

$$l = \{q + \lambda\omega : \lambda \in \mathbb{R}\}. \tag{2.39}$$

The above definitions hold when the screw motion consists of a nonzero rotation followed by translation.

In the case of zero rotation, the axis of the screw must be defined differently: we take the axis as the line through the origin in the direction v (i.e., v is a vector of magnitude 1), as shown in Figure 2.7b. By convention, the pitch of this screw is ∞ and the magnitude is the amount of translation along the direction v. Collecting these, we have the following definition of a screw:

Definition 2.2. Screw motion
A *screw* S consists of an *axis* l, a *pitch* h, and a *magnitude* M. A *screw motion* represents rotation by an amount $\theta = M$ about the axis l followed by translation by an amount $h\theta$ parallel to the axis l. If $h = \infty$ then the corresponding screw motion consists of a pure translation along the axis of the screw by a distance M.

To compute the rigid body transformation associated with a screw, we analyze the motion of a point $p \in \mathbb{R}^3$, as shown in Figure 2.8. The final location of the point is given by

$$gp = q + e^{\widehat{\omega}\theta}(p - q) + h\theta\omega$$

or, in homogeneous coordinates,

$$g \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})q + h\theta\omega \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}.$$

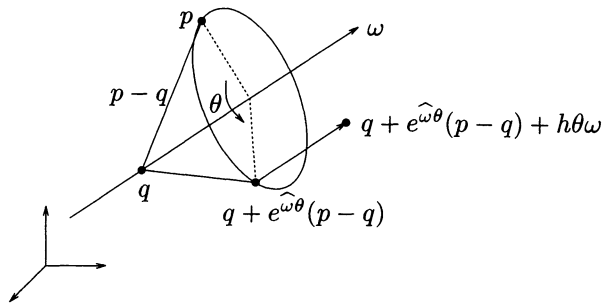


Figure 2.8: Generalized screw motion (with nonzero rotation).

Since this relationship must hold for all $p \in \mathbb{R}^3$, the rigid motion given by the screw is

$$g = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})q + h\theta\omega \\ 0 & 1 \end{bmatrix}. \tag{2.40}$$

As in the last section, this transformation maps points attached to the rigid body from their initial coordinates ($\theta = 0$) to their final coordinates, and all points are specified with respect to the fixed reference frame.

Note that the rigid body displacement given in equation (2.40) has the same form as the exponential of a twist, given in equation (2.36):

$$e^{\widehat{\xi}\theta} = \begin{bmatrix} e^{\widehat{\omega}\theta} & (I - e^{\widehat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}.$$

In fact, if we choose $v = -\omega \times q + h\omega$, then $\xi = (v, \omega)$ generates the screw motion in equation (2.40) (assuming $\|\omega\| = 1$, $\theta \neq 0$). In the case of a pure rotation, $h = 0$ and the twist associated with a screw motion is simply $\xi = (-\omega \times q, \omega)$. In the instance that the screw corresponds to pure translation, we let θ be the amount of translation, and the rigid body motion described by this "screw" is

$$g = \begin{bmatrix} I & \theta v \\ 0 & 1 \end{bmatrix}, \tag{2.41}$$

which is precisely the motion generated by $\exp(\widehat{\xi}\theta)$ with $\xi = (v, 0)$. *Thus, we see that a screw motion corresponds to motion along a constant twist by an amount equal to the magnitude of the screw.*

In fact, we can go one step further and define a screw associated with every twist. Let $\widehat{\xi} \in se(3)$ be a twist with twist coordinates $\xi = (v, \omega) \in \mathbb{R}^6$. We do not assume that $\|\omega\| = 1$, allowing both translation plus rotation as well as pure translation. The following are the screw coordinates of a twist:

1. *Pitch:*

$$h = \frac{\omega^T v}{\|\omega\|^2}. \tag{2.42}$$

The pitch of a twist is the ratio of translational motion to rotational motion. If $\omega = 0$, we say that ξ has infinite pitch.

2. *Axis:*

$$l = \begin{cases} \{\frac{\omega \times v}{\|\omega\|^2} + \lambda\omega : \lambda \in \mathbb{R}\}, & \text{if } \omega \neq 0 \\ \{0 + \lambda v : \lambda \in \mathbb{R}\}, & \text{if } \omega = 0. \end{cases} \tag{2.43}$$

The axis l is a directed line through a point. For $\omega \neq 0$, the axis is a line in the ω direction going through the point $\frac{\omega \times v}{\|\omega\|^2}$. For $\omega = 0$, the axis is a line in the v direction going through the origin.

3. *Magnitude:*

$$M = \begin{cases} \|\omega\|, & \text{if } \omega \neq 0 \\ \|v\|, & \text{if } \omega = 0. \end{cases} \tag{2.44}$$

The magnitude of a screw is the net rotation if the motion contains a rotational component, or the net translation otherwise. If we choose $\|\omega\| = 1$ (or $\|v\| = 1$ when $\omega = 0$), then a twist $\widehat{\xi}\theta$ has magnitude $M = \theta$.

We next show that given a screw, we can define a twist which realizes the screw motion and has the proper geometric attributes. It suffices to prove that we can define a twist with a given set of attributes, since any twist with those attributes will generate the correct screw motion.

Proposition 2.10. Screw motions correspond to twists
Given a screw with axis l, pitch h, and magnitude M, there exists a unit magnitude twist ξ such that the rigid motion associated with the screw is generated by the twist $M\xi$.

Proof. The proof is by construction. We split the proof into the usual cases: pure translation and translation plus rotation. For consistency, we generate a screw of the form $\widehat{\xi}\theta$, where $\theta = M$. We will assume that q is a point on the axis of the screw.

Case 1 ($h = \infty$). Let $l = \{q + \lambda v : \|v\| = 1, \lambda \in \mathbb{R}\}$, $\theta = M$, and define

$$\widehat{\xi} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}.$$

The rigid body motion $\exp(\widehat{\xi}\theta)$ corresponds to pure translation along the screw axis by an amount θ.

Case 2 (h finite). Let $l = \{q + \lambda\omega : \|\omega\| = 1, \lambda \in \mathbb{R}\}$, $\theta = M$, and define

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times q + h\omega \\ 0 & 0 \end{bmatrix}.$$

The fact that the rigid body motion $\exp(\hat{\xi}\theta)$ is the appropriate screw motion is verified by direct calculation. □

There are several important special cases of screw motion of which we shall make frequent use. A *zero pitch screw* is a screw motion for which $h = 0$, corresponding to a pure rotation about an axis. Zero pitch screws are used to model the action of a revolute joint of a manipulator. The axis of the screw corresponds to the axis of rotation of the joint. An *infinite pitch screw* is a motion for which $h = \infty$, as previously mentioned. This case corresponds to a pure translation and is the model for the action of a prismatic joint. The axis of the screw is defined to be a line through the origin which points in the direction of translation (a line through any other point could also be used). The magnitude of the screw gives the amount of the displacement. Finally, we define a *unit twist* to be a twist such that either $\|\omega\| = 1$, or $\omega = 0$ and $\|v\| = 1$; that is, a unit twist has magnitude $M = 1$. Unit twists are useful since they allow us to express rigid motions due to revolute and prismatic joints as $g = \exp(\hat{\xi}\theta)$, where θ corresponds to the amount of rotation or translation.

Some comments about the point q on the axis of the screw in the formulas above are in order. For instance, it is important to note that the formulas do not change for different choices of points on the axis of the screw. Thus, if $q' = q + \lambda\omega$ is some other point on the axis of the screw, the formula in equation (2.40) would be unchanged. It is also instructive to verify that for points on the axis of the screw, the screw motion is purely translational of magnitude $h\theta$, as may be verified by applying equation (2.40) to points on the axis.

The geometric meaning of a screw is expressed succinctly in the following theorem. Its proof follows directly from the definition of the attributes of a twist.

Theorem 2.11 (Chasles). *Every rigid body motion can be realized by a rotation about an axis combined with a translation parallel to that axis.*

As mentioned previously, it is important to keep in mind that the exponential of a twist represents the *relative* motion of a rigid body. As a mapping, $\exp(\hat{\xi}\theta)$ takes points from their initial coordinates, $p(0) \in \mathbb{R}^3$, to their coordinates after the rigid motion is applied:

$$p(\theta) = e^{\hat{\xi}\theta} p(0).$$

Both $p(0)$ and $p(\theta)$ are specified with respect to a single reference frame. If a coordinate frame B is attached to a rigid body undergoing a screw

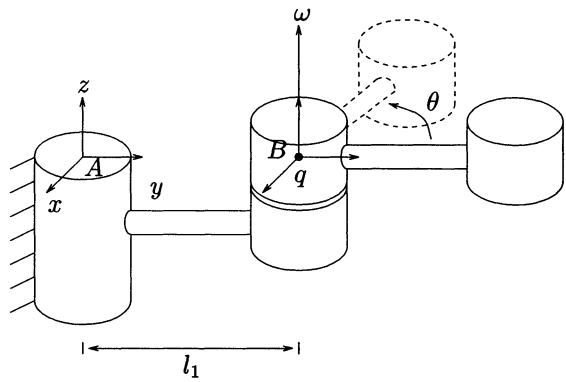


Figure 2.9: Rigid body motion generated by rotation about a fixed axis.

motion, the instantaneous configuration of the coordinate frame B, relative to a fixed frame A, is given by

$$g_{ab}(\theta) = e^{\widehat{\xi\theta}} g_{ab}(0). \tag{2.45}$$

This transformation can be interpreted as follows: multiplication by $g_{ab}(0)$ maps the coordinates of a point relative to the B frame into A's coordinates, and the exponential map transforms the point to its final location (still in A coordinates).

Example 2.3. Rotation about a line
Consider the motion of a rigid body rotating about a fixed axis in space, as shown in Figure 2.9. This motion corresponds to a zero-pitch screw about an axis in the $\omega = (0, 0, 1)$ direction passing through the point $q = (0, l_1, 0)$. The corresponding twist is

$$\xi = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix} = \begin{bmatrix} l_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The exponential of this twist is given by

$$e^{\widehat{\xi\theta}} = \begin{bmatrix} e^{\widehat{\omega\theta}} & (I - e^{\widehat{\omega\theta}})(\omega \times v) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & l_1 \sin\theta \\ \sin\theta & \cos\theta & 0 & l_1(1 - \cos\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

When applied to the homogeneous representation of a point, this matrix maps the coordinates of a point on the rigid body, specified relative to the frame A with $\theta = 0$, to the coordinates of the same point after rotating by θ radians about the axis.

The rigid transformation which maps points in B coordinates to A coordinates—and hence describes the configuration of the rigid body—is given by $g_{ab}(\theta) = \exp(\widehat{\xi}\theta)g_{ab}(0)$ where

$$g_{ab}(0) = \begin{bmatrix} I & \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}.$$

Taking the exponential and performing the matrix multiplication yields

$$g_{ab} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

which can be verified by inspection.

4 Velocity of a Rigid Body

In this section, we derive a formula for the velocity of a rigid body whose motion is given by $g(t)$, a curve parameterized by time t in $SE(3)$. This is not such a naive question as in the case of a single particle following a curve $q(t) \in \mathbb{R}^3$, where the velocity of the particle is $v_q(t) = \frac{d}{dt}q(t)$, because this notion of velocity cannot be generalized since $SE(3)$ is not Euclidean. In particular, the quantity $\dot{g}(t) \notin SE(3)$ and $\dot{g}(t) \notin se(3)$ and the question of its connection with rotational and translational velocity needs to be handled with care. Further, the definition of velocity needs to relate to our informal understanding of rotational and translational velocity. We will show that the proper representation of rigid body velocity is through the use of twists.

4.1 Rotational velocity

Consider first the case of pure rotational motion in $\mathbb{R}^3$. Let $R_{ab}(t) \in SO(3)$ be a curve representing a trajectory of an object frame B, with origin at the origin of frame A, but rotating relative to the fixed frame A. We call A the *spatial* coordinate frame and B the *body* coordinate frame.[1] Any point q attached to the rigid body follows a path in spatial coordinates given by

$$q_a(t) = R_{ab}(t)q_b.$$

[1]The word "spatial" is sometimes used to differentiate between planar motions in $\mathbb{R}^2$ and general (spatial) motions in $\mathbb{R}^3$. In this chapter we reserve the word spatial to mean "relative to a fixed (inertial) coordinate frame."

Note that the coordinates q_b are fixed in the body frame. The velocity of the point in spatial coordinates is

$$v_{q_a}(t) = \frac{d}{dt} q_a(t) = \dot{R}_{ab}(t) q_b. \tag{2.46}$$

Thus $\dot{R}_{ab}$ maps the body coordinates of a point to the spatial velocity of that point. This representation of the rotational velocity is somewhat inefficient, since it requires nine numbers to describe the velocity of a rotating body. One may use the special structure in the matrix $\dot{R}_{ab}$ to derive a more compact representation. To this end, we rewrite equation (2.46) as

$$v_{q_a}(t) = \dot{R}_{ab}(t) R_{ab}^{-1}(t) R_{ab}(t) q_b. \tag{2.47}$$

The following lemma shows that $\dot{R}_{ab}(t) R_{ab}^{-1}(t) \in so(3)$; i.e., it is skew-symmetric.

Lemma 2.12. *Given $R(t) \in SO(3)$, the matrices $\dot{R}(t)R^{-1}(t) \in \mathbb{R}^{3\times 3}$ and $R^{-1}(t)\dot{R}(t) \in \mathbb{R}^{3\times 3}$ are skew-symmetric.*

Proof. Differentiating the identity

$$R(t)R(t)^T = I$$

we have, dropping the dependence of the matrices on t,

$$\dot{R}R^T + R\dot{R}^T = 0,$$

so that

$$\dot{R}R^T = -(\dot{R}R^T)^T.$$

Hence, $\dot{R}R^{-1} = \dot{R}R^T$ is a skew-symmetric matrix. The proof that $R^{-1}\dot{R}$ is skew-symmetric follows by differentiating the identity $R^T R = I$. □

Lemma 2.12 allows us to represent the velocity of a rotating body using a 3-vector. We define the *instantaneous spatial angular velocity*, denoted $\omega_{ab}^s \in \mathbb{R}^3$, as

$$\widehat{\omega}_{ab}^s := \dot{R}_{ab} R_{ab}^{-1}. \tag{2.48}$$

The vector ω_{ab}^s corresponds to the instantaneous angular velocity of the object as seen from the spatial (A) coordinate frame. Similarly, we define the *instantaneous body angular velocity*, denoted $\omega_{ab}^b \in \mathbb{R}^3$, as

$$\widehat{\omega}_{ab}^b := R_{ab}^{-1} \dot{R}_{ab}. \tag{2.49}$$

The body angular velocity describes the angular velocity as viewed from the instantaneous body (B) coordinate frame. From these two equations, it follows that the relationship between the two angular velocities is

$$\widehat{\omega}_{ab}^b = R_{ab}^{-1} \widehat{\omega}_{ab}^s R_{ab} \qquad \text{or} \qquad \omega_{ab}^b = R_{ab}^{-1} \omega_{ab}^s. \tag{2.50}$$

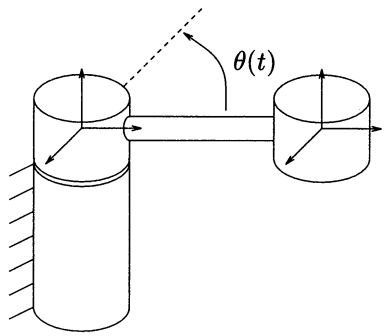


Figure 2.10: Rotational motion of a one degree of freedom manipulator.

Thus the body angular velocity can be determined from the spatial angular velocity by rotating the angular velocity vector into the instantaneous body frame.

Returning now to equation (2.47), we can express the velocity of a point in terms of the instantaneous angular velocity of the rigid body. Substituting equation (2.48) into equation (2.47),

$$v_{q_a}(t) = \widehat{\omega}^s_{ab} R_{ab}(t) q_b = \omega^s_{ab}(t) \times q_a(t). \tag{2.51}$$

Alternatively, using equation (2.50), the velocity of the point in body frame is given by

$$v_{q_b}(t) := R^T_{ab}(t) v_{q_a}(t) = \omega^b_{ab}(t) \times q_b. \tag{2.52}$$

Equations (2.51) and (2.52) constitute a compact description of the velocity of all particles of the body in terms of the body and spatial angular velocities, ω^b_{ab} and ω^s_{ab}.

Example 2.4. Rotational motion of a one degree of freedom manipulator

Consider the motion of the one degree of freedom manipulator shown in Figure 2.10. Let $\theta(t)$ be the angle of rotation about some reference configuration. The trajectory of the manipulator is given by

$$R(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 \\ \sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The spatial velocity is

$$\widehat{\omega}^s = \dot{R}R^T = \begin{bmatrix} -\dot{\theta}\sin\theta & -\dot{\theta}\cos\theta & 0 \\ \dot{\theta}\cos\theta & -\dot{\theta}\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

hence,

$$\omega^s = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}.$$

The body velocity is

$$\widehat{\omega}^b = R^T \dot{R} = \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{or} \quad \omega^b = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}.$$

4.2 Rigid body velocity

Let us now consider the general case where $g_{ab}(t) \in SE(3)$ is a one-parameter curve (parameterized by time) representing a trajectory of a rigid body: more specifically, the rigid body motion of the frame B attached to the body, relative to a fixed or inertial frame A. As in the case of rotation, $\dot{g}_{ab}(t)$ by itself is not particularly useful, but the two terms $\dot{g}_{ab}g_{ab}^{-1}$ and $g_{ab}^{-1}\dot{g}_{ab}$ have some special significance. With

$$g_{ab}(t) = \begin{bmatrix} R_{ab}(t) & p_{ab}(t) \\ 0 & 1 \end{bmatrix},$$

we have that

$$\dot{g}_{ab}g_{ab}^{-1} = \begin{bmatrix} \dot{R}_{ab} & \dot{p}_{ab} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_{ab}^T & -R_{ab}^T p_{ab} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{R}_{ab}R_{ab}^T & -\dot{R}_{ab}R_{ab}^T p_{ab} + \dot{p}_{ab} \\ 0 & 0 \end{bmatrix},$$

which has the form of a twist. By analogy to the rotational velocity, we define the *spatial velocity* $\widehat{V}_{ab}^s \in se(3)$ as

$$\widehat{V}_{ab}^s = \dot{g}_{ab}g_{ab}^{-1} \qquad V_{ab}^s = \begin{bmatrix} v_{ab}^s \\ \omega_{ab}^s \end{bmatrix} = \begin{bmatrix} -\dot{R}_{ab}R_{ab}^T p_{ab} + \dot{p}_{ab} \\ (\dot{R}_{ab}R_{ab}^T)^\vee \end{bmatrix}. \tag{2.53}$$

The spatial velocity $\widehat{V}_{ab}^s$ can be used to find the velocity of a point in spatial coordinates. The coordinates of a point q attached to the rigid body in spatial coordinates are given by

$$q_a(t) = g_{ab}(t)q_b.$$

Differentiating yields

$$v_{q_a} = \dot{q}_a = \dot{g}_{ab}q_b = \dot{g}_{ab}g_{ab}^{-1}q_a$$

and thus,

$$v_{q_a} = \widehat{V}_{ab}^s q_a = \omega_{ab}^s \times q_a + v_{ab}^s. \tag{2.54}$$

The interpretation of the components of the spatial velocity of a rigid motion is somewhat unintuitive. The angular component, ω_{ab}^s, is the instantaneous angular velocity of the body as viewed in the spatial frame. The linear component, v_{ab}^s, is *not the velocity of the origin of the body frame*, which is apparent from equation (2.53). Rather, $v_{ab}^s(t)$ is the velocity of a (possibly imaginary) point on the rigid body which is traveling through the origin of the spatial frame at time t. That is, if one stands at the origin of the spatial frame and measures the instantaneous velocity of a point attached to the rigid body and traveling through the origin at that instant, this is $v_{ab}^s(t)$.

A somewhat more natural interpretation of the spatial velocity is obtained by using the relationship between twists and screws described in the previous section. The screw associated with the twist $\widehat{V}_{ab}^s$ gives the instantaneous axis, pitch, and magnitude of the rigid motion relative to the spatial frame.

It is also possible to specify the velocity of a rigid body with respect to the (instantaneous) body frame. We define

$$\widehat{V}_{ab}^b = g_{ab}^{-1}\dot{g}_{ab} = \begin{bmatrix} R_{ab}^T\dot{R}_{ab} & R_{ab}^T\dot{p}_{ab} \\ 0 & 0 \end{bmatrix} \qquad V_{ab}^b = \begin{bmatrix} v_{ab}^b \\ \omega_{ab}^b \end{bmatrix} = \begin{bmatrix} R_{ab}^T\dot{p}_{ab} \\ (R_{ab}^T\dot{R}_{ab})^\vee \end{bmatrix} \tag{2.55}$$

to be the *body velocity* of a rigid motion $g_{ab}(t) \in SE(3)$. The velocity of the point in the body frame is given by

$$v_{q_b} = g_{ab}^{-1}v_{q_a} = g_{ab}^{-1}\dot{g}_{ab}q_b = \widehat{V}_{ab}^b(t)q_b.$$

Thus, the action of $\widehat{V}_{ab}^b$ is to take the body coordinates of a point, q_b, and return the velocity of that point written in body coordinates, v_{q_b}:

$$v_{q_b} = \widehat{V}_{ab}^b q_b = \omega_{ab}^b \times q_b + v_{ab}^b. \tag{2.56}$$

The interpretation of the body velocity is straightforward: v_{ab}^b is the velocity of the origin of the body coordinate frame relative to the spatial frame, as viewed in the current body frame. ω_{ab}^b is the angular velocity of the coordinate frame, also as viewed in the current body frame. Note that the body velocity is *not* the velocity of the body relative to the body frame; this latter quantity is always zero.

The spatial and body velocity of a rigid motion are related by a similarity transformation. To calculate this relationship, we note that

$$\widehat{V}_{ab}^s = \dot{g}_{ab}g_{ab}^{-1} = g_{ab}(g_{ab}^{-1}\dot{g}_{ab})g_{ab}^{-1} = g_{ab}\,\widehat{V}_{ab}^b\,g_{ab}^{-1}.$$

Alternatively, we can write

$$\begin{aligned} \omega_{ab}^s &= R_{ab}\omega_{ab}^b \\ v_{ab}^s &= -\omega_{ab}^s \times p_{ab} + \dot{p}_{ab} = p_{ab} \times (R_{ab}\omega_{ab}^b) + R_{ab}v_{ab}^b. \end{aligned}$$

In either case, we may summarize the calculation as

$$V_{ab}^s = \begin{bmatrix} v_{ab}^s \\ \omega_{ab}^s \end{bmatrix} = \begin{bmatrix} R_{ab} & \widehat{p}_{ab} R_{ab} \\ 0 & R_{ab} \end{bmatrix} \begin{bmatrix} v_{ab}^b \\ \omega_{ab}^b \end{bmatrix}. \tag{2.57}$$

The 6×6 matrix which transforms twists from one coordinate frame to another is referred to as the *adjoint transformation* associated with g, written Ad_g. Thus, given $g \in SE(3)$ which maps one coordinate system into another, $\mathrm{Ad}_g : \mathbb{R}^6 \to \mathbb{R}^6$ is given as

$$\boxed{\mathrm{Ad}_g = \begin{bmatrix} R & \widehat{p}R \\ 0 & R \end{bmatrix}} \tag{2.58}$$

In the calculation that we have just performed, Ad_g maps body velocity twist coordinates to spatial velocity twist coordinates. Ad_g is invertible, and its inverse is given by

$$\mathrm{Ad}_g^{-1} = \begin{bmatrix} R^T & -(R^T p)^\wedge R^T \\ 0 & R^T \end{bmatrix} = \begin{bmatrix} R^T & -R^T \widehat{p} \\ 0 & R^T \end{bmatrix} = \mathrm{Ad}_{g^{-1}}$$

(see Exercise 14).

We shall make frequent use of the adjoint transformations throughout the book. The calculations performed above give the following useful characterization of the adjoint associated with a rigid transformation $g \in SE(3)$:

Lemma 2.13. *If $\widehat{\xi} \in se(3)$ is a twist with twist coordinates $\xi \in \mathbb{R}^6$, then for any $g \in SE(3)$, $g\widehat{\xi}g^{-1}$ is a twist with twist coordinates $\mathrm{Ad}_g \xi \in \mathbb{R}^6$.*

It will often be convenient to define velocity without explicit reference to coordinate frames. For a rigid body with configuration $g \in SE(3)$, we define the spatial velocity as

$$\widehat{V}^s = \dot{g}g^{-1} \qquad V^s = \begin{bmatrix} v^s \\ \omega^s \end{bmatrix} = \begin{bmatrix} -\dot{R}R^T p + \dot{p} \\ (\dot{R}R^T)^\vee \end{bmatrix} \tag{2.59}$$

and the body velocity as

$$\widehat{V}^b = g^{-1}\dot{g} \qquad V^b = \begin{bmatrix} v^b \\ \omega^b \end{bmatrix} = \begin{bmatrix} R^T \dot{p} \\ (R^T \dot{R})^\vee \end{bmatrix}. \tag{2.60}$$

The body and spatial velocities are related by the adjoint transformation,

$$V^s = \mathrm{Ad}_g V^b. \tag{2.61}$$

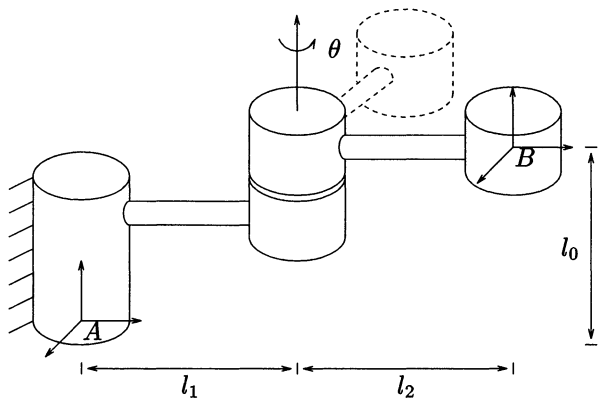


Figure 2.11: Rigid body motion generated by rotation about a fixed axis.

Example 2.5. One degree of freedom manipulator
Consider the one degree of freedom manipulator shown in Figure 2.11. The configuration of the coordinate frame B relative to the fixed frame A is given by

$$g(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 & -l_2\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) & 0 & l_1 + l_2\cos\theta(t) \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where we drop all subscripts for simplicity. The spatial velocity of the rotating rigid body is given by

$$V^s = \begin{bmatrix} v^s \\ \omega^s \end{bmatrix} \qquad \begin{aligned} v^s &= -\dot{R}R^T p + \dot{p} \\ \omega^s &= (\dot{R}R^T)^\vee. \end{aligned}$$

Using the calculation of ω^s from the previous example, we have

$$v^s = \begin{bmatrix} l_1\dot{\theta} \\ 0 \\ 0 \end{bmatrix} \qquad \omega^s = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}.$$

Note that v^s is precisely the velocity of a point attached to the rigid body as it travels through the origin of the A coordinate frame.

The body velocity is

$$V^b = \begin{bmatrix} v^b \\ \omega^b \end{bmatrix} \qquad \begin{aligned} v^b &= R^T\dot{p} \\ \omega^b &= (R^T\dot{R})^\vee, \end{aligned}$$

which gives

$$v^b = \begin{bmatrix} -l_2\dot{\theta} \\ 0 \\ 0 \end{bmatrix} \qquad \omega^b = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}.$$

The body velocity can be interpreted by imagining the velocity of the origin of the B coordinate frame, as seen in the B coordinates. Thus, the linear velocity is always in the $-x$ direction and the angular velocity is always in the z direction. The magnitude of the linear component of the velocity is dependent on the length of the link connecting the B frame to the joint.

4.3 Velocity of a screw motion

In the previous example, we calculated the spatial velocity of a rigid motion generated by a screw action, $\exp(\widehat{\xi}\theta)$. Referring back to Example 2.3 in the previous section, we see that the spatial velocity V^s in the example above is identical to ξ when $\dot{\theta} = 1$. Consider the more general case where

$$g_{ab}(\theta) = e^{\widehat{\xi}\theta} g_{ab}(0)$$

represents the configuration of coordinate frame B relative to frame A. Using the fact that for a constant twist $\widehat{\xi}$,

$$\frac{d}{dt}\left(e^{\widehat{\xi}\theta}\right) = \widehat{\xi}\dot{\theta}e^{\widehat{\xi}\theta}$$

(see Exercise 8), the spatial velocity for this rigid body motion is

$$\begin{aligned}\widehat{V}^s_{ab} &= \dot{g}_{ab}(\theta)g^{-1}_{ab}(\theta) \\ &= \left(\widehat{\xi}\dot{\theta}e^{\widehat{\xi}\theta}g_{ab}(0)\right)\left(g^{-1}_{ab}(0)e^{-\widehat{\xi}\theta}\right) \\ &= \widehat{\xi}\dot{\theta}.\end{aligned}$$

Thus, the spatial velocity corresponding to this motion is precisely the velocity generated by the screw.

The body velocity of a screw motion can be calculated in a similar manner:

$$\begin{aligned}\widehat{V}^b_{ab} &= g^{-1}_{ab}(\theta)\dot{g}_{ab}(\theta) \\ &= \left(g^{-1}_{ab}(0)e^{-\widehat{\xi}\theta}\right)\left(\widehat{\xi}\dot{\theta}e^{\widehat{\xi}\theta}g_{ab}(0)\right) \\ &= \left(g^{-1}_{ab}(0)\widehat{\xi}g_{ab}(0)\right)\dot{\theta} = \left(\mathrm{Ad}_{g^{-1}_{ab}(0)}\,\xi\right)^{\wedge}\dot{\theta}.\end{aligned}$$

For $\dot{\theta} = 1$, $\widehat{V}^b_{ab}$ is a constant vector in the moving body frame. The direction of the body velocity twist is given by the adjoint transformation generated by the initial configuration of the rigid body, $g^{-1}_{ab}(0)$. In particular, if $g_{ab}(0) = I$, i.e., the body frame and spatial frame coincide at $\theta = 0$, then $V^s_{ab} = V^b_{ab} = \xi\dot{\theta}$, where ξ is the constant twist which generates the screw motion.

4.4 Coordinate transformations

Just as we can compose rigid body transformations to find $g_{ac} \in SE(3)$ given $g_{ab}, g_{bc} \in SE(3)$, it is possible to determine the velocity of one coordinate frame relative to a third given the relative velocities between the first and second and second and third coordinate frames. We state the main results as a set of propositions.

Proposition 2.14. Transformation of spatial velocities
Consider the motion of three coordinate frames, A, B, and C. The following relation exists between their spatial velocities:

$$V_{ac}^s = V_{ab}^s + \mathrm{Ad}_{g_{ab}} V_{bc}^s.$$

Proof. The configuration of frame C relative to A is given by

$$g_{ac} = g_{ab} g_{bc}.$$

By definition and the chain rule,

$$\begin{aligned} \widehat{V}_{ac}^s &= \dot{g}_{ac} g_{ac}^{-1} \\ &= (\dot{g}_{ab} g_{bc} + g_{ab} \dot{g}_{bc})(g_{bc}^{-1} g_{ab}^{-1}) \\ &= \dot{g}_{ab} g_{ab}^{-1} + g_{ab} (\dot{g}_{bc} g_{bc}^{-1}) g_{ab}^{-1} \\ &= \widehat{V}_{ab}^s + g_{ab} \widehat{V}_{bc}^s g_{ab}^{-1}, \end{aligned}$$

and converting to twist coordinates,

$$V_{ac}^s = V_{ab}^s + \mathrm{Ad}_{g_{ab}} V_{bc}^s.$$

□

Proposition 2.15. Transformation of body velocities
Consider motion of three coordinate frames, A, B, and C. The following relation exists between their relative body velocities:

$$V_{ac}^b = \mathrm{Ad}_{g_{bc}^{-1}} V_{ab}^b + V_{bc}^b.$$

Proof. Application of the chain rule, as above. □

Propositions 2.14 and 2.15 are used to transform the velocity of a rigid body between different coordinate frames. Often, two of the coordinate frames are stationary with respect to each other and the velocity relationships can be simplified. As an example, if A and B are two inertial frames which are fixed relative to each other, then the spatial velocity of a frame C satisfies

$$V_{ac}^s = \mathrm{Ad}_{g_{ab}} V_{bc}^s. \tag{2.62}$$

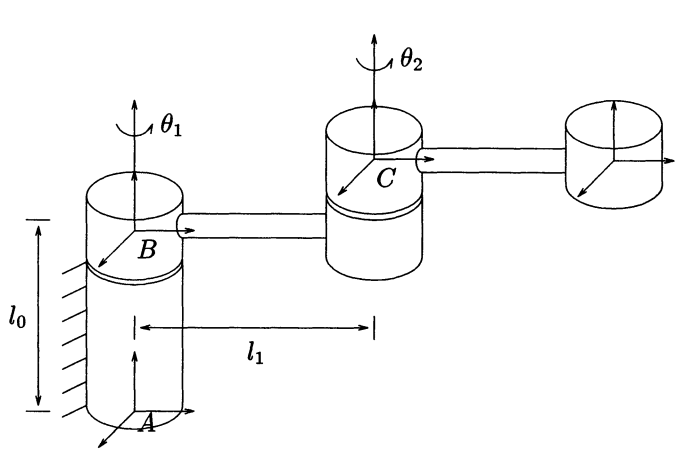


Figure 2.12: Two degree of freedom manipulator.

The corresponding relationship for body velocities is

$$V_{ac}^b = V_{bc}^b, \tag{2.63}$$

since the body velocity is independent of the inertial frame with respect to which it is measured.

The transformation rules given by Propositions 2.14 and 2.15 can also be applied to constant twists, such as those used to model revolute and prismatic joints. If ξ is a twist which represents the motion of a screw and we move the screw by applying a rigid body motion $g \in SE(3)$, the new twist can be obtained using equation (2.62). We interpret g as a fixed rigid motion and equate ξ with a spatial velocity vector. In this case, $\dot{g} = 0$ and hence

$$\xi' = \mathrm{Ad}_g \, \xi \qquad \text{or} \qquad \widehat{\xi'} = g\widehat{\xi}g^{-1}. \tag{2.64}$$

This formula is of tremendous importance in the chapters to come, where we will need to keep track of the different twist axes corresponding to the joints of a robot when they are moved.

Example 2.6. Velocity of a two-link mechanism
Consider the two degree of freedom manipulator shown in Figure 2.12. We wish to find the velocity of frame C relative to A, given the joint velocities $\dot{\theta}_1, \dot{\theta}_2 \in \mathbb{R}$. Since each motion is a screw motion, we write

$$V_{ab}^s = \begin{bmatrix} v_{ab} \\ \omega_{ab} \end{bmatrix} \dot{\theta}_1 \qquad v_{ab} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \omega_{ab} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$V_{bc}^s = \begin{bmatrix} v_{bc} \\ \omega_{bc} \end{bmatrix} \dot{\theta}_2 \qquad v_{bc} = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad \omega_{bc} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

We also calculate $\mathrm{Ad}_{g_{ab}}$:

$$\mathrm{Ad}_{g_{ab}} = \begin{bmatrix} R_{ab} & \begin{pmatrix} 0 \\ 0 \\ l_0 \end{pmatrix}^\wedge R_{ab} \\ 0 & R_{ab} \end{bmatrix}.$$

Using Proposition 2.14,

$$V_{ac}^s = V_{ab}^s + \mathrm{Ad}_{g_{ab}} V_{bc}^s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} l_1 \cos\theta_1 \\ l_1 \sin\theta_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}_2.$$

Note that the velocity consists of two components, one from each of the joints, and that they add together linearly.

A few other identities between body and spatial velocities will prove useful in subsequent chapters. We give them here in the form of a lemma. Their proof is left as an exercise.

Lemma 2.16. Rigid body velocity identities
Using the notation given above for the velocity of one coordinate frame relative to another, the following relationships hold:

$$V_{ab}^b = -V_{ba}^s$$
$$V_{ab}^b = -\mathrm{Ad}_{g_{ba}} V_{ba}^b.$$

5 Wrenches and Reciprocal Screws

In this section we consider forces and moments acting on rigid bodies and use this to introduce the notion of screw systems and reciprocal screws.

5.1 Wrenches

A generalized force acting on a rigid body consists of a linear component (pure force) and an angular component (pure moment) acting at a point. We can represent this generalized force as a vector in $\mathbb{R}^6$:

$$F = \begin{bmatrix} f \\ \tau \end{bmatrix} \qquad \begin{array}{ll} f \in \mathbb{R}^3 & \text{linear component} \\ \tau \in \mathbb{R}^3 & \text{rotational component} \end{array}$$

We will refer to a force/moment pair as a *wrench*.

The values of the wrench vector $F \in \mathbb{R}^6$ depend on the coordinate frame in which the force and moment are represented. If B is a coordinate frame attached to a rigid body, then we write $F_b = (f_b, \tau_b)$ for a wrench applied at the origin of B, with f_b and τ_b specified with respect to the B coordinate frame.

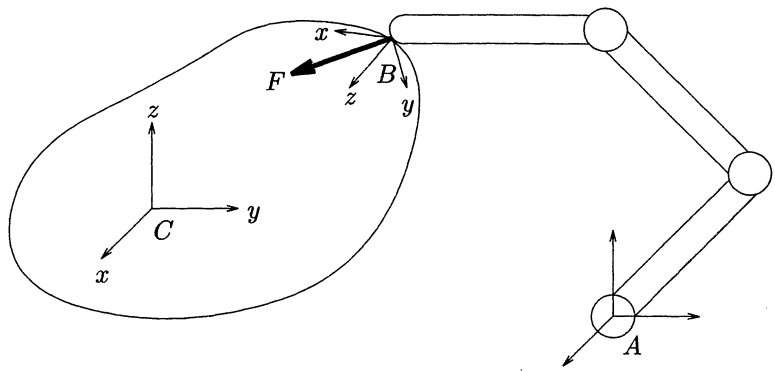


Figure 2.13: Transformation of wrenches between coordinate frames.

Wrenches combine naturally with twists to define instantaneous work. Consider the motion of a rigid body parameterized by $g_{ab}(t)$, where A is an inertial frame and B is a frame attached to the rigid body. Let $V_{ab}^b \in \mathbb{R}^6$ represent the instantaneous body velocity of the rigid body and let F_b represent an applied wrench. Both of these quantities are represented relative to the B coordinate frame and their dot product is the infinitesimal work:

$$\delta W = V_{ab}^b \cdot F_b = (v \cdot f + \omega \cdot \tau).$$

The net work generated by applying the wrench F_b through a twist V_{ab}^b over a time interval $[t_1, t_2]$ is given by

$$W = \int_{t_1}^{t_2} V_{ab}^b \cdot F_b \, dt.$$

Two wrenches are said to be *equivalent* if they generate the same work for every possible rigid body motion. Equivalent wrenches can be used to rewrite a given wrench in terms of a wrench applied at a different point (and with respect to a different coordinate frame). An example of this is shown in Figure 2.13: given the wrench F_b applied at the origin of contact coordinate frame B, we wish to determine the equivalent wrench applied at the origin of the object coordinate frame C. In order to compute the equivalent wrench, we use the instantaneous work performed by the wrench as the body undergoes an arbitrary rigid motion. Let $g_{bc} = (p_{bc}, R_{bc})$ be the configuration of frame C relative to B. By equating the instantaneous work done by the wrench F_b and the wrench F_c over an arbitrary interval of time, we have that

$$V_{ac}^b \cdot F_c = V_{ab}^b \cdot F_b = \left(\mathrm{Ad}_{g_{bc}} V_{ac}^b\right)^T F_b = V_{ac}^b \cdot \mathrm{Ad}_{g_{bc}}^T F_b,$$

and since V_{ac}^b is free,

$$F_c = \mathrm{Ad}_{g_{bc}}^T F_b. \tag{2.65}$$

Equation (2.65) transforms a wrench applied at the origin of the B frame into an equivalent wrench applied at the origin of the C frame. The components of F_c are specified relative to the C coordinate frame. Expanding equation (2.65),

$$\begin{bmatrix} f_c \\ \tau_c \end{bmatrix} = \begin{bmatrix} R_{bc}^T & 0 \\ -R_{bc}^T \widehat{p}_{bc} & R_{bc}^T \end{bmatrix} \begin{bmatrix} f_b \\ \tau_b \end{bmatrix}, \tag{2.66}$$

we see that the adjoint transformation rotates the force and torque vectors from the B frame into the C frame and includes an additional torque of the form $-p_{bc} \times f_b$, which is the torque generated by applying a force f_b at a distance $-p_{bc}$.

It is also possible to represent a wrench with respect to a coordinate frame which is not inside the rigid body. Consider for example the coordinate frame A shown in Figure 2.13. The wrench F written in A's coordinate frame is given by

$$F_a = \mathrm{Ad}_{g_{ba}}^T F_b.$$

This wrench represents the equivalent force/moment pair applied as if the coordinate frame A were rigidly attached to the object. This is *not* the same as simply rewriting the components of F_b in A's coordinates, since the point of application for F_a is the origin of the A frame and not the origin of the B frame.

If several wrenches are all applied to a single rigid body, then the net wrench acting on the rigid body can be constructed by adding the wrench vectors. In order for this addition to make sense, all of the wrenches must be represented with respect to the same frame. Thus, given a set of wrenches F_i, each wrench is first written as an equivalent wrench relative to a single coordinate frame and then the equivalent wrenches are added to determine the net wrench acting on the rigid body. This helps explain why equivalent wrenches include a shift of origin: one can only add wrenches if they represent forces and torques applied at a single point (such as the center of mass or a fixed spatial frame).

A net wrench F acting on a rigid body with configuration $g_{ab} \in SE(3)$ has two natural representations. The *body* representation of the wrench is written as F_b and represents the equivalent force and moment applied at the origin of the B frame (and written in B's coordinates). The *spatial* representation of the wrench is the equivalent wrench written in A's coordinate frame. These representations are analogous to the spatial and body representations of the velocity of a rigid body.

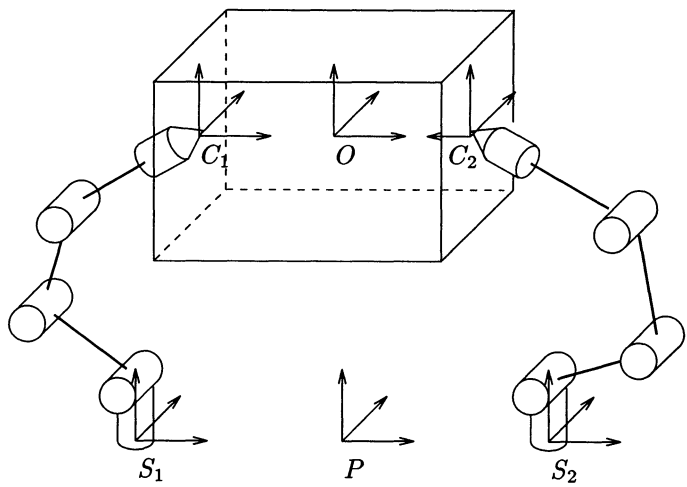


Figure 2.14: Coordinate frames for a simple grasping example.

As with velocities, it will be convenient to define the spatial and body representations of a wrench without explicit reference to a given set of coordinate frames. If $g \in SE(3)$ is the configuration of a rigid body, then we write F^b for the body wrench and F^s for the spatial wrench. These wrenches are related by the transpose of the adjoint matrix:

$$F^b = \mathrm{Ad}_g^T F^s. \tag{2.67}$$

This notation mirrors that used for body and spatial velocities of a rigid body allowing the instantaneous work performed by a wrench F moving through a rigid motion with instantaneous velocity V to be written as

$$\delta W = V^b \cdot F^b = V^s \cdot F^s.$$

We leave the proof of this statement as an exercise.

Example 2.7. Preview of multifingered grasping
Consider the multifingered grasp shown in Figure 2.14. Let F_{c_i} be the wrench exerted by the i^{th} finger on the grasped object, represented in the frame C_i. The net wrench on the body, in the body coordinate frame O, is given by

$$F_o = \sum \mathrm{Ad}_{g_{oc_i}^{-1}}^T F_{c_i}.$$

This is the basic calculation which is used in grasping to determine the net effect of forces applied at the contact points between the fingers and the object.

5.2 Screw coordinates for a wrench

As with twists, it is possible to generate a wrench by applying a force along an axis in space and simultaneously applying a torque about the same axis. The dual of Chasles' theorem, which showed that *every* twist could be generated by a screw, is called *Poinsot's theorem.* It asserts that every wrench is equivalent to a force and torque applied along the same axis. We begin by defining the notion of a wrench acting along a screw.

With respect to some fixed spatial coordinate frame A, let S be a screw with axis $l = \{q + \lambda\omega : \lambda \in \mathbb{R}\}$, $\|\omega\| = 1$, pitch h, and magnitude M. We construct a wrench from this screw by applying a force of magnitude M along the directed line l and a torque of magnitude hM about the line. If $h = \infty$, we generate a wrench by applying a pure torque about l. The resulting wrench, in A's coordinates, is given by

$$\begin{aligned} F &= M \begin{bmatrix} \omega \\ -\omega \times q + h\omega \end{bmatrix} && h \text{ finite} \\ F &= M \begin{bmatrix} 0 \\ \omega \end{bmatrix} && h = \infty, \end{aligned} \tag{2.68}$$

where the term $-\omega \times q$ accounts for the offset between the axis of the screw and the origin of A. We call F the *wrench along the screw* S. Note that F (and q and ω) are all specified with respect to the fixed coordinate frame A and hence F represents the spatial wrench applied to the rigid body. (We omit the use of subscripts in this section since all quantities are specified with respect to a single coordinate frame.)

To find the screw coordinates for a wrench, we solve equation (2.68) for ω, q, h, and M given $F = (f, \tau)$. This leads to the following theorem:

Theorem 2.17 (Poinsot). *Every collection of wrenches applied to a rigid body is equivalent to a force applied along a fixed axis plus a torque about the same axis.*

Proof. The proof is constructive. Let $F = (f, \tau)$ be the net wrench applied to the object. We ignore the trivial case, $F = 0$.

Case 1: ($f = 0$, pure torque). Set $M = \|\tau\|$, $\omega = \tau/M$, and $h = \infty$. Equation (2.68) verifies that these are the appropriate screw coordinates.

Case 2: ($f \neq 0$). Set $M = \|f\|$, and $\omega = f/M$. It remains to solve

$$M(q \times \omega + h\omega) = \tau$$

for q and h. One solution is given by

$$h = \frac{f^T \tau}{\|f\|^2} \qquad q = \frac{f \times \tau}{\|f\|^2}.$$

This solution is not unique since any $q' = q + \lambda\omega$ will also satisfy equation (2.68). □

Using Poinsot's theorem, we can define the screw coordinates of a wrench, $F = (f, \tau)$:

1. *Pitch:*

$$h = \frac{f^T \tau}{\|f\|^2} \tag{2.69}$$

The pitch of a wrench is the ratio of angular torque to linear force. If $f = 0$, we say that F has infinite pitch.

2. *Axis:*

$$l = \begin{cases} \{\frac{f \times \tau}{\|f\|^2} + \lambda f : \lambda \in \mathbb{R}\}, & \text{if } f \neq 0 \\ \{0 + \lambda\tau : \lambda \in \mathbb{R}\}, & \text{if } f = 0 \end{cases} \tag{2.70}$$

The axis l is a directed line through a point. For $f \neq 0$, the axis is a line in the f direction going through the point $q = \frac{f \times \tau}{\|f\|^2}$. For $f = 0$, the axis is a line in the τ direction going through the origin.

3. *Magnitude:*

$$M = \begin{cases} \|f\|, & \text{if } f \neq 0 \\ \|\tau\|, & \text{if } f = 0 \end{cases} \tag{2.71}$$

The magnitude of a screw is the net linear force, if the motion contains a linear component, or the net torque, otherwise.

The dual nature of twists and wrenches is evident in the screw coordinates for twists and wrenches. For example, a zero pitch twist corresponds to pure rotation, while a zero pitch wrench corresponds to a pure force (no angular component).

5.3 Reciprocal screws

The dot product between twists and wrenches gives the instantaneous power associated with moving a rigid body through an applied force. As in the previous subsection, we carry out all calculations relative to a single coordinate frame and omit the use of subscripts. A wrench F is said to be *reciprocal* to a twist V if the instantaneous power is zero: $F \cdot V = 0$. Since both twists and wrenches can be represented by screws, we can use this to define the notion of reciprocal screws:

Definition 2.3. Reciprocal screws
Two screws S_1 and S_2 are *reciprocal* if the twist V about S_1 and the wrench F along S_2 are reciprocal.

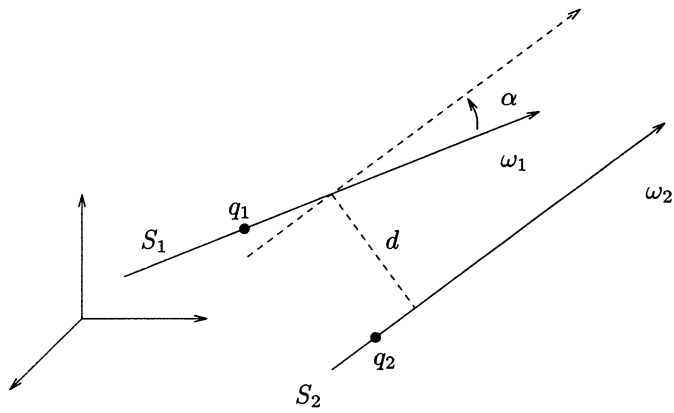


Figure 2.15: Notation for reciprocal screws.

Classically, reciprocal screws are defined by using the *reciprocal product* between screws. Let S_i be a screw with axis $l_i = \{q_i + \lambda\omega_i : \lambda \in \mathbb{R}\}$, pitch h_i, and magnitude M_i. Given two screws S_1 and S_2, we define the distance d between the screws as the minimum distance between l_1 and l_2; this distance will be achieved along a line perpendicular to both l_1 and l_2. We denote this line as dn where n is a unit vector and $d > 0$. The angle α between S_1 and S_2 is the angle between the vectors ω_1 and ω_2,

$$\alpha = \text{atan2}(\omega_1 \times \omega_2 \cdot n,\ \omega_1 \cdot \omega_2)$$

(see Figure 2.15). The reciprocal product between two screws is defined as

$$S_1 \odot S_2 = M_1 M_2 \big((h_1 + h_2)\cos\alpha - d\sin\alpha\big). \tag{2.72}$$

Proposition 2.18. Characterization of reciprocal screws
Two screws S_1 and S_2 are reciprocal if and only if

$$S_1 \odot S_2 = 0.$$

Proof. We consider only the case where h_1 and h_2 are finite. The other cases are left as exercises. Let V be the twist about the screw S_1 and F be the wrench along the screw S_2:

$$V = M_1 \begin{bmatrix} q_1 \times \omega_1 + h_1\omega_1 \\ \omega_1 \end{bmatrix} \qquad F = M_2 \begin{bmatrix} \omega_2 \\ q_2 \times \omega_2 + h_2\omega_2 \end{bmatrix}.$$

Without loss of generality we can assume that q_1 and q_2 are the points at which the axes are closest and hence q_2 can be rewritten as $q_2 = q_1 + dn$ where n is the unit normal vector connecting the two axes. The

instantaneous work between V and F is

$$\begin{aligned} V \cdot F &= M_1 M_2 \big(\omega_2 \cdot (q_1 \times \omega_1 + h_1\omega_1) + \omega_1 \cdot (q_2 \times \omega_2 + h_2\omega_2)\big) \\ &= M_1 M_2 \big(\omega_2 \cdot q_1 \times \omega_1 + h_1\omega_1 \cdot \omega_2 + \omega_1 \cdot (q_1 + dn) \times \omega_2 + h_2\omega_1 \cdot \omega_2\big) \\ &= M_1 M_2 \big((h_1 + h_2)\cos\alpha - d\sin\alpha\big), \end{aligned}$$

which is precisely the reciprocal product. Hence, by definition, the screws are reciprocal if and only if the reciprocal product is zero. □

If we represent screws using twist coordinates, then we can define the reciprocal product directly in terms of the components of the twists. Let $V_1, V_2 \in \mathbb{R}^6$ be two arbitrary twists. Then we define the reciprocal product between V_1 and V_2 as

$$V_1 \odot V_2 = v_1^T \omega_2 + v_2^T \omega_1.$$

A similar relationship holds if we associate screws with wrenches.

Reciprocal screws play an important role in analyzing the kinematic properties of mechanisms. For example, in a grasping context we can view the wrenches applied to an object as a set of constraining screws and ask if there are any instantaneous rigid motions (twists) that do not violate the constraints. Such twists, if they exist, correspond to motions of the grasped object which cannot be restricted by the fingers. This specific situation is considered in detail in Chapter 5, but we can give some preliminary indications of how the analysis might proceed using the concept of a system of screws.

As a motivating example, consider the grasping situation depicted in Figure 2.16. Suppose we constrain the motion of a rigid body by applying normal forces at several points around the rigid body. We would like to ascertain if there are any motions of the rigid body which cannot be resisted by these forces. Let $\{S_1, \ldots, S_k\}$ represent the screws corresponding to the wrenches. Suppose that there exists another screw S_f such that $S_f \odot S_i = 0$. Then, interpreting S_f as a twist and each S_i as a wrench, we see that motion along S_f causes no work to be performed against any of the wrenches. Hence, the wrenches cannot resist this type of motion and the object is free to move (instantaneously) along S_f.

If we interpret a set of screws $\{S_1, \ldots, S_k\}$ as twists, then the twists form a linear space over the reals and hence we can talk about scaling and adding screws by interpreting this in terms of regular addition and multiplication on twists. We call the set of screws $\{S_1, \ldots, S_k\}$ a *system of screws* and we define addition and scaling of screws by associating each screw with a unique twist.

It follows immediately from the definition of the reciprocal product that if S is reciprocal to S_1 and S_2, then it is reciprocal to any linear combination of S_1 and S_2 (with the linear combination performed in twist coordinates). Using this linearity property, we can define the set of all

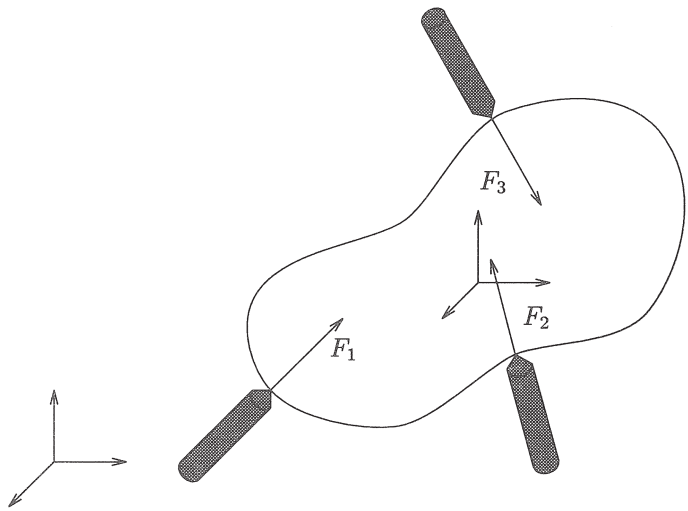


Figure 2.16: A set of pure forces acting on a rigid body.

screws which are reciprocal to a given system of screws as the *reciprocal screw system.* A reciprocal screw system defines a linear subspace of twists. If we interpret a screw system as a set of wrenches (or constrained directions), then the reciprocal screw system describes the instantaneous motions which are possible under the constraints. Conversely, if we interpret the screw system as a set of twists, then the reciprocal screw system is the set of wrenches which cause no net motion of the object. Both of these interpretations follow directly from the definition of the reciprocal product between a twist and a wrench.

In addition to applications in grasping, screw systems and reciprocal screw systems can be also used to analyze the mobility of mechanisms, as we shall see in detail in the next chapter. The following proposition is one of the main tools in this type of analysis. Its proof follows directly from the fact that the space of twists is a 6-dimensional linear space and that screws can be naturally associated with this linear space.

Proposition 2.19. Dimensionality of reciprocal screw systems *Let r be the dimension of system of screws $\{S_1, \ldots, S_k\}$ (determined by converting the screws into either twists or wrenches) and let n be the dimension of the corresponding reciprocal system. Then,*

$$r + n = 6.$$

Applying this proposition to the example in Figure 2.16, we see that the subspace of twists which cannot be resisted is at least 3-dimensional. It may have greater dimension if the applied normal forces do not generate independent wrenches.

6 Summary

The following are the key concepts covered in this chapter:

1. The *configuration* of a rigid body is represented as an element $g \in SE(3)$. An element $g \in SE(3)$ may also be viewed as a mapping $g : \mathbb{R}^3 \to \mathbb{R}^3$ which preserves distances and angles between points. In homogeneous coordinates, we write

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \qquad \begin{matrix} R \in SO(3) \\ p \in \mathbb{R}^3. \end{matrix}$$

   The same representation can also be used for a rigid body transformation between two coordinate frames.

2. *Rigid body transformations* can be represented as the exponentials of twists:

$$g = \exp(\widehat{\xi}\theta) \qquad \widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}, \quad \begin{matrix} \widehat{\omega} \in so(3), \\ v \in \mathbb{R}^3, \theta \in \mathbb{R}. \end{matrix}$$

   The twist coordinates of $\widehat{\xi}$ are $\xi = (v, \omega) \in \mathbb{R}^6$.

3. A twist $\xi = (v, \omega)$ is associated with a *screw* motion having attributes

$$\text{pitch:} \qquad h = \frac{\omega^T v}{\|\omega\|^2};$$

$$\text{axis:} \qquad l = \begin{cases} \{\frac{\omega \times v}{\|\omega\|^2} + \lambda\omega : \lambda \in \mathbb{R}\}, & \text{if } \omega \neq 0 \\ \{0 + \lambda v : \lambda \in \mathbb{R}\}, & \text{if } \omega = 0; \end{cases}$$

$$\text{magnitude:} \qquad M = \begin{cases} \|\omega\|, & \text{if } \omega \neq 0 \\ \|v\|, & \text{if } \omega = 0. \end{cases}$$

   Conversely, given a screw we can write the associated twist. Two special cases are *pure rotation* about an axis $l = \{q + \lambda\omega\}$ by an amount θ and *pure translation* along an axis $l = \{0 + \lambda v\}$:

$$\xi = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix} \theta \quad \text{(pure rotation)} \qquad \xi = \begin{bmatrix} v \\ 0 \end{bmatrix} \theta \quad \text{(pure translation)}.$$

4. The *velocity* of a rigid motion $g(t) \in SE(3)$ can be specified in two ways. The *spatial velocity*,

$$\widehat{V}^s = \dot{g}g^{-1},$$

is a twist which gives the velocity of the rigid body as measured by an observer at the origin of the reference frame. The *body velocity*,

$$\widehat{V}^b = g^{-1}\dot{g},$$

is the velocity of the object in the instantaneous body frame. These velocities are related by the *adjoint transformation*

$$V^s = \mathrm{Ad}_g V^b \qquad \mathrm{Ad}_g = \begin{bmatrix} R & \widehat{p}R \\ 0 & R \end{bmatrix},$$

which maps $\mathbb{R}^6 \to \mathbb{R}^6$. To transform velocities between coordinate frames, we use the relations

$$\begin{aligned} V^s_{ac} &= V^s_{ab} + \mathrm{Ad}_{g_{ab}} V^s_{bc} \\ V^b_{ac} &= \mathrm{Ad}_{g^{-1}_{bc}} V^b_{ab} + V^b_{bc}, \end{aligned}$$

where V^s_{ab} is the spatial velocity of coordinate frame B relative to frame A and V^b_{ab} is the body velocity.

5. *Wrenches* are represented as a force, moment pair

$$F = (f, \tau) \in \mathbb{R}^6.$$

If B is a coordinate frame attached to a rigid body, then we write $F_b = (f_b, \tau_b)$ for a wrench applied at the origin of B, with f_b and τ_b specified with respect to the B coordinate frame. If C is a second coordinate frame, then we can write F_b as an *equivalent wrench* applied at C:

$$F_c = \mathrm{Ad}^T_{g_{bc}} F_b.$$

For a rigid body with configuration g_{ab}, $F^s := F_a$ is called the *spatial* wrench and $F^b := F_b$ is called the *body* wrench.

6. A wrench $F = (f, \tau)$ is associated with a screw having attributes

pitch: $$h = \frac{f^T \tau}{\|f\|^2};$$

axis: $$l = \begin{cases} \{\frac{f \times \tau}{\|f\|^2} + \lambda f : \lambda \in \mathbb{R}\}, & \text{if } f \neq 0 \\ \{0 + \lambda \tau : \lambda \in \mathbb{R}\}, & \text{if } f = 0; \end{cases}$$

magnitude: $$M = \begin{cases} \|f\|, & \text{if } f \neq 0 \\ \|\tau\|, & \text{if } f = 0. \end{cases}$$

7. A wrench F and a twist V are *reciprocal* if $F \cdot V = 0$. Two screws S_1 and S_2 are reciprocal if the twist V_1 about S_1 and the wrench F_2

along S_2 are reciprocal. The *reciprocal product* between two screws is given by

$$S_1 \odot S_2 = V_1 \cdot F_2 = V_1 \odot V_2 = v_1 \cdot \omega_2 + v_2^T \omega_1$$

where $V_i = (v_i, \omega_i)$ represents the twist associated with the screw S_i. Two screws are reciprocal if the reciprocal product between the screws is zero.

8. A *system of screws* $\{S_1, \ldots, S_k\}$ describes the vector space of all linear combinations of the screws $\{S_1, \ldots, S_k\}$. A *reciprocal screw system* is the set of all screws which are reciprocal to S_i. The dimensions of a screw system and its reciprocal system sum to 6 (in $SE(3)$).

All of the concepts presented in this chapter can also be applied to planar rigid body motions (see Exercises 10 and 11).

7 Bibliography

The treatment of rigid motion described here, particularly the geometry of twists, was inspired by the work of Paden [86]. The use of exponential coordinates for representing robotic motion was introduced by Brockett [12]. Brockett's derivation also forms the basis of the next chapter. Related treatments can be found in the classical work by Ball [6] and the more recent texts by Hunt [42], Bottema and Roth [10], Duffy [28], Angeles [1], and McCarthy [71]. A more abstract version of the developments of this chapter can be made in the framework of matrix Lie groups and is presented in Appendix A.

8 Exercises

1. Let $a, b, c \in \mathbb{R}^3$ be 3-vectors and let $\cdot$ and $\times$ denote the dot product and cross product in $\mathbb{R}^3$. Verify the following identities:

   (a) $a \cdot (b \times c) = (a \times b) \cdot c$

   (b) $a \times (b \times c) = (a \cdot c)b - (a \cdot b)c$

2. Using the homogeneous representation, show that $SE(3)$ satisfies the axioms of a group, with the group multiplication given by the usual matrix multiplication.

3. *Properties of rotation matrices*
   Let $R \in SO(3)$ be a rotation matrix generated by rotating about a unit vector ω by θ radians. That is, R satisfies $R = \exp(\widehat{\omega}\theta)$.

   (a) Show that the eigenvalues of $\widehat{\omega}$ are 0, i, and $-i$, where $i = \sqrt{-1}$. What are the corresponding eigenvectors?

   (b) Show that the eigenvalues of R are 1, $e^{i\theta}$, and $e^{-i\theta}$. What is the eigenvector whose eigenvalue is 1?

   (c) Let $R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$ be a rotation matrix. Show that $\det R = r_1^T(r_2 \times r_3)$.

4. *Properties of skew-symmetric matrices*
   Show that the following properties of skew-symmetric matrices are true:

   (a) If $R \in SO(3)$ and $\omega \in \mathbb{R}^3$, then $R\widehat{\omega}R^T = (R\omega)\widehat{}$.

   (b) If $R \in SO(3)$ and $v, w \in \mathbb{R}^3$, then $R(v \times w) = (Rv) \times (Rw)$.

   (c) Show that $so(3)$ is a vector space. Determine its dimension and give a basis for $so(3)$.

5. *Cayley parameters*
   Another parameterization of $SO(3)$, which does not involve transcendental functions, is *Cayley's parameterization.* Let a be a vector in $\mathbb{R}^3$ and let $\widehat{a}$ be the associated 3×3 skew-symmetric matrix.

   (a) Show that $R_a = (I - \widehat{a})^{-1}(I + \widehat{a}) \in SO(3)$.

   (b) Verify that

$$R_a = \frac{1}{1+\|a\|^2} \begin{bmatrix} 1+a_1^2-a_2^2-a_3^2 & 2(a_1a_2 - a_3) & 2(a_1a_3 + a_2) \\ 2(a_1a_2 + a_3) & 1-a_1^2+a_2^2-a_3^2 & 2(a_2a_3 - a_1) \\ 2(a_1a_3 - a_2) & 2(a_2a_3 + a_1) & 1-a_1^2-a_2^2+a_3^2 \end{bmatrix}$$

   (c) Given a rotation matrix R, compute the Cayley parameters a.

6. *Unit quaternions*
Let $Q = (q_0, \vec{q})$ and $P = (p_0, \vec{p})$ be quaternions, where $q_0, p_0 \in \mathbb{R}$ are the scalar parts of Q and P and $\vec{q}, \vec{p}$ are the vector parts.

   (a) Show that the set of *unit* quaternions satisfies the axioms of a group.

   (b) Let x be a point and let X be a quaternion whose scalar part is zero and whose vector part is equal to x (such a quaternion is called a *pure* quaternion). Show that if Q is a unit quaternion, the product QXQ^* is a pure quaternion and the vector part of QXQ^* satisfies

$$(q_0^2 - \vec{q} \cdot \vec{q})\vec{x} + 2\Big(q_0(\vec{q} \times \vec{x}) + (x \cdot \vec{q})\vec{q}\Big).$$

   Verify that the vector part describes the point to which x is rotated under the rotation associated with Q.

   (c) Show that the set of unit quaternions is a two-to-one covering of $SO(3)$. That is, for each $R \in SO(3)$, there exist two distinct unit quaternions which can be used to represent this rotation.

   (d) Compare the number of additions and multiplications needed to perform the following operations:

      i. Compose two rotation matrices.
      ii. Compose two quaternions.
      iii. Apply a rotation matrix to a vector.
      iv. Apply a quaternion to a vector [as in part (b)].

   Count a subtraction as an addition, and a division as a multiplication.

   (e) Show that a rigid body rotating at unit velocity about a unit vector in $\omega \in \mathbb{R}^3$ can be represented by the quaternion differential equations

$$\dot{Q} \cdot Q^* = (0, \omega/2),$$

   where $\cdot$ represents quaternion multiplication.

7. A rigid body moving in $\mathbb{R}^2$ has three degrees of freedom (two components of translation and one of rotation), a rigid body moving in $\mathbb{R}^3$ has six degrees of freedom (three each of translation and rotation). Show that a rigid body moving in $\mathbb{R}^n$ will have $\frac{1}{2}(n + n^2)$ degrees of freedom. How many are translational and how many are rotational?

8. *Properties of the matrix exponential*
Let Λ be a matrix in $\mathbb{R}^{n \times n}$. The exponential of Λ is defined as

$$e^{\Lambda} = I + \Lambda + \frac{\Lambda^2}{2!} + \frac{\Lambda^3}{3!} + \cdots$$

(a) Choose a matrix norm and show that the above series converges.

(b) Let $g \in \mathbb{R}^{n\times n}$ be an invertible matrix. Show the following equality:

$$ge^{\Lambda}g^{-1} = e^{g\Lambda g^{-1}}.$$

(c) Verify that

$$\frac{d}{dt}e^{\Lambda\theta} = (\Lambda\dot{\theta})e^{\Lambda\theta} = e^{\Lambda\theta}(\Lambda\dot{\theta}).$$

9. *Projection maps and proof of Proposition 2.9*
This problem completes the proof of Proposition 2.9 using the properties of projection maps on linear spaces. Assume $\omega \in so(3)$ and $\|\omega\| = 1$.

(a) Given a vector $\omega \in \mathbb{R}^3$, let N_ω denote the subspace spanned by ω and $N_\omega^\perp$ denote the orthogonal complement. Show that

$$\text{image } \widehat{\omega} = N_\omega^\perp \qquad \text{and} \qquad \text{kernel } \widehat{\omega} = N_\omega.$$

(b) Let $V \subset \mathbb{R}^n$ be a linear subspace. A *projection map* is a linear mapping $P_V : \mathbb{R}^n \to V$ which satisfies image $(P_V) = V$ and $P_V(x) = x$ for all $x \in V$. Show that

$$P_{N_\omega} = \omega\omega^T \qquad \text{and} \qquad P_{N_\omega^\perp} = (I - \omega\omega^T)$$

are both projection maps.

(c) Calculate the null space of $I - e^{\widehat{\omega}\theta}$ for $\omega \in so(3)$ and $\theta \in (0, 2\pi)$ and show that $(I - e^{\widehat{\omega}\theta}) : N_\omega^\perp \to N_\omega^\perp$ is bijective.

(d) Let $A = (I - e^{\widehat{\omega}\theta})\widehat{\omega} + \omega\omega^T\theta$, where $\theta \in (0, 2\pi)$. Show that $A : \mathbb{R}^3 \to \mathbb{R}^3$ is invertible.

10. *Planar rotational motion*
Let $SO(2)$ be the set of all 2×2 orthogonal matrices with determinant equal to $+1$.

(a) Show that $SO(2)$ can be identified with the $\mathbb{S}^1$, the unit circle in $\mathbb{R}^2$.

(b) Let $\omega \in \mathbb{R}$ be a real number and define $\widehat{\omega} \in so(2)$ as the skew-symmetric matrix

$$\widehat{\omega} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}.$$

Show that

$$e^{\widehat{\omega}\theta} = \begin{bmatrix} \cos\omega\theta & -\sin\omega\theta \\ \sin\omega\theta & \cos\omega\theta \end{bmatrix}.$$

Is the exponential map $\exp : so(2) \to SO(3)$ surjective? injective?

(c) Show that for $R \in SO(2)$ and $\widehat{\omega} \in so(2)$, $R\widehat{\omega}R^T = \widehat{\omega}$.

11. *Planar rigid body transformations*
A transformation $g = (p, R) \in SE(2)$ consists of a translation $p \in \mathbb{R}^2$ and a 2×2 rotation matrix R. We represent this in homogeneous coordinates as a 3×3 matrix:

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}.$$

A twist $\widehat{\xi} \in se(2)$ can be represented by a 3×3 matrix of the form:

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \qquad \widehat{\omega} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \qquad \omega \in \mathbb{R}, v \in \mathbb{R}^2.$$

The twist coordinates for $\widehat{\xi} \in se(2)$ have the form $\xi = (v, \omega) \in \mathbb{R}^3$. Note that v is a vector in the plane and ω is a scalar.

(a) Show that the exponential of a twist in $se(2)$ gives a rigid body transformation in $SE(2)$. Consider both the pure translation case, $\xi = (v, 0)$, and the general case, $\xi = (v, \omega), \omega \neq 0$.

(b) Show that the planar twists which correspond to pure rotation about a point q and pure translation in a direction v are given by

$$\xi = \begin{bmatrix} q_y \\ -q_x \\ 1 \end{bmatrix} \text{ (pure rotation)} \qquad \xi = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \text{ (pure translation).}$$

(c) Show that every planar rigid body motion can be described as either pure rotation about a point (called the *pole* of the motion) or pure translation.

(d) Show that the matrices $\widehat{V}^s = \dot{g}g^{-1}$ and $\widehat{V}^b = g^{-1}\dot{g}$ are both twists. Define and interpret the spatial velocity $V^s \in \mathbb{R}^3$ and the body velocity $V^b \in \mathbb{R}^3$.

(e) The adjoint transformation is used to map body velocities $V^b \in \mathbb{R}^3$ into spatial velocities $V^s \in \mathbb{R}^3$. Show that the adjoint transformation for planar rigid motions is given by

$$\mathrm{Ad}_g = \begin{bmatrix} R & \begin{bmatrix} p_y \\ -p_x \end{bmatrix} \\ 0 & 1 \end{bmatrix}.$$

12. Verify that for $\omega \in \mathbb{R}^3$, $\|\omega\| \neq 1$

$$e^{\widehat{\omega}\theta} = I + \frac{\widehat{\omega}}{\|\omega\|}\sin(\|\omega\|\theta) + \frac{\widehat{\omega}^2}{\|\omega^2\|}\big(1 - \cos(\|\omega\|\theta)\big).$$

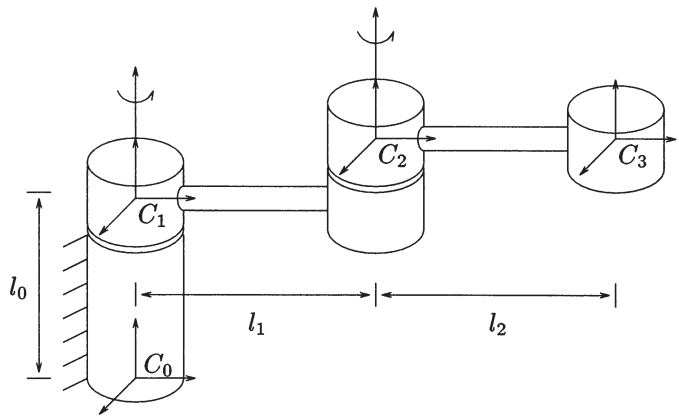


Figure 2.17: A two degree of freedom manipulator.

13. Let $\xi_a = (-\omega_a \times q_a + h\omega_a, \omega_a)$ be the twist associated with a screw having pitch h and axis $l = \{q_a + \lambda\omega_a : \lambda \in \mathbb{R}\}$, where all quantities are specified relative to a coordinate frame A.

    (a) Let B be a second coordinate frame with configuration $g_{ab} \in SE(3)$. Show that the representation of the twist relative to B is given by
    $$\xi_b = \mathrm{Ad}_{g_{ab}}^{-1} \xi_a = \mathrm{Ad}_{g_{ba}} \xi_a.$$

    (b) Suppose instead that we move the screw via a rigid body transformation $g \in SE(3)$. Show that the transformed screw can be represented by the twist
    $$\xi_a' = \mathrm{Ad}_g \xi_a,$$
    still relative to the A coordinate frame.

14. Use homogeneous representations to show that the following identities hold:

    (a) $(\mathrm{Ad}_g)^{-1} = \mathrm{Ad}_{g^{-1}}$ for all $g \in SE(3)$.

    (b) $\mathrm{Ad}_{g_1 g_2} = \mathrm{Ad}_{g_1} \mathrm{Ad}_{g_2}$ for all $g_1, g_2 \in SE(3)$.

15. Prove Proposition 2.15: $V_{ac}^b = \mathrm{Ad}_{g_{bc}^{-1}} V_{ab}^b + V_{bc}^b$.

16. Figure 2.17 shows a two degree of freedom manipulator. Let l_0, l_1, l_2 be the link length parameters and θ_1, θ_2 the joint angle variables of link 1 and link 2, respectively.

    (a) Express the position and orientation of frame C_3 relative to frame C_0 in terms of the joint angle variables and the link parameters.

(b) Compute the spatial velocity of C_3 relative to C_0 as functions of the joint angles and the joint rates.

(c) Compute the body velocity of C_3 relative to C_0 as functions of the joint angles and the joint rates.

(d) *Optional:* Find the spatial velocity of the origin of C_3 and use this to check your answer for parts (b) and (c).

You may want to use a symbolic math package, such as the one described in Appendix B, to carry out the computations in this exercise.

17. *Frame invariance and reciprocal screw systems*
An operator is said to be *frame invariant* if it does not depend on the choice of coordinate frame used to carry out the calculations. Operations which are frame invariant can be computed relative to *any* coordinate frame, which can simplify calculations.

(a) Show that the reciprocal product between two screws is frame invariant.

(b) Show that the inner product between two twists is *not* frame invariant.

(c) Calculate a basis for the system of screws reciprocal to a zero-pitch screw through a point q. Give a geometric interpretation for the screws which form your basis. (Hint: perform your calculations relative to a specially chosen frame.)

(d) Calculate a basis for the system of screws reciprocal to an infinite pitch screw. Give a geometric interpretation for the screws which form your basis.

(e) Using reciprocal screws, show that three parallel, coplanar, zero-pitch screws are dependent. That is, exhibit a system of four independent screws which are reciprocal to each of the coplanar screws.

18. *Hybrid representation of velocity*
A seemingly natural way of representing the velocity of a rigid body is to use $\dot{p}$ to represent the linear velocity and ω^s to represent the angular velocity. We call $V_{ab}^h = (\dot{p}, \omega^s)$ the *hybrid velocity* of a rigid body.

(a) Show that the hybrid velocity is related to the body velocity by the relationship

$$V^h = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} V^b$$

and hence represents the velocity of the rigid body relative to a frame attached to the origin of the rigid body, but whose orientation remains fixed relative to the inertial frame.

(b) Consider the motion of three coordinates frames, A, B, and C. Show that the following relationship holds between their hybrid velocities:

$$V_{ac}^h = \mathrm{Ad}_{(-R_{ab}p_{bc})} V_{ab}^h + \mathrm{Ad}_{R_{ab}} V_{bc}^h.$$

where Ad_p denotes the adjoint map corresponding to a pure translation by p and Ad_R denotes the adjoint map corresponding to pure rotation.

(c) Show that the hybrid velocity of a rigid body is independent of the position of the spatial frame, but not its orientation.

(d) Show that the hybrid velocity of a rigid body is independent of the orientation of the body frame, but not its position.

(e) Interpret a wrench in hybrid coordinates and calculate the change of basis formulas for a change in spatial and/or body frames.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 3

Manipulator Kinematics

DOI: 10.1201/9781315136370-3
This chapter has been made available under a CC-BY 4.0 license.

The kinematics of a robot manipulator describes the relationship between the motion of the joints of the manipulator and the resulting motion of the rigid bodies which form the robot. This chapter gives a description of the kinematics for a general n degree of freedom, open-chain robot manipulator using the tools presented in Chapter 2. We also present a brief treatment of redundant and parallel manipulators using this same framework.

1 Introduction

Most modern manipulators consist of a set of rigid links connected together by a set of joints. Motors are attached to the joints so that the overall motion of the mechanism can be controlled to perform a given task. A tool, typically a gripper of some sort, is attached to the end of the robot to interact with the environment.

Although any type of joint mechanism can be used to connect the links of a robot, traditionally the joints are chosen from a set of six mechanisms called *lower pairs*. These special types of joint mechanisms correspond to subgroups of the special Euclidean group $SE(3)$. They represent revolute, prismatic, helical, cylindrical, spherical, and planar joints.

The *revolute*, *prismatic*, and *helical* joints each correspond to screw motions, with the helical joint corresponding to a general screw with finite, nonzero pitch. A *cylindrical joint* has two independent degrees of freedom and is typically constructed by combining a revolute and a prismatic joint such that their axes are coincident. Revolute and prismatic joints are by far the most common type of joint encountered in robotics.

A *spherical joint* joint is a mechanism which is capable of arbitrary rotations. Passive spherical joints often consist of a ball inserted into a socket, and are therefore referred to as *ball and socket joints*. Unfortu-

type of mechanism does not work well if the joint is to exert forces and torques, and hence actuated spherical joints are most often constructed by combining three revolute joints (with motors) such that their axes all intersect at a point. The orientation of the joint is then given by

$$R = e^{\widehat{\omega}_1\theta_1} e^{\widehat{\omega}_2\theta_2} e^{\widehat{\omega}_3\theta_3},$$

where $\omega_1, \omega_2, \omega_3 \in \mathbb{R}^3$ represent the directions of the three axes. This is very similar to an Euler angle parameterization of orientation and has the same limitations in terms of singularities of the mechanism. Spherical mechanisms are often used as wrists in robot manipulators to allow arbitrary orientation of the gripper or tool at the end of the robot.

Planar joints allow for arbitrary translation and rotation in the plane. Along with helical joints, they are the least commonly used of the lower-pair mechanisms. A planar joint can be built from a revolute joint attached to two independent prismatic joints. The motion of a planar joint is restricted to $SE(2)$, regarded as a 3-dimensional subgroup of $SE(3)$.

Modern robot manipulators, and kinematic mechanisms in general, are typically constructed by connecting different lower-pair joints together using rigid links. Since each of the joints restricts the motion of adjacent links to a subgroup of $SE(3)$, the tools developed in the last chapter provide a natural starting point for the analysis of such mechanisms. In this chapter and the next, we concentrate on the kinematics, dynamics, and control of open-chain robot manipulators, in which a number of links are attached serially by a set of actuated joints. By controlling the forces and torques on each of the links, we seek to move the robot in a specified way.

The heart of the formulation which we present here is the *product of exponentials formula*, which represents the kinematics of an open-chain mechanism as the product of exponentials of twists. This setting works whenever the joints of the robot consist of either revolute, prismatic, or helical joints, which is the case for practically all commercially available robot manipulators. It provides a global, geometric representation of the kinematics of a manipulator which greatly simplifies the analysis of the mechanism and provides a very structured parameterization for open-chain robots.

This chapter is organized as follows: Section 2 contains a derivation of the product of exponentials formula for the forward kinematics of an arbitrary open-chain manipulator. We concentrate on the most general case, where the end-effector configuration lies in $SE(3)$. Section 3 discusses the inverse problem of finding a set of joint angles which causes the end-effector to have a desired configuration. We make extensive use of a set of subproblems originally proposed by Paden and Kahan which are very closely related to the exponential representation of rigid body motion. Section 4 derives the velocity and force relationships between

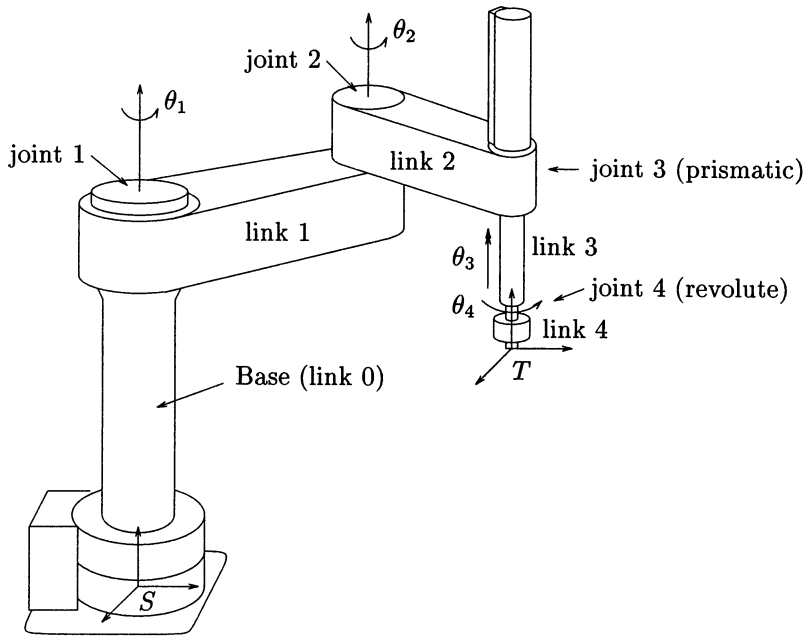


Figure 3.1: Numbering conventions for an AdeptOne robot.

the end-effector and the joints, and introduces the manipulator Jacobian for a robot. Finally, in Section 5 we extend some of the main results of this chapter to redundant manipulators and parallel mechanisms.

2 Forward Kinematics

2.1 Problem statement

The *forward kinematics* of a robot determines the configuration of the end-effector (the gripper or tool mounted on the end of the robot) given the relative configurations of each pair of adjacent links of the robot. In this section, we restrict ourselves to open-chain manipulators in which the links form a single serial chain and each pair of links is connected either by a revolute joint or a prismatic (sliding) joint. To fix notation, we number the joints from 1 to n, starting at the base, and number the links such that joint i connects links $i-1$ and i. Link 0 is taken to be the base of the manipulator and link n is attached rigidly to the end-effector. Figure 3.1 illustrates our choice of notational conventions for an AdeptOne robot (a type of SCARA manipulator).

The *joint space* Q of a manipulator consists of all possible values of the joint variables of the robot. This is also the configuration space of

the robot, since specifying the joint angles specifies location of all of the links of the robot. For revolute joints, the joint variable is given by an angle $\theta_i \in [0, 2\pi)$ with the angle 2π equated to the angle 0. This set of joint angles is naturally associated with a unit circle in the plane, denoted $\mathbb{S}^1$, and hence we write $\theta_i \in \mathbb{S}^1$ for revolute joints. We measure all joint angles using a right-handed coordinate system, so that an angle about a directed axis is positive if it represents a clockwise rotation as viewed along the direction of the axis. Prismatic joints are described by a linear displacement $\theta_i \in \mathbb{R}$ along a directed axis, where positive displacement is taken along the direction of the axis.

The joint space Q is the Cartesian product between each of these individual joint spaces. The number of degrees of freedom of an open-chain manipulator is equal to the the number of joints in the manipulator. For the four degree of freedom SCARA robot of Figure 3.1, for instance, we have $\theta = (\theta_1, \theta_2, \theta_3, \theta_4) \in \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{R} \times \mathbb{S}^1 = Q$. For manipulators with multiple revolute joints, we use $\mathbb{T}^p$ to represent the p-torus, defined to be the Cartesian product of p copies of $\mathbb{S}^1$: $\mathbb{T}^p = \mathbb{S}^1 \times \cdots \times \mathbb{S}^1$. The joint space of a manipulator with p revolute joints and r prismatic joints is $Q = \mathbb{T}^p \times \mathbb{R}^r$ and has $p + r$ degrees of freedom. In practice, Q may be defined to be a subset of this unrestricted joint space in order to account for joint constraints such as finite displacements and rotations.

We attach two coordinate frames to the manipulator, as illustrated in Figure 3.1. The *base frame*, S, is attached to a point on the manipulator which is stationary with respect to link 0. Usually, S is attached directly to link 0, although this need not be the case, as we shall see later. (The reason for the use of the letter S instead of B is to avoid confusing the base frame with the body frame, which is ordinarily attached to a moving object.) The *tool frame*, T, is attached to the end-effector of the robot, so that the tool frame moves when the joints of the robot move.

The forward kinematics problem can now be formalized. For simplicity, we refer to all joint variables as angles, although both angles and displacements are allowed, depending on the joint type. Given a set of joint angles $\theta \in Q$, we wish to determine the configuration of the tool frame T relative to the base frame S. The forward kinematics is represented by a mapping $g_{st} : Q \to SE(3)$ which describes this relationship. The goal of this section is to show how to explicitly construct g_{st} for a given open-chain robot manipulator and explore the structure of this mapping.

Classically, the forward kinematics map for an open-chain manipulator is constructed by composing the rigid motions due to the individual joints. Consider, for example, the two degree of freedom manipulator shown in Figure 3.2. To compute the configuration of the tool frame T relative to the base frame S, we concatenate the rigid motions between

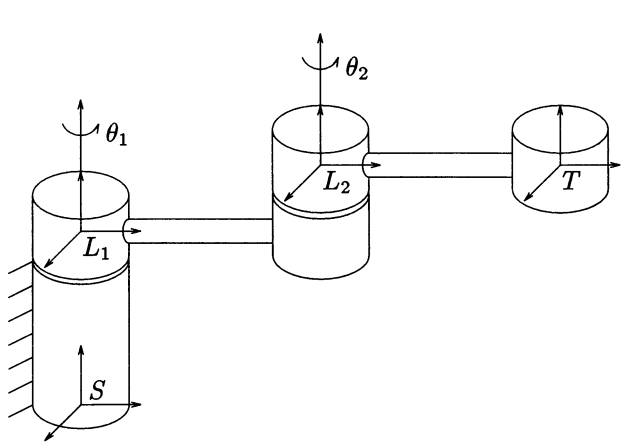


Figure 3.2: A two degree of freedom manipulator.

adjacent frames:

$$g_{st}(\theta_1, \theta_2) = g_{sl_1}(\theta_1) g_{l_1 l_2}(\theta_2) g_{l_2 t}.$$

The mapping $g_{st} : \mathbb{T}^2 \to SE(3)$ represents the forward kinematics of the manipulator: it gives the end-effector configuration as a function of the joint angles.

This procedure is easily extended to any open-chain mechanism. If we define $g_{l_{i-1} l_i}(\theta_i)$ as the transformation between the adjacent link frames, then the overall kinematics are given by

$$g_{st}(\theta) = g_{sl_1}(\theta_1) g_{l_1 l_2}(\theta_2) \cdots g_{l_{n-1} l_n}(\theta_n) g_{l_n t}. \tag{3.1}$$

Equation (3.1) is a general formula for the forward kinematics map of an open-chain manipulator in terms of the relative transformations between adjacent link frames.

2.2 The product of exponentials formula

A more geometric description of the kinematics can be obtained by using the fact that motion of the individual joints is generated by a twist associated with the joint axis. Recall that if ξ is a twist, then the rigid motion associated with rotating and translating along the axis of the twist is given by

$$g_{ab}(\theta) = e^{\widehat{\xi}\theta} g_{ab}(0).$$

If ξ corresponds to a prismatic (infinite pitch) joint, then $\theta \in \mathbb{R}$ is the amount of translation; otherwise, $\theta \in \mathbb{S}^1$ measures the angle of rotation about the axis.

Consider again the two degree of freedom manipulator shown in Figure 3.2. Suppose that we fix the first joint and consider the configuration of the tool frame as a function of θ_2 only. This is a simple revolute (zero-pitch) screw motion about the axis of the second joint and hence we can write

$$g_{st}(\theta_2) = e^{\widehat{\xi}_2\theta_2} g_{st}(0),$$

where ξ_2 is the twist corresponding to rotation about the second joint. Next, fix θ_2 and move only θ_1. By composition, the end-effector configuration becomes

$$g_{st}(\theta_1, \theta_2) = e^{\widehat{\xi}_1\theta_1} g_{st}(\theta_2) = e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} g_{st}(0), \tag{3.2}$$

where ξ_1 is the twist associated with the first joint. Equation (3.2) is an alternative formula for the manipulator forward kinematics. Note that ξ_1 and ξ_2 are constant twists obtained by evaluating the screw motion for each joint at the $\theta_1 = \theta_2 = 0$ configuration of the manipulator.

The simple form of equation (3.2) appears to rely on moving θ_2 first, followed by θ_1. This allowed us to represent the joint motions as twists about constant axes. To show that this representation does not depend on the order in which we move the joints, we can derive the forward kinematics by moving θ_1 first, and then θ_2. In this case,

$$g_{st}(\theta_1) = e^{\widehat{\xi}_1\theta_1} g_{st}(0)$$

is the motion due to moving θ_1 with θ_2 fixed. This motion moves the axis of θ_2, and rotation of the second link occurs around a *new* axis,

$$\xi_2' = \mathrm{Ad}_{e^{\widehat{\xi}_1\theta_1}} \xi_2.$$

Using the properties of the matrix exponential (see Exercise 8 in Chapter 2), the rigid body transformation

$$e^{\widehat{\xi}_2'\theta_2} = e^{\widehat{\xi}_1\theta_1} \left(e^{\widehat{\xi}_2\theta_2}\right) e^{-\widehat{\xi}_1\theta_1}$$

describes motion about the new axis. Thus,

$$\begin{aligned} g_{st}(\theta_1, \theta_2) &= e^{\widehat{\xi}_2'\theta_2} e^{\widehat{\xi}_1\theta_1} g_{st}(0) \\ &= e^{\widehat{\xi}_1\theta_1} \left(e^{\widehat{\xi}_2\theta_2}\right) e^{-\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_1\theta_1} g_{st}(0) \\ &= e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} g_{st}(0), \end{aligned}$$

as before.

We can generalize this procedure to find the forward kinematics map for an arbitrary open-chain manipulator with n degrees of freedom. Let S be a frame attached to the base of the manipulator and T be a frame

attached to the last link of the manipulator. Define the *reference configuration* of the manipulator to be the configuration of the manipulator corresponding to $\theta = 0$ and let $g_{st}(0)$ represent the rigid body transformation between T and S when the manipulator is in its reference configuration. For each joint, construct a twist ξ_i which corresponds to the screw motion for the i^{th} joint with all other joint angles held fixed at $\theta_j = 0$. For a revolute joint, the twist ξ_i has the form

$$\xi_i = \begin{bmatrix} -\omega_i \times q_i \\ \omega_i \end{bmatrix},$$

where $\omega_i \in \mathbb{R}^3$ is a unit vector in the direction of the twist axis and $q_i \in \mathbb{R}^3$ is any point on the axis.[1] For a prismatic joint,

$$\xi_i = \begin{bmatrix} v_i \\ 0 \end{bmatrix},$$

where $v_i \in \mathbb{R}^3$ is a unit vector pointing in the direction of translation. All vectors and points are specified relative to the base coordinate frame S.

Combining the individual joint motions, the forward kinematics map, $g_{st} : Q \to SE(3)$, is given by

$$\boxed{g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)} \tag{3.3}$$

The ξ_i's must be numbered sequentially starting from the base, but $g_{st}(\theta)$ gives the configuration of the tool frame independently of the order in which the rotations and translations are actually performed. Equation (3.3) is called the *product of exponentials formula* for the manipulator forward kinematics.

Example 3.1. SCARA forward kinematics
Consider the SCARA manipulator shown in Figure 3.3. It consists of four joints—three revolute and one prismatic (note that we have chosen to order the joints differently than for the AdeptOne robot in Figure 3.1). We let $\theta = 0$ correspond to the fully extended configuration and attach base and tool frames as shown in the figure.

The transformation between tool and base frames at $\theta = 0$ is given by

$$g_{st}(0) = \begin{bmatrix} I & \begin{pmatrix} 0 \\ l_1 + l_2 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix}.$$

[1] We choose the convention $-\omega \times q$ instead of $q \times \omega$ since the former can be correctly interpreted in both the spatial and planar cases (see Exercise 11 in Chapter 2).

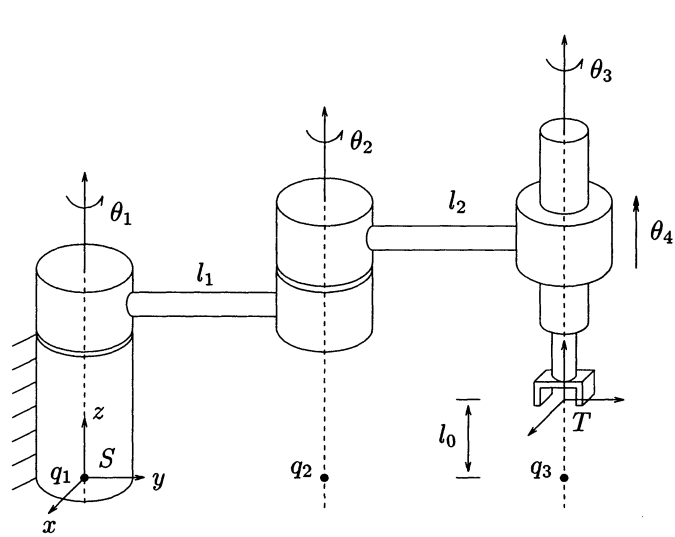


Figure 3.3: SCARA manipulator in its reference configuration.

To construct the twists for the revolute joints, note that

$$\omega_1 = \omega_2 = \omega_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and we can choose axis points

$$q_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad q_2 = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} \qquad q_3 = \begin{bmatrix} 0 \\ l_1 + l_2 \\ 0 \end{bmatrix}.$$

This yields twists

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} l_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_3 = \begin{bmatrix} l_1+l_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

The prismatic joint points in the z direction and has an associated twist

$$\xi_4 = \begin{bmatrix} v_4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The forward kinematics map of the manipulator has the form

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} e^{\widehat{\xi}_3\theta_3} e^{\widehat{\xi}_4\theta_4} g_{st}(0) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix}.$$

The individual exponentials are given by

$$e^{\widehat{\xi}_1\theta_1} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{\widehat{\xi}_2\theta_2} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_1\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_1(1-\cos\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{\widehat{\xi}_3\theta_3} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & (l_1+l_2)\sin\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & (l_1+l_2)(1-\cos\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$e^{\widehat{\xi}_4\theta_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Expanding the terms in the product of exponentials formula yields

$$\begin{aligned} g_{st}(\theta) &= \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix} \\ R(\theta) &= \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ p(\theta) &= \begin{bmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ l_0+\theta_4 \end{bmatrix}. \end{aligned} \tag{3.4}$$

Example 3.2. Elbow manipulator forward kinematics
Consider the elbow manipulator shown in Figure 3.4. The mechanism consists of two intersecting axes at the shoulder, an elbow, and a spherical wrist (modeled as three intersecting axes). The reference configuration ($\theta = 0$) is fully extended, as shown.

The forward kinematics is computed by calculating the individual twist motions for each joint. The transformation between the tool and base frames at $\theta = 0$ is given by

$$g_{st}(0) = \begin{bmatrix} I & \begin{pmatrix} 0 \\ l_1+l_2 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix}.$$

The first two joints have twists

$$\xi_1 = \begin{bmatrix} -\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ l_0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} -\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ l_0 \end{pmatrix} \\ \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ -l_0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}.$$

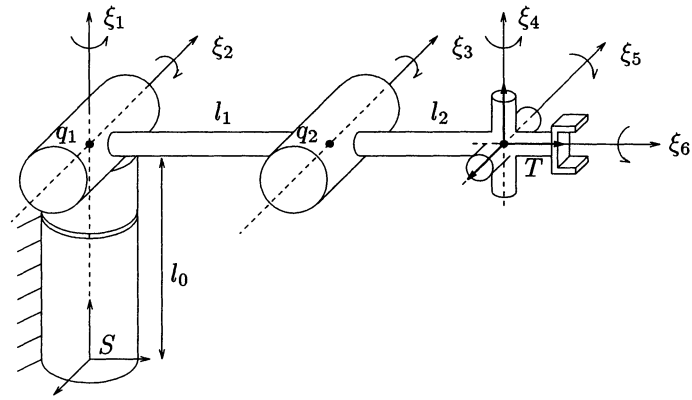


Figure 3.4: Elbow manipulator.

Note that we have used $q_1 = (0, 0, l_0)$ for the first twist; we could just as well have used the origin or any other point on the axis of the twist. The other twists are calculated in a similar manner:

$$\xi_3 = \begin{bmatrix} 0 \\ -l_0 \\ l_1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \qquad \xi_4 = \begin{bmatrix} l_1+l_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_5 = \begin{bmatrix} 0 \\ -l_0 \\ l_1+l_2 \\ -1 \\ 0 \\ 0 \end{bmatrix} \qquad \xi_6 = \begin{bmatrix} -l_0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

The full forward kinematics are

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_6\theta_6} g_{st}(0) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix}$$

where

$$R(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$p(\theta) = \begin{bmatrix} -\sin\theta_1(l_1\cos\theta_2 + l_2\cos(\theta_2+\theta_3)) \\ \cos\theta_1(l_1\cos\theta_2 + l_2\cos(\theta_2+\theta_3)) \\ l_0 - l_1\sin\theta_2 - l_2\sin(\theta_2+\theta_3) \end{bmatrix},$$

and, using the notation $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$, $s_{ij} =$

$\sin(\theta_i + \theta_j)$,

$$
\begin{aligned}
r_{11} &= c_6(c_1c_4 - s_1c_{23}s_4) + s_6(s_1s_{23}c_5 + s_1c_{23}c_4s_5 + c_1s_4s_5) \\
r_{12} &= -c_5(s_1c_{23}c_4 + c_1s_4) + s_1s_{23}s_5 \\
r_{13} &= c_6(-c_5s_1s_{23} - (c_{23}c_4s_1 + c_1s_4)s_5) + (c_1c_4 - c_{23}s_1s_4)s_6
\end{aligned}
$$

$$
\begin{aligned}
r_{21} &= c_6(c_4s_1 + c_1c_{23}s_4) - (c_1c_5s_{23} + (c_1c_{23}c_4 - s_1s_4)s_5)s_6 \\
r_{22} &= c_5(c_1c_{23}c_4 - s_1s_4) - c_1s_{23}s_5 \\
r_{23} &= c_6(c_1c_5s_{23} + (c_1c_{23}c_4 - s_1s_4)s_5) + (c_4s_1 + c_1c_{23}s_4)s_6
\end{aligned}
$$

$$
\begin{aligned}
r_{31} &= -(c_6s_{23}s_4) - (c_{23}c_5 - c_4s_{23}s_5)s_6 \\
r_{32} &= -(c_4c_5s_{23}) - c_{23}s_5 \\
r_{33} &= c_6(c_{23}c_5 - c_4s_{23}s_5) - s_{23}s_4s_6.
\end{aligned}
$$

2.3 Parameterization of manipulators via twists

Using the product of exponentials formula, the kinematics of a manipulator is completely characterized by the twist coordinates for each of the joints. We now consider some issues related to parameterizing robot motion using twists.

Choice of base frame and reference configuration

In the examples above, we chose the base frame for the manipulator to be at the base of the robot. Other choices of the base frame are possible, and can sometimes lead to simplified calculations. One natural choice is to place the base frame coincident with the tool frame in the reference configuration. That is, we choose a base frame which is fixed relative to the base of the robot and which lines up with the tool frame when $\theta = 0$. This simplifies calculations since $g_{st}(0) = I$ with this choice of base frame and hence

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_n\theta_n}.$$

A further degree of freedom in specifying the manipulator kinematics is the choice of the reference configuration for the manipulator. Recall that the reference configuration was the configuration corresponding to setting all of the joint variables to 0. By adding or subtracting a fixed offset from each joint variable, we can define *any* configuration of the manipulator as the reference configuration. The twist coordinates for the individual joints of a manipulator depend on the choice of reference configuration (as well as base frame) and so the reference configuration is usually chosen such that the kinematic analysis is as simple as possible. For example, a common choice is to define the reference configuration

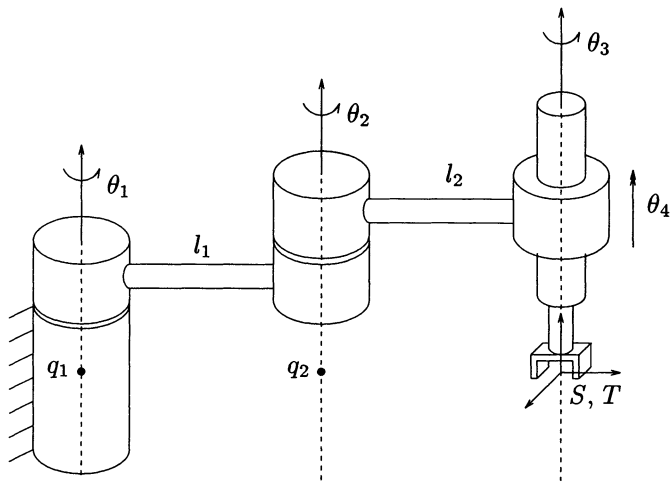


Figure 3.5: SCARA manipulator in its reference configuration, with base frame coincident with tool frame.

such that points on the twist axes for the joints have a simple form, as in all of the examples above.

Example 3.3. SCARA forward kinematics with alternate base frame

Consider the SCARA manipulator with base frame coincident with the tool frame at $\theta = 0$, as shown in Figure 3.5. The twists are now calculated with respect to the new base frame:

$$\xi_1 = \begin{bmatrix} -\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ -l_1 - l_2 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} -l_1 - l_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\xi_2 = \begin{bmatrix} -\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ -l_2 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} -l_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

and similar calculations yield:

$$\xi_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Expanding the product of exponentials formula gives

$$g_{st}(\theta) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$p(\theta) = \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1+\theta_2) \\ -l_1 - l_2 + l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2) \\ \theta_4 \end{bmatrix}.$$

Note that $g_{st}(0) = I$, which is consistent with the fact that the base and tool frames are coincident at $\theta = 0$. Compare this formula with the kinematics map derived in Example 3.1.

Relationship with Denavit-Hartenberg parameters

Given a base frame S and a tool frame T, the coordinates of the twists corresponding to each joint of a robot manipulator provide a complete parameterization of the kinematics of the manipulator. An alternative parameterization, which is the de facto standard in robotics, is the use of *Denavit-Hartenberg parameters* [25]. In this section, we discuss the relationships between these two parameterizations and their relative merits.

Denavit-Hartenberg parameters are obtained by applying a set of rules which specify the position and orientation of frames L_i attached to each link of the robot and then constructing the homogeneous transformations between frames, denoted $g_{l_{i-1}l_i}$. By convention, we identify the base frame S with L_0. The kinematics of the mechanism can be written as

$$g_{st}(\theta) = g_{l_0 l_1}(\theta_1) g_{l_1 l_2}(\theta_2) \cdots g_{l_{n-1} l_n}(\theta_n) g_{l_n t}, \tag{3.6}$$

just as in equation (3.1). Each of the transformations g_{l_{i-1},l_i} has the form

$$g_{l_{i-1},l_i} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \cos\alpha_i & \sin\phi_i \sin\alpha_i & a_i \cos\phi_i \\ \sin\phi_i & \cos\phi_i \cos\alpha_i & -\cos\phi_i \sin\alpha_i & a_i \sin\phi_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the four scalars α_i, a_i, d_i, and ϕ_i are the parameters for the i^{th} link. For revolute joints, ϕ_i corresponds to the joint variable θ_i, while for prismatic joints, d_i corresponds to the joint variable θ_i. Denavit-Hartenberg parameters are available for standard industrial robots and are used by most commercial robot simulation and programming systems.

It may seem somewhat surprising that only four parameters are needed to specify the relative link displacements, since the twists for each joint have six independent parameters. This is achieved by cleverly choosing

the link frames so that certain cancellations occur. In fact, it is possible to give physical interpretations to the various parameters based on the relationships between adjacent link frames. An excellent discussion can be found in Spong and Vidyasagar [111].

There is not a simple one-to-one mapping between the twist coordinates for the joints of a robot manipulator and the Denavit-Hartenberg parameters. This is because the twist coordinates for each joint are specified with respect to a single base frame and hence do not directly represent the relative motions of each link with respect to the previous link. To see this, let $\xi_{i-1,i}$ be the twist for the i^{th} link *relative to the previous link frame.* Then, $g_{l_{i-1}l_i}$ is given by

$$g_{l_{i-1}l_i} = e^{\widehat{\xi}_{i-1,i}\theta_i} g_{l_{i-1}l_i}(0) \tag{3.7}$$

and the forward kinematics map becomes

$$g_{st}(\theta) = e^{\widehat{\xi}_{01}\theta_1} g_{l_0 l_1}(0)\, e^{\widehat{\xi}_{12}\theta_2} g_{l_1 l_2}(0) \cdots e^{\widehat{\xi}_{n-1,n}\theta_n} g_{l_{n-1}l_n}(0). \tag{3.8}$$

This is evidently not the same as the product of exponentials formula, though it bears some resemblance to it.

The relationship between the twists ξ_i and the pairs $g_{l_{i-1}l_i}(0)$ and $\xi_{i-1,i}$ can be determined using the adjoint mapping. We first rewrite equation (3.8) as

$$\begin{aligned} g_{st}(\theta) = {} & e^{\widehat{\xi}_{01}\theta_1} \left(g_{l_0 l_1}(0) e^{\widehat{\xi}_{12}\theta_2} g_{l_0 l_1}^{-1}(0) \right) \cdot \\ & \left(g_{l_0 l_2}(0) e^{\widehat{\xi}_{23}\theta_3} g_{l_0 l_2}^{-1}(0) \right) \cdots \left(g_{l_0 l_{n-1}}(0) e^{\widehat{\xi}_{n-1,n}\theta_n} g_{l_0 l_{n-1}}^{-1}(0) \right) g_{l_0 t}(0). \end{aligned} \tag{3.9}$$

We can simplify this equation by making use of the relationship

$$g e^{\widehat{\xi}\theta} g^{-1} = e^{(\mathrm{Ad}_g \xi)^\wedge \theta}$$

to obtain

$$g_{st}(\theta) = e^{\widehat{\xi}_{01}\theta_1} e^{(\mathrm{Ad}_{g_{l_0 l_1}(0)}\, \xi_{12})^\wedge \theta_2} \cdots e^{(\mathrm{Ad}_{g_{l_0 l_{n-1}}(0)}\, \xi_{n-1,n})^\wedge \theta_n} g_{l_0 t}(0).$$

It follows immediately that

$$\xi_i = \mathrm{Ad}_{g_{l_0 l_{i-1}}(0)}\, \xi_{i-1,i}. \tag{3.10}$$

This formula verifies that the twist ξ_i is the joint twist for the i^{th} joint in its reference configuration and written relative to the base coordinate frame.

Given the Denavit-Hartenberg parameters for a manipulator, the corresponding twists ξ_i can be determined by first parameterizing $g_{i-1,i}$ using exponential coordinates, as in equation (3.7), and then applying

equation (3.10). However, in almost all instances it is substantially easier to construct the joint twists ξ_i directly, by writing down the direction of the joint axes and, in the case of revolute joints, choosing a convenient point on each axis. Indeed, one of the most attractive features of the product of exponentials formula is its usage of only two coordinate frames, the base frame S, and the tool frame T. This property, combined with the geometric significance of the twists ξ_i, make the product of exponentials representation a superior alternative to the use of Denavit-Hartenberg parameters.

2.4 Manipulator workspace

The *workspace* of a manipulator is defined as the set of all end-effector configurations which can be reached by some choice of joint angles. If Q is the configuration space of a manipulator and $g_{st} : Q \to SE(3)$ is the forward kinematics map, then the workspace W is defined as the set

$$W = \{g_{st}(\theta) : \theta \in Q\} \subset SE(3). \tag{3.11}$$

The workspace is used when planning a task for the manipulator to execute; all desired motions of the manipulator must remain within the workspace. We refer to this notion of workspace as the *complete workspace* of a manipulator.

Characterizing the workspace as a subset of $SE(3)$ is often somewhat difficult to interpret. Instead, one can consider the set of positions (in $\mathbb{R}^3$) which can be reached by some choice of joint angles. This set is called the *reachable workspace* and is defined as

$$W_R = \{p(\theta) : \theta \in Q\} \subset \mathbb{R}^3, \tag{3.12}$$

where $p(\theta) : Q \to \mathbb{R}^3$ is the position component of the forward kinematics map g_{st}. The reachable workspace is the volume of $\mathbb{R}^3$ which can be reached at *some* orientation.

Since the reachable workspace does not consider ability to arbitrarily orient the end-effector, for some tasks it is not a useful measure of the range of a manipulator. The *dextrous workspace* of a manipulator is the volume of space which can be reached by the manipulator with arbitrary orientation:

$$W_D = \{p \in \mathbb{R}^3 : \forall R \in SO(3), \exists\, \theta \text{ with } g_{st}(\theta) = (p, R)\} \subset \mathbb{R}^3. \tag{3.13}$$

The dextrous workspace is useful in the context of task planning since it allows the orientation of the end-effector to be ignored when positioning objects in the dextrous workspace.

For a general robot manipulator, the dextrous workspace can be very difficult to calculate. A common feature of industrial manipulators is to

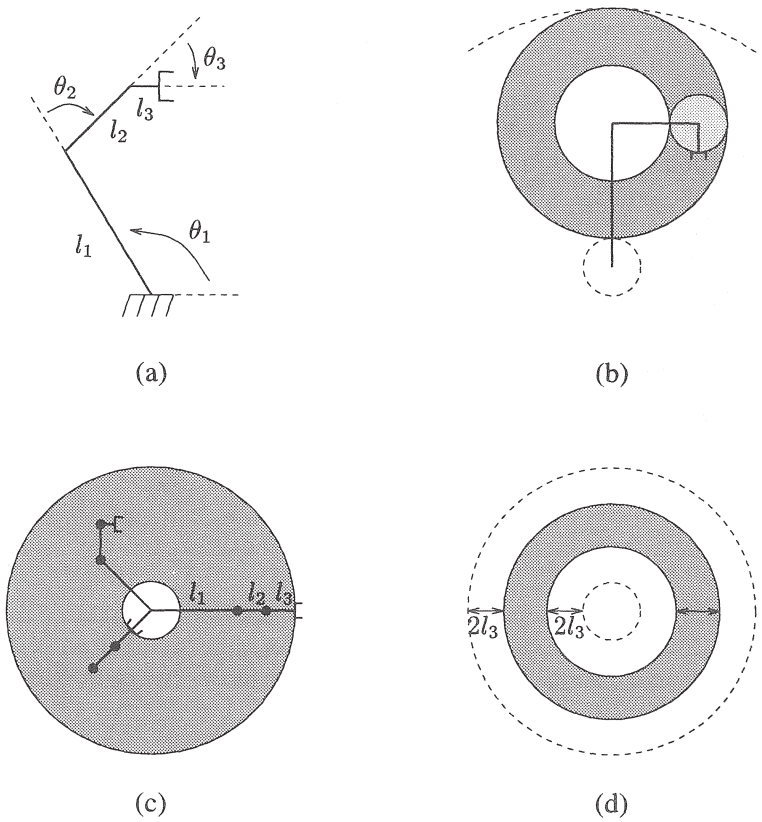


Figure 3.6: Workspace calculations for a planar three-link robot (a). The construction of the workspace is illustrated in (b). The reachable workspace is shown in (c) and the dextrous workspace is shown in (d).

place a spherical wrist at the end of the manipulator, as in the elbow manipulator given in Example 3.2. Recall that a spherical wrist consists of three orthogonal revolute axes which intersect at a point. If the end-effector frame is placed at the origin of the wrist axes, then the spherical wrist can be used to achieve any orientation at a given end-effector position. Hence, for a manipulator with a spherical wrist, the dextrous workspace is equal to the reachable workspace, $W_D = W_R$. Furthermore, the complete workspace for the end-effector satisfies $W = W_R \times SO(3)$. This analysis only holds when the end-effector frame is placed at the center of the spherical wrist; if an offset is present, the analysis becomes more complex.

Example 3.4. Workspace for a planar three-link robot
Consider the planar manipulator shown in Figure 3.6a. Let $g = (x, y, \phi)$

represent the position and orientation of the end-effector. The forward kinematics of the mechanism can be derived using the product of exponentials formula, but are more easily derived using plane geometry:

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \phi &= \theta_1 + \theta_2 + \theta_3. \end{aligned} \tag{3.14}$$

We take $l_1 > l_2 > l_3$, and assume $l_1 > l_2 + l_3$.

The reachable workspace is calculated by ignoring the orientation of the end-effector. To generate it, we first take θ_1 and θ_2 as fixed. The set of reachable points becomes a circle of radius l_3 formed by sweeping θ_3 through all angles (see Figure 3.6b). We now let θ_2 vary through all angles to get an annulus with inner radius $l_2 - l_3$ and outer radius $l_2 + l_3$ centered at the end of the first link. Finally, we generate the reachable workspace by sweeping the annulus through all values of θ_1, to give the reachable workspace. The final construction is shown in Figure 3.6c. W_R is an annulus with inner radius $l_1 - l_2 - l_3$ and outer radius $l_1 + l_2 + l_3$.

The dextrous workspace for this manipulator is somewhat subtle. Although the manipulator has the planar equivalent of a spherical wrist, the end-effector frame is not aligned with the center of the wrist. This reduces the size of the dextrous workspace by $2l_3$ on the inner and outer edges, as shown in Figure 3.6d.

3 Inverse Kinematics

We now consider the inverse kinematics problem: given a desired configuration for the tool frame, find joint angles which achieve that configuration. That is, given a forward kinematics map $g_{st} : Q \to SE(3)$ and a desired configuration $g_d \in SE(3)$, we would like to solve the equation

$$g_{st}(\theta) = g_d \tag{3.15}$$

for some $\theta \in Q$. This problem may have multiple solutions, a unique solution, or no solution.

3.1 A planar example

To illustrate some of the issues in inverse kinematics, we first consider the inverse kinematics of the planar two-link manipulator shown in Figure 3.7a. The forward kinematics can be determined using plane geometry:

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2). \end{aligned}$$

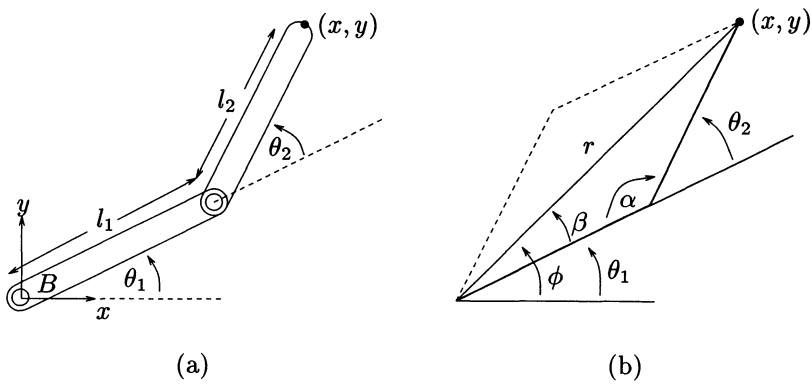


Figure 3.7: Inverse kinematics of a planar two-link manipulator.

The inverse problem is to solve for θ_1 and θ_2, given x and y. A standard trick is to solve the problem using polar coordinates, (r, ϕ), as shown in Figure 3.7b. From this viewpoint, θ_2 is determined by $r = \sqrt{x^2 + y^2}$, and the law of cosines gives

$$\theta_2 = \pi \pm \alpha \qquad \alpha = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2}\right). \tag{3.16}$$

If $\alpha \neq 0$, there are two distinct values of θ_2 which give the appropriate radius; the second is referred to as the "flip solution" and is shown as a dashed line in Figure 3.7b. The complete solution is given by solving for ϕ and using this to determine θ_1. This problem must be solved for each possible value of θ_2, yielding

$$\theta_1 = \text{atan2}(y, x) \pm \beta \qquad \beta = \cos^{-1}\left(\frac{r^2 + l_1^2 - l_2^2}{2l_1 r}\right),$$

where the sign used for β agrees with that used for α.

This planar example illustrates several important features of inverse kinematics problems. In solving an inverse kinematics problem, one first divides the problem into specific subproblems, such as solving for θ_2 given r and then using θ_1 to rotate the end-effector to the proper position. Each subproblem may have zero, one, or many solutions depending on the desired end-effector location. If the configuration is outside of the workspace of the manipulator, then no solution can exist and one of the subproblems must fail to have a solution (consider what happens if $r > l_1 + l_2$ in the example above). Multiple solutions occur when the desired configuration is within the workspace but there are multiple joint configurations which all map to the same end-effector location. If a subproblem generates multiple solutions, then we must complete the solution procedure for all joint angles generated by the subproblem.

Traditionally, inverse kinematics solutions are separated into classes: closed-form solutions and numerical solutions. Closed-form solutions, such as the one given above, allow for fast and efficient calculation of the joint angles which give a desired end-effector configuration. Numerical solutions rely on an interactive procedure to solve equation (3.15). Most industrial manipulators have closed-form solutions. These solutions are obtained in a manner similar to that described above: using geometric and algebraic identities, solve the set of nonlinear, coupled, algebraic equations which define the inverse kinematics problem. An introduction to classical techniques for solving inverse kinematics problems is given in Craig [21].

3.2 Paden-Kahan subproblems

Using the product of exponentials formula for the forward kinematics map, it is possible to develop a geometric algorithm to solve the inverse kinematics problem. This method was originally presented by Paden [86] and built on the unpublished work of Kahan [47].

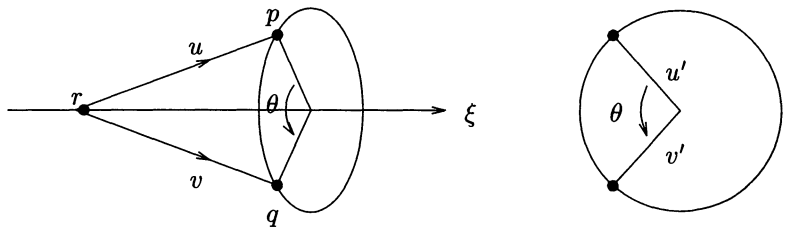
To solve the inverse kinematics problem, we first solve a number of subproblems which occur frequently in inverse solutions for common manipulator designs. One then seeks to reduce the full inverse kinematics problem into appropriate subproblems whose solutions are known. One feature of the subproblems presented here is that they are both geometrically meaningful and numerically stable. Note that this set of subproblems is by no means exhaustive and there may exist manipulators which cannot be solved using these canonical problems. Additional subproblems are explored in the exercises.

For each of the subproblems presented below, we give a statement of the geometric problem to be solved and a detailed solution. On a first reading of this section, it may be difficult to understand the relevance of the specific subproblems presented here. We recommend that the first-time reader skip the solutions until she sees how the subproblems are used in the examples presented later in this section.

Subproblem 1. Rotation about a single axis
Let ξ be a zero-pitch twist with unit magnitude and $p, q \in \mathbb{R}^3$ two points. Find θ such that

$$e^{\widehat{\xi}\theta} p = q. \tag{3.17}$$

Solution. This problem corresponds to rotating a point p about a given axis ξ until it coincides with a second point q, as shown in Figure 3.8a. Let r be a point on the axis of ξ. Define $u = (p - r)$ to be the vector between r and p, and $v = (q - r)$ the vector between r and q. It follows from equation (3.17) and the fact that $\exp(\widehat{\xi}\theta)r = r$ (since r is on the



(a) (b)

Figure 3.8: Subproblem 1: (a) Rotate p about the axis of ξ until it is coincident with q. (b) The projection of u and v onto the plane normal to the twist axis.

axis of ξ) that

$$e^{\widehat{\omega}\theta}u = v, \tag{3.18}$$

where we have used the fact that u and v are vectors, and hence we have $\exp(\widehat{\xi}\theta)u = \exp(\widehat{\omega}\theta)u$.

To determine when the problem has a solution, define u' and v' to be the projections of u and v onto the plane perpendicular to the axis of ξ. If $\omega \in \mathbb{R}^3$ is a unit vector in the direction of the axis of ξ, then

$$u' = u - \omega\omega^T u \qquad \text{and} \qquad v' = v - \omega\omega^T v.$$

The problem has a solution only if the projections of u and v onto the ω-axis and onto the plane perpendicular to ω have equal lengths. More formally, if we project equation (3.18) onto the span of ω and the null space of ω^T, we obtain the necessary conditions

$$\omega^T u = \omega^T v \qquad \text{and} \qquad \|u'\| = \|v'\|. \tag{3.19}$$

If equation (3.19) is satisfied, then we can find θ by looking only at the projected vectors u' and v' as shown in Figure 3.8b. If $u' \neq 0$, then we can determine θ using the relationships

$$\left.\begin{aligned} u' \times v' &= \omega \sin\theta \|u'\|\|v'\| \\ u' \cdot v' &= \cos\theta \|u'\|\|v'\| \end{aligned}\right. \quad \Longrightarrow \quad \theta = \text{atan2}(\omega^T(u' \times v'), u'^T v').$$

If $u' = 0$, then there are an infinite number of solutions since, in this case, $p = q$ and both points lie on the axis of rotation. □

Subproblem 2. Rotation about two subsequent axes

Let ξ_1 and ξ_2 be two zero-pitch, unit magnitude twists with intersecting axes and $p, q \in \mathbb{R}^3$ two points. Find θ_1 and θ_2 such that

$$e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} p = q. \tag{3.20}$$

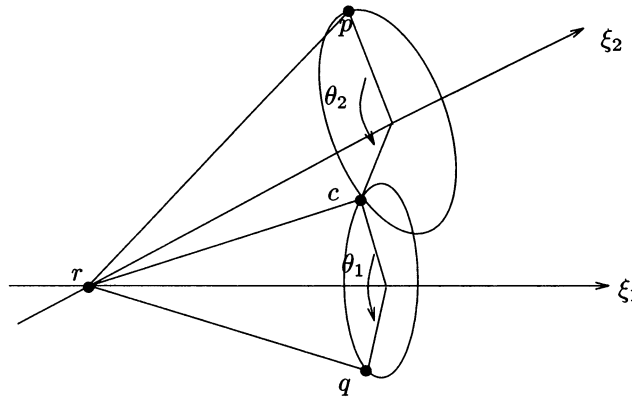


Figure 3.9: Subproblem 2: Rotate p about the axis of ξ_2 followed by a rotation around the axis of ξ_1 such that the final location of p is coincident with q.

Solution. This problem corresponds to rotating a point p first about the axis of ξ_2 by θ_2 and then about the axis of ξ_1 by θ_1, so that the final location of p is coincident with the point q. This motion is depicted in Figure 3.9. If the axes of ξ_1 and ξ_2 coincide, this problem reduces to Subproblem 1 and any θ_1, θ_2 such that $\theta_1 + \theta_2 = \theta$ is a solution, where θ is the solution to Subproblem 1.

If the two axes are not parallel, i.e., $\omega_1 \times \omega_2 \neq 0$, then let c be a point such that

$$e^{\widehat{\xi}_2\theta_2}p = c = e^{-\widehat{\xi}_1\theta_1}q. \tag{3.21}$$

In other words, c represents the point to which p is rotated about the axis of ξ_2 by θ_2. Let r be the point of intersection of the two axes, so that

$$e^{\widehat{\xi}_2\theta_2}(p-r) = c - r = e^{-\widehat{\xi}_1\theta_1}(q-r). \tag{3.22}$$

As before, define vectors $u = (p-r)$, $v = (q-r)$, and $z = (c-r)$. Substituting these expressions into equation (3.22) gives

$$e^{\widehat{\omega}_2\theta_2}u = z = e^{-\widehat{\omega}_1\theta_1}v,$$

which implies that

$$\omega_2^T u = \omega_2^T z \qquad \text{and} \qquad \omega_1^T v = \omega_1^T z \tag{3.23}$$

and $\|u\|^2 = \|z\|^2 = \|v\|^2$. Furthermore, since ω_1, ω_2, and $\omega_1 \times \omega_2$ are linearly independent, we can write

$$z = \alpha\omega_1 + \beta\omega_2 + \gamma(\omega_1 \times \omega_2) \tag{3.24}$$

and

$$\|z\|^2 = \alpha^2 + \beta^2 + 2\alpha\beta\omega_1^T\omega_2 + \gamma^2\|\omega_1 \times \omega_2\|^2. \tag{3.25}$$

Substituting equation (3.24) into equation (3.23) gives a system of two equations in two unknowns:

$$\begin{aligned} \omega_2^T u &= \alpha\omega_2^T\omega_1 + \beta \\ \omega_1^T v &= \alpha + \beta\omega_1^T\omega_2 \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \alpha &= \frac{(\omega_1^T\omega_2)\omega_2^T u - \omega_1^T v}{(\omega_1^T\omega_2)^2 - 1} \\ \beta &= \frac{(\omega_1^T\omega_2)\omega_1^T v - \omega_2^T u}{(\omega_1^T\omega_2)^2 - 1}. \end{aligned}$$

Solving equation (3.25) for γ^2 and using $\|z\|^2 = \|u\|^2$ yields

$$\gamma^2 = \frac{\|u\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T\omega_2}{\|\omega_1 \times \omega_2\|^2}.$$

This equation may have zero, one, or two real solutions. In the case that a solution exists, we can find z—and hence c—given α, β, and γ.

To find θ_1 and θ_2, we solve

$$e^{\widehat{\xi}_2\theta_2}p = c \qquad \text{and} \qquad e^{-\widehat{\xi}_1\theta_1}q = c$$

using Subproblem 1. If there are multiple solutions for c, each of these solutions gives a value for θ_1 and θ_2. Two solutions exist in the case where the circles in Figure 3.9 intersect at two points, one solution when the circles are tangential, and none when the circles fail to intersect. □

Subproblem 3. Rotation to a given distance
Let ξ be a zero-pitch, unit magnitude twist; $p, q \in \mathbb{R}^3$ two points; and δ a real number > 0. Find θ such that

$$\|q - e^{\widehat{\xi\theta}}p\| = \delta. \tag{3.26}$$

Solution. This problem corresponds to rotating a point p about axis ξ until the point is a distance δ from q, as shown in Figure 3.10a. A solution exists if the circle defined by rotating p around ξ intersects the sphere of radius δ centered at q.

To find the explicit solution, we again consider the projection of all points onto the plane perpendicular to ω, the direction of the axis of ξ. Let r be a point on the axis of ξ, and define $u = (p - r)$ and $v = (q - r)$ so that

$$\|v - e^{\widehat{\omega\theta}}u\|^2 = \delta^2. \tag{3.27}$$

The projections of u and v are

$$u' = u - \omega\omega^T u \qquad \text{and} \qquad v' = v - \omega\omega^T v.$$

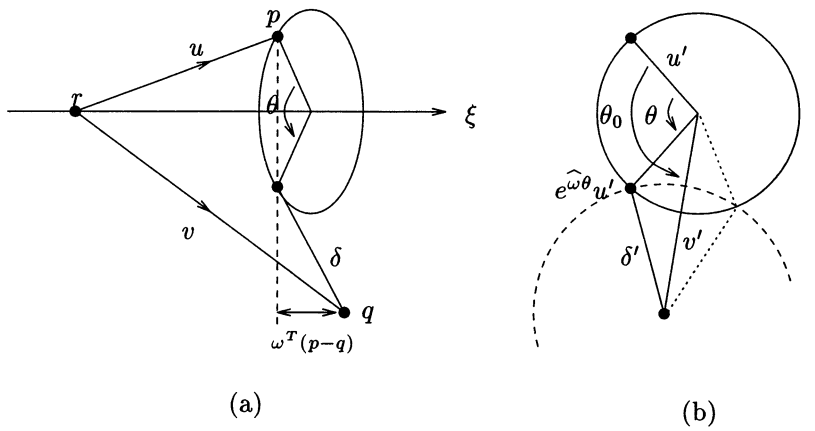


Figure 3.10: Subproblem 3: (a) Rotate p about the axis of ξ until it is a distance δ from q. (b) The projection onto the plane perpendicular to the axis. The dashed line is the "flip" solution.

We can also "project" δ by subtracting the component of $p - q$ in the ω direction,

$$\delta'^2 = \delta^2 - |\omega^T(p - q)|^2,$$

so that equation (3.27) becomes

$$\|v' - e^{\widehat{\omega}\theta}u'\|^2 = \delta'^2$$

(see Figure 3.10b). If we let θ_0 be the angle between the vectors u' and v', we have

$$\theta_0 = \text{atan2}(\omega^T(u' \times v'), u'^T v'). \tag{3.28}$$

We can now use the law of cosines to solve for the angle $\phi = \theta_0 - \theta$. The triangle formed by the center of the axis, $\exp(\widehat{\omega}\theta)u'$, and v' satisfies

$$\|u'\|^2 + \|v'\|^2 - 2\|u'\|\|v'\| \cos\phi = \delta'^2$$

and therefore

$$\theta = \theta_0 \pm \cos^{-1}\left(\frac{\|u'\|^2 + \|v'\|^2 - \delta'^2}{2\|u'\|\|v'\|}\right). \tag{3.29}$$

Equation (3.29) has either zero, one, or two solutions, depending on the number of points in which the circle of radius $\|u'\|$ intersects the circle of radius δ'. □

3.3 Solving inverse kinematics using subproblems

Given the solutions to the subproblems presented above, we must now find techniques for converting the complete inverse kinematics problem into the appropriate subproblems.

The basic technique for simplification is to apply the kinematic equations to special points, such as the intersection of two or more axes. This is a potentially powerful operation since $\exp(\widehat{\xi}\theta)p = p$ if p is on the axis of a revolute twist ξ. Using this, we can eliminate the dependence of certain joint angles by appropriate selection of points. For example, if we wish to solve

$$e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}e^{\widehat{\xi}_3\theta_3} = g,$$

with ξ_1, ξ_2, and ξ_3 all zero-pitch twists, then applying both sides to a point p on the axis of ξ_3 yields

$$e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}p = gp,$$

which can be solved using Subproblem 2 (in the case that ξ_1 and ξ_2 intersect).

Another common trick for reducing a problem to a subproblem is to subtract a point from both sides of an equation and take the norm of the result. Since rigid motions preserve norm, some dependencies can be eliminated. For example, if we wish to solve

$$e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}e^{\widehat{\xi}_3\theta_3} = g$$

for ξ_3, and ξ_1, ξ_2 intersect at a point q, then we can apply both sides of the equation to a point p that is not on the axis of ξ_3 and subtract the point q. Taking the norm of the result yields

$$\begin{aligned}\delta := \|gp - q\| &= \|e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}e^{\widehat{\xi}_3\theta_3}p - q\| \\ &= \|e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}(e^{\widehat{\xi}_3\theta_3}p - q)\| \\ &= \|e^{\widehat{\xi}_3\theta_3}p - q\|\end{aligned}$$

which is Subproblem 3.

We now show how the subproblems of Section 3.2 can be used to solve the inverse kinematics of some common manipulators. Additional examples appear in the exercises.

Example 3.5. Elbow manipulator inverse kinematics
The elbow manipulator in Figure 3.11 consists of a three degree of freedom manipulator with a spherical wrist. This special structure simplifies the inverse kinematics and fits nicely with the subproblems presented earlier.

The equation we wish to solve is

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1}\cdots e^{\widehat{\xi}_6\theta_6}g_{st}(0) = g_d,$$

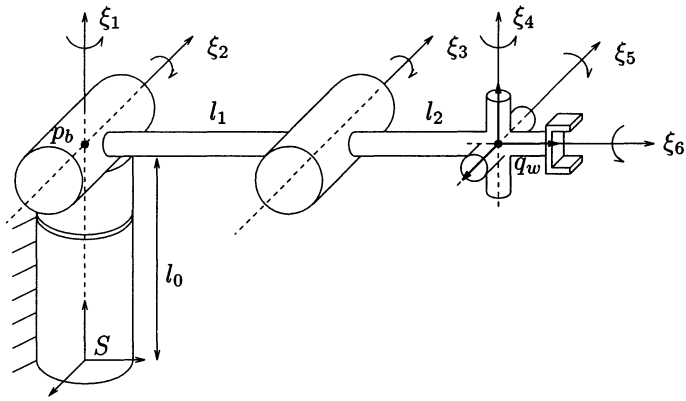


Figure 3.11: Elbow manipulator.

where $g_d \in SE(3)$ is the desired configuration of the tool frame. Post-multiplying this equation by $g_{st}^{-1}(0)$ isolates the exponential maps:

$$e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_6\theta_6} = g_d g_{st}^{-1}(0) =: g_1. \tag{3.30}$$

We determine the requisite joint angles in four steps.

Step 1 *(solve for the elbow angle, θ_3).* Apply both sides of equation (3.30) to a point $p_w \in \mathbb{R}^3$ which is the common point of intersection for the wrist axes. Since $\exp(\widehat{\xi}\theta)p_w = p_w$ if p_w is on the axis of ξ, this yields

$$e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} e^{\widehat{\xi}_3\theta_3} p_w = g_1 p_w. \tag{3.31}$$

Subtract from both sides of equation (3.31) a point p_b which is at the intersection of the first two axes, as shown in Figure 3.11:

$$e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} e^{\widehat{\xi}_3\theta_3} p_w - p_b = e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} \left(e^{\widehat{\xi}_3\theta_3} p_w - p_b \right) = g_1 p_w - p_b. \tag{3.32}$$

Using the property that the distance between points is preserved by rigid motions, take the magnitude of both sides of equation (3.32):

$$\| e^{\widehat{\xi}_3\theta_3} p_w - p_b \| = \| g_1 p_w - p_b \|. \tag{3.33}$$

This equation is in the form required for Subproblem 3, with $p = p_w$, $q = p_b$, and $\delta = \| g_1 p_w - p_b \|$. Applying Subproblem 3, we solve for θ_3.

Step 2 *(solve for the base joint angles).* Since θ_3 is known, equation (3.31) becomes

$$e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} (e^{\widehat{\xi}_3\theta_3} p_w) = g_1 p_w. \tag{3.34}$$

Applying Subproblem 2 with $p = \exp(\widehat{\xi}_3\theta_3)p_w$ and $q = g_1 p_w$ gives the values for θ_1 and θ_2.

Step 3 (solve for two of three wrist angles). The remaining kinematics can be written as

$$e^{\widehat{\xi}_4\theta_4}e^{\widehat{\xi}_5\theta_5}e^{\widehat{\xi}_6\theta_6} = e^{-\widehat{\xi}_3\theta_3}e^{-\widehat{\xi}_2\theta_2}e^{-\widehat{\xi}_1\theta_1}g_d g_{st}^{-1}(0) =: g_2. \tag{3.35}$$

Apply both sides of equation (3.35) to a point p which is on the axis of ξ_6 but *not* on the ξ_4, ξ_5 axes. This gives

$$e^{\widehat{\xi}_4\theta_4}e^{\widehat{\xi}_5\theta_5}p = g_2 p. \tag{3.36}$$

Apply Subproblem 2 to find θ_4 and θ_5.

Step 4 (solve for the remaining wrist angle). The only remaining unknown is θ_6. Rearranging the kinematics equation and applying both sides to any point p which is not on the axis of ξ_6,

$$e^{\widehat{\xi}_6\theta_6}p = e^{-\widehat{\xi}_5\theta_5}e^{-\widehat{\xi}_4\theta_4}\cdots e^{-\widehat{\xi}_1\theta_1}g_d g_{st}^{-1}(0)p =: q. \tag{3.37}$$

Apply Subproblem 1 to find θ_6.

At the end of this procedure, θ_1 through θ_6 are determined. There are a maximum of eight possible solutions, due to multiple solutions for equations (3.33), (3.34), and (3.36). Note that the overall procedure is to first solve for the three angles which determine the position of the center of the wrist and then solve for the wrist angles.

Example 3.6. Inverse kinematics of a SCARA manipulator
Consider the four degree of freedom SCARA manipulator shown in Figure 3.12. From the forward kinematics derived in Example 3.1 on page 87, the tool configuration has the form

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1}\cdots e^{\widehat{\xi}_4\theta_4}g_{st}(0) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & x \\ \sin\phi & \cos\phi & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} =: g_d \tag{3.38}$$

and hence we can solve the inverse kinematics given x, y, z, and ϕ as in equation (3.38). We begin by solving for θ_4. Applying both sides of equation (3.38) to the origin of the tool frame gives

$$p(\theta) = \begin{bmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ l_0 + \theta_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

where $p(\theta)$ is the position component of the forward kinematics, given by equation (3.4). From the form of $p(\theta)$, we see that $\theta_4 = z - l_0$. Notice that finding θ_4 did not make use of any of the previously defined subproblems.

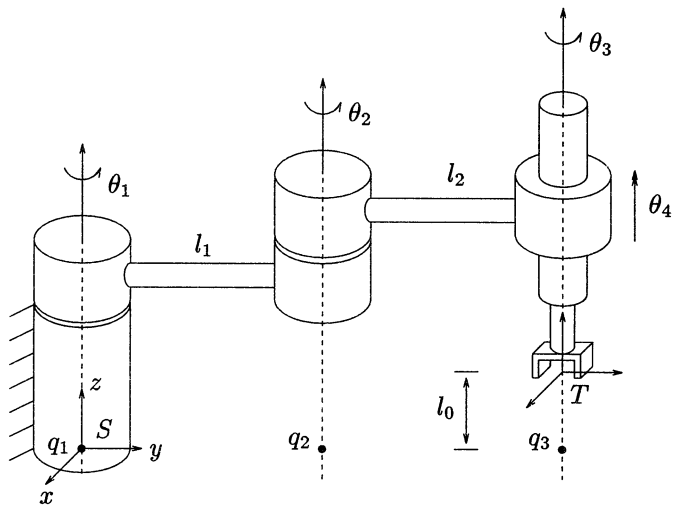


Figure 3.12: SCARA manipulator in its reference configuration.

Once θ_4 is known, we can rearrange equation (3.38) to read

$$e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}e^{\widehat{\xi}_3\theta_3} = g_d g_{st}^{-1}(0)e^{-\widehat{\xi}_4\theta_4} =: g_1. \tag{3.39}$$

Let p be a point on the axis of ξ_3 and q be a point on the axis of ξ_1. Applying equation (3.39) to p, subtracting q from both sides, and applying norms,

$$\begin{aligned} \|e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}p - q\| &= \|e^{\widehat{\xi}_1\theta_1}(e^{\widehat{\xi}_2\theta_2}p - q)\| \\ &= \|e^{\widehat{\xi}_2\theta_2}p - q\| = \|g_1 p - q\| =: \delta. \end{aligned} \tag{3.40}$$

Application of Subproblem 3 gives the value of θ_2. θ_1 can now be found by applying equation (3.39) to a point p' on the axis of ξ_3 and solving Subproblem 1:

$$e^{\widehat{\xi}_1\theta_1}e^{\widehat{\xi}_2\theta_2}e^{\widehat{\xi}_3\theta_3}p' = e^{\widehat{\xi}_1\theta_1}\left(e^{\widehat{\xi}_2\theta_2}p'\right) = g_1 p'.$$

Finally, we rearrange equation (3.39), shifting the (known) θ_1 and θ_2 terms to the right-hand side:

$$e^{\widehat{\xi}_3\theta_3} = e^{-\widehat{\xi}_2\theta_2}e^{-\widehat{\xi}_1\theta_1}g_d g_{st}^{-1}(\theta)e^{-\widehat{\xi}_4\theta_4}. \tag{3.41}$$

Applying equation (3.41) to any point p which is not on the axis of ξ_3 and solving Subproblem 1 a final time gives θ_3 and completes the solution.

There are a maximum of two possible solutions for the SCARA manipulator, corresponding to the solutions of equation (3.40).

3.4 General solutions to inverse kinematics problems

In the preceding sections, we have developed an elegant set of geometric and conceptual subproblems in terms of which the inverse kinematics solution of a large number of manipulators can be decomposed. The set of subproblems and their extensions given in the exercises turn out to be useful for decomposing the solution of the inverse kinematics equations, primarily when the robot has at least some intersecting axes.

The question of how to solve the inverse kinematics problem in general, (that is, in the absence of any intersections of the axes of the manipulator) for both planar and spatial mechanisms is an extremely active area of current research. In particular, there are many interesting questions about the number of inverse kinematics solutions and how the computations can be mechanized in real-time. In this subsection, we give a brief summary of some of the newest and most general approaches in this regard, drawn from [58], [65] and [96]. Our development closely parallels that of Manocha and Canny [65]. The approaches are primarily based on classic elimination theory from algebraic geometry; this is a systematic procedure for simultaneously eliminating $n-1$ of the variables in a system of n polynomials in n variables to obtain a single polynomial in one variable. This procedure is a general procedure but also a "brute-force" procedure, in that it only takes very simple properties of the manipulator kinematics into account (unlike the solutions based on the subproblems listed above). We illustrate this procedure, sometimes referred to as *dialytical elimination*, in the following example of three nonhomogeneous polynomials in three variables.

Example 3.7. Dialytical elimination for three polynomials in three variables

We will assume that the three polynomials f_1, f_2, f_3 are nonhomogeneous in x_1, x_2, x_3 and are of the form

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 & e_1 & g_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 & g_2 \\ a_3 & b_3 & c_3 & d_3 & e_3 & g_3 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_1 x_3 \\ x_2^2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Here, $a_i, b_i, c_i, d_i, e_i, g_i$, for $i = 1, \ldots, 3$ are all real numbers. To solve this system of three polynomials in x_1, x_2, x_3 for their common zeros, we eliminate x_2, x_3 to get a single polynomial in x_1 in two steps:

Step 1. We express the above equation as a system of equations in x_2, x_3

with coefficients being polynomials in x_1 as follows:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} d_1 & e_1 + c_1 x_1 & b_1 x_1 & a_1 x_1^2 + g_1 \\ d_2 & e_2 + c_2 x_1 & b_2 x_1 & a_2 x_1^2 + g_2 \\ d_3 & e_3 + c_3 x_1 & b_3 x_1 & a_3 x_1^2 + g_3 \end{bmatrix} \begin{bmatrix} x_2^2 \\ x_3 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Step 2. We try to generate more polynomials which are independent[2] of f_1, f_2, f_3 until we have as many unknown monomials[3] in x_2, x_3 as there are equations. For example, multiplying f_1, f_2, f_3 by x_2 yields the following set of polynomials which are independent of f_1, f_2, f_3:

$$\begin{bmatrix} x_2 f_1 \\ x_2 f_2 \\ x_2 f_3 \end{bmatrix} = \begin{bmatrix} d_1 & e_1 + c_1 x_1 & b_1 x_1 & a_1 x_1^2 + g_1 \\ d_2 & e_2 + c_2 x_1 & b_2 x_1 & a_2 x_1^2 + g_2 \\ d_3 & e_3 + c_3 x_1 & b_3 x_1 & a_3 x_1^2 + g_3 \end{bmatrix} \begin{bmatrix} x_2^3 \\ x_2 x_3 \\ x_2^2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Combining these equations with those for f_1, f_2, f_3 yields the following set of six equations in the six monomials $x_2^3, x_2^2, x_2 x_3, x_2, x_3, 1$:

$$A(x_1) := \begin{bmatrix} d_1 & e_1+c_1x_1 & b_1x_1 & a_1x_1^2+g_1 & 0 & 0 \\ d_2 & e_2+c_2x_1 & b_2x_1 & a_2x_1^2+g_2 & 0 & 0 \\ d_3 & e_3+c_3x_1 & b_3x_1 & a_3x_1^2+g_3 & 0 & 0 \\ 0 & 0 & d_1 & b_1x_1 & e_1+c_1x_1 & a_1x_1^2+g_1 \\ 0 & 0 & d_2 & b_2x_1 & e_2+c_2x_1 & a_2x_1^2+g_2 \\ 0 & 0 & d_3 & b_3x_1 & e_3+c_3x_1 & a_3x_1^2+g_3 \end{bmatrix} \begin{bmatrix} x_2^3 \\ x_2x_3 \\ x_2^2 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For this equation to have a solution, it is clear that the determinant of the matrix $A(x_1)$ multiplying the entries $x_2^3, x_2x_3, x_2^2, x_2, x_3, 1$ should have zero determinant. Thus, the system of equations $f_1 = f_2 = f_3 = 0$ is equivalent to $\det A(x_1) = 0$. In general, the degree of this polynomial in x_1 is high (generically, i.e., for almost all values of $a_i, \ldots, g_i$, its degree is six) and, furthermore, solutions need not be real. However, the degree of the polynomial is an upper bound on the number of real solutions and once a real number x_1 has been found, one can read off the solution for x_2, x_3 by scaling the element in the null space of $A(x_1)$ to have 1 as its last entry. If the null space of the matrix $A(x_1)$ has dimension greater than one, corresponding to a multiplicity of roots for the polynomial $\det A(x_1) = 0$, then there may be a multiplicity of solutions for x_2, x_3 corresponding to the value(s) of x_1 for which the null space of the matrix $A(x_1)$ has dimension greater than 1. While the exact multiplicity is a subtle question in [65], it is stated that an upper bound for the multiplicity of the solutions x_2, x_3 associated with that value of x_1 is the dimension of the null space of $A(x_1)$.

[2]Independence is meant in a technical sense, over the ring of polynomials in x_1.

[3]A monomial is a single polynomial; for example, x_2x_3 or x_1^3.

A general procedure for six degree of freedom manipulators

To apply this procedure to the inverse kinematics of a six-link manipulator, some preliminary work is necessary. The kinematics are given by

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_6\theta_6} g_{st}(0)$$

and the inverse kinematics problem is to solve for $\theta_1, \ldots, \theta_6$ given $g_d \in SE(3)$. We rewrite the kinematics in terms of the Denavit-Hartenberg parameterization, as in equation (3.6):

$$g_{st}(\theta) = g_{l_0 l_1}(\theta_1) g_{l_1 l_2}(\theta_2) \cdots g_{l_5 l_6}(\theta_n) g_{l_6 t} = g_d.$$

By proper choice of link frames, each $g_{l_{i-1}l_i}$ has the form

$$g_{l_{i-1},l_i} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \cos\alpha_i & \sin\phi_i \sin\alpha_i & a_i \cos\phi_i \\ \sin\phi_i & \cos\phi_i \cos\alpha_i & -\cos\phi_i \sin\alpha_i & a_i \sin\phi_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.42}$$

Given a desired g_d for the end-effector position and orientation, we rewrite this equation as

$$g_{l_2 l_3}(\theta_3) g_{l_3 l_4}(\theta_4) g_{l_4 l_5}(\theta_5) = g_{l_1 l_2}^{-1}(\theta_2) g_{l_0 l_1}^{-1}(\theta_1)\, g_d\, g_{l_6 t}^{-1} g_{l_5 l_6}^{-1}(\theta_6). \tag{3.43}$$

What we have done is to take the inverses of $g_{l_5 l_6}(\theta_6)$, $g_{l_0 l_1}(\theta_1)$, and $g_{l_1 l_2}(\theta_2)$ and moved them to the right-hand side. The reason for this clever rearrangement and the choice of the Denavit-Hartenberg parameterization in the kinematics will become clear in Step 1 of the procedure which follows. By way of notation, we write $g_d g_{l_6 t}^{-1}$ as

$$g_d g_{l_6 t}^{-1} = \left[\begin{array}{ccc|c} l_x & m_x & n_x & q_x \\ l_y & m_y & n_y & q_y \\ l_z & m_z & n_z & q_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{3.44}$$

Hence, $l, m, n \in \mathbb{R}^3$ are the columns of the rotational component of $g_d g_{l_6 t}^{-1}$ and and $q \in \mathbb{R}^3$ is the translational component.

To bring to bear the machinery of algebraic geometry in an equation involving sines and cosines of angles, we define $c_i := \cos\theta_i$ and $s_i := \sin\theta_i$ for $i = 1, \ldots, 6$ and think of equation (3.43) as being polynomial in s_i and c_i. Indeed, by direct inspection we may see that the entries of each exponential are *unary* (i.e., of degree 1 or less) in c_i and s_i. Part of the reason for rewriting the kinematics as in equation (3.43) is to be able to reduce the order of the polynomial in s_i and c_i. Both the left-hand side and right-hand side of (3.43) are now cubic in s_i and c_i. The details of

the procedure to eliminate all the variables except for θ_3 is sketched as a number of steps. The details of the proofs and justification of the steps is quite involved and, for them, the reader is referred to the papers cited above.

Step 1. Verify that the third and fourth columns of equation (3.43) are independent of θ_6. This is the step where the Denavit-Hartenberg parameterization comes in handy. Indeed, from taking the inverse of the formula for $g_{l_6 t}(\theta_6)$ from equation (3.42), it follows that

$$g_{l_6 t}^{-1}(\theta_6) = \begin{bmatrix} c_{\theta_6} & s_{\theta_6} & 0 & a_6 \\ -c_{\alpha_6} s_{\theta_6} & c_{\alpha_6} c_{\theta_6} & s_{\alpha_6} & -d_6 s_{\alpha_6} \\ s_{\alpha_6} s_{\theta_6} & -s_{\alpha_6} c_{\theta_6} & c_{\alpha_6} & -d_6 c_{\alpha_6} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thus, the last two columns on the right-hand side of equation (3.43) are independent of θ_6.

Step 2. Equate the third and fourth column of the left-hand side and right hand side of equation (3.43). This yields the following six equations in $\theta_i, i = 1, \ldots, 5$:

$$\begin{aligned}
\text{EQ1}: &\quad c_3 f_1 + s_3 f_2 = c_2 h_1 + s_2 h_2 - a_2 \\
\text{EQ2}: &\quad s_3 f_1 - c_3 f_2 = -\lambda_2(s_2 h_1 - c_2 h_2) + \mu_2(h_3 - d_2) \\
\text{EQ3}: &\quad f_3 = \mu_2(s_2 h_1 - c_2 h_2) + \lambda_2(h_3 - d_2) \\
\text{EQ4}: &\quad c_3 r_1 + s_3 r_2 = c_2 n_1 + s_2 n_2 \\
\text{EQ5}: &\quad s_3 r_1 - c_3 r_3 = -\lambda_2(s_2 n_1 - c_2 n_2) + \mu_2 n_3 \\
\text{EQ6}: &\quad r_3 = \mu_3(s_2 n_1 - c_2 n_2) + \lambda_2 n_3,
\end{aligned} \tag{3.45}$$

where

$$\begin{array}{lll}
f_1 = c_4 g_1 + s_4 g_2 + a_3 & f_2 = \mu_3 g_3 - \lambda_3(s_4 g_1 - c_4 g_2) & f_3 = \mu_3(s_4 g_1 - c_4 g_2) + \lambda_3 g_3 + d_3 \\
r_1 = c_4 m_1 + s_4 m_2 & r_2 = \mu_3 m_3 - \lambda_3(s_4 m_1 - c_4 m_2) & r_3 = \mu_3(s_4 m_1 - c_4 m_2) + \lambda_3 m_3 \\
g_1 = c_5 a_5 + a_4 & g_2 = \mu_4 d_5 - \lambda_4 s_5 a_5 & g_3 = -s_5 \mu_4 a_5 + \lambda_4 d_5 + d_4 \\
m_1 = s_5 \mu_5 & m_2 = c_5 \lambda_4 \mu_5 + \mu_4 \lambda_5 & m_3 = -c_5 \mu_4 \mu_5 + \lambda_4 \lambda_5 \\
h_1 = c_1 p + s_1 q - a_1 & h_2 = \mu_1(r - d_1) - \lambda_1(s_1 p - c_1 q) & h_3 = \mu_1(s_1 p - c_1 q) + \lambda_1(r - d_1) \\
n_1 = c_1 u + s_1 v & n_2 = \mu_1 w - \lambda_1(s_1 u - c_1 v) & n_3 = \mu_1(s_1 u - c_1 v) + \lambda_1 w.
\end{array}$$

The scalars a_i and d_i are the Denavit-Hartenberg parameters of the links, and $\mu_i = \sin\alpha_i$, $\lambda_i = \cos\alpha_i$. The desired configuration enters through the following parameters:

$$\begin{array}{ll}
p = -l_x a_6 - (m_x \mu_6 + n_x \lambda_6) d_6 + q_x & u = m_x \mu_6 + n_x \lambda_6 \\
q = -l_y a_6 - (m_y \mu_6 + n_y \lambda_6) d_6 + q_y & v = m_y \mu_6 + n_y \lambda_6 \\
r = -l_z a_6 - (m_z \mu_6 + n_z \lambda_6) d_6 + q_z & w = m_z \mu_6 + n_z \lambda_6.
\end{array}$$

Recall that the $l = (l_x, l_y, l_z)$, $m = (m_x, m_y, m_z)$, $n = (n_x, n_y, n_z)$, and $q = (q_x, q_y, q_z)$ were defined in equation (3.44).

Step 3. Now, rearrange equations EQ1–EQ6, with the obvious definitions for $h, f, n, r \in \mathbb{R}^3$ from the above definitions for their components, to get two sets of three equations:

$$\begin{aligned} p &= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} h = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\lambda_2 & \mu_2 \\ 0 & \mu_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} c_3 & s_3 & 0 \\ s_3 & -c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} f + \begin{bmatrix} a_2 \\ 0 \\ d_2 \end{bmatrix} \\ l &= \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\lambda_2 & \mu_2 \\ 0 & \mu_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} c_3 & s_3 & 0 \\ s_3 & -c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} r. \end{aligned} \tag{3.46}$$

The left-hand sides of p and l are linear combinations of 1, c_2, s_2, c_1, s_1, c_1c_2, s_1c_2, and s_1s_2. The right-hand sides are linear combinations in functions of s_3, c_3 of 1, c_5, s_5, c_4, s_4, c_4c_5, s_4c_5, and s_4s_5.

Step 4. From a lengthy calculation, using the kinematics, it follows that

$$l \quad p \quad p \cdot p \quad p \cdot l \quad p \times l \quad (p \cdot p)l - 2(p \cdot l)p$$

have the same functional form as p and l in terms of their dependence on linear combinations of the same variables as the left- and right-hand sides of p and l. Between them, these represent $3+3+1+1+3+3 = 14$ equations. Combine these 14 equations to get an equation of the form

$$P(s_3, c_3) \begin{bmatrix} s_4s_5 \\ s_4c_5 \\ c_4s_5 \\ c_4c_5 \\ s_4 \\ c_4 \\ s_5 \\ c_5 \\ 1 \end{bmatrix} = Q \begin{bmatrix} s_1s_2 \\ s_1c_2 \\ c_1s_2 \\ c_1c_2 \\ s_1 \\ c_1 \\ s_2 \\ c_2 \end{bmatrix}. \tag{3.47}$$

In this equation, $P(s_3, c_3) \in \mathbb{R}^{14\times 9}$ is a function of s_3, c_3 and $Q \in \mathbb{R}^{14\times 8}$ is a constant matrix.

Step 5. If the rank of the matrix $Q \in \mathbb{R}^{14\times 8}$ is 8, use any 8 of the 14 equations in equation (3.47) to solve for the eight variables

$$s_1s_2 \quad s_1c_2 \quad c_1s_2 \quad c_1c_2 \quad s_1 \quad c_1 \quad s_2 \quad c_2$$

in terms of the variables

$$s_4s_5 \quad s_4c_5 \quad c_4s_5 \quad c_4c_5 \quad s_4 \quad c_4 \quad s_5 \quad c_5.$$

Use this in the remaining six equations of (3.47) to get six equations of the form

$$\Sigma(c_3, s_3) \begin{bmatrix} s_4s_5 \\ s_4c_5 \\ c_4s_5 \\ c_4c_5 \\ s_4 \\ c_4 \\ s_5 \\ c_5 \\ 1 \end{bmatrix} = 0, \tag{3.48}$$

where $\Sigma(c_3, s_3) \in \mathbb{R}^{6\times 9}$.

If the rank of Q is an integer k less than 8, then one can only solve for k of the variables $s_1s_2, s_1c_2, c_1s_2, s_1, c_1, s_2, c_2$ and we are left with six equations in $9 + 8 - k$ variables in place of (3.48). We will not describe the algorithm in detail for this scenario, for details we refer the reader to [65].

Step 6. We will assume that Q has rank 8, so that the equation of (3.48) holds. To make the equations (3.48) polynomial rather than trigonometric, use the substitutions

$$s_i = \frac{2x_i}{1+x_i^2} \qquad c_i = \frac{1-x_i^2}{1+x_i^2}$$

for $i = 3, 4, 5$, where

$$x_i = \tan(\frac{\theta_i}{2}).$$

Use this in equation (3.48) and multiply each equation by $(1 + x_4^2)$ and $(1+x_5^2)$ to clear the denominators. Now, multiply the first four equations by $(1 + x_3^2)$ to get

$$\Sigma_1(x_3) \begin{bmatrix} x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_5^2 \\ x_5 \\ 1 \end{bmatrix} = 0. \tag{3.49}$$

The first four equations of equation (3.49) are quadratic in x_3 and the last two are rational functions of x_3, with denominators $1+x_3^2$. However, the determinant of any 6×6 submatrix of Σ is a polynomial in x_3. (This last fact needs proof.)

Step 7. We now use dialytic elimination to eliminate x_4, x_5 as in the previous example by multiplying the six equations in equation (3.49) by x_4 and then appending them to the original set to get 12 equations of the form

$$\bar{\Sigma} := \begin{bmatrix} \Sigma_1(x_3) & 0 \\ 0 & \Sigma_1(x_3) \end{bmatrix} \begin{bmatrix} x_4^3x_5^2 \\ x_4^3x_5 \\ x_4^3 \\ x_4^2x_5^2 \\ x_4^2x_5 \\ x_4^2 \\ x_4x_5^2 \\ x_4x_5 \\ x_4 \\ x_5^2 \\ x_5 \\ 1 \end{bmatrix} = 0. \tag{3.50}$$

The matrix $\Sigma_1(x_3)$ is in $\mathbb{R}^{6\times 9}$ and the zero matrices are of dimension 6 $\times$ 3 so that the overall matrix on the left-hand side of equation (3.50) is in $\mathbb{R}^{12\times 12}$.

Step 8. The polynomial equation $\det \bar{\Sigma} = 0$ is a polynomial of order 16 in x_3. Once we solve for x_3, we solve for x_4, x_5 from the determination of the null space of $\bar{\Sigma}$ as in the Example 3.4. Then, we solve for θ_1, θ_2 from equation (3.47). This yields the solutions to all the angles except θ_6. For this purpose, we return to the form of the kinematics

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_6\theta_6} g_{st}(0) = g_d,$$

with all the variables known except θ_6. By choosing any point p, we may write this equation as

$$e^{\widehat{\xi}_6\theta_6} p = e^{-\widehat{\xi}_5\theta_5} \cdots e^{-\widehat{\xi}_1\theta_1} g_d g_{st}^{-1} p =: q$$

Now we use Subproblem 1 from the preceding subsection to solve for θ_6.

Number of inverse kinematics solutions

The set of steps outlined above results in a 16th order polynomial for x_3 and hence θ_3. This polynomial may or may not have 16 *real roots.* In the instance of the elbow manipulator, we saw that the maximum number of feasible solutions was eight. One may verify that if the three wrist axes of the elbow manipulator do not intersect at a point, the number of inverse kinematics solutions for this manipulator may be as high as 16. Indeed, 16 is the upper bound of number of solutions of any open-link spatial mechanism with six degrees of freedom. This number is a sharp bound—that is to say, it is achievable (see, for example, [66])—and is far superior to those obtained from a brute force elimination technique using Bezout's theorem. Indeed, this bound of 16 is a tribute to the ingenuity of Lee and Liang [58] who were the first to notice the set of tricks presented above and put to rest a number of erroneous conjectures that had been populating the literature up to that point. Not the least of their tricks is the rewriting of the kinematic equations as equation (3.43).

The algorithm as stated above has some drawbacks, as pointed out by Manocha and Canny [65]. The majority of the drawbacks are numerical, owing to the computation of determinants of ill-conditioned matrices, though there are some conceptual ones as well. For an example of a conceptual drawback, we have the modification of the procedure in Step 5 above for choosing fewer eight independent equations out of 14 to eliminate as many of the cosines and sines of θ_1, θ_2 when the rank of the matrix Q is less than 8 to produce an over determined set of equations. The aforementioned paper shows that a general computational technique of converting the dialytical elimination technique into a generalized eigenvalue problem can be brought to bear on the inverse kinematics problem

so as to improve its numerical conditioning and speed of computation. These computational details are of tremendous importance when real-time computation of the inverse kinematics solutions needs to be done in the context of obstacle avoidance for robots in cluttered environments.

4 The Manipulator Jacobian

In addition to the relationship between the joint angles and the end-effector configuration, one often makes use of the relationship between the joint and end-effector velocities. In this section, we derive a formula for this relationship and study its structure and properties. We also study the dual relationship between end-effector wrenches and joint torques.

Traditionally, one describes the Jacobian for a manipulator by differentiating the forward kinematics map. This works when the forward kinematics is represented as a mapping $g : \mathbb{R}^n \to \mathbb{R}^p$, in which case the Jacobian is the linear map $\frac{\partial g}{\partial \theta}(\theta) : \mathbb{R}^n \to \mathbb{R}^p$. However, if we represent the forward kinematics more completely as $g : \mathbb{R}^n \to SE(3)$, the Jacobian is not so easily obtained. The problem is that $\frac{\partial g}{\partial \theta}(\theta)$ is not a natural quantity since g is a matrix-valued function. Of course, one could always choose coordinates for $SE(3)$, but this description only holds locally. More importantly, choosing a local parameterization for $SE(3)$ destroys the natural geometric structure of rigid body motion.

To correct this problem, we use the tools of Chapter 2 to write the Jacobian of the forward kinematics map in terms of twists. We shall see that the product of exponentials formula leads to a very natural and explicit description of the manipulator Jacobian, which highlights the geometry of the mechanism and has none of the drawbacks of a local representation.

4.1 End-effector velocity

Let $g_{st} : Q \to SE(3)$ be the forward kinematics map for a manipulator. If the joints move along a path $\theta(t) \in Q$, then the end-effector traverses a path $g_{st}(\theta(t)) \in SE(3)$. The instantaneous spatial velocity of the end-effector is given by the twist

$$\widehat{V}_{st}^s = \dot{g}_{st}(\theta) g_{st}^{-1}(\theta).$$

Applying the chain rule,

$$\widehat{V}_{st}^s = \sum_{i=1}^{n} \left(\frac{\partial g_{st}}{\partial \theta_i} \dot{\theta}_i \right) g_{st}^{-1}(\theta) = \sum_{i=1}^{n} \left(\frac{\partial g_{st}}{\partial \theta_i} g_{st}^{-1}(\theta) \right) \dot{\theta}_i, \qquad (3.51)$$

and we see that the end-effector velocity is linearly related to the velocity of the individual joints. In twist coordinates, equation (3.51) can be

written as

$$V_{st}^s = J_{st}^s(\theta)\dot{\theta},$$

where

$$J_{st}^s(\theta) = \left[\left(\frac{\partial g_{st}}{\partial \theta_1} g_{st}^{-1} \right)^\vee \quad \cdots \quad \left(\frac{\partial g_{st}}{\partial \theta_n} g_{st}^{-1} \right)^\vee \right]. \tag{3.52}$$

We call the matrix $J_{st}^s(\theta) \in \mathbb{R}^{6\times n}$ the *spatial manipulator Jacobian.* At each configuration θ, it maps the joint velocity vector into the corresponding velocity of the end-effector.

If we represent the forward kinematics using the product of exponentials formula, we can obtain a more explicit and elegant formula for J_{st}^s. Let

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)$$

represent the mapping $g_{st} : Q \to SE(3)$, where $\widehat{\xi}_i \in se(3)$ are unit twists. Then,

$$\begin{aligned}
\left(\frac{\partial g_{st}}{\partial \theta_i} \right) g_{st}^{-1} &= e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}} \frac{\partial}{\partial \theta_i} \left(e^{\widehat{\xi}_i\theta_i} \right) e^{\widehat{\xi}_{i+1}\theta_{i+1}} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)\, g_{st}^{-1} \\
&= e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}} \left(\widehat{\xi}_i\right) e^{\widehat{\xi}_i\theta_i} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)\, g_{st}^{-1} \\
&= e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}} \left(\widehat{\xi}_i\right) e^{-\widehat{\xi}_{i-1}\theta_{i-1}} \cdots e^{-\widehat{\xi}_1\theta_1}
\end{aligned}$$

and, converting to twist coordinates,

$$\left(\frac{\partial g_{st}}{\partial \theta_i} g_{st}^{-1} \right)^\vee = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)} \xi_i. \tag{3.53}$$

The spatial manipulator Jacobian becomes

$$\boxed{\begin{aligned} J_{st}^s(\theta) &= \begin{bmatrix} \xi_1 & \xi_2' & \cdots & \xi_n' \end{bmatrix} \\ \xi_i' &= \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)} \xi_i \end{aligned}} \tag{3.54}$$

$J_{st}^s(\theta) : \mathbb{R}^n \to \mathbb{R}^6$ is a configuration-dependent matrix which maps joint velocities to end-effector velocities.

Equation (3.54) shows that the manipulator Jacobian has a very special structure. By virtue of the definition of ξ', the i^{th} column of the Jacobian depends only on $\theta_1, \ldots, \theta_{i-1}$. In other words, the contribution of the i^{th} joint velocity to the end-effector velocity is independent of the configuration of later joints in the chain. Furthermore, the i^{th} column of the Jacobian,

$$\xi_i' = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)} \xi_i,$$

corresponds to the i^{th} joint twist, ξ_i, transformed by the rigid transformation $\exp(\widehat{\xi}_1\theta_1)\cdots\exp(\widehat{\xi}_{i-1}\theta_{i-1})$. This is precisely the rigid body transformation which takes the i^{th} joint frame from its reference configuration to the current configuration of the manipulator. Thus, *the i^{th} column of the spatial Jacobian is the i^{th} joint twist, transformed to the current manipulator configuration.* This powerful structural property means that we can calculate $J^s_{st}(\theta)$ "by inspection," as we shall see shortly in an example.

It is also possible to define a *body manipulator Jacobian*, J^b_{st}, which is defined by the relationship

$$V^b_{st} = J^b_{st}(\theta)\dot{\theta}.$$

A calculation similar to that performed previously yields

$$\boxed{\begin{aligned} J^b_{st}(\theta) &= [\xi^\dagger_1 \quad \cdots \quad \xi^\dagger_{n-1} \quad \xi^\dagger_n] \\ \xi^\dagger_i &= \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_i\theta_i}\cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)\right)} \xi_i \end{aligned}} \tag{3.55}$$

The columns of J^b_{st} correspond to the joint twists written with respect to the tool frame at the current configuration. Note that $g_{st}(0)$ appears explicitly; choosing S such that $g_{st}(0) = I$ simplifies the calculation of J^b_{st}. The spatial and body Jacobians are related by an adjoint transformation:

$$J^s_{st}(\theta) = \mathrm{Ad}_{g_{st}(\theta)} J^b_{st}(\theta). \tag{3.56}$$

The spatial and body manipulator Jacobians can be used to compute the instantaneous velocity of a point attached to the end-effector. Let q^b represent a point attached to the end-effector, written in body (tool) coordinates. The velocity of q^b, also in body coordinates, is given by

$$v^b_q = \widehat{V}^b_{st} q^b = \left(J^b_{st}(\theta)\dot{\theta}\right)^\wedge q^b.$$

Similarly, if we represent our point relative to the spatial (base) frame, then

$$v^s_q = \widehat{V}^s_{st} q^s = \left(J^s_{st}(\theta)\dot{\theta}\right)^\wedge q^s.$$

If we desire the velocity of the origin of the tool frame, then $q^b = 0$ but $q^s = g_{st}(\theta)q^b = p(\theta)$, the position component of the forward kinematics map. Thus, using homogeneous coordinates explicitly,

$$v^s_q = \begin{bmatrix} \dot{p}(\theta) \\ 0 \end{bmatrix} = R_{st}\widehat{V}^b_{st}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \widehat{V}^s_{st}\begin{bmatrix} p(\theta) \\ 1 \end{bmatrix}$$

are all equivalent expressions for the velocity of the origin of the tool frame.

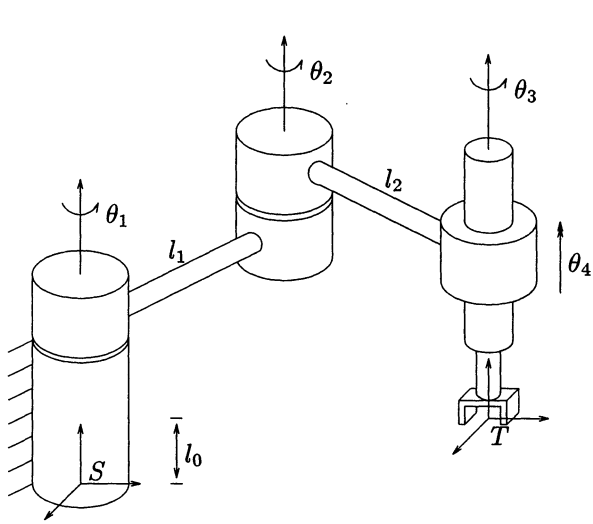


Figure 3.13: SCARA manipulator in non-reference configuration.

The relationship between joint velocity and end-effector velocity can be used to move a robot manipulator from one end-effector configuration to another without calculating the inverse kinematics for the manipulator. If J_{st} is invertible, then we can write

$$\dot{\theta}(t) = (J^s_{st}(\theta))^{-1} V^s_{st}(t). \tag{3.57}$$

If $V^s_{st}(t)$ is known, equation (3.57) is an ordinary differential equation for θ. To move the end-effector between two configurations g_1 and g_2, we pick *any* workspace path $g(t)$ with $g(0) = g_1$ and $g(T) = g_2$, calculate the spatial velocity $\widehat{V}^s = \dot{g}g^{-1}$, and integrate equation (3.57) over the interval $[0, T]$.

Example 3.8. Jacobian for a SCARA robot
Consider the SCARA robot at an arbitrary configuration $\theta \in Q$, as shown in Figure 3.13. The spatial Jacobian can be evaluated by writing the twists associated with each joint in its current configuration. For the SCARA, the directions of the twists are fixed and only the points through which the axes of the twists pass are functions of θ. By inspection,

$$q'_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad q'_2 = \begin{bmatrix} -l_1 \sin\theta_1 \\ l_1 \cos\theta_1 \\ 0 \end{bmatrix} \qquad q'_3 = \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ 0 \end{bmatrix}$$

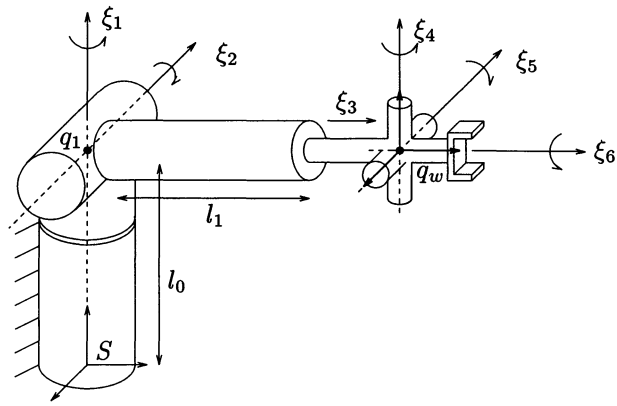


Figure 3.14: An idealized version of the Stanford arm.

are points on the axes. Calculating the associated twists yields

$$J^s_{st} = \begin{bmatrix} 0 & l_1 \cos\theta_1 & l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & 0 \\ 0 & l_1 \sin\theta_1 & l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

As a check, one can calculate the linear velocity of the end-effector using the formula $(J^s_{st}\dot{\theta})\hat{}\, p(\theta)$ and verify that it agrees with $\dot{p}(\theta)$.

Example 3.9. Jacobian for the Stanford arm
The Stanford arm, shown in Figure 3.14, is a six degree of freedom robot with two revolute joints at the base, a prismatic joint, and a spherical wrist. It is very similar to an elbow manipulator, with the "elbow" replaced by a prismatic joint.

The spatial manipulator Jacobian for the Stanford arm is computed by determining the location and directions of the joint twists as a function of the joint angles. For example, the first two joints pass through the point $q_1 = q_2 = (0, 0, l_0)$ and point in the directions $\omega_1 = (0, 0, 1)$ and $\omega_2' = (-\cos\theta_1, -\sin\theta_1, 0)$. This gives joint twists

$$\xi_1 = \begin{bmatrix} -\omega_1 \times q_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_2' = \begin{bmatrix} -\omega_2' \times q_1 \\ \omega_2' \end{bmatrix} = \begin{bmatrix} l_0 \sin\theta_1 \\ -l_0 \cos\theta_1 \\ 0 \\ -\cos\theta_1 \\ -\sin\theta_1 \\ 0 \end{bmatrix}.$$

The third joint is prismatic and hence we care only about its direction. Taking into account the change in orientation due to the first two joints,

we have

$$\xi_3' = \begin{bmatrix} e^{\widehat{\mathbf{z}}\theta_1} e^{-\widehat{\mathbf{x}}\theta_2} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin\theta_1 \cos\theta_2 \\ \cos\theta_1 \cos\theta_2 \\ -\sin\theta_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v_3' \\ 0 \end{bmatrix}.$$

Finally, we compute the twists corresponding to the wrist. The wrist is located at the point

$$q_w' = \begin{bmatrix} 0 \\ 0 \\ l_0 \end{bmatrix} + e^{\widehat{\mathbf{z}}\theta_1} e^{-\widehat{\mathbf{x}}\theta_2} \begin{bmatrix} 0 \\ l_1 + \theta_3 \\ 0 \end{bmatrix} = \begin{bmatrix} -(l_1 + \theta_3) \sin\theta_1 \cos\theta_2 \\ (l_1 + \theta_3) \cos\theta_1 \cos\theta_2 \\ l_0 - (l_1 + \theta_3) \sin\theta_2 \end{bmatrix}.$$

The directions of the wrist axes depend on θ_1 and θ_2 as well as the preceding wrist axes. These are given by

$$\omega_4' = e^{\widehat{\mathbf{z}}\theta_1} e^{-\widehat{\mathbf{x}}\theta_2} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s_1 s_2 \\ c_1 s_2 \\ c_2 \end{bmatrix}$$

$$\omega_5' = e^{\widehat{\mathbf{z}}\theta_1} e^{-\widehat{\mathbf{x}}\theta_2} e^{\widehat{\mathbf{z}}\theta_4} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -c_1 c_4 + s_1 c_2 s_4 \\ -s_1 c_4 - c_1 c_2 s_4 \\ s_2 s_4 \end{bmatrix}$$

$$\omega_6' = e^{\widehat{\mathbf{z}}\theta_1} e^{-\widehat{\mathbf{x}}\theta_2} e^{\widehat{\mathbf{z}}\theta_4} e^{-\widehat{\mathbf{x}}\theta_5} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -c_5(s_1 c_2 c_4 + c_1 s_4) + s_1 s_2 s_5 \\ c_5(c_1 c_2 c_4 - s_1 s_4) - c_1 s_2 s_5 \\ -s_2 c_4 c_5 - c_2 s_5 \end{bmatrix}.$$

Combining these directions with our calculations for q_w', we can now write the complete manipulator Jacobian:

$$J_{st}^s = \begin{bmatrix} 0 & -\omega_2' \times q_1 & v_3' & -\omega_4' \times q_w' & -\omega_5' \times q_w' & -\omega_6' \times q_w' \\ \omega_1 & \omega_2' & 0 & \omega_4' & \omega_5' & \omega_6' \end{bmatrix}, \tag{3.58}$$

where the various quantities are defined above.

Note that we were able to calculate the entire manipulator Jacobian without explicitly differentiating the forward kinematics map.

As a final comment, we re-emphasize that the manipulator Jacobian differs from the Jacobian of a mapping $f : \mathbb{R}^n \to \mathbb{R}^p$. For instance, in Example 3.8 (the SCARA manipulator), it is possible to define the configuration of the end-effector as $x = (p(\theta), \phi(\theta))$, where $p(\theta)$ is the xyz position of the tool frame and $\phi(\theta) = \theta_1 + \theta_2 + \theta_3$ is the angle of rotation of the tool frame about the z-axis. The kinematics is then represented by the mapping $x = f(\theta)$ and by the chain rule

$$\dot{x} = \frac{\partial f}{\partial \theta} \dot{\theta}.$$

The matrix $\frac{\partial f}{\partial \theta}$ is the Jacobian of the mapping $f : Q \to \mathbb{R}^4$, but it is *not* the manipulator Jacobian (body or spatial). In particular, the columns of $\frac{\partial f}{\partial \theta}$ cannot be interpreted as the instantaneous twist axes corresponding to each joint.

Similarly, for a general manipulator, one can choose a local parameterization for $SE(3)$ and write the kinematics as $f : Q \to \mathbb{R}^6$. Once again, the Jacobian of the mapping $f : Q \to \mathbb{R}^6$ has no direct geometric interpretation, even though it has the same dimensions as the manipulator Jacobian. Furthermore, for manipulators which generate full rotation of the end-effector, the parameterization of $SE(3)$ by a vector of six numbers introduces singularities which are solely an artifact of the parameterization. These singularities may lead to false conclusions about the ability of the manipulator to reach certain configurations or achieve certain velocities. The use of the manipulator Jacobian, as we have defined it here, avoids these difficulties.

4.2 End-effector forces

The manipulator Jacobian can also be used to describe the relationship between wrenches applied at the end-effector and joint torques. This relationship is fundamental in understanding how to program robots to interact with their environment by application of forces. We shall see that the duality of wrenches and twists discussed in Chapter 2 extends to manipulator kinematics.

To derive the relationship between wrenches and torques, we calculate the work associated with applying a wrench through a displacement of the end-effector. If we let $g_{st}(\theta(t))$ represent the motion of the end-effector, the net work performed by applying a (body) wrench F_t over an interval of time $[t_1, t_2]$ is

$$W = \int_{t_1}^{t_2} V_{st}^b \cdot F_t \, dt,$$

where V_{st}^b is the body velocity of the end-effector. The work will be the same as that performed by the joints (assuming no friction), and hence

$$\int_{t_1}^{t_2} \dot{\theta} \cdot \tau \, dt = W = \int_{t_1}^{t_2} V_{st}^b \cdot F_t \, dt.$$

Since this relationship must hold for any choice of time interval, the integrands must be equal. Using the manipulator Jacobian to relate V_{st}^b to $\dot{\theta}$, we have

$$\dot{\theta}^T \tau = \dot{\theta}^T (J_{st}^b)^T F_t.$$

It follows that since $\dot{\theta}$ is free,

$$\tau = (J_{st}^b)^T F_t. \tag{3.59}$$

This equation relates the end-effector wrench to the joint torques by giving the torques that are equivalent to a (body) wrench applied at the end-effector.

A similar analysis can be used to derive the relationship between a spatial wrench F_s applied at the end-effector and the corresponding joint torques:

$$\tau = (J_{st}^s)^T F_s. \tag{3.60}$$

The full derivation of this equation is left as an exercise.

The interpretation of the Jacobian transpose as a mapping from end-effector forces to joint torques must be made carefully. If the Jacobian is square and full rank, there are no difficulties. However, in all other cases, the relationship can be misleading. We defer the discussion of singularities to the next section, and consider only the case when J_{st} is not square.

The formulas given by equations (3.59) and (3.60) describe the force relationship that must hold between the end-effector forces and joint torques. We can use these equations to ask two separate questions:

1. If we apply an end-effector force, what joint torques are required to resist that force?

2. If we apply a set of joint torques, what is the resulting end-effector wrench (assuming that the wrench is resisted by some external agent)?

Equation (3.59) answers the first question in all cases. However, in manipulation tasks, we are often more interested in answering the second question, which can be recast as: what joint torques must be applied to generate a given end-effector wrench?

If the number of joints is larger than the dimension of the workspace, then we say the manipulator is *kinematically redundant.* In this case, we can generically find a vector of joint torques which generates the appropriate end-effector force, as given by equation (3.59). However, since there are more joints than the minimum number required, internal motions may exist which allow the manipulator to move while keeping the position of the end-effector fixed. Redundant manipulators are discussed in more detail in Section 5.

If, on the other hand, the number of joints is smaller than the dimension of the workspace, then there may be no torque which satisfies equation (3.59) for arbitrary end-effector wrenches, and therefore some end-effector wrenches cannot be applied. They can, however, be *resisted* by the manipulator. This is a consequence of our assumption that the allowable motion of the manipulator is completely parameterized by the joint angles θ. If a wrench causes no joint torques, it must be resisted by

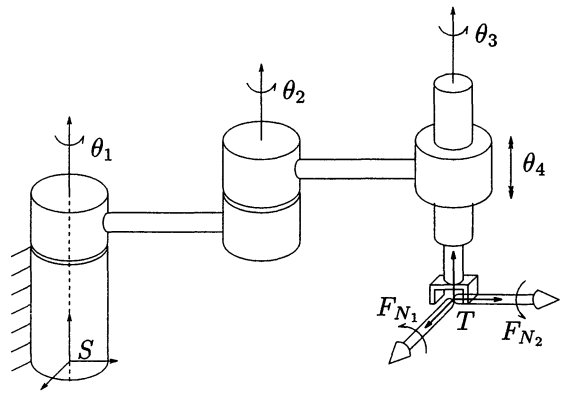


Figure 3.15: End-effector wrenches which generate no joint torques.

structural forces generated by the mechanism. Such a situation occurs when F lies in the null space of J_{st}^T. In this case, the force balance equation is satisfied with $\tau = 0$; the resisting forces are supplied completely by the robot's mechanical structure.

Example 3.10. End-effector forces for a SCARA robot
Consider again the SCARA robot from the Example 3.8. The transpose of the spatial manipulator Jacobian is

$$(J_{st}^s)^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ l_1 c_1 & l_1 s_1 & 0 & 0 & 0 & 1 \\ l_1 c_1 + l_2 c_{12} & l_1 s_1 + l_2 s_{12} & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The null space of this matrix is spanned by

$$F_{N_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad F_{N_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix};$$

hence, workspace torques about the x- and y-axes of the manipulator cannot be applied by the manipulator. For example, twisting the manipulator as shown in Figure 3.15 generates no joint torques and no motion of the manipulator.

4.3 Singularities

At a given configuration, the manipulator Jacobian describes the relationship between the instantaneous velocity of the end-effector and the joint velocities:

$$V_{bt}^s = J_{bt}^s(\theta)\dot{\theta}.$$

A *singular configuration* of a robot manipulator is a configuration at which the manipulator Jacobian drops rank. For a six degree of freedom manipulator in $SE(3)$, the Jacobian fails to be invertible at singular points and hence the manipulator is not able to achieve instantaneous motion in certain directions. Near singular configurations, the size of the joint velocities required to maintain a desired end-effector velocity in certain directions can be extremely large.

If a manipulator has fewer than six degrees of freedom, a singular configuration corresponds to a configuration in which the number of degrees of freedom of the end-effector drops. This is again characterized by the manipulator Jacobian dropping rank, i.e., two or more of the columns of $J^s_{st}(\theta) \in \mathbb{R}^{6\times n}$ become linearly dependent. Since most manipulators are designed for tasks in which all of the degrees of freedom are needed, singular configurations should usually be avoided, if possible.

Singularities also affect the size of the end-effector forces that the manipulator can apply. At a singular configuration, some end-effector wrenches will lie in the null space of the Jacobian transpose. These wrenches can be balanced without applying *any* joint torques, as the mechanism will generate the opposing wrenches. On the other hand, *applying* an end-effector wrench in a singular direction is not possible. Such a wrench could be balanced by any wrench in the singular direction, and hence it can be balanced by the zero wrench. If no other external forces are present, no forces will be generated.

In order to avoid these difficulties, it is necessary to identify singular configurations of a manipulator. We concentrate on classifying several common singularities for six degree of freedom manipulators and show how these can be determined by analyzing the geometry of the system. The cases presented here can be extended to consider more general open-chain manipulators. For each of the geometric conditions given below, we give a sketch of the proof of singularity. To illustrate some of the different ways in which singularities can be analyzed, we use a different proof technique for each example.

Example 3.11. Two collinear revolute joints
The Jacobian for a six degree of freedom manipulator is singular if there exist two revolute joints with twists

$$\xi_1 = \begin{bmatrix} -\omega_1 \times q_1 \\ \omega_1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} -\omega_2 \times q_2 \\ \omega_2 \end{bmatrix}$$

which satisfy the following conditions:

1. The axes are parallel: $\omega_1 = \pm\omega_2$.

2. The axes are collinear: $\omega_i \times (q_1 - q_2) = 0$, $i = 1, 2$.

Proof. In analyzing the singularity of a matrix, we are permitted to pre- or post-multiply the matrix by a nonsingular matrix of the proper dimensions. Pre-multiplication by a nonsingular matrix can be used to add one row to another or switch two rows, while post-multiplication can be used to perform the same operations on columns.

Assume, without loss of generality, that the columns of the Jacobian which are linearly dependent are the *first* two columns of J. The Jacobian has the form

$$J(\theta) = \begin{bmatrix} -\omega_1 \times q_1 & -\omega_2 \times q_2 & \cdots \\ \omega_1 & \omega_2 & \cdots \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

and we can assume $\omega_1 = \omega_2$ by negating the second column if necessary. Subtracting column 1 from column 2 yields

$$J(\theta) \sim \begin{bmatrix} -\omega_1 \times q_1 & -\omega_2 \times (q_2 - q_1) & \cdots \\ \omega_1 & 0 & \cdots \end{bmatrix},$$

where the symbol $\sim$ denotes equivalence of two matrices (up to elementary column operations). Using condition 2, the second column is zero, so that $J(\theta)$ is singular. □

This type of singularity is common in spherical wrist assemblies that are composed of three mutually orthogonal revolute joints whose axes intersect at a point. By rotating the second joint in the wrist, it is possible to align the first and third axes and the manipulator Jacobian becomes singular. In this configuration, rotation about the axis normal to the plane defined by the first and second joints is not possible.

Example 3.12. Three parallel coplanar revolute joint axes
The Jacobian for a six degree of freedom manipulator is singular if there exist three revolute joints which satisfy the following conditions:

1. The axes are parallel: $\omega_i = \pm\omega_j$ for $i, j = 1, 2, 3$.
2. The axes are coplanar: there exists a plane with unit normal n such that $n^T\omega_i = 0$ and

$$n^T(q_i - q_j) = 0, \qquad i, j = 1, 2, 3.$$

Proof. Another type of transformation which can be used in analyzing singularities is to change the frame of reference used to express the twists that form the columns of the Jacobian. A change of coordinates affects twists (and hence the Jacobian) by pre-multiplying by the adjoint matrix corresponding to the change of basis. Since the adjoint is an invertible transformation ($\mathrm{Ad}_g^{-1} = \mathrm{Ad}_{g^{-1}}$), this does not affect the singularity of the matrix.

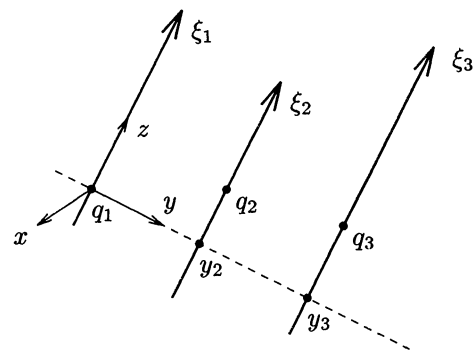


Figure 3.16: Three coplanar, parallel, revolute twists.

After an initial column permutation, assume $J(\theta)$ has the form

$$J(\theta) = \begin{bmatrix} -\omega_1 \times q_1 & -\omega_2 \times q_2 & -\omega_3 \times q_3 & \cdots \\ \omega_1 & \omega_2 & \omega_3 & \cdots \end{bmatrix}.$$

Attach a coordinate frame to the point q_1 with the z-axis of the frame pointing in the direction of ω_1 (see Figure 3.16). Further, choose the frame such that the plane formed by the axes is the yz plane in the new coordinates. Thus, each axis has a point of intersection which lies on the y-axis. Call these points $y_1(=0)$, y_2, and y_3. Now, with respect to this frame, the Jacobian has the form

$$\mathrm{Ad}_g J(\theta) = \begin{bmatrix} 0 & \pm y_2 & \pm y_3 & \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \cdots \\ 1 & \pm 1 & \pm 1 & \end{bmatrix}.$$

The first three columns are clearly linearly dependent. □

The elbow manipulator exhibits this singularity in its reference configuration (see Figure 3.11).

Example 3.13. Four intersecting revolute joint axes
The Jacobian for a six degree of freedom manipulator is singular if there exist four revolute joint axes that intersect at a point q:

$$\omega_i \times (q_i - q) = 0, \quad i = 1, \ldots, 4.$$

Proof. This example is trivial if we choose a frame whose origin is at the common point of intersection of the four revolute twists. However, we can also show singularity by making use of reciprocal screw systems. Recall

from Section 5.3 of Chapter 2 that a wrench is reciprocal to a twist when the inner product between the wrench and the twist is zero (indicating that no work is done by applying the wrench and moving along the twist). Since we are in a 6-dimensional space, if we can show that the dimension of the reciprocal system is sufficiently large (three for this example), then we can show singularity of the system of twists. This technique works well when there are a large number of twists and hence the size of the reciprocal system is small.

For this example, we make use of the following fact: every revolute twist is reciprocal to a pure force, in any direction, applied to a point on the axis of the revolute twist. To see this, it suffices to consider a twist and wrench through the origin:

$$V = \begin{bmatrix} 0 \\ \omega \end{bmatrix} \qquad F = \begin{bmatrix} f \\ 0 \end{bmatrix} \qquad \Longrightarrow \qquad V^T F = 0.$$

It is left to the reader to verify that this case generalizes appropriately.

We can now use this fact to construct the reciprocal system for the four twists which intersect at a point. Since any pure force through this point corresponds to a reciprocal wrench, it follows that the dimension of the reciprocal system is three and hence the four twists must be singular. □

This type of singularity occurs in the inverse elbow manipulator (see Exercise 4) when the final joint axis intersects the shoulder adding a fourth axis as shown.

The singularities given here and in the exercises are by no means exhaustive. However, they do occur frequently and are often easy to determine just by examining the geometry of the manipulator. It is also possible for a manipulator to exhibit different types of singularities at a single configuration. In this case, depending on the number and type of the singularities, the manipulator may lose the ability to move in several different directions at once. For example, if the arm of the elbow manipulator shown in Figure 3.4 is held vertically over the base, it exhibits all three of the singularities we have just illustrated. However, it still has four degrees of freedom (instead of three) since two of the singularities restrict motion in the same direction.

In addition to singularities of the manipulator Jacobian, a robot can also lose degrees of freedom when the joint variables are constrained to lie in a closed interval. In this case, a loss of freedom of motion can occur when one or more of the joints is at the limit of its travel. At such a configuration, motion past the joint limit is not allowed and the motion of the end-effector is restricted.

4.4 Manipulability

As we saw in the previous section, when a manipulator is at a singular configuration there are directions of movement which require high joint rates and forces. Near a singularity, movement may also be difficult in certain directions. The *manipulability* of a robot describes its ability to move freely in all directions in the workspace.

Manipulability measures can be divided into two rough classes:

1. The ability to *reach* a certain position or set of positions
2. The ability to *change* the position or orientation at a given configuration

The first of these measures is directly related to the workspace of a manipulator. Depending on the task, we may want to use the complete, reachable, or dextrous workspaces to characterize the manipulability of a manipulator. The second class of measures concerns the manipulability of a manipulator around a given configuration; that is, it is a local property.

To study local manipulability, we examine the Jacobian of the manipulator, which relates infinitesimal joint motions to infinitesimal workspace motions. Throughout this section we write J for the manipulator Jacobian J_{st}. Either the spatial or body Jacobian can be used, but the body Jacobian is preferred since the body velocity of the end-effector is independent of the choice of base frame.

There are many different local manipulability measures that have been proposed in the literature and which are useful in different situations. We present a small sample of some of the more common measures here. Many of these measures rely on the singular values of J. Recall that for a matrix $A \in \mathbb{R}^{p \times n}$, the singular values of A are the square roots of the eigenvalues of $A^T A$. We write $\sigma(A)$ for the set of singular values of A and $\lambda(A)$ to denote the set of eigenvalues of A. The maximum singular value of a matrix is equal to the induced two-norm of the matrix:

$$\sigma_{\max}(A) = \max_{\|x\|_2=1} \|Ax\|_2 = \|A\|_2.$$

If a matrix is singular, then at least one of its singular values is zero.

Example 3.14. Minimum singular value of J

$$\mu_1(\theta) = \sigma_{\min}(J(\theta))$$

The minimum singular value of the Jacobian corresponds to the minimum workspace velocity that can be produced by a unit joint velocity vector. The corresponding eigenvector gives the direction (twist) in which the motion of the end-effector is most limited. At a singular configuration, the minimum singular value of J is zero.

Example 3.15. Inverse of the condition number of J

$$\mu_2(\theta) = \frac{\sigma_{\min}(J(\theta))}{\sigma_{\max}(J(\theta))}$$

The condition number of a matrix A is defined as the ratio of the maximum singular value of A to the minimum singular value of A. For the Jacobian, the inverse condition number gives a measure of the sensitivity of the magnitude of the end-effector velocity V to the direction of the joint velocity vector $\dot{\theta}$. It provides a normalized measure of the minimum singular value of J. At a singular configuration, the inverse condition number is zero.

Example 3.16. Determinant of J

$$\mu_3(\theta) = \det J(\theta)$$

The determinant of the Jacobian measures the volume of the velocity ellipsoid (in the workspace) generated by unit joint velocity vectors. It is important to note that $\mu_3(\theta)$ does not contain information about the condition number of J. In particular, since $\det J(\theta)$ is the product of the singular values of $J(\theta)$, it can be large even if $\sigma_{\min}(J(\theta))$ is small, by having a large $\sigma_{\max}(J(\theta))$.

These manipulability measures can be used to provide an alternate definition for the dextrous workspace of a manipulator. For any of the measures given above, define the set W_D' as

$$W_D' = \{g_{st}(\theta) : \theta \in Q, \mu_i(\theta) \neq 0\} \subset SE(3). \tag{3.61}$$

W_D' is the set of end-effector configurations for which the manipulator can move infinitesimally in any direction. Note that W_D' is a subset of $SE(3)$, unlike our previous definition (in equation (3.13)) which consisted of the subset of $\mathbb{R}^3$ at which the manipulator could achieve *any* orientation.

Additional manipulability measures are given in the exercises.

5 Redundant and Parallel Manipulators

In this section, we briefly consider some other kinematic mechanisms that occur frequently in robotic manipulation. We focus on two particular types of structures—redundant manipulators and parallel manipulators—and indicate how to extend some of the results of this chapter to cover these cases.

5.1 Redundant manipulators

In order to perform a given task, a robot must have enough degrees of freedom to accomplish that task. In the analysis presented so far, we have concentrated on the case in which the robot has precisely the required degrees of freedom. A *kinematically redundant* manipulator has more than the minimal number of degrees of freedom required to complete a set of tasks.

A redundant manipulator can have an infinite number of joint configurations which give the same end-effector configuration. The extra degrees of freedom present in redundant manipulators can be used to avoid obstacles and kinematic singularities or to optimize the motion of the manipulator relative to a cost function. Additionally, if joint limits are present, redundant manipulators can be used to increase the workspace of the manipulator.

The derivation of the forward kinematics of a redundant manipulator is no different from the derivation presented in Section 2. Using the product of exponentials formula,

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0),$$

where n is greater than p, the dimension of the workspace ($p = 3$ for planar manipulators and $p = 6$ for spatial manipulators). The Jacobian of a redundant manipulator has the form

$$J^s_{st}(\theta) = \begin{bmatrix} \xi_1 & \xi'_2 & \cdots & \xi'_n \end{bmatrix},$$

where ξ'_i is the twist corresponding to the i^{th} joint axis in the current configuration. $J^s_{st} \in \mathbb{R}^{p \times n}$ has more columns than rows.

The inverse kinematics problem for a redundant manipulator is ill-posed: there may exist infinitely many configurations of the robot which give the desired end-effector configuration. In fact, if we keep the end-effector configuration fixed, the robot is still free to move along any trajectory which satisfies

$$g_{st}(\theta(t)) = g_d, \tag{3.62}$$

where $g_d \in SE(3)$ is the desired configuration of the end-effector. The set of all θ which satisfy this equation is called the *self-motion manifold* for the configuration g_d. Differentiating equation (3.62), we obtain

$$J^s_{st}(\theta(t))\dot{\theta} = (\dot{g}_{st} g^{-1}_{st})^\vee = 0.$$

Thus, the motions which are allowed must have joint velocities which lie in the null space of the manipulator Jacobian. A motion along the self-motion manifold is called an *internal motion.*

More generally, given an end-effector path $g(t)$, we would like to find a corresponding joint trajectory $\theta(t)$. Since there may be an infinite number of joint trajectories which give the requisite end-effector path, additional criteria are used to choose among them. One common solution is to choose the minimum joint velocity which gives the desired workspace velocity. This is achieved by choosing

$$\dot{\theta} = J_{st}^{\dagger}(\theta) V_{st},$$

where $J^{\dagger} = J^T(JJ^T)^{-1}$ is the Moore-Penrose generalized inverse of J. The properties of this and other kinematic redundancy resolution algorithms are discussed briefly in Chapter 7.

The manipulator Jacobian can also be used to relate joint torques to end-effector wrenches for redundant manipulators. Since the links of the manipulator are free to move even when the end-effector is fixed, a thorough understanding of the relationship between joint forces and end-effector wrenches requires a study of the dynamics of the manipulator. In particular, the possible existence of internal motions, combined with the inertial coupling between the links, can cause forces to be applied to the end-effector even if no joint torques are applied. We defer a complete discussion of this situation until Chapter 6, in which we study the dynamics of constrained systems in full detail. Using the results of that chapter, it will be possible to show that when a manipulator is in *static equilibrium*, the previous relationship,

$$\tau = J_{st}^T F, \tag{3.63}$$

still holds. This relationship gives the joint torques necessary to produce a given end-effector wrench when the system is stationary. Either the body or spatial Jacobian can be used, as long as the wrench F is represented appropriately.

Example 3.17. Self-motion manifold for a planar manipulator
Consider the planar manipulator shown in Figure 3.17a. Holding the position of the end-effector fixed, the system obeys the following kinematic constraints:

$$\begin{aligned} l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) &= x \\ l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) &= y. \end{aligned}$$

This is a set of two equations in three variables and hence there exist multiple solutions. A self-motion manifold for this manipulator is shown in Figure 3.17b.

The Jacobian for the mapping $p : \theta \mapsto (x, y)$ is

$$\frac{\partial p}{\partial \theta} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix}, \tag{3.64}$$

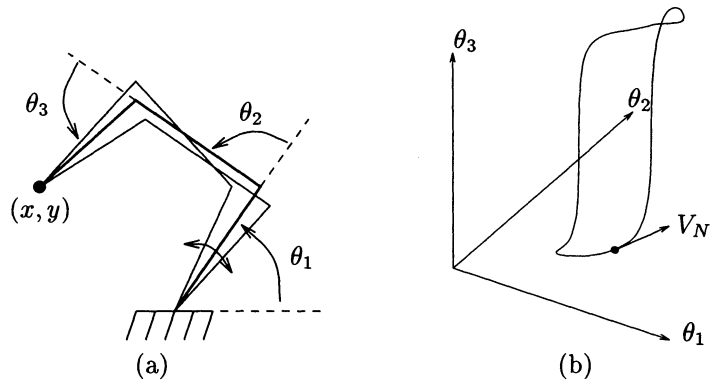


Figure 3.17: Self-motion manifold for a redundant planar manipulator.

where $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ and similarly for c_{ijk}. The Jacobian has a null space spanned by the vector

$$V_N = \begin{bmatrix} l_2 l_3 \sin\theta_3 \\ -l_2 l_3 \sin\theta_3 - l_1 l_3 \sin(\theta_2 + \theta_3) \\ l_1 l_2 \sin\theta_2 + l_1 l_3 \sin(\theta_2 + \theta_3) \end{bmatrix}.$$

Any velocity $\dot{\theta} = \alpha V_N$ is tangent to the self-motion manifold and maintains the position of the end-effector. One such velocity is shown as an arrow on Figure 3.17b.

5.2 Parallel manipulators

A *parallel* manipulator is one in which two or more series chains connect the end-effector to the base of the robot. An example is shown in Figure 3.18. Parallel manipulators can offer advantages over open-chain manipulators in terms of rigidity of the mechanism and placement of the actuators. For example, the manipulator in Figure 3.18 can be completely actuated by controlling only the first link in each chain, eliminating the need to place motors at the distal links of the manipulator. Parallel manipulators are also called *closed-chain* manipulators, since they contain one or more closed kinematic chains.

Structure equation for a parallel mechanism

The forward kinematics for a parallel manipulator are described by equating the end-effector location specified by each chain. Suppose we have a manipulator with n_1 joints in the first chain (including the end-effector) and n_2 joints in the second chain (including the end-effector). Then, the

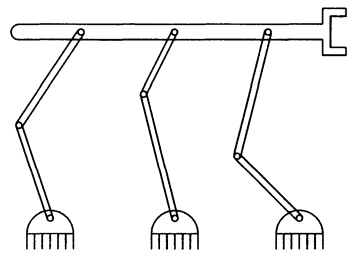


Figure 3.18: A parallel manipulator consisting of three series chains connected to a single end-effector.

forward kinematics is described in exponential coordinates as

$$g_{st} = e^{\widehat{\xi}_{11}\theta_{11}} \cdots e^{\widehat{\xi}_{1n_1}\theta_{1n_1}} g_{st}(0) = e^{\widehat{\xi}_{21}\theta_{21}} \cdots e^{\widehat{\xi}_{2n_2}\theta_{2n_2}} g_{st}(0), \quad (3.65)$$

where all quantities are specified relative to a single base and tool frame. Equation (3.65) is called the *structure equation* (or loop equation) for the manipulator and introduces constraints between the possible joint angles of the manipulator. It is because of these constraints that we can control the end-effector location by specifying only a subset of the joint variables: the other joint variables must take on values such that equation (3.65) is satisfied.

Since the joint variables are constrained by equation (3.65), the joint space for a parallel manipulator is not simply the Cartesian product of the individual joint spaces, as in the open-chain case. Rather, it is the subset $Q' \subset Q$ which also satisfies equation (3.65). Determining the dimension of Q', and hence the number of degrees of freedom for the parallel manipulator, requires careful inspection of the number of joints and links in the mechanism.

Let N be the number of links in the mechanism, g the number of joints, and f_i the number of degrees of freedom for the i^{th} joint. The number of degrees of freedom can be obtained by taking the total degrees of freedom for all of the links and subtracting the number of constraints imposed by the joints attached to the links. If all of the joints define independent constraints, the number of degrees of freedom for the mechanism is

$$F = 6N - \sum_{i=1}^{g}(6 - f_i) = 6(N - g) + \sum_{i=1}^{g} f_i. \quad (3.66)$$

Equation (3.66) is called *Gruebler's formula.* Gruebler's formula only holds when the constraints imposed by the joints are independent. For planar motions, Gruebler's formula holds with 6 replaced by 3.

Although the forward kinematics of a parallel manipulator is complicated by the closed loop nature of the mechanism, the inverse problem

is no more difficult than in the open-chain case. Namely, the inverse kinematics problem for a parallel manipulator is solved by considering the inverse problem for each open-chain mechanism which connects the ground to the end-effector. This can be done using the methods presented in Section 3.

Velocity and force relationships

The velocity of the end-effector of a parallel manipulator is related to the velocity of the joints of the manipulator by differentiating the structure equation (3.65). This gives a Jacobian matrix for each chain:

$$V_{st}^s = J_1^s \begin{bmatrix} \dot{\theta}_{11} \\ \vdots \\ \dot{\theta}_{1n_1} \end{bmatrix} = J_2^s \begin{bmatrix} \dot{\theta}_{21} \\ \vdots \\ \dot{\theta}_{2n_2} \end{bmatrix}, \tag{3.67}$$

where

$$J_1^s = \begin{bmatrix} \xi_{11} & \xi_{12}' & \cdots & \xi_{1n_1}' \end{bmatrix} \qquad J_2^s = \begin{bmatrix} \xi_{21} & \xi_{22}' & \cdots & \xi_{2n_2}' \end{bmatrix}.$$

Since the mechanism contains closed kinematic chains, not all joint velocities can be specified independently.

The manipulator Jacobian can be written in more conventional form by stacking the Jacobians for each chain:

$$\begin{bmatrix} J_1^s & 0 \\ 0 & J_2^s \end{bmatrix} \dot{\theta} = \begin{bmatrix} I \\ I \end{bmatrix} V_{st}^s. \tag{3.68}$$

This equation has a form very similar to one which we shall use in Chapter 5 to describe the kinematics of a multifingered grasp. Indeed, if we view each of the chains as grasping the end-effector link via a prismatic or revolute joint, then a parallel manipulator is very similar to a multifingered hand grasping an object.

The relationship between joint torques and end-effector forces for a parallel manipulator is more complicated than for an open-chain manipulator. The basic problem is that two or more chains can fight against each other and apply forces which cause no net end-effector wrench. A set of joint torques which causes no net end-effector wrench is called an *internal force*. We defer a discussion of internal forces until Chapter 5, where they arise naturally in the context of grasping.

Singularities

Determining the singularities of parallel mechanisms is more involved than it is for serial mechanisms. Consider a general parallel mechanism

with k chains and let Θ_i denote the joint variables in the i^{th} chain. The Jacobian of the structure equations has the form

$$V_{st}^s = J_1(\Theta_1)\dot{\Theta}_1 = \cdots = J_k(\Theta_k)\dot{\Theta}_k. \tag{3.69}$$

We say that an end-effector velocity is *admissible* at a configuration $\Theta = (\Theta_1, \ldots, \Theta_k)$ if there exist $\dot{\Theta}_i$ which satisfy equation (3.69). The set of admissible velocities forms a linear space, since it is the intersection of the range spaces of $J_i(\theta)$, $i = 1, \ldots, k$. Hence, we can define the *rank of the structure equations*, ρ, at a configuration Θ as the dimension of the space of admissible velocities,

$$\rho = \dim \bigcap_{i=1}^{k} \mathcal{R}(J_i(\theta)), \tag{3.70}$$

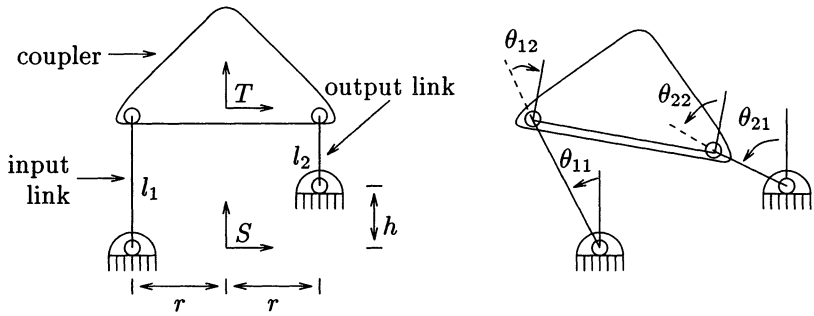
where $\mathcal{R}(A)$ denotes the range space of the matrix A.

A parallel mechanism is *kinematically singular* at a point if the rank of the structure equations drops at that point. In this case, the tool loses the ability to move instantaneously in some direction. This is analogous to our description of singularities in serial mechanisms. However, at this point we have not yet identified which joints in the mechanism are actuated and which are passive. If a parallel mechanism is fully actuated, then these are the only types of singularities that can occur. However, in most instances, only some of the joints of a parallel manipulator are actuated and this can lead to additional singularities. We call this second type of singularity an *actuator singularity* and give an example of it when we study the Stewart platform in Section 5.4.

5.3 Four-bar linkage

To illustrate some of the concepts introduced above, we consider the four-bar linkage shown in Figure 3.19. The mechanism consists of three rigid bodies connected together by revolute joints. The links attached to the ground frame are called the *input* and *output* links, the rigid body which connects the input and output links is called the *coupler*. It is called a four-bar mechanism even though there are only three links, since historically the ground frame is considered to be the fourth link.

Four-bar linkages are usually studied in the context of mechanism synthesis. For example, we might try to find a mechanism such that the input and output links satisfy a given functional relationship, creating a type of mechanical computer. Alternatively, one might wish to design the mechanism so that a point on the coupler traces a specified path or passes through a set of points. Many other variations exist, but for all of these problems one tries to choose the kinematic parameters which describe the mechanism—the lengths of the links, shape of the coupler,



(a) Reference configuration (b) Angle description

Figure 3.19: Four-bar linkage.

location of the joints—so that a given task is performed. In this section, we bypass the synthesis problem and concentrate on kinematic analysis of a given mechanism.

The number of degrees of freedom of a four-bar mechanism is given by Gruebler's formula:

$$\begin{aligned} N &= 3 \text{ links} \\ g &= 4 \text{ joints} \qquad \Longrightarrow \qquad F = 3(3-4)+4 = 1. \\ f_i &= 1 \text{ DOF/link} \end{aligned}$$

The fact that there is only one degree of freedom helps explain the terminology of the input and output links. Note that we used the planar version of Gruebler's formula to calculate the mobility of the mechanism. A quick calculation shows that the spatial version of the formula gives $F = 6(3-4)+4 = -2$ (!). We leave the resolution of this apparent paradox as an exercise.

To write the structure equations, we must first assign base and tool frames and choose a reference configuration. The tool and base frames are assigned as shown in the figure. We choose the reference configuration ($\theta = 0$) to be the configuration shown in Figure 3.19a. Note that this is not the usual reference configuration if we had considered each of the kinematic chains as independent two-link robots. However, it will be convenient to have the reference configuration satisfy the kinematic constraints and hence we define the angles as shown in Figure 3.19b. With respect to this configuration, the structure equations have the form

$$g_{st} = e^{\hat{\xi}_{11}\theta_{11}} e^{\hat{\xi}_{12}\theta_{12}} g_{st}(0) = e^{\hat{\xi}_{21}\theta_{21}} e^{\hat{\xi}_{22}\theta_{22}} g_{st}(0).$$

Note that in the plane this gives three constraints in terms of four variables, leaving one degree of freedom as expected.

The twists can be calculated using the formulas for twists in the plane derived in the exercises at the end of Chapter 2. In particular, a revolute

joint in the plane through a point $q = (q_x, q_y)$ is described by a planar twist

$$\xi = \begin{bmatrix} q_y \\ -q_x \\ 1 \end{bmatrix} \in \mathbb{R}^3.$$

This yields

$$\xi_{11} = \begin{bmatrix} 0 \\ r \\ 1 \end{bmatrix} \quad \xi_{12} = \begin{bmatrix} l_1 \\ r \\ 1 \end{bmatrix} \quad \xi_{21} = \begin{bmatrix} h \\ -r \\ 1 \end{bmatrix} \quad \xi_{22} = \begin{bmatrix} h + l_2 \\ -r \\ 1 \end{bmatrix}$$

$$g_{st}(0) = \begin{bmatrix} I & \begin{bmatrix} 0 \\ h+l_2 \end{bmatrix} \\ 0 & 1 \end{bmatrix}.$$

Expanding the product of exponentials formula gives

$$\begin{aligned} -r - l_1 \sin\theta_{11} + r\cos(\theta_{11} + \theta_{12}) &= x = r - l_2 \sin\theta_{21} - r\cos(\theta_{21} + \theta_{22}) \\ l_1 \cos\theta_{11} + r\sin(\theta_{11} + \theta_{12}) &= y = h + l_2 \cos\theta_{21} - r\sin(\theta_{21} + \theta_{22}) \\ \theta_{11} + \theta_{12} &= \phi = \theta_{21} + \theta_{22} \end{aligned}$$

where ϕ is the angle the tool frame makes with the horizontal. From the form of this equation, it is clear that solving for the forward kinematics is a complicated task, though it turns out that in many cases it can be done in closed form.

The Jacobian of the structure equation has the form

$$V_{st}^s = \begin{bmatrix} \xi_{11} & \xi_{12}' \end{bmatrix} \begin{bmatrix} \dot\theta_{11} \\ \dot\theta_{12} \end{bmatrix} = \begin{bmatrix} \xi_{21} & \xi_{22}' \end{bmatrix} \begin{bmatrix} \dot\theta_{21} \\ \dot\theta_{22} \end{bmatrix}.$$

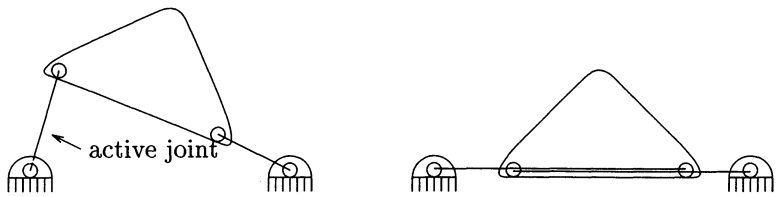
To calculate the individual columns of the Jacobian, we write the twists at the current configuration of the manipulator. Thus,

$$V_{st}^s = \begin{bmatrix} 0 & l_1\cos\theta_{11} \\ r & r + l_1\sin\theta_{11} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot\theta_{11} \\ \dot\theta_{12} \end{bmatrix} = \begin{bmatrix} h & h + l_2\cos\theta_{21} \\ -r & -r + l_2\sin\theta_{21} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot\theta_{21} \\ \dot\theta_{22} \end{bmatrix}.$$

This gives the velocity constraints on the system. Since the individual Jacobians for each chain only have two columns, it is clear that the dimension of the space of admissible velocities is at most two. To examine the mobility more closely, we rearrange the Jacobian to isolate the actuated and passive joints.

Suppose we take $\theta = \theta_{11}$ as the actuated joint and let $\alpha = (\theta_{12}, \theta_{21}, \theta_{22})$ represent the passive joints. The Jacobian of the structure equation can be rearranged as

$$\xi_{11}\dot\theta = \begin{bmatrix} -\xi_{12}' & \xi_{21} & \xi_{22}' \end{bmatrix} \dot\alpha \tag{3.71}$$



(a) Singular configuration (b) Uncertainty configuration

Figure 3.20: Potential singular configurations for four-bar mechanisms.

(this type of rearrangement works only in the special case where we have two serial chains or a single kinematic loop). The form of this equation suggests that if we specify the velocity of the actuated joint θ, then we can solve for the velocity of the passive joints if the right-hand side of equation (3.71) is nonsingular.

The right-hand side of equation (3.71) corresponds to the twists generated by three parallel, revolute joints. We know from our study of singularities of twists that if the three axes are coplanar in addition to being parallel, then the twists are linearly dependent. In the planar case, this means that if the passive joints are collinear, then the right-hand side of equation (3.71) loses rank and the mechanism may not be able to move. However, this condition is not sufficient since it may happen that ξ_{11} is in the range of $\{\xi'_{12}, \xi_{21}, \xi'_{22}\}$ even though they are singular. These two different situations are shown in Figure 3.20. Note that switching the role of the input and output links changes the singular configurations of the mechanism. For example, the singularity shown in Figure 3.20a is only a singularity if the left-hand link is chosen as the input link (since this choice gives three collinear passive joints).

The configuration shown in Figure 3.20b is known as an *uncertainty configuration* in the kinematics literature. In this case, it is actually possible for the mechanism to move instantaneously in two independent directions. Examining the structure equations, we see that

$$\begin{bmatrix} 0 & 0 \\ r & r-l_1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -r & -r+l_1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{21} \\ \dot{\theta}_{22} \end{bmatrix}$$

and hence

$$\dot{\theta}_{N_1} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \text{and} \quad \dot{\theta}_{N_2} = \begin{bmatrix} r-l_1 \\ -r \\ r-l_1 \\ -r \end{bmatrix}$$

represent two independent, instantaneously admissible velocities. These two independent velocities exist only if the mechanism is perfectly aligned

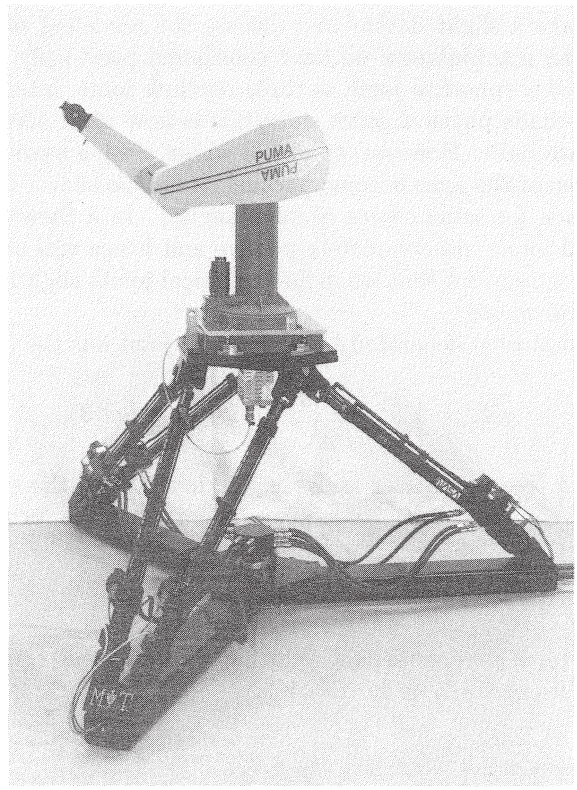


Figure 3.21: A Stewart platform with a PUMA robot attached. (Photo courtesy of Steve Dubowsky, MIT)

and hence uncertainty configurations rarely occur in practice.

5.4 Stewart platform

Another common example of a parallel mechanism is the Stewart platform, an example of which is shown in Figure 3.21. The mechanism consists of two rigid bodies, connected by a set of prismatic joints. Each prismatic joint is connected to the rigid body by a spherical joint, allowing complete rotational motion. Only the prismatic joints are actuated.

Stewart platforms are commonly used in aircraft flight simulators to move an aircraft cockpit along the motion indicated by the (simulated) dynamics of the system. Although the concept of a Stewart platform is quite old (it was studied by Stewart in the 1950s [112]), it is only recently that the kinematics for a general Stewart platform have been solved in complete generality.

In order to write the structure equation for a Stewart platform, we

must first take a slight detour and discuss the modeling of a spherical joint. For the manipulators we have considered previously, we have always modeled a spherical joint as three revolute joints intersecting at a point. This made physical sense since this is how most actuated spherical wrists are built. However, spherical wrists always have singularities when any two of the axes become parallel (it can be shown that this will always happen for some choice of joint angles). In a Stewart platform, the spherical joints are completely passive and hence will never become singular. This requires that we define spherical joints slightly differently than spherical wrists.

The rigid motion generated by a spherical joint has the form

$$g(R) = \begin{bmatrix} R & (I - R)q \\ 0 & 1 \end{bmatrix} \qquad R \in SO(3),$$

where R is a free parameter and q is the location of the center of the wrist. Similarly, the velocity of a spherical joint has the form

$$V^s = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix} \qquad \omega \in \mathbb{R}^3,$$

where ω is a free parameter (the velocity). To cast this equation into a more useful framework, we rewrite V^s as

$$V^s = \begin{bmatrix} -e_1 \times q & -e_2 \times q & -e_3 \times q \\ e_1 & e_2 & e_3 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix},$$

where e_i is the i^{th} unit vector in $\mathbb{R}^3$. Notice that the columns of the matrix which defines V^s are *never* linearly dependent.

We can now write the structure equations for the Stewart platform. Let $g_{s_i}(R_{s_i})$ represent the orientation of the i^{th} spherical joint attached to the base and $g_{t_i}(R_{t_i})$ represent the i^{th} spherical joint attached to the tool. Then, the structure equation for the Stewart platform is given by

$$g_{st} = g_{s_1}(R_{s_1})e^{\widehat{\xi}_1\theta_1}g_{t_1}(R_{t_1})g_{st}(0) = \cdots = g_{s_6}(R_{s_6})e^{\widehat{\xi}_6\theta_6}g_{t_6}(R_{t_6})g_{st}(0), \tag{3.72}$$

where ξ_i and θ_i model the motion of the i^{th} prismatic joint.

Solving the forward kinematics for a Stewart platform is a very difficult problem due to the large number and complicated form of the constraints. Abstractly, given the length of the links, we can solve the structure equations to find the orientations of the ball and socket joints and then determine the configuration of the tool frame. As the problem has been specified here, there is an extra degree of freedom in each link corresponding to rotation of the prismatic joint about its own axis. This further complicates the forward kinematics problem.

The inverse kinematics problem for the Stewart platform is remarkably simple. Given the desired configuration of the platform, we find the locations of the pivot points and solve for the distance between each base and the appropriate pivot. Let q_{s_i} be the location of the i^{th} pivot point on the base and q_{t_i} be the location of the tool pivot point (written relative to the base and tool frames, respectively). Then, the extension of the prismatic joints is given by

$$\theta_i = \|q_{s_i} - g_{st} q_{t_i}\|.$$

It is possible to derive this result in a manner similar to that used in solving the subproblems of Section 3; but in this case, the solution is obvious from the geometry of the manipulator.

We may now study the mobility of the Stewart platform by calculating the Jacobian of the structure equation. Taking the Jacobian of equation (3.72) yields

$$\begin{aligned} V_{st}^s &= \begin{bmatrix} \xi_{s_1,1} & \xi_{s_1,2} & \xi_{s_1,3} & \xi_1' & \xi_{t_1,1}' & \xi_{t_1,2}' & \xi_{t_1,3}' \end{bmatrix} \begin{bmatrix} \omega_{s_1} \\ \dot{\theta}_1 \\ \omega_{t_1} \end{bmatrix} \\ &= \cdots \\ &= \begin{bmatrix} \xi_{s_6,1} & \xi_{s_6,2} & \xi_{s_6,3} & \xi_6' & \xi_{t_6,1}' & \xi_{t_6,2}' & \xi_{t_6,3}' \end{bmatrix} \begin{bmatrix} \omega_{s_6} \\ \dot{\theta}_6 \\ \omega_{t_6} \end{bmatrix}. \end{aligned} \tag{3.73}$$

It can be shown that the Jacobian matrices are never singular as long as θ_i is nonzero (see Exercise 22). Hence, all tool velocities are admissible.

However, an interesting problem occurs when the tool frame and base frame are coplanar. In this case, the actuated joints can only generate forces in the plane, and hence the mechanism cannot resist (or apply) nonplanar forces or torques. Note that the mechanism is still not kinematically singular: the joints can accommodate any motion of the tool frame. However, the *actuated* joints cannot generate the wrenches necessary to actually achieve any motion. This is an example of the second class of singularity that was mentioned previously. We call this type of singularity an *actuator singularity* since it corresponds to a failure of the actuated joints to be able to generate arbitrary wrenches in the tool frame. This type of singularity is very closely related to the failure of the force-closure conditions which occur in grasping.

A geometric interpretation of this singularity in the Stewart platform can be obtained by noting that the system of wrenches which can be applied to the tool frame is given by the set of all zero-pitch (pure force) wrenches generated by the actuated joints through the points q_{t_i}. Since the prismatic joints generate zero-pitch wrenches, we can use the previously derived examples of singularities of zero-pitch screws to locate some

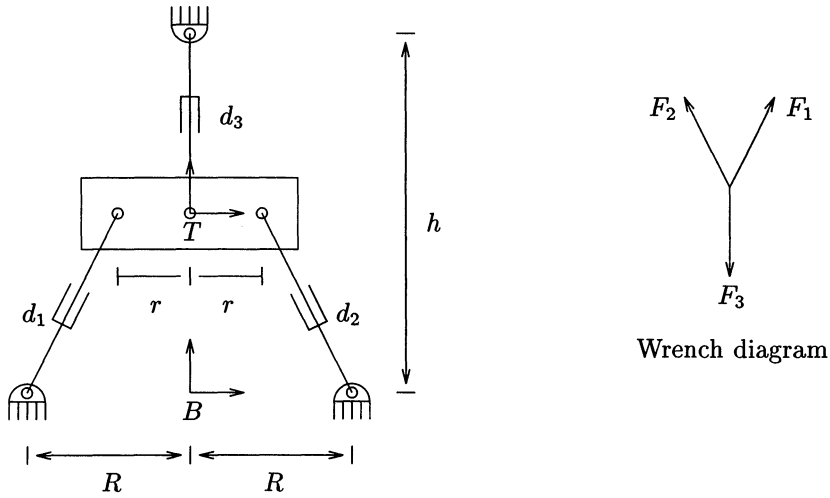


Figure 3.22: A planar version of the Stewart platform.

of the singularities of the Stewart platform. In this context, when the base and tool frames are coplanar, we get a singularity because we have four (actually six) coplanar, zero-pitch screws. A separate singularity occurs whenever two of the prismatic joints are collinear.

Example 3.18. Singularities for a planar Stewart platform
Consider the planar parallel mechanism shown in Figure 3.22. Three actuated prismatic joints are used to control the position and orientation of the platform. The revolute joints at the end of each link are passive. Kinematically, this mechanism shares some of the properties of the Stewart platform.

We concentrate on the singularities of the mechanism. As with the Stewart platform, it can be shown that there are no kinematic singularities where the dimension of the space of achievable velocities drops rank (it is always three). Since the actuated joints are all prismatic, the wrenches generated by the joints correspond to zero-pitch screws. In the plane, it can be shown that three zero-pitch screws intersecting at a point are singular. This is exactly the configuration in which the mechanism is drawn in Figure 3.22 (see the wrench diagram to the right).

Hence, in this configuration, it is not possible for the mechanism to generate pure torques around the point of common intersection. This is clear if we write down the wrenches relative to a coordinate frame attached at the intersection point. In this set of coordinates, we have

$$F = \begin{bmatrix} v_1 & v_2 & v_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix},$$

where v_i is the direction of the i^{th} prismatic axis and f_i is the force exerted by the i^{th} actuator. It is clear that we cannot generate a pure torque around the intersection point since

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \notin \text{range} \begin{bmatrix} v_1 & v_2 & v_3 \\ 0 & 0 & 0 \end{bmatrix}.$$

6 Summary

The following are the key concepts covered in this chapter:

1. The *forward kinematics* of a manipulator is described by a mapping $g_{st} : Q \to SE(3)$ which describes the end-effector configuration as a function of the robot joint variables. For open-chain manipulators consisting of revolute and prismatic joints, the kinematics can be written using the *product of exponentials formula*:

$$g_{st}(\theta) = e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} \cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0),$$

   where ξ_i is the twist corresponding to the i^{th} joint axis in the reference ($\theta = 0$) configuration.

2. The *(complete) workspace* of a manipulator is the set of end-effector configurations which can be reached by some choice of joint angles. The *reachable workspace* defines end-effector positions which can be reached at some orientation. The *dextrous workspace* defines end-effector positions which can be reached at any orientation.

3. The *inverse kinematics* of a manipulator describes the relationship between the end-effector configuration and the joint angles which achieve that configuration. For many manipulators, we can find the inverse kinematics by making use of the following subproblems:

Subproblem 1:	$e^{\widehat{\xi}\theta} p = q$	rotate one point onto another
Subproblem 2:	$e^{\widehat{\xi}_1\theta_1} e^{\widehat{\xi}_2\theta_2} p = q$	rotate about two intersecting twists
Subproblem 3:	$\|q - e^{\widehat{\xi}\theta} p\| = \delta$	move one point to a specified distance from another

   To find a complete solution, we apply the manipulator kinematics to a set of points which reduce the complete problem into an appropriate set of subproblems.

4. The *manipulator Jacobian* relates the joint velocities $\dot{\theta}$ to the end-effector velocity V_{st} and the joint torques τ to the end-effector wrench F:

$$\begin{aligned} V^s_{st} &= J^s_{st}(\theta)\dot{\theta} \qquad \tau = (J^s_{st})^T F_s \qquad \text{(spatial)} \\ V^b_{st} &= J^b_{st}(\theta)\dot{\theta} \qquad \tau = (J^b_{st})^T F_t \qquad \text{(body).} \end{aligned}$$

If the manipulator kinematics is written using the product of exponentials formula, then the manipulator Jacobians have the form:

$$J^s_{st}(\theta) = \begin{bmatrix} \xi_1 & \xi'_2 & \cdots & \xi'_n \end{bmatrix} \qquad \xi'_i = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1\theta_1}\cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)}\xi_i$$

$$J^b_{st}(\theta) = \begin{bmatrix} \xi^\dagger_1 & \cdots & \xi^\dagger_{n-1} & \xi^\dagger_n \end{bmatrix} \qquad \xi^\dagger_i = \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_i\theta_i}\cdots e^{\widehat{\xi}_n\theta_n} g_{st}(0)\right)}\xi_i.$$

5. A configuration is *singular* if the manipulator Jacobian loses rank at that configuration. Examples for a general six degree of freedom arm include:

   (a) Two collinear revolute joints

   (b) Three parallel, coplanar revolute joint axes

   (c) Four intersecting revolute joint axes

   The *manipulability* of a robot provides a measure of the nearness to singularity.

6. A manipulator is *kinematically redundant* if it has more than the minimally required degrees of freedom. The *self-motion manifold* describes the set of joint values which can be used to achieve a desired configuration of the end-effector. *Internal motions* correspond to motions along the self-motion manifold and satisfy

$$J_{st}(\theta)\dot{\theta} = 0.$$

7. A *parallel manipulator* has multiple kinematic chains connecting the base to the end-effector. For the case of two chains, the kinematics satisfies the structure equation

$$g_{st} = e^{\widehat{\xi}_{11}\theta_{11}}\cdots e^{\widehat{\xi}_{1n_1}\theta_{1n_1}} g_{st}(0) = e^{\widehat{\xi}_{21}\theta_{21}}\cdots e^{\widehat{\xi}_{2n_2}\theta_{2n_2}} g_{st}(0),$$

   where ξ_{ij} is twist for the the j^{th} joint on the i^{th} chain. The Jacobian of the structure equation has the form

$$V^s_{st} = J^s_1(\Theta_1)\dot{\Theta}_1 = J^s_2(\Theta_2)\dot{\Theta}_2,$$

   where $\Theta_i = (\theta_{i1}, \ldots, \theta_{in_i})$. A kinematic singularity occurs when the dimension of the space of admissible forces drops rank. Other singularities can occur when the set of end-effector forces which can be generated by the actuated joints drops rank.

7 Bibliography

There is a vast literature on robot kinematics, including a number of textbooks devoted to analysis, design, and control of manipulators. For an introductory treatment of the topics presented here, consult the textbook by Craig [21]. See also [35, 36, 80, 91, 122]. The product of exponentials formula was initially described by Brockett [12]; the presentation given here was inspired by the dissertation of Paden [86]. A selection of advanced topics in the flavor of the tools presented in this section can be found in a collection of papers edited by Brockett [13].

In terms of bounds on the number of inverse kinematic solutions to a six degree of freedom manipulator, Rastegar, Roth and Scheinman [101] established a bound of 32 using a non-constructive proof. This bound was made constructive by Duffy and Crane [29] and reduced to 16 by Primrose [95]. However, it was Lee and Liang [58] who gave a constructive procedure for finding the inverse kinematic solutions for a general manipulator. The procedure has been refined by Roth and Raghavan [97] and Manocha and Canny [65], whose account we follow in this chapter. Manseur and Doty [66] gave an example of a robot with 16 inverse kinematic solutions.

The treatment of parallel mechanisms given here is not the standard one. For a classical treatment of four-bar and other parallel mechanisms, see, for example, Hunt [42]. A detailed description of the four-bar synthesis problem, along with analytical and graphical solution techniques, can be found in the book by Erdman and Sandor [30], in addition to other textbooks on kinematics and design of mechanisms.

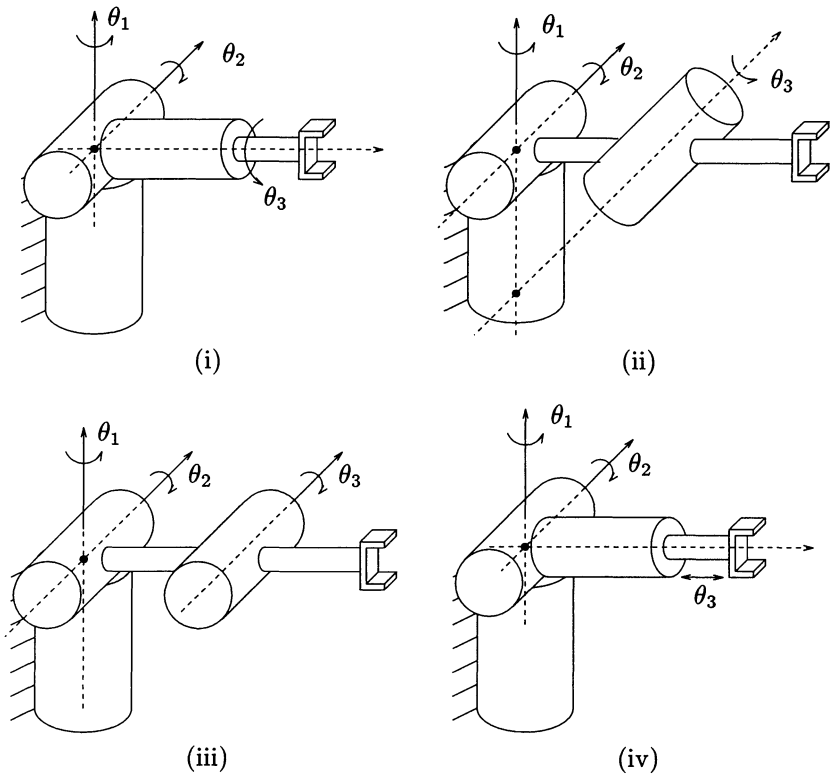


Figure 3.23: Some simple three degree of freedom manipulators.

8 Exercises

1. Draw the twists axes for the manipulators shown in Chapter 1.

2. Show that the forward kinematics map for a manipulator is independent of the order in which rotations and translations are performed.

3. For each of the three degree of freedom manipulators shown in Figure 3.23:

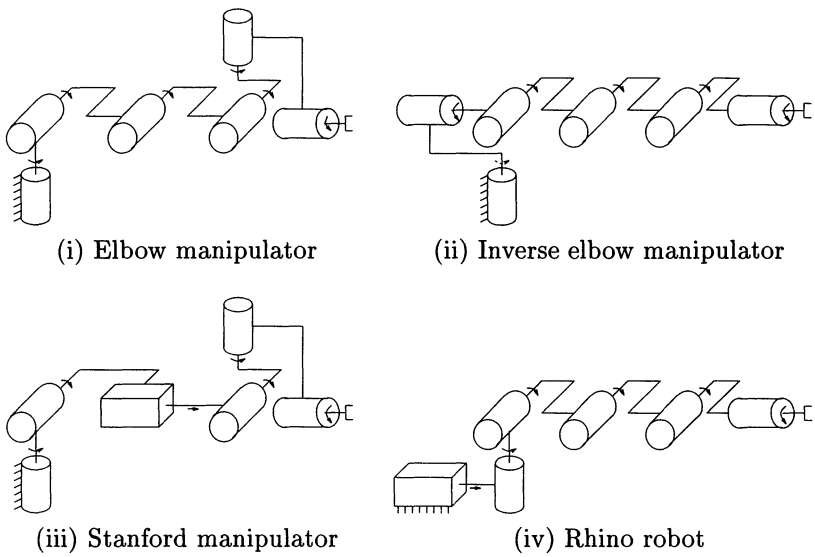
   (a) Find the forward kinematics map.

   (b) Solve the inverse kinematics problem using the Paden-Kahan subproblems.

   (c) Derive the spatial and body Jacobians.

4. For each of the manipulators shown schematically in Figure 3.24:

   (a) Find the forward kinematics map.



(i) Elbow manipulator (ii) Inverse elbow manipulator

(iii) Stanford manipulator (iv) Rhino robot

Figure 3.24: Sample manipulators. Revolute joints are represented by cylinders; prismatic joints are represented by rectangular boxes.

(b) Solve the inverse kinematics problem using the Paden-Kahan subproblems.

(c) Derive the spatial and body Jacobians.

(d) Give a geometric description of the singular configurations.

(e) Describe the reachable and dextrous workspaces and calculate the number of inverse kinematic solutions in different regions of the workspace.

(Note that some of these problems have already been solved in the examples.)

5. *Subproblem 2': Rotation about two non-intersecting axes*
Solve Subproblem 2 when the two axes ξ_1 and ξ_2 do not intersect. Use this subproblem to solve the inverse kinematics for the elbow manipulator in Example 3.5 when the first two joints do not intersect at a point.

6. *Subproblem 4: Rotation about two axes to given distances*
Let ξ_1, ξ_2 be two zero-pitch unit magnitude twists with intersecting axes, and p, q_1, and q_2 be points in $\mathbb{R}^3$ (see Figure 3.25). Find θ_1 and θ_2 such that

$$\|e^{\widehat{\xi_1}\theta_1} e^{\widehat{\xi_2}\theta_2} p - q_1\| = \delta_1$$

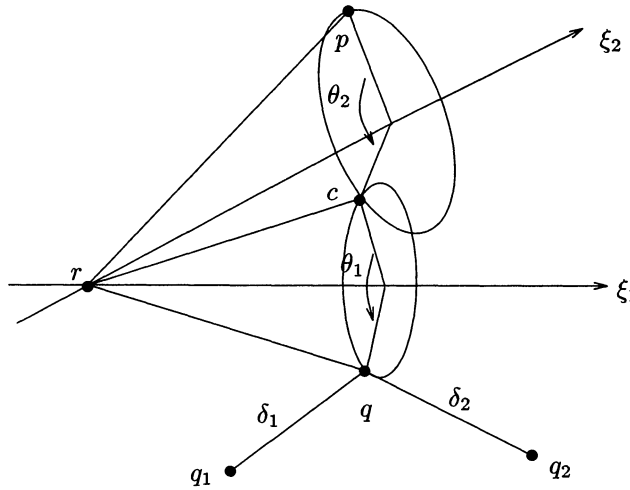


Figure 3.25: Subproblem 4: Rotate p about the axis of ξ_1 followed by a rotation about the axis of ξ_2 such that the final location of p is δ_1 from q_1 and δ_2 from q_2.

and

$$\|e^{\widehat{\xi_1}\theta_1}\, e^{\widehat{\xi_2}\theta_2} p - q_2\| = \delta_2.$$

(Hint: Find a point q such that $q = e^{\widehat{\xi_1}\theta_1} e^{\widehat{\xi_2}\theta_2} p$, and q is on the intersection of the three spheres centered at, respectively, q_1, q_2, and r, of radii δ_1, δ_2, and $\|p - r\|$.)

7. *Subproblem 5: Translation to a given distance*
   Let ξ be an infinite-pitch unit-magnitude twist; $p, q \in \mathbb{R}^3$ two points; and δ a real number > 0. Find θ such that

$$\|q - e^{\widehat{\xi}\theta} p\| = \delta.$$

   Use this subproblem to solve for the extension of the prismatic joint in the SCARA robot in Example 3.6.

8. Show that the spatial velocity of a manipulator does not depend on the location of the tool frame (as long as it moves with the end-effector).

9. *Singular values of a matrix*
   Let $A : \mathbb{R}^n \to \mathbb{R}^p$ represent a linear map and assume that r is the rank of A. Thus, $r \leq \min(n, p)$. Show that there exist matrices $U \in \mathbb{R}^{p \times p}, V \in \mathbb{R}^{n \times n}$ and $\Sigma \in \mathbb{R}^{p \times n}$ such that

$$A = U \Sigma V^T, \tag{3.74}$$

where

(a) The columns of V are orthonormal, i.e., $V^TV = I$. Further, it may be partitioned as

$$V = \begin{bmatrix} v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \end{bmatrix} = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$$

so that the range space of $A^T : \mathbb{R}^p \to \mathbb{R}^n$, denoted $\mathcal{R}(A^T)$, is spanned by the columns of V_1, and the null space of A, denoted $\eta(A)$, is spanned by the columns of V_2.

(b) The columns of U are orthonormal, i.e., $U^TU = I$, and it may be partitioned as

$$U = \begin{bmatrix} u_1 & \cdots & u_r & u_{r+1} & \cdots & u_p \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$$

so that $\mathcal{R}(A)$ is spanned by the columns of U_1, and $\eta(A^T)$ is spanned by the columns of U_2.

(c) Σ is a matrix of dimension $p \times n$ of the form

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}$$

with

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} \in \mathbb{R}^{r \times r}, \qquad \sigma_1 \geq \cdots \geq \sigma_r > 0$$

The σ_i are called the *singular values* of A, and $\Sigma \in \mathbb{R}^{p \times n}$ is the representation of A in terms of the V basis for $\mathbb{R}^n$ and the U basis for $\mathbb{R}^p$.

10. Let $J(\theta) : \mathbb{R}^n \to \mathbb{R}^6$ be the Jacobian of a manipulator. Show that the manipulability measure $\mu_3(\theta)$ is given by the product of the singular values of $J(\theta)$; that is,

$$\mu_3(\theta) = \prod_{i=1}^{6} \sigma_i(\theta).$$

Thus, $\mu_3(\theta)$ is zero if and only if the Jacobian is singular.

11. Let $A : \mathbb{R}^n \to \mathbb{R}^p$ be of rank r and have singular value decomposition (3.74). Let B_1 denote the ball of unit radius in $\mathbb{R}^n$; that is,

$$B_1 = \{x \in \mathbb{R}^n, \|x\| \leq 1\}.$$

Use the description of the matrices U, V of Exercise 9 to find the map under A of B_1. Distinguish between the cases that $p \geq n$ and $p \leq n$ and also when $r < \min(n, p)$.

12. Let $J(\theta) : \mathbb{R}^n \to \mathbb{R}^p$ be the Jacobian of a manipulator ($p = 3$ or 6). Assume that a task is modeled by an ellipsoid in the task space with its principal axes of length $\alpha_1, \cdots, \alpha_p$. Let $E_\beta \subset \mathbb{R}^p$ be an ellipsoid of size scaled by β, namely

$$E_\beta := \{y : \left(\frac{y_1}{\alpha_1}\right)^2 + \cdots + \left(\frac{y_p}{\alpha_p}\right)^2 \leq \beta\}$$

Define a manipulability measure on $J(\theta)$ which takes into account the task requirement as

$$\mu_t(\theta) := \max\{\beta : J(\theta)(B_1) \subset E_\beta\}.$$

Characterize $\mu_t(\theta)$ in terms of the singular values of $J(\theta)$ and lengths of the principal axes, $\alpha_1, \ldots, \alpha_p$.

13. *Isotropic points*
A point in a manipulator's workspace is said to be *isotropic* if the condition number of the Jacobian is 1.

    (a) Calculate conditions under which a two-link planar manipulator has isotropic points and sketch their location in the plane.

    (b) Compute the isotropic points for an elbow manipulator without a wrist.

    (c) Discuss why isotropic points are useful for tasks which involve applying forces against the environment.

14. Euler angles can be used to represent rotations via the product of exponentials formula. If we think of (α, β, γ) as joints angles of a robot manipulator, then we can find the singularities of an Euler angle parameterization by calculating the Jacobian of the "forward kinematics," where we are concerned only with the rotation portion of the forward kinematics map. Use this point of view to find singularities for the following classes of Euler angles:

    (a) ZYZ Euler angles

    (b) ZYX Euler angles

    (c) XYZ Euler angles

15. *Kinematic singularity: four coplanar revolute joints*
Four revolute joint axes with twists $\xi_i = (q_i \times \omega_i, \omega_i)$, $i = 1, \ldots, 4$, are said to be coplanar if there exists a plane with unit normal n such that:

    (a) Each axis direction is orthogonal to n: $n^T \omega_i = 0$, $i = 1, \ldots, 4$.

(b) The vector from q_i to q_j is orthogonal to n: $n^T(q_i - q_j) = 0$, $i = 1, \ldots, 4$.

Show that when four of its revolute joint axes are coplanar, a six degree of freedom manipulator is at a singular configuration. Give an example of a manipulator exhibiting such a singularity.

16. *Kinematic singularity: six revolute joints intersecting along a line*
Six revolute joint axes with twists $\xi_i = (q_i \times \omega_i, \omega_i)$, $i = 1, \ldots, 6$, intersect along a line $(p \times n, n)$ if there exist constants $\gamma_i, \beta_i \in \mathbb{R}$, $i = 1, ...6$, such that

$$q_i + \gamma_i \omega_i = p + \beta_i n.$$

Show that when the six revolute joint axes of a six degree of freedom manipulator intersect along a line, the manipulator is at a singular configuration.

17. *Kinematic singularity: prismatic joint perpendicular to two parallel coplanar revolute joints*
A prismatic joint with twist $\xi_3 = (v_3, 0)$ is normal to a plane containing two parallel revolute axes $\xi_i = (q_i \times \omega_i, \omega_i)$, $i = 1, 2$, if

(a) $v_3^T \omega_i = 0$, $i = 1, 2$

(b) $v_3^T (q_1 - q_2) = 0$

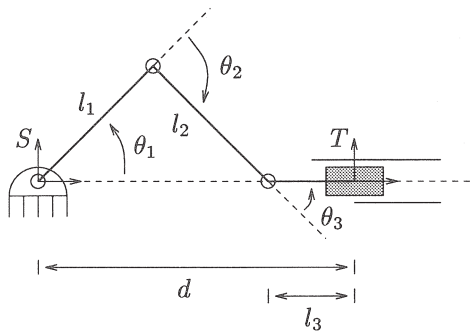
(c) $\omega_1 = \pm \omega_2$

Show that when this occurs, a six degree of freedom manipulator is at a singular configuration. Give an example of a manipulator exhibiting such a singularity.

18. In general, the manipulator Jacobian depends on the choice of base and tool frames. Determine which of the manipulability measures described in Section 4.4 is independent of the choice of base and/or tool frames.

19. Show that if a manipulator is at a singular configuration, then there exists an end-effector wrench F which can be balanced without applying any joint torques. How is the wrench related to the twists which form the columns of the Jacobian?

20. Consider the slider-crank mechanism shown below:



(a) Calculate the number of degrees of freedom of the mechanism. Explain why the spatial version of Gruebler's formula cannot be used.

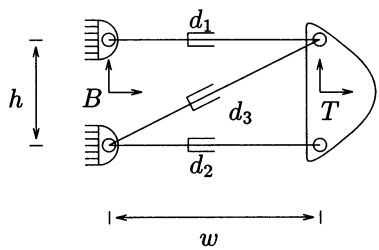
(b) Calculate the structure equations for the mechanism.

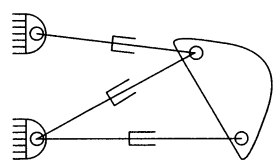
(c) Calculate the Jacobian of the structure equations; give explicit expressions for the instantaneous twists for each of the joints.

(d) Find the singular configurations of the mechanism if d is the active variable.

(e) Find the singular configurations if θ_1 is treated as the active variable. Under what conditions (on l_1, l_2, l_3) do singular configurations exist?

21. The figure below shows a planar parallel manipulator called a "variable geometry truss." Three actuated prismatic joints are used to control the position and orientation of the platform. The revolute joints at the end of each link are passive. Assume that that there are no actuator limits.





Another configuration

(a) Use Gruebler's formula to calculate the number of degrees of freedom of the mechanism.

(b) Write the structure equations for the mechanism. Be sure to clearly define your zero configuration.

(c) Given $g_{st} = ([x, y], R_\phi)$, find explicit expressions for d_1, d_2, and d_3.

(d) Find the spatial Jacobian of the structure equations. Give an explicit answer. Use the fact that some links intersect at a point to minimize extra calculations.

(e) Find the singular configurations of the mechanism. In addition to kinematic singularities, also identify any actuator singularities.

22. *Stewart platform*
Consider the Stewart platform shown in Figure 3.21. Let θ_i represent the displacement of the i^{th} prismatic actuator.

(a) Use Gruebler's formula to compute the number of degrees of freedom of the mechanism.

(b) Show that if $\theta_i > 0$ for all i, then the mechanism is not at a singular configuration.

(c) Suppose that we replace the spherical joints in the Stewart platform with U-joints (a U-joint consists of two orthogonal revolute joints which intersect at a point). Use Gruebler's formula to compute the number of degrees of freedom of the mechanism.

(d) Derive the structure equations for the mechanism in part (c). Are there any singular configurations?

Chapter 4

Robot Dynamics and Control

DOI: 10.1201/9781315136370-4
This chapter has been made available under a CC-BY 4.0 license.

This chapter presents an introduction to the dynamics and control of robot manipulators. We derive the equations of motion for a general open-chain manipulator and, using the structure present in the dynamics, construct control laws for asymptotic tracking of a desired trajectory. In deriving the dynamics, we will make explicit use of twists for representing the kinematics of the manipulator and explore the role that the kinematics play in the equations of motion. We assume some familiarity with dynamics and control of physical systems.

1 Introduction

The kinematic models of robots that we saw in the last chapter describe how the motion of the joints of a robot is related to the motion of the rigid bodies that make up the robot. We implicitly assumed that we could command arbitrary joint level trajectories and that these trajectories would be faithfully executed by the real-world robot. In this chapter, we look more closely at how to execute a given joint trajectory on a robot manipulator.

Most robot manipulators are driven by electric, hydraulic, or pneumatic actuators, which apply torques (or forces, in the case of linear actuators) at the joints of the robot. The *dynamics* of a robot manipulator describes how the robot moves in response to these actuator forces. For simplicity, we will assume that the actuators do not have dynamics of their own and, hence, we can command arbitrary torques at the joints of the robot. This allows us to study the inherent mechanics of robot manipulators without worrying about the details of how the joints are actuated on a particular robot.

We will describe the dynamics of a robot manipulator using a set of nonlinear, second-order, ordinary differential equations which depend on the kinematic and inertial properties of the robot. Although in principle these equations can be derived by summing all of the forces acting on the coupled rigid bodies which form the robot, we shall rely instead on a Lagrangian derivation of the dynamics. This technique has the advantage of requiring only the kinetic and potential energies of the system to be computed, and hence tends to be less prone to error than summing together the inertial, Coriolis, centrifugal, actuator, and other forces acting on the robot's links. It also allows the structural properties of the dynamics to be determined and exploited.

Once the equations of motion for a manipulator are known, the inverse problem can be treated: the *control* of a robot manipulator entails finding actuator forces which cause the manipulator to move along a given trajectory. If we have a perfect model of the dynamics of the manipulator, we can find the proper joint torques directly from this model. In practice, we must design a feedback control law which updates the applied forces in response to deviations from the desired trajectory. Care is required in designing a feedback control law to insure that the overall system converges to the desired trajectory in the presence of initial condition errors, sensor noise, and modeling errors.

In this chapter, we primarily concentrate on one of the simplest robot control problems, that of regulating the position of the robot. There are two basic ways that this problem can be solved. The first, referred to as *joint space control*, involves converting a given task into a desired path for the joints of the robot. A control law is then used to determine joint torques which cause the manipulator to follow the given trajectory. A different approach is to transform the dynamics and control problem into the task space, so that the control law is written in terms of the end-effector position and orientation. We refer to this approach as *workspace control.*

A much harder control problem is one in which the robot is in contact with its environment. In this case, we must regulate not only the position of the end-effector but also the forces it applies against the environment. We discuss this problem briefly in the last section of this chapter and defer a more complete treatment until Chapter 6, after we have introduced the tools necessary to study constrained systems.

2 Lagrange's Equations

There are many methods for generating the dynamic equations of a mechanical system. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation or analysis. We will use a Lagrangian analysis for our derivation, which

relies on the energy properties of mechanical systems to compute the equations of motion. The resulting equations can be computed in closed form, allowing detailed analysis of the properties of the system.

2.1 Basic formulation

Consider a system of n particles which obeys Newton's second law—the time rate of change of a particle's momentum is equal to the force applied to a particle. If we let F_i be the applied force on the i^{th} particle, m_i be the particle's mass, and r_i be its position, then Newton's law becomes

$$F_i = m_i \ddot{r}_i \qquad r_i \in \mathbb{R}^3, i = 1, \ldots, n. \tag{4.1}$$

Our interest is not in a set of independent particles, but rather in particles which are attached to one another and have limited degrees of freedom. To describe this interconnection, we introduce *constraints* between the positions of our particles. Each constraint is represented by a function $g_j \colon \mathbb{R}^{3n} \to \mathbb{R}$ such that

$$g_j(r_1, \ldots, r_n) = 0 \qquad j = 1, \ldots, k. \tag{4.2}$$

A constraint which can be written in this form, as an algebraic relationship between the positions of the particles, is called a *holonomic* constraint. More general constraints between rigid bodies—involving $\dot{r}_i$—can also occur, as we shall discover when we study multifingered hands.

A constraint acts on a system of particles through application of *constraint forces*. The constraint forces are determined in such a way that the constraint in equation (4.2) is always satisfied. If we view the constraint as a smooth surface in $\mathbb{R}^n$, the constraint forces are normal to this surface and restrict the velocity of the system to be tangent to the surface at all times. Thus, we can rewrite our system dynamics as a vector equation

$$F = \begin{bmatrix} m_1 I & & 0 \\ & \ddots & \\ 0 & & m_n I \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \vdots \\ \ddot{r}_n \end{bmatrix} + \sum_{j=1}^{k} \Gamma_j \lambda_j, \tag{4.3}$$

where the vectors $\Gamma_1, \ldots, \Gamma_k \in \mathbb{R}^{3n}$ are a basis for the forces of constraint and λ_j is the scale factor for the j^{th} basis element. We do not require that $\Gamma_1, \ldots, \Gamma_k$ be orthonormal. For constraints of the form in equation (4.2), Γ_j can be taken as the gradient of g_j, which is perpendicular to the level set $g_j(r) = 0$.

The scalars $\lambda_1, \ldots, \lambda_k$ are called *Lagrange multipliers*. Formally, we determine the Lagrange multipliers by solving the $3n + k$ equations in equations (4.2) and (4.3) for the $3n + k$ variables $r \in \mathbb{R}^{3n}$ and $\lambda \in \mathbb{R}^k$. The λ_i values only give the relative magnitudes of the constraint forces since the vectors Γ_j are not necessarily orthonormal.

This approach to dealing with constraints is intuitively simple but computationally complex, since we must keep track of the state of all particles in the system even though they are not capable of independent motion. A more appealing approach is to describe the motion of the system in terms of a smaller set of variables that completely describes the configuration of the system. For a system of n particles with k constraints, we seek a set of $m = 3n - k$ variables $q_1, \ldots, q_m$ and smooth functions $f_1, \ldots, f_n$ such that

$$\begin{array}{c} r_i = f_i(q_1, \ldots, q_m) \\ i = 1, \ldots, n \end{array} \iff \begin{array}{c} g_j(r_1, \ldots, r_n) = 0 \\ j = 1, \ldots, k. \end{array} \tag{4.4}$$

We call the q_i's a set of *generalized coordinates* for the system. For a robot manipulator consisting of rigid links, these generalized coordinates are almost always chosen to be the angles of the joints. The specification of these angles uniquely determines the position of all of the particles which make up the robot.

Since the values of the generalized coordinates are sufficient to specify the position of the particles, we can rewrite the equations of motion for the system in terms of the generalized coordinates. To do so, we also express the external forces applied to the system in terms of components along the generalized coordinates. We call these forces *generalized forces* to distinguish them from physical forces, which are always represented as vectors in $\mathbb{R}^3$. For a robot manipulator with joint angles acting as generalized coordinates, the generalized forces are the torques applied about the joint axes.

To write the equations of motion, we define the *Lagrangian*, L, as the difference between the kinetic and potential energy of the system. Thus,

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q),$$

where T is the kinetic energy and V is the potential energy of the system, both written in generalized coordinates.

Theorem 4.1. Lagrange's equations
The equations of motion for a mechanical system with generalized coordinates $q \in \mathbb{R}^m$ and Lagrangian L are given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i \qquad i = 1, \ldots, m, \tag{4.5}$$

where Υ_i is the external force acting on the i^{th} generalized coordinate.

The equations in (4.5) are called *Lagrange's equations*. We will often write them in vector form as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \Upsilon,$$

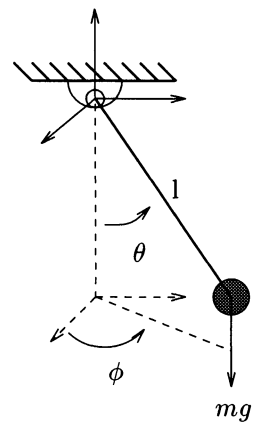


Figure 4.1: Idealized spherical pendulum. The configuration of the system is described by the angles θ and ϕ.

where $\frac{\partial L}{\partial \dot{q}}$, $\frac{\partial L}{\partial q}$, and Υ are to be formally regarded as row vectors, though we often write them as column vectors for notational convenience. A proof of Theorem 4.1 can be found in most books on dynamics of mechanical systems (e.g., [100]).

Lagrange's equations are an elegant formulation of the dynamics of a mechanical system. They reduce the number of equations needed to describe the motion of the system from n, the number of particles in the system, to m, the number of generalized coordinates. Note that if there are *no* constraints, then we can choose q to be the components of r, giving $T = \frac{1}{2}\sum m_i \|\dot{r}_i^2\|$, and equation (4.5) then reduces to equation (4.1). In fact, rearranging equation (4.5) as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} + \Upsilon$$

is just a restatement of Newton's law in generalized coordinates:

$$\frac{d}{dt}(\text{momentum}) = \text{applied force}.$$

The motion of the individual particles can be recovered through application of equation (4.4).

Example 4.1. Dynamics of a spherical pendulum
Consider an idealized spherical pendulum as shown in Figure 4.1. The system consists of a point with mass m attached to a spherical joint by a massless rod of length l. We parameterize the configuration of the point mass by two scalars, θ and ϕ, which measure the angular displacement from the z- and x-axes, respectively. We wish to solve for the motion of the mass under the influence of gravity.

We begin by deriving the Lagrangian for the system. The position of the mass, relative to the origin at the base of the pendulum, is given by

$$r(\theta, \phi) = \begin{bmatrix} l \sin\theta \cos\phi \\ l \sin\theta \sin\phi \\ -l \cos\theta \end{bmatrix}. \tag{4.6}$$

The kinetic energy is

$$T = \frac{1}{2} m l^2 \|\dot{r}\|^2 = \frac{1}{2} m l^2 \left(\dot{\theta}^2 + (1 - \cos^2\theta) \dot{\phi}^2 \right)$$

and the potential energy is

$$V = -mgl \cos\theta,$$

where $g \approx 9.8$ m/sec^2 is the gravitational constant. Thus, the Lagrangian is given by

$$L(q, \dot{q}) = \frac{1}{2} m l^2 \left(\dot{\theta}^2 + (1 - \cos^2\theta) \dot{\phi}^2 \right) + mgl \cos\theta,$$

where $q = (\theta, \phi)$.

Substituting L into Lagrange's equations gives

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} &= \frac{d}{dt} \left(m l^2 \dot{\theta} \right) = m l^2 \ddot{\theta} \\ \frac{\partial L}{\partial \theta} &= m l^2 \sin\theta \cos\theta \, \dot{\phi}^2 - mgl \sin\theta \\ \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} &= \frac{d}{dt} \left(m l^2 \sin^2\theta \, \dot{\phi} \right) = m l^2 \sin^2\theta \, \ddot{\phi} + 2 m l^2 \sin\theta \cos\theta \, \dot{\theta} \dot{\phi} \\ \frac{\partial L}{\partial \phi} &= 0 \end{aligned}$$

and the overall dynamics satisfy

$$\begin{bmatrix} ml^2 & 0 \\ 0 & ml^2 \sin^2\theta \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} -ml^2 \sin\theta \cos\theta \, \dot{\phi}^2 \\ 2 ml^2 \sin\theta \cos\theta \, \dot{\theta} \dot{\phi} \end{bmatrix} + \begin{bmatrix} mgl \sin\theta \\ 0 \end{bmatrix} = 0. \tag{4.7}$$

Given the initial position and velocity of the point mass, equation (4.7) uniquely determines the subsequent motion of the system. The motion of the mass in $\mathbb{R}^3$ can be retrieved from equation (4.6).

2.2 Inertial properties of rigid bodies

To apply Lagrange's equations to a robot, we must calculate the kinetic and potential energy of the robot links as a function of the joint angles

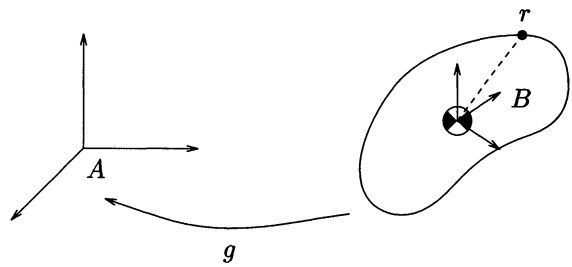


Figure 4.2: Coordinate frames for calculating the kinetic energy of a moving rigid body.

and velocities. This, in turn, requires that we have a model for the mass distribution of the links. Since each link is a rigid body, its kinetic and potential energy can be defined in terms of its total mass and its moments of inertia about the center of mass.

Let $V \subset \mathbb{R}^3$ be the volume occupied by a rigid body, and $\rho(r)$, $r \in V$ be the mass distribution of the body. If the object is made from a homogeneous material, then $\rho(r) = \rho$, a constant. The mass of the body is the volume integral of the mass density:

$$m = \int_V \rho(r)\, dV.$$

The center of mass of the body is the weighted average of the density:

$$\bar{r} = \frac{1}{m} \int_V \rho(r) r\, dV.$$

Consider the rigid object shown in Figure 4.2. We compute the kinetic energy as follows: fix the body frame at the mass center of the object and let (p, R) be a trajectory of the object relative to an inertial frame, where we have dropped all subscripts to simplify notation. Let $r \in \mathbb{R}^3$ be the coordinates of a body point relative to the body frame. The velocity of the point in the inertial frame is given by

$$\dot{p} + \dot{R}\, r$$

and the kinetic energy of the object is given by the following volume integral:

$$T = \frac{1}{2} \int_V \rho(r) \|\dot{p} + \dot{R}r\|^2\, dV. \tag{4.8}$$

Expanding the product in the kinetic energy integral yields

$$T = \frac{1}{2} \int_V \rho(r) \left(\|\dot{p}\|^2 + 2\dot{p}^T \dot{R} r + \|\dot{R}r\|^2 \right) dV.$$

The first term of the above expression gives the translational kinetic energy. The second term vanishes because the body frame is placed at the mass center of the object and

$$\int_V \rho(r)(\dot{p}^T \dot{R})r \, dV = (\dot{p}^T \dot{R}) \int_V \rho(r) r \, dV = 0.$$

The last term can be simplified using properties of rotation and skew-symmetric matrices:

$$\begin{aligned}
\frac{1}{2}\int_V \rho(r)(\dot{R}r)^T(\dot{R}r)\, dV &= \frac{1}{2}\int_V \rho(r)(R\widehat{\omega}r)^T(R\widehat{\omega}r)\, dV \\
&= \frac{1}{2}\int_V \rho(r)(\widehat{r}\omega)^T(\widehat{r}\omega)\, dV \\
&= \frac{1}{2}\omega^T\left(\int_V \rho(r)\widehat{r}^T\widehat{r}dV\right)\omega =: \frac{1}{2}\omega^T\mathcal{I}\omega,
\end{aligned}$$

where $\omega \in \mathbb{R}^3$ is the *body* angular velocity. The symmetric matrix $\mathcal{I} \in \mathbb{R}^{3\times 3}$ defined by

$$\mathcal{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = -\int_V \rho(r)\widehat{r}^2\, dV$$

is called the *inertia tensor* of the object expressed in the body frame. It has entries

$$\begin{aligned}
I_{xx} &= \int_V \rho(r)(y^2 + z^2)\, dx\, dy\, dz \\
I_{xy} &= -\int_V \rho(r)(xy)\, dx\, dy\, dz,
\end{aligned}$$

and the other entries are defined similarly.

The total kinetic energy of the object can now be written as the sum of a translational component and a rotational component,

$$\begin{aligned}
T &= \frac{1}{2}m\|\dot{p}\|^2 + \frac{1}{2}\omega^T\mathcal{I}\omega \\
&= \frac{1}{2}(V^b)^T\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix}V^b =: \frac{1}{2}(V^b)^T\mathcal{M}V^b,
\end{aligned} \tag{4.9}$$

where $\widehat{V}^b = g^{-1}\dot{g} \in se(3)$ is the body velocity, and $\mathcal{M}$ is called the *generalized inertia matrix* of the object, expressed in the body frame. The matrix $\mathcal{M}$ is symmetric and positive definite.

Example 4.2. Generalized inertia matrix for a homogeneous bar Consider a homogeneous rectangular bar with mass m, length l, width w, and height h, as shown in Figure 4.3. The mass density of the bar is

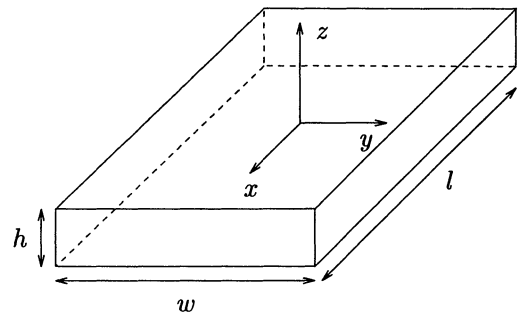


Figure 4.3: A homogeneous rectangular bar.

$\rho = \frac{m}{lwh}$. We attach a coordinate frame at the center of mass of the bar, with the coordinate axes aligned with the principal axes of the bar.

The inertia tensor is evaluated using the previous formula:

$$\begin{aligned} I_{xx} &= \int_V \frac{m}{lwh}\left(y^2+z^2\right)\,dV = \frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\int_{-l/2}^{l/2}\left(y^2+z^2\right)\,dx\,dy\,dz \\ &= \frac{m}{lwh}\left(\frac{1}{12}\left(lw^3h+lwh^3\right)\right) = \frac{m}{12}(w^2+h^2), \\ I_{xy} &= -\int_V \frac{m}{lwh}(xy)\,dV = -\frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\int_{-l/2}^{l/2}(xy)\,dx\,dy\,dz \\ &= -\frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\left(\frac{1}{2}x^2y\Big|_{-l/2}^{l/2}\right)\,dy\,dz = 0. \end{aligned}$$

The other entries are calculated in the same manner and we have:

$$\mathcal{I} = \begin{bmatrix} \frac{m}{12}(w^2+h^2) & 0 & 0 \\ 0 & \frac{m}{12}(l^2+h^2) & 0 \\ 0 & 0 & \frac{m}{12}(l^2+w^2) \end{bmatrix}.$$

The inertia tensor is diagonal by virtue of the fact that we aligned the coordinate axes with the principal axes of the box.

The generalized inertia matrix is given by

$$\mathcal{M} = \begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{m}{12}(w^2+h^2) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{m}{12}(l^2+h^2) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{m}{12}(l^2+w^2) \end{bmatrix}.$$

The block diagonal structure of this matrix relies on attaching the body coordinate frame at center of mass (see Exercise 3).

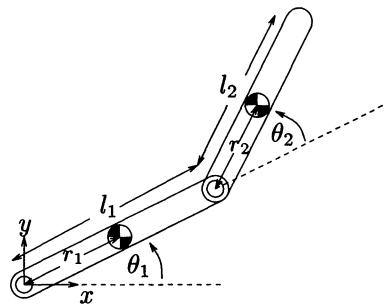


Figure 4.4: Two-link planar manipulator.

2.3 Example: Dynamics of a two-link planar robot

To illustrate how Lagrange's equations apply to a simple robotic system, consider the two-link planar manipulator shown in Figure 4.4. Model each link as a homogeneous rectangular bar with mass m_i and moment of inertia tensor

$$\mathcal{I}_i = \begin{bmatrix} I_{xi} & 0 & 0 \\ 0 & I_{yi} & 0 \\ 0 & 0 & I_{zi} \end{bmatrix}$$

relative to a frame attached at the center of mass of the link and aligned with the principle axes of the bar. Letting $v_i \in \mathbb{R}^3$ be the translational velocity of the center of mass for the i^{th} link and $\omega_i \in \mathbb{R}^3$ be the angular velocity, the kinetic energy of the manipulator is

$$T(\theta, \dot{\theta}) = \frac{1}{2} m_1 \|v_1\|^2 + \frac{1}{2} \omega_1^T \mathcal{I}_1 \omega_1 + \frac{1}{2} m_2 \|v_2\|^2 + \frac{1}{2} \omega_2^T \mathcal{I}_2 \omega_2.$$

Since the motion of the manipulator is restricted to the xy plane, $\|v_i\|$ is the magnitude of the xy velocity of the center of mass and ω_i is a vector in the direction of the z-axis, with $\|\omega_1\| = \dot{\theta}_1$ and $\|\omega_2\| = \dot{\theta}_1 + \dot{\theta}_2$.

We solve for the kinetic energy in terms of the generalized coordinates by using the kinematics of the mechanism. Let $p_i = (x_i, y_i, 0)$ denote the position of the i^{th} center of mass. Letting r_1 and r_2 be the distance from the joints to the center of mass for each link, as shown in the figure, we have

$$\begin{aligned}
\bar{x}_1 &= r_1 c_1 & \dot{\bar{x}}_1 &= -r_1 s_1 \dot{\theta}_1 \\
\bar{y}_1 &= r_1 s_1 & \dot{\bar{y}}_1 &= r_1 c_1 \dot{\theta}_1 \\
\bar{x}_2 &= l_1 c_1 + r_2 c_{12} & \dot{\bar{x}}_2 &= -(l_1 s_1 + r_2 s_{12})\dot{\theta}_1 - r_2 s_{12} \dot{\theta}_2 \\
\bar{y}_2 &= l_1 s_1 + r_2 s_{12} & \dot{\bar{y}}_2 &= (l_1 c_1 + r_2 c_{12})\dot{\theta}_1 + r_2 c_{12} \dot{\theta}_2,
\end{aligned}$$

where $s_i = \sin\theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$, and similarly for c_i and c_{ij}. The

kinetic energy becomes

$$T(\theta,\dot{\theta}) = \frac{1}{2}m_1(\dot{\tilde{x}}_1^2 + \dot{\tilde{y}}_1^2) + \frac{1}{2}\mathcal{I}_{z1}\dot{\theta}_1^2 + \frac{1}{2}m_2(\dot{\tilde{x}}_2^2 + \dot{\tilde{y}}_2^2) + \frac{1}{2}\mathcal{I}_{z2}(\dot{\theta}_1+\dot{\theta}_2)^2$$
$$= \frac{1}{2}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}^T \begin{bmatrix}\alpha+2\beta c_2 & \delta+\beta c_2\\ \delta+\beta c_2 & \delta\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}, \tag{4.10}$$

where

$$\alpha = \mathcal{I}_{z1} + \mathcal{I}_{z2} + m_1 r_1^2 + m_2(l_1^2 + r_2^2)$$
$$\beta = m_2 l_1 r_2$$
$$\delta = \mathcal{I}_{z2} + m_2 r_2^2.$$

Finally, we can substitute the Lagrangian $L = T$ into Lagrange's equations to obtain (after some calculation)

$$\begin{bmatrix}\alpha+2\beta c_2 & \delta+\beta c_2\\ \delta+\beta c_2 & \delta\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix}-\beta s_2\dot{\theta}_2 & -\beta s_2(\dot{\theta}_1+\dot{\theta}_2)\\ \beta s_2\dot{\theta}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}. \tag{4.11}$$

The first term in this equation represents the inertial forces due to acceleration of the joints, the second represents the Coriolis and centrifugal forces, and the right-hand side is the applied torques.

2.4 Newton-Euler equations for a rigid body

Lagrange's equations provide a very general method for deriving the equations of motion for a mechanical system. However, implicit in the derivation of Lagrange's equations is the assumption that the configuration space of the system can be parameterized by a subset of $\mathbb{R}^n$, where n is the number of degrees of freedom of the system. For a rigid body with configuration $g \in SE(3)$, Lagrange's equations cannot be directly used to determine the equations of motion unless we choose a local parameterization for the configuration space (for example, using Euler angles to parameterize the orientation of the rigid body). Since all parameterizations of $SE(3)$ are singular at some configuration, such a derivation can only hold locally.

In this section, we give a global characterization of the dynamics of a rigid body subject to external forces and torques. We begin by reviewing the standard derivation of the equations of rigid body motion and then examine the dynamics in terms of twists and wrenches.

Let $g = (p, R) \in SE(3)$ be the configuration of a coordinate frame attached to the center of mass of a rigid body, relative to an inertial frame. Let f represent a force applied at the center of mass, with the coordinates of f specified relative to the inertial frame. The translational

equations of motion are given by Newton's law, which can written in terms of the linear momentum $m\dot{p}$ as

$$f = \frac{d}{dt}(m\dot{p}).$$

Since the mass of the rigid body is constant, the translational motion of the center of mass becomes

$$f = m\ddot{p}. \tag{4.12}$$

These equations are independent of the angular motion of the rigid body because we have used the center of mass to represent the position of the body.

Similarly, the equations describing angular motion can be derived independently of the linear motion of the system. Consider the rotational motion of a rigid body about a point, subject to an externally applied torque τ. To derive the equations of motion, we equate the change in angular momentum to the applied torque. The angular momentum relative to an inertial frame is given by $\mathcal{I}'\omega^s$, where

$$\mathcal{I}' = R\mathcal{I}R^T$$

is the instantaneous inertia tensor relative to the inertial frame and ω^s is the spatial angular velocity. The angular equations of motion become

$$\tau = \frac{d}{dt}(\mathcal{I}'\omega^s) = \frac{d}{dt}(R\mathcal{I}R^T\omega^s),$$

where $\tau \in \mathbb{R}^3$ is specified relative to the inertial frame. Expanding the right-hand side of this equation, we have

$$\begin{aligned}\tau &= R\mathcal{I}R^T\dot{\omega}^s + \dot{R}\mathcal{I}R^T\omega^s + R\mathcal{I}\dot{R}^T\omega^s \\ &= \mathcal{I}'\dot{\omega}^s + \dot{R}R^T\mathcal{I}'\omega^s + \mathcal{I}'R\dot{R}^T\omega^s \\ &= \mathcal{I}'\dot{\omega}^s + \omega^s \times \mathcal{I}'\omega^s - \mathcal{I}'\omega^s \times \omega^s,\end{aligned}$$

where the last equation follows by differentiating the identity $RR^T = I$ and using the definition of ω^s. The last term of this equation is zero, and hence the dynamics are given by

$$\mathcal{I}'\dot{\omega}^s + \omega^s \times \mathcal{I}'\omega^s = \tau. \tag{4.13}$$

Equation (4.13) is called *Euler's equation.*

Equations (4.12) and (4.13) describe the dynamics of a rigid body in terms of a force and torque applied at the center of mass of the object. However, the coordinates of the force and torque vectors are not written relative to a body-fixed frame attached at the center of mass, but rather

with respect to an inertial frame. Thus the pair $(f, \tau) \in \mathbb{R}^6$ is not the wrench applied to the rigid body, as defined in Chapter 2, since the point of application is not the origin of the inertial coordinate frame. Similarly, the velocity pair $(\dot{p}, \omega^s)$ does not correspond to the spatial or body velocity, since $\dot{p}$ is not the correct expression for the linear velocity term in either body or spatial coordinates.

In order to express the dynamics in terms of twists and wrenches, we rewrite Newton's equation using the body velocity $v^b = R^T \dot{p}$ and body force $f^b = R^T f$. Expanding the right-hand side of equation (4.12),

$$\frac{d}{dt}(m\dot{p}) = \frac{d}{dt}(mRv^b) = Rm\dot{v}^b + \dot{R}mv^b,$$

and pre-multiplying by R^T, the translational dynamics become

$$m\dot{v}^b + \omega^b \times mv^b = f^b. \tag{4.14}$$

Equation (4.14) is Newton's law written in body coordinates.

Similarly, we can write Euler's equation in terms of the body angular velocity $\omega^b = R^T \omega^s$ and the body torque $\tau^b = R^T \tau$. A straightforward computation shows that

$$\mathcal{I}\dot{\omega}^b + \omega^b \times \mathcal{I}\omega^b = \tau^b. \tag{4.15}$$

Equation (4.15) is Euler's equation, written in body coordinates. Note that in body coordinates the inertia tensor is constant and hence we use $\mathcal{I}$ instead of $\mathcal{I}' = R\mathcal{I}R^T$.

Combining equations (4.14) and (4.15) gives the equations of motion for a rigid body subject to an external wrench F applied at the center of mass and specified with respect to the body coordinate frame:

$$\boxed{\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b} \tag{4.16}$$

This equation is called the *Newton-Euler equation* in body coordinates. It gives a global description of the equations of motion for a rigid body subject to an external wrench. Note that the linear and angular motions are coupled since the linear velocity in body coordinates depends on the current orientation.

It is also possible to write the Newton-Euler equations relative to a spatial coordinate frame. This version is explored in Exercises 4 and 5. Once again the equations for linear and angular motion are coupled, so that the translational motion still depends on the rotational motion. In this book we shall always write the Newton-Euler equations in body coordinates, as in equation (4.16).

3 Dynamics of Open-Chain Manipulators

We now derive the equations of motion for an open-chain robot manipulator. We shall use the kinematics formulation presented in the previous chapter to write the Lagrangian for the robot in terms of the joint angles and joint velocities. Using this form of the dynamics, we explore several fundamental properties of robot manipulators which are of importance when proving the stability of robot control laws.

3.1 The Lagrangian for an open-chain robot

To calculate the kinetic energy of an open-chain robot manipulator with n joints, we sum the kinetic energy of each link. For this we define a coordinate frame, L_i, attached to the center of mass of the i^{th} link. Let

$$g_{sl_i}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0)$$

represent the configuration of the frame L_i relative to the base frame of the robot, S. The body velocity of the center of mass of the i^{th} link is given by

$$V^b_{sl_i} = J^b_{sl_i}(\theta)\dot{\theta},$$

where $J^b_{sl_i}$ is the body Jacobian corresponding to g_{sl_i}. $J^b_{sl_i}$ has the form

$$J^b_{sl_i}(\theta) = \begin{bmatrix} \xi_1^\dagger & \cdots & \xi_i^\dagger & 0 & \cdots & 0 \end{bmatrix},$$

where

$$\xi_j^\dagger = \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_j\theta_j} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0)\right)} \xi_j \qquad j \leq i$$

is the j^{th} instantaneous joint twist relative to the i^{th} link frame. To streamline notation, we write $J^b_{sl_i}$ as J_i for the remainder of this section.

The kinetic energy of the i^{th} link is

$$T_i(\theta,\dot{\theta}) = \frac{1}{2}(V^b_{sl_i})^T \mathcal{M}_i V^b_{sl_i} = \frac{1}{2}\dot{\theta}^T J_i^T(\theta)\mathcal{M}_i J_i(\theta)\dot{\theta}, \tag{4.17}$$

where $\mathcal{M}_i$ is the generalized inertia matrix for the i^{th} link. Now the total kinetic energy can be written as

$$T(\theta,\dot{\theta}) = \sum_{i=1}^{n} T_i(\theta,\dot{\theta}) =: \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}. \tag{4.18}$$

The matrix $M(\theta) \in \mathbb{R}^{n\times n}$ is the *manipulator inertia matrix*. In terms of the link Jacobians, J_i, the manipulator inertia matrix is defined as

$$M(\theta) = \sum_{i=1}^{n} J_i^T(\theta)\mathcal{M}_i J_i(\theta). \tag{4.19}$$

To complete our derivation of the Lagrangian, we must calculate the potential energy of the manipulator. Let $h_i(\theta)$ be the height of the center of mass of the i^{th} link (height is the component of the position of the center of mass opposite the direction of gravity). The potential energy for the i^{th} link is

$$V_i(\theta) = m_i g h_i(\theta),$$

where m_i is the mass of the i^{th} link and g is the gravitational constant. The total potential energy is given by the sum of the contributions from each link:

$$V(\theta) = \sum_{i=1}^{n} V_i(\theta) = \sum_{i=1}^{n} m_i g h_i(\theta).$$

Combining this with the kinetic energy, we have

$$L(\theta, \dot{\theta}) = \sum_{i=1}^{n} \Big(T_i(\theta, \dot{\theta}) - V_i(\theta) \Big) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} - V(\theta).$$

3.2 Equations of motion for an open-chain manipulator

Let $\theta \in \mathbb{R}^n$ be the joint angles for an open-chain manipulator. The Lagrangian is of the form

$$L(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} - V(\theta),$$

where $M(\theta)$ is the manipulator inertia matrix and $V(\theta)$ is the potential energy due to gravity. It will be convenient to express the kinetic energy as a sum,

$$L(\theta, \dot{\theta}) = \frac{1}{2} \sum_{i,j=1}^{n} M_{ij}(\theta) \dot{\theta}_i \dot{\theta}_j - V(\theta). \tag{4.20}$$

The equations of motion are given by substituting into Lagrange's equations,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = \Upsilon_i,$$

where we let Υ_i represent the actuator torque and other nonconservative, generalized forces acting on the i^{th} joint. Using equation (4.20), we have

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} = \frac{d}{dt} \Big(\sum_{j=1}^{n} M_{ij} \dot{\theta}_j \Big) = \sum_{j=1}^{n} \Big(M_{ij} \ddot{\theta}_j + \dot{M}_{ij} \dot{\theta}_j \Big)$$

$$\frac{\partial L}{\partial \theta_i} = \frac{1}{2} \sum_{j,k=1}^{n} \frac{\partial M_{kj}}{\partial \theta_i} \dot{\theta}_k \dot{\theta}_j - \frac{\partial V}{\partial \theta_i}.$$

The $\dot{M}_{ij}$ term can now be expanded in terms of partial derivatives to yield

$$\sum_{j=1}^{n} M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} \left(\frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_j\dot{\theta}_k - \frac{1}{2}\frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j \right) + \frac{\partial V}{\partial \theta_i}(\theta) = \Upsilon_i \qquad i = 1, \ldots, n.$$

Rearranging terms, we can write

$$\sum_{j=1}^{n} M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k + \frac{\partial V}{\partial \theta_i}(\theta) = \Upsilon_i \qquad i = 1, \ldots, n, \tag{4.21}$$

where Γ_{ijk} is given by

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}(\theta)}{\partial \theta_k} + \frac{\partial M_{ik}(\theta)}{\partial \theta_j} - \frac{\partial M_{kj}(\theta)}{\partial \theta_i} \right). \tag{4.22}$$

Equation (4.21) is a second-order differential equation in terms of the manipulator joint variables. It consists of four pieces: inertial forces, which depend on the acceleration of the joints; centrifugal and Coriolis forces, which are quadratic in the joint velocities; potential forces, of the form $\frac{\partial V}{\partial \theta_i}$; and external forces, Υ_i.

The centrifugal and Coriolis terms arise because of the non-inertial frames which are implicit in the use of generalized coordinates. In the classical mechanics literature, one identifies terms of the form $\dot{\theta}_i\dot{\theta}_j$, $i \neq j$ as Coriolis forces and terms of the form $\dot{\theta}_i^2$ as centrifugal forces. The functions Γ_{ijk} are called the *Christoffel symbols* corresponding to the inertia matrix $M(\theta)$.

The external forces can be divided into two components. Let τ_i represent the applied torque at the joint and define $-N_i(\theta, \dot{\theta})$ to be any other forces which act on the i^{th} generalized coordinate, including conservative forces arising from a potential as well as frictional forces. (The reason for the negative sign in the definition of N_i will become apparent in a moment.) As an example, if the manipulator has viscous friction at the joints, then N_i would be defined as

$$-N_i(\theta, \dot{\theta}) = -\frac{\partial V}{\partial \theta_i} - \beta\dot{\theta}_i,$$

where β is the damping coefficient. Other forces acting on the manipulator, such as forces applied at the end-effector, can also be included by reflecting them to the joints (via the transpose of the appropriate Jacobian).

In order to put the equations of motion back into vector form, we define the matrix $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$ as

$$C_{ij}(\theta, \dot{\theta}) = \sum_{k=1}^{n} \Gamma_{ijk}\dot{\theta}_k = \frac{1}{2}\sum_{k=1}^{n} \left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i} \right)\dot{\theta}_k. \tag{4.23}$$

We call the matrix C the *Coriolis matrix* for the manipulator; the vector $C(\theta,\dot{\theta})\dot{\theta}$ gives the Coriolis and centrifugal force terms in the equations of motion. Note that there are other ways to define the matrix $C(\theta,\dot{\theta})$ such that $C_{ij}(\theta,\dot{\theta})\dot{\theta}_j = \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k$. However, this particular choice has important properties which we shall later exploit.

Equation (4.21) can now be rewritten as

$$\boxed{M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau} \tag{4.24}$$

where τ is the vector of actuator torques and $N(\theta,\dot{\theta})$ includes gravity terms and other forces which act at the joints. This is a second-order vector differential equation for the motion of the manipulator as a function of the applied joint torques. The matrices M and C, which summarize the inertial properties of the manipulator, have some important properties which we shall use in the sequel:

Lemma 4.2. Structural properties of the robot equations of motion

Equation (4.24) *satisfies the following properties:*

1. $M(\theta)$ *is symmetric and positive definite.*
2. $\dot{M} - 2C \in \mathbb{R}^{n\times n}$ *is a skew-symmetric matrix.*

Proof. Positive definiteness of the inertia matrix follows directly from its definition and the fact that the kinetic energy of the manipulator is zero only if the system is at rest. To show property 2, we calculate the components of the matrix $\dot{M} - 2C$:

$$\begin{aligned}(\dot{M} - 2C)_{ij} &= \dot{M}_{ij}(\theta) - 2C_{ij}(\theta) \\ &= \sum_{k=1}^{n} \frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k - \frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k - \frac{\partial M_{ik}}{\partial \theta_j}\dot{\theta}_k + \frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k \\ &= \sum_{k=1}^{n} \frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k - \frac{\partial M_{ik}}{\partial \theta_j}\dot{\theta}_k.\end{aligned}$$

Switching i and j shows $(\dot{M} - 2C)^T = -(\dot{M} - 2C)$. Note that the skew-symmetry property depends upon the particular definition of C given in equation (4.23). □

Property 2 is often referred to as the *passivity* property since it implies, among other things, that in the absence of friction the net energy of the robot system is conserved (see Exercise 8). The passivity property is important in the proof of many control laws for robot manipulators.

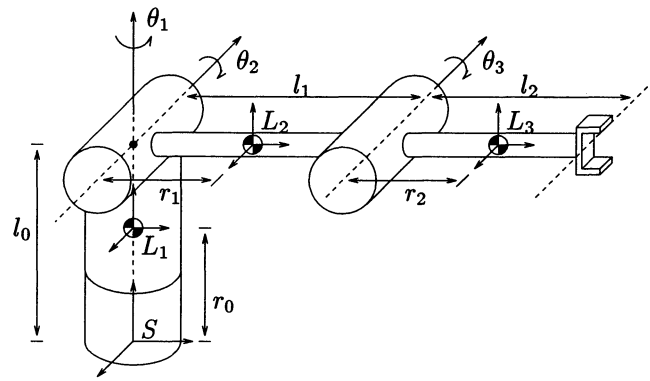


Figure 4.5: Three-link, open-chain manipulator.

Example 4.3. Dynamics of a three-link manipulator
To illustrate the formulation presented above, we calculate the dynamics of the three-link manipulator shown in Figure 4.5. The joint twists were computed in Chapter 3 (for the elbow manipulator) and are given by

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} 0 \\ -l_0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \qquad \xi_3 = \begin{bmatrix} 0 \\ -l_0 \\ l_1 \\ -1 \\ 0 \\ 0 \end{bmatrix}.$$

To each link we attach a frame L_i at the center of mass and aligned with principle inertia axes of the link, as shown in the figure:

$$g_{sl_1}(0) = \begin{bmatrix} I & \begin{pmatrix} 0 \\ 0 \\ r_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix} \quad g_{sl_2}(0) = \begin{bmatrix} I & \begin{pmatrix} 0 \\ r_1 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix} \quad g_{sl_3}(0) = \begin{bmatrix} I & \begin{pmatrix} 0 \\ l_1+r_2 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix}.$$

With this choice of link frames, the link inertia matrices have the general form

$$\mathcal{M}_i = \left[\begin{array}{ccc|ccc} m_i & & & & & \\ & m_i & & & 0 & \\ & & m_i & & & \\ \hline & & & I_{xi} & & \\ & 0 & & & I_{yi} & \\ & & & & & I_{zi} \end{array}\right],$$

where m_i is the mass of the object and I_{xi}, I_{yi}, and I_{zi} are the moments of inertia about the x-, y-, and z-axes of the i^{th} link frame.

To compute the manipulator inertia matrix, we first compute the body Jacobians corresponding to each link frame. A detailed, but straightfor-

ward, calculation yields

$$J_1 = J^b_{sl_1(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad J_2 = J^b_{sl_2(0)} = \begin{bmatrix} -r_1c_2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -r_1 & 0 \\ 0 & -1 & 0 \\ -s_2 & 0 & 0 \\ c_2 & 0 & 0 \end{bmatrix}$$

$$J_3 = J^b_{sl_3(0)} = \begin{bmatrix} -l_2c_2 - r_2c_{23} & 0 & 0 \\ 0 & l_1s_3 & 0 \\ 0 & -r_2 - l_1c_3 & -r_2 \\ 0 & -1 & -1 \\ -s_{23} & 0 & 0 \\ c_{23} & 0 & 0 \end{bmatrix}.$$

The inertia matrix for the system is given by

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} = J_1^T \mathcal{M}_1 J_1 + J_2^T \mathcal{M}_2 J_2 + J_3^T \mathcal{M}_3 J_3.$$

The components of M are given by

$$\begin{aligned} M_{11} &= I_{y2}s_2^2 + I_{y3}s_{23}^2 + I_{z1} + I_{z2}c_2^2 + I_{z3}c_{23}^2 \\ &\quad + m_2r_1^2c_2^2 + m_3(l_1c_2 + r_2c_{23})^2 \\ M_{12} &= 0 \\ M_{13} &= 0 \\ \\ M_{21} &= 0 \\ M_{22} &= I_{x2} + I_{x3} + m_3l_1^2 + m_2r_1^2 + m_3r_2^2 + 2m_3l_1r_2c_3 \\ M_{23} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3 \\ \\ M_{31} &= 0 \\ M_{32} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3 \\ M_{33} &= I_{x3} + m_3r_2^2. \end{aligned}$$

Note that several of the moments of inertia of the different links do not appear in this expression. This is because the limited degrees of freedom of the manipulator do not allow arbitrary rotations of each joint around each axis.

The Coriolis and centrifugal forces are computed directly from the inertia matrix via the formula

$$C_{ij}(\theta, \dot{\theta}) = \sum_{k=1}^{n} \Gamma_{ijk}\dot{\theta}_k = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i}\right)\dot{\theta}_k.$$

A very messy calculation shows that the nonzero values of Γ_{ijk} are given

by:

$$\begin{aligned}
\Gamma_{112} &= (I_{y2} - I_{z2} - m_2 r_1^2)c_2 s_2 + (I_{y3} - I_{z3})c_{23}s_{23} \\
&\quad - m_3(l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\Gamma_{113} &= (I_{y3} - I_{z3})c_{23}s_{23} - m_3 r_2 s_{23}(l_1 c_2 + r_2 c_{23}) \\
\Gamma_{121} &= (I_{y2} - I_{z2} - m_2 r_1^2)c_2 s_2 + (I_{y3} - I_{z3})c_{23}s_{23} \\
&\quad - m_3(l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\\
\Gamma_{131} &= (I_{y3} - I_{z3})c_{23}s_{23} - m_3 r_2 s_{23}(l_1 c_2 + r_2 c_{23}) \\
\\
\Gamma_{211} &= (I_{z2} - I_{y2} + m_2 r_1^2)c_2 s_2 + (I_{z3} - I_{y3})c_{23}s_{23} \\
&\quad + m_3(l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\Gamma_{223} &= -l_1 m_3 r_2 s_3 \\
\Gamma_{232} &= -l_1 m_3 r_2 s_3 \\
\Gamma_{233} &= -l_1 m_3 r_2 s_3 \\
\\
\Gamma_{311} &= (I_{z3} - I_{y3})c_{23}s_{23} + m_3 r_2 s_{23}(l_1 c_2 + r_2 c_{23}) \\
\\
\Gamma_{322} &= l_1 m_3 r_2 s_3
\end{aligned}$$

Finally, we compute the effect of gravitational forces on the manipulator. These forces are written as

$$N(\theta, \dot{\theta}) = \frac{\partial V}{\partial \theta},$$

where $V : \mathbb{R}^n \to \mathbb{R}$ is the potential energy of the manipulator. For the three-link manipulator under consideration here, the potential energy is given by

$$V(\theta) = m_1 g h_1(\theta) + m_2 g h_2(\theta) + m_3 g h_3(\theta),$$

where h_i is the the height of the center of mass for the i^{th} link. These can be found using the forward kinematics map

$$g_{sl_i}(\theta) = e^{\widehat{\xi}_1 \theta_1} \dots e^{\widehat{\xi}_i \theta_i} g_{sl_i}(0),$$

which gives

$$\begin{aligned}
h_1(\theta) &= r_0 \\
h_2(\theta) &= l_0 - r_1 \sin\theta_2 \\
h_3(\theta) &= l_0 - l_1 \sin\theta_2 - r_2 \sin(\theta_2 + \theta_3).
\end{aligned}$$

Substituting these expressions into the potential energy and taking the

derivative gives

$$N(\theta,\dot{\theta}) = \frac{\partial V}{\partial \theta} = \begin{bmatrix} 0 \\ -(m_2 g r_1 + m_3 g l_1)\cos\theta_2 - m_3 r_2 \cos(\theta_2+\theta_3)) \\ -m_3 g r_2 \cos(\theta_2+\theta_3)) \end{bmatrix}. \tag{4.25}$$

This completes the derivation of the dynamics.

3.3 Robot dynamics and the product of exponentials formula

The formulas and properties given in the last section hold for any mechanical system with Lagrangian $L = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} - V(\theta)$. If the forward kinematics are specified using the product of exponentials formula, then it is possible to get more explicit formulas for the inertia and Coriolis matrices. In this section we derive these formulas, based on the treatments given by Brockett et al. [15] and Park et al. [88].

In addition to the tools introduced in Chapters 2 and 3, we will make use of one additional operation on twists. Recall, first, that in $so(3)$ the cross product between two vectors $\omega_1, \omega_2 \in \mathbb{R}^3$ yields a third vector, $\omega_1 \times \omega_2 \in \mathbb{R}^3$. It can be shown by direct calculation that the cross product satisfies

$$(\omega_1 \times \omega_2)^\wedge = \widehat{\omega}_1\widehat{\omega}_2 - \widehat{\omega}_2\widehat{\omega}_1.$$

By direct analogy, we define the *Lie bracket* on $se(3)$ as

$$[\widehat{\xi}_1, \widehat{\xi}_2] = \widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1.$$

A simple calculation verifies that the right-hand side of this equation has the form of a twist, and hence $[\widehat{\xi}_1, \widehat{\xi}_2] \in se(3)$.

If $\xi_1, \xi_2 \in \mathbb{R}^6$ represent the coordinates for two twists, we define the bracket operation $[\cdot,\cdot] : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ as

$$[\xi_1, \xi_2] = \left(\widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1\right)^\vee. \tag{4.26}$$

This is a generalization of the cross product on $\mathbb{R}^3$ to vectors in $\mathbb{R}^6$. The following properties of the Lie bracket are also generalizations of properties of the cross product:

$$[\xi_1, \xi_2] = -[\xi_2, \xi_1]$$
$$[\xi_1, [\xi_2, \xi_3]] + [\xi_2, [\xi_3, \xi_1]] + [\xi_3, [\xi_1, \xi_2]] = 0.$$

A more detailed (and abstract) description of the Lie bracket operation on $se(3)$ is given in Appendix A. For this chapter we shall only need the formula given in equation (4.26)

We now define some additional notation which we use in the sequel. Let $A_{ij} \in \mathbb{R}^{6\times 6}$ represent the adjoint transformation given by

$$A_{ij} = \begin{cases} \mathrm{Ad}^{-1}_{(e^{\xi_{j+1}\theta_{j+1}} \cdots e^{\xi_i \theta_i})} & i > j \\ I & i = j \\ 0 & i < j. \end{cases} \tag{4.27}$$

Using this notation, the j^{th} column of the body Jacobian for the i^{th} link is given by $\mathrm{Ad}_{g_{sl_i}^{-1}} A_{ij}\xi_j$:

$$J_i(\theta) = \mathrm{Ad}_{g_{sl_i(0)}^{-1}} \begin{bmatrix} A_{i1}\xi_1 & \cdots & A_{ii}\xi_i & 0 & \cdots & 0 \end{bmatrix}.$$

We combine $\mathrm{Ad}_{g_{sl_i(0)}^{-1}}$ with the link inertia matrix by defining the transformed inertia matrix for the i^{th} link:

$$\mathcal{M}_i' = \mathrm{Ad}^T_{g_{sl_i(0)}^{-1}} \mathcal{M}_i \mathrm{Ad}_{g_{sl_i(0)}^{-1}}. \tag{4.28}$$

The matrix $\mathcal{M}_i'$ represents the inertia of the i^{th} link reflected into the base frame of the manipulator.

Using these definitions, we can obtain formulas for the inertial quantities which appear in the equation of motion. We state the results as a proposition.

Proposition 4.3. Formulas for inertia and Coriolis matrices *Using the notation defined above, the inertia and Coriolis matrices for an open-chain manipulator are given by*

$$\begin{aligned} M_{ij}(\theta) &= \sum_{l=\max(i,j)}^{n} \xi_i^T A_{li}^T \mathcal{M}_l' A_{lj} \xi_j \\ C_{ij}(\theta) &= \frac{1}{2}\sum_{k=1}^{n} \left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i} \right) \dot{\theta}_k, \end{aligned} \tag{4.29}$$

where

$$\begin{aligned} \frac{\partial M_{ij}}{\partial \theta_k} = \sum_{l=\max(i,j)}^{n} \Big(& [A_{k-1,i}\xi_i, \xi_k]^T A_{lk}^T \mathcal{M}_l' A_{lj}\xi_j \\ & + \xi_i^T A_{li}^T \mathcal{M}_l' A_{lk} [A_{k-1,j}\xi_j, \xi_k] \Big). \end{aligned} \tag{4.30}$$

This proposition shows that all of the dynamic attributes of the manipulator can be determined directly from the joint twists ξ_i, the link frames $g_{sl_i(0)}$, and the link inertia matrices $\mathcal{M}_i$. The matrices A_{ij} are

the only expressions in equations (4.29) and (4.30) which depend on the current configuration of the manipulator.

Proof. The only term which needs to be calculated in order to prove the proposition is $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$. For $i \geq j$, let $g_{ij} \in SE(3)$ be the rigid transformation given by

$$g_{ij} = \begin{cases} e^{-\widehat{\xi}_i\theta_i} \cdots e^{-\widehat{\xi}_{j+1}\theta_{j+1}} & i > j \\ I & i = j, \end{cases}$$

so that $A_{ij} = \mathrm{Ad}_{g_{ij}}$. Using this notation, if k is an integer such that $i \geq k \geq j$, then $g_{ij} = g_{ik}g_{kj}$. We now proceed to calculate $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$ for $i \geq k \geq j$:

$$\begin{aligned} \frac{\partial}{\partial\theta_k}(A_{lj}\xi_j) &= \left(\frac{\partial}{\partial\theta_k}\left(g_{lj}\,\widehat{\xi}_j\,g_{lj}^{-1}\right)\right)^\vee = \left(\frac{\partial g_{lj}}{\partial\theta_k}\,\widehat{\xi}_j\,g_{lj}^{-1} + g_{lj}\,\widehat{\xi}_j\,\frac{\partial g_{lj}^{-1}}{\partial\theta_k}\right)^\vee \\ &= \left(-g_{l,k}\,\widehat{\xi}_k\,g_{k-1,l}\,\widehat{\xi}_j\,g_{lj}^{-1} + g_{lj}\,\widehat{\xi}_j\,g_{k-1,j}^{-1}\,\widehat{\xi}_k\,g_{lk}^{-1}\right)^\vee \\ &= \mathrm{Ad}_{g_{lk}}\left(-\widehat{\xi}_k\,g_{k-1,l}\,\widehat{\xi}_j\,g_{kl}^{-1} + g_{k-1,j}\,\widehat{\xi}_j\,g_{k-1,j}^{-1}\,\widehat{\xi}_k\right)^\vee \\ &= A_{lk}[A_{k-1,j}\xi_j,\ \xi_k]. \end{aligned}$$

For all other values of k, $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$ is zero. The proposition now follows by direct calculation. □

Example 4.4. Dynamics of an idealized SCARA manipulator Consider the SCARA manipulator shown in Figure 4.6. The joint twists are given by

$$\xi_1 = \begin{bmatrix} 0\\0\\0\\0\\0\\1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} l_1\\0\\0\\0\\0\\1 \end{bmatrix} \qquad \xi_3 = \begin{bmatrix} l_1+l_2\\0\\0\\0\\0\\1 \end{bmatrix} \qquad \xi_4 = \begin{bmatrix} 0\\0\\1\\0\\0\\0 \end{bmatrix}.$$

Assuming that the link frames are initially aligned with the base frame and are located at the centers of mass of the links, the transformed link inertia matrices have the form

$$\mathcal{M}_i' = \begin{bmatrix} I & 0\\ -\widehat{p}_i & I \end{bmatrix}\begin{bmatrix} m_iI & 0\\ 0 & \mathcal{I} \end{bmatrix}\begin{bmatrix} I & \widehat{p}_i\\ 0 & I \end{bmatrix} = \begin{bmatrix} m_iI & m_i\widehat{p}_i\\ -m_i\widehat{p}_i & \mathcal{I} \end{bmatrix},$$

where p_i is the location of the origin of the i^{th} link frame relative to the base frame S.

Given the joint twists ξ_i and transformed link inertias $\mathcal{M}_i'$, the dynamics of the manipulator can be computed using the formulas in Proposition 4.3. This task is considerably simplified using the software described

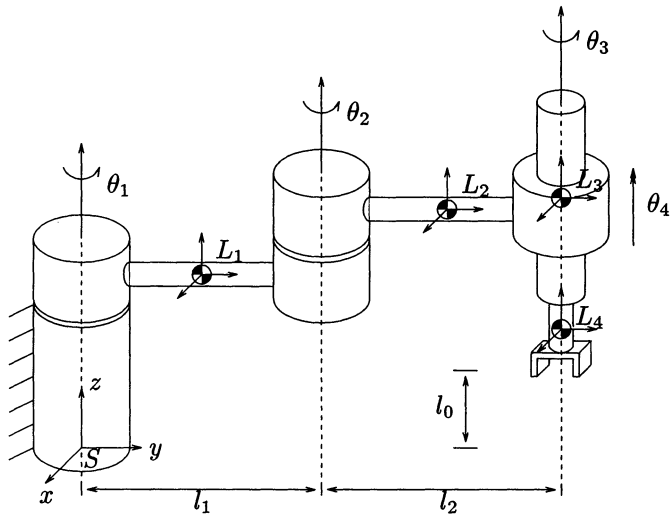


Figure 4.6: SCARA manipulator in its reference configuration.

in Appendix B, so we omit a detailed computation and present only the final result. The inertia matrix $M(\theta) \in \mathbb{R}^{4\times 4}$ is given by

$$M(\theta) = \begin{bmatrix} \alpha + \beta + 2\gamma\cos\theta_2 & \beta + \gamma\cos\theta_2 & \delta & 0 \\ \beta + \gamma\cos\theta_2 & \beta & \delta & 0 \\ \delta & \delta & \delta & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix},$$

where

$$\begin{aligned} \alpha &= I_{z1} + r_1^2 m_1 + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4 \\ \beta &= I_{z2} + I_{z3} + I_{z4} + l_2^2 m_3 + l_2^2 m_4 + m_2 r_2^2 \\ \gamma &= l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 m_2 r_2 \\ \delta &= I_{z3} + I_{z4}. \end{aligned}$$

The Coriolis matrix is given by

$$C(\theta,\dot\theta) = \begin{bmatrix} -\gamma\sin\theta_2\,\dot\theta_2 & -\gamma\sin\theta_2\,(\dot\theta_1 + \dot\theta_2) & 0 & 0 \\ \gamma\sin\theta_2\,\dot\theta_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The only remaining term in the dynamics is the gravity term, which can be determined by inspection since only θ_4 affects the potential energy of

the manipulator. Hence,

$$N(\theta, \dot{\theta}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_4 g \end{bmatrix}.$$

Friction and other nonconservative forces can also be included in N.

4 Lyapunov Stability Theory

In this section we review the tools of Lyapunov stability theory. These tools will be used in the next section to analyze the stability properties of a robot controller. We present a survey of the results that we shall need in the sequel, with no proofs. The interested reader should consult a standard text, such as Vidyasagar [118] or Khalil [50], for details.

4.1 Basic definitions

Consider a dynamical system which satisfies

$$\dot{x} = f(x, t) \qquad x(t_0) = x_0 \qquad x \in \mathbb{R}^n. \tag{4.31}$$

We will assume that $f(x, t)$ satisfies the standard conditions for the existence and uniqueness of solutions. Such conditions are, for instance, that $f(x, t)$ is Lipschitz continuous with respect to x, uniformly in t, and piecewise continuous in t. A point $x^* \in \mathbb{R}^n$ is an *equilibrium point* of (4.31) if $f(x^*, t) \equiv 0$. Intuitively and somewhat crudely speaking, we say an equilibrium point is *locally stable* if all solutions which start near x^* (meaning that the initial conditions are in a neighborhood of x^*) remain near x^* for all time. The equilibrium point x^* is said to be *locally asymptotically stable* if x^* is locally stable and, furthermore, all solutions starting near x^* tend towards x^* as $t \to \infty$. We say somewhat crude because the time-varying nature of equation (4.31) introduces all kinds of additional subtleties. Nonetheless, it is intuitive that a pendulum has a locally stable equilibrium point when the pendulum is hanging straight down and an unstable equilibrium point when it is pointing straight up. If the pendulum is damped, the stable equilibrium point is locally asymptotically stable.

By shifting the origin of the system, we may assume that the equilibrium point of interest occurs at $x^* = 0$. If multiple equilibrium points exist, we will need to study the stability of each by appropriately shifting the origin.

Definition 4.1. Stability in the sense of Lyapunov
The equilibrium point $x^* = 0$ of (4.31) is *stable (in the sense of Lyapunov)* at $t = t_0$ if for any $\epsilon > 0$ there exists a $\delta(t_0, \epsilon) > 0$ such that

$$\|x(t_0)\| < \delta \quad \Longrightarrow \quad \|x(t)\| < \epsilon, \quad \forall t \geq t_0. \tag{4.32}$$

Lyapunov stability is a very mild requirement on equilibrium points. In particular, it does not require that trajectories starting close to the origin tend to the origin asymptotically. Also, stability is defined at a time instant t_0. *Uniform stability* is a concept which guarantees that the equilibrium point is not losing stability. We insist that for a uniformly stable equilibrium point x^*, δ in the Definition 4.1 not be a function of t_0, so that equation (4.32) may hold for all t_0. Asymptotic stability is made precise in the following definition:

Definition 4.2. Asymptotic stability
An equilibrium point $x^* = 0$ of (4.31) is *asymptotically stable* at $t = t_0$ if

1. $x^* = 0$ is stable, and
2. $x^* = 0$ is locally attractive; i.e., there exists $\delta(t_0)$ such that

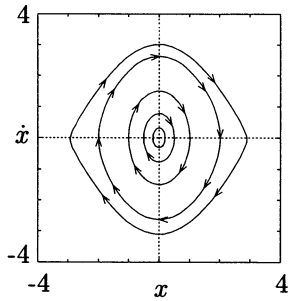
$$\|x(t_0)\| < \delta \quad \Longrightarrow \quad \lim_{t \to \infty} x(t) = 0. \tag{4.33}$$

As in the previous definition, asymptotic stability is defined at t_0. *Uniform asymptotic stability* requires:

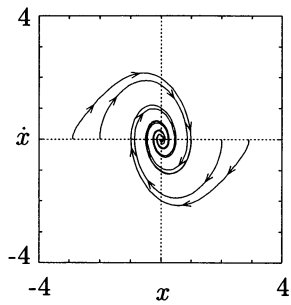
1. $x^* = 0$ is uniformly stable, and
2. $x^* = 0$ is uniformly locally attractive; i.e., there exists δ independent of t_0 for which equation (4.33) holds. Further, it is required that the convergence in equation (4.33) is uniform.

Finally, we say that an equilibrium point is *unstable* if it is not stable. This is less of a tautology than it sounds and the reader should be sure he or she can negate the definition of stability in the sense of Lyapunov to get a definition of instability. In robotics, we are almost always interested in uniformly asymptotically stable equilibria. If we wish to move the robot to a point, we would like to actually converge to that point, not merely remain nearby. Figure 4.7 illustrates the difference between stability in the sense of Lyapunov and asymptotic stability.

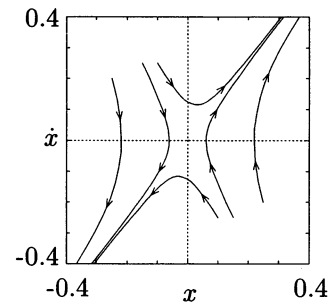
Definitions 4.1 and 4.2 are *local* definitions; they describe the behavior of a system near an equilibrium point. We say an equilibrium point x^* is *globally* stable if it is stable for all initial conditions $x_0 \in \mathbb{R}^n$. Global stability is very desirable, but in many applications it can be difficult to achieve. We will concentrate on local stability theorems and indicate where it is possible to extend the results to the global case. Notions



(a) Stable in the sense of Lyapunov



(b) Asymptotically stable



(c) Unstable (saddle)

Figure 4.7: Phase portraits for stable and unstable equilibrium points.

of uniformity are only important for time-varying systems. Thus, for time-invariant systems, stability implies uniform stability and asymptotic stability implies uniform asymptotic stability.

It is important to note that the definitions of asymptotic stability do not quantify the rate of convergence. There is a strong form of stability which demands an exponential rate of convergence:

Definition 4.3. Exponential stability, rate of convergence
The equilibrium point $x^* = 0$ is an *exponentially stable* equilibrium point of (4.31) if there exist constants $m, \alpha > 0$ and $\epsilon > 0$ such that

$$\|x(t)\| \leq me^{-\alpha(t-t_0)}\|x(t_0)\| \tag{4.34}$$

for all $\|x(t_0)\| \leq \epsilon$ and $t \geq t_0$. The largest constant α which may be utilized in (4.34) is called the *rate of convergence.*

Exponential stability is a strong form of stability; in particular, it implies uniform, asymptotic stability. Exponential convergence is important in applications because it can be shown to be robust to perturbations and is essential for the consideration of more advanced control algorithms, such as adaptive ones. A system is *globally exponentially stable* if the

bound in equation (4.34) holds for all $x_0 \in \mathbb{R}^n$. Whenever possible, we shall strive to prove global, exponential stability.

4.2 The direct method of Lyapunov

Lyapunov's direct method (also called the second method of Lyapunov) allows us to determine the stability of a system without explicitly integrating the differential equation (4.31). The method is a generalization of the idea that if there is some "measure of energy" in a system, then we can study the rate of change of the energy of the system to ascertain stability. To make this precise, we need to define exactly what one means by a "measure of energy." Let B_ϵ be a ball of size ϵ around the origin, $B_\epsilon = \{x \in \mathbb{R}^n : \|x\| < \epsilon\}$.

Definition 4.4. Locally positive definite functions (lpdf)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a *locally positive definite function* if for some $\epsilon > 0$ and some continuous, strictly increasing function $\alpha : \mathbb{R}_+ \to \mathbb{R}$,

$$V(0,t) = 0 \quad \text{and} \quad V(x,t) \geq \alpha(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0. \quad (4.35)$$

A locally positive definite function is locally like an energy function. Functions which are globally like energy functions are called positive definite functions:

Definition 4.5. Positive definite functions (pdf)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a *positive definite function* if it satisfies the conditions of Definition 4.4 and, additionally, $\alpha(p) \to \infty$ as $p \to \infty$.

To bound the energy function from above, we define decrescence as follows:

Definition 4.6. Decrescent functions
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is *decrescent* if for some $\epsilon > 0$ and some continuous, strictly increasing function $\beta : \mathbb{R}_+ \to \mathbb{R}$,

$$V(x,t) \leq \beta(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0 \quad (4.36)$$

Using these definitions, the following theorem allows us to determine stability for a system by studying an appropriate energy function. Roughly, this theorem states that when $V(x,t)$ is a locally positive definite function and $\dot{V}(x,t) \leq 0$ then we can conclude stability of the equilibrium point. The time derivative of V is taken along the trajectories of the system:

$$\dot{V}\Big|_{\dot{x}=f(x,t)} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f.$$

In what follows, by $\dot{V}$ we will mean $\dot{V}|_{\dot{x}=f(x,t)}$.

Table 4.1: Summary of the basic theorem of Lyapunov.

	Conditions on $V(x,t)$	Conditions on $-\dot{V}(x,t)$	Conclusions
1	lpdf	≥ 0 locally	Stable
2	lpdf, decrescent	≥ 0 locally	Uniformly stable
3	lpdf, decrescent	lpdf	Uniformly asymptotically stable
4	pdf, decrescent	pdf	Globally uniformly asymptotically stable

Theorem 4.4. Basic theorem of Lyapunov
Let $V(x,t)$ be a non-negative function with derivative $\dot{V}$ along the trajectories of the system.

1. *If $V(x,t)$ is locally positive definite and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is locally stable (in the sense of Lyapunov).*

2. *If $V(x,t)$ is locally positive definite and decrescent, and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is uniformly locally stable (in the sense of Lyapunov).*

3. *If $V(x,t)$ is locally positive definite and decrescent, and $-\dot{V}(x,t)$ is locally positive definite, then the origin of the system is uniformly locally asymptotically stable.*

4. *If $V(x,t)$ is positive definite and decrescent, and $-\dot{V}(x,t)$ is positive definite, then the origin of the system is globally uniformly asymptotically stable.*

The conditions in the theorem are summarized in Table 4.1.

Theorem 4.4 gives sufficient conditions for the stability of the origin of a system. It does not, however, give a prescription for determining the Lyapunov function $V(x,t)$. Since the theorem only gives sufficient conditions, the search for a Lyapunov function establishing stability of an equilibrium point could be arduous. However, it is a remarkable fact that the converse of Theorem 4.4 also exists: if an equilibrium point is stable, then there exists a function $V(x,t)$ satisfying the conditions of the theorem. However, the utility of this and other converse theorems is limited by the lack of a computable technique for generating Lyapunov functions.

Theorem 4.4 also stops short of giving explicit rates of convergence of solutions to the equilibrium. It may be modified to do so in the case of exponentially stable equilibria.

Theorem 4.5. Exponential stability theorem
$x^* = 0$ is an exponentially stable equilibrium point of $\dot{x} = f(x,t)$ if and only if there exists an $\epsilon > 0$ and a function $V(x,t)$ which satisfies

$$\alpha_1\|x\|^2 \leq V(x,t) \leq \alpha_2\|x\|^2$$
$$\dot{V}|_{\dot{x}=f(x,t)} \leq -\alpha_3\|x\|^2$$
$$\|\frac{\partial V}{\partial x}(x,t)\| \leq \alpha_4\|x\|$$

for some positive constants $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and $\|x\| \leq \epsilon$.

The rate of convergence for a system satisfying the conditions of Theorem 4.5 can be determined from the proof of the theorem [103]. It can be shown that

$$m \leq \left(\frac{\alpha_2}{\alpha_1}\right)^{1/2} \qquad \alpha \geq \frac{\alpha_3}{2\alpha_2}$$

are bounds in equation (4.34). The equilibrium point $x^* = 0$ is globally exponentially stable if the bounds in Theorem 4.5 hold for all x.

4.3 The indirect method of Lyapunov

The indirect method of Lyapunov uses the linearization of a system to determine the local stability of the original system. Consider the system

$$\dot{x} = f(x,t) \tag{4.37}$$

with $f(0,t) = 0$ for all $t \geq 0$. Define

$$A(t) = \left.\frac{\partial f(x,t)}{\partial x}\right|_{x=0} \tag{4.38}$$

to be the Jacobian matrix of $f(x,t)$ with respect to x, evaluated at the origin. It follows that for each fixed t, the remainder

$$f_1(x,t) = f(x,t) - A(t)x$$

approaches zero as x approaches zero. However, the remainder may not approach zero *uniformly*. For this to be true, we require the stronger condition that

$$\lim_{\|x\|\to 0} \sup_{t\geq 0} \frac{\|f_1(x,t)\|}{\|x\|} = 0. \tag{4.39}$$

If equation (4.39) holds, then the system

$$\dot{z} = A(t)z \tag{4.40}$$

is referred to as the (uniform) *linearization* of equation (4.31) about the origin. When the linearization exists, its stability determines the local stability of the original nonlinear equation.

Theorem 4.6. Stability by linearization
Consider the system (4.37) *and assume*

$$\lim_{\|x\|\to 0} \sup_{t\geq 0} \frac{\|f_1(x,t)\|}{\|x\|} = 0.$$

Further, let $A(\cdot)$ *defined in equation* (4.38) *be bounded. If 0 is a uniformly asymptotically stable equilibrium point of* (4.40) *then it is a* locally *uniformly asymptotically stable equilibrium point of* (4.37).

The preceding theorem requires *uniform* asymptotic stability of the linearized system to prove uniform asymptotic stability of the nonlinear system. Counterexamples to the theorem exist if the linearized system is not uniformly asymptotically stable.

If the system (4.37) is time-invariant, then the indirect method says that if the eigenvalues of

$$A = \left.\frac{\partial f(x)}{\partial x}\right|_{x=0}$$

are in the open left half complex plane, then the origin is asymptotically stable.

This theorem proves that *global* uniform asymptotic stability of the linearization implies *local* uniform asymptotic stability of the original nonlinear system. The estimates provided by the proof of the theorem can be used to give a (conservative) bound on the domain of attraction of the origin. Systematic techniques for estimating the bounds on the regions of attraction of equilibrium points of nonlinear systems is an important area of research and involves searching for the "best" Lyapunov functions.

4.4 Examples

We now illustrate the use of the stability theorems given above on a few examples.

Example 4.5. Linear harmonic oscillator
Consider a damped harmonic oscillator, as shown in Figure 4.8. The dynamics of the system are given by the equation

$$M\ddot{q} + B\dot{q} + Kq = 0, \tag{4.41}$$

where M, B, and K are all positive quantities. As a state space equation we rewrite equation (4.41) as

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -(K/M)q - (B/M)\dot{q} \end{bmatrix}. \tag{4.42}$$

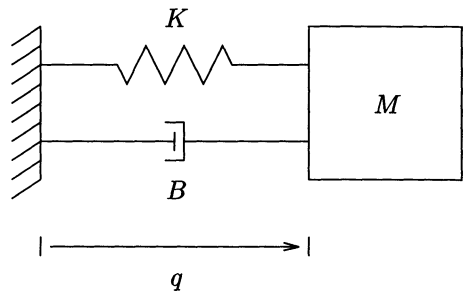


Figure 4.8: Damped harmonic oscillator.

Define $x = (q, \dot{q})$ as the state of the system.

Since this system is a linear system, we can determine stability by examining the poles of the system. The Jacobian matrix for the system is

$$A = \begin{bmatrix} 0 & 1 \\ -K/M & -B/M \end{bmatrix},$$

which has a characteristic equation

$$\lambda^2 + (B/M)\lambda + (K/M) = 0.$$

The solutions of the characteristic equation are

$$\lambda = \frac{-B \pm \sqrt{B^2 - 4KM}}{2M},$$

which always have negative real parts, and hence the system is (globally) exponentially stable.

We now try to apply Lyapunov's direct method to determine exponential stability. The "obvious" Lyapunov function to use in this context is the energy of the system,

$$V(x,t) = \frac{1}{2}M\dot{q}^2 + \frac{1}{2}Kq^2. \tag{4.43}$$

Taking the derivative of V along trajectories of the system (4.41) gives

$$\dot{V} = M\dot{q}\ddot{q} + Kq\dot{q} = -B\dot{q}^2. \tag{4.44}$$

The function $-\dot{V}$ is quadratic but not locally positive definite, since it does not depend on q, and hence we cannot conclude exponential stability. It is still possible to conclude *asymptotic* stability using Lasalle's invariance principle (described in the next section), but this is obviously conservative since we already know that the system is exponentially stable.

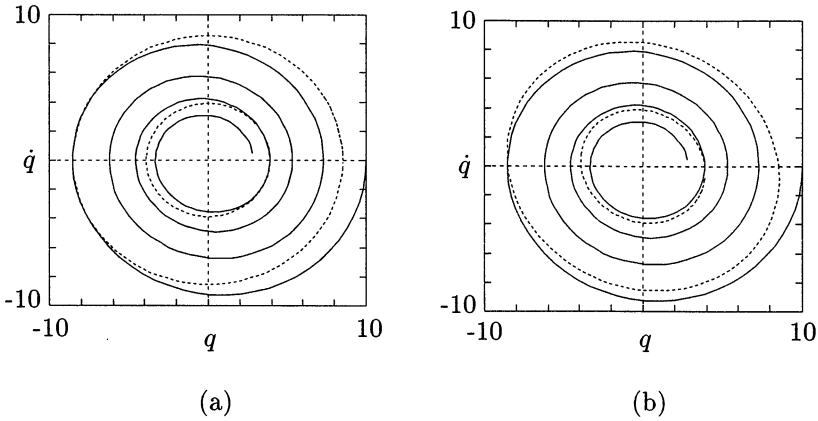


Figure 4.9: Flow of damped harmonic oscillator. The dashed lines are the level sets of the Lyapunov function defined by (a) the total energy and (b) a skewed modification of the energy.

The reason that Lyapunov's direct method fails is illustrated in Figure 4.9a, which shows the flow of the system superimposed with the level sets of the Lyapunov function. The level sets of the Lyapunov function become tangent to the flow when $\dot{q} = 0$, and hence it is not a valid Lyapunov function for determining exponential stability.

To fix this problem, we skew the level sets slightly, so that the flow of the system crosses the level surfaces transversely. Define

$$V(x,t) = \frac{1}{2}\begin{bmatrix} q \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} K & \epsilon M \\ \epsilon M & M \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \frac{1}{2}\dot{q}M\dot{q} + \frac{1}{2}qKq + \epsilon\dot{q}Mq,$$

where ϵ is a small positive constant such that V is still positive definite. The derivative of the Lyapunov function becomes

$$\begin{aligned} \dot{V} &= \dot{q}M\ddot{q} + qK\dot{q} + \epsilon M\dot{q}^2 + \epsilon qM\ddot{q} \\ &= (-B + \epsilon M)\dot{q}^2 + \epsilon(-Kq^2 - Bq\dot{q}) = -\begin{bmatrix} q \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} \epsilon K & \frac{1}{2}\epsilon B \\ \frac{1}{2}\epsilon B & B - \epsilon M \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix}. \end{aligned}$$

The function $\dot{V}$ can be made negative definite for ϵ chosen sufficiently small (see Exercise 11) and hence we can conclude *exponential* stability. The level sets of this Lyapunov function are shown in Figure 4.9b.

This same technique is used in the stability proofs for the robot control laws contained in the next section.

Example 4.6. Nonlinear spring mass system with damper
Consider a mechanical system consisting of a unit mass attached to a

nonlinear spring with a velocity-dependent damper. If x_1 stands for the position of the mass and x_2 its velocity, then the equations describing the system are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -f(x_2) - g(x_1). \end{aligned} \tag{4.45}$$

Here f and g are smooth functions modeling the friction in the damper and restoring force of the spring, respectively. We will assume that f, g are both passive; that is,

$$\begin{aligned} \sigma f(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0] \\ \sigma g(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0] \end{aligned}$$

and equality is only achieved when $\sigma = 0$. The candidate for the Lyapunov function is

$$V(x) = \frac{x_2^2}{2} + \int_0^{x_1} g(\sigma)\, d\sigma.$$

The passivity of g guarantees that $V(x)$ is a locally positive definite function. A short calculation verifies that

$$\dot{V}(x) = -x_2 f(x_2) \leq 0 \quad \text{when } |x_2| \leq \sigma_0.$$

This establishes the stability, but not the asymptotic stability of the origin. Actually, the origin is asymptotically stable, but this needs Lasalle's principle, which is discussed in the next section.

4.5 Lasalle's invariance principle

Lasalle's theorem enables one to conclude asymptotic stability of an equilibrium point even when $-\dot{V}(x,t)$ is not locally positive definite. However, it applies only to autonomous or periodic systems. We will deal with the autonomous case and begin by introducing a few more definitions. We denote the solution trajectories of the autonomous system

$$\dot{x} = f(x) \tag{4.46}$$

as $s(t, x_0, t_0)$, which is the solution of equation (4.46) at time t starting from x_0 at t_0.

Definition 4.7. ω limit set
The set $S \subset \mathbb{R}^n$ is the *ω limit set* of a trajectory $s(\cdot, x_0, t_0)$ if for every $y \in S$, there exists a strictly increasing sequence of times t_n such that

$$s(t_n, x_0, t_0) \to y$$

as $t_n \to \infty$.

Definition 4.8. Invariant set
The set $M \subset \mathbb{R}^n$ is said to be an (positively) *invariant set* if for all $y \in M$ and $t_0 \geq 0$, we have

$$s(t, y, t_0) \in M \quad \forall t \geq t_0.$$

It may be proved that the ω limit set of every trajectory is closed and invariant. We may now state Lasalle's principle.

Theorem 4.7. Lasalle's principle
Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally positive definite function such that on the compact set $\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}$ we have $\dot{V}(x) \leq 0$. Define

$$S = \{x \in \Omega_c : \dot{V}(x) = 0\}.$$

As $t \to \infty$, the trajectory tends to the largest invariant set inside S; i.e., its ω limit set is contained inside the largest invariant set in S. In particular, if S contains no invariant sets other than $x = 0$, then 0 is asymptotically stable.

A global version of the preceding theorem may also be stated. An application of Lasalle's principle is as follows:

Example 4.7. Nonlinear spring mass system with damper
Consider the same example as in equation (4.45), where we saw that with

$$V(x) = \frac{x_2^2}{2} + \int_0^{x_1} g(\sigma)\, d\sigma,$$

we obtained

$$\dot{V}(x) = -x_2 f(x_2).$$

Choosing $c = \min(V(-\sigma_0, 0), V(\sigma_0, 0))$ so as to apply Lasalle's principle, we see that

$$\dot{V}(x) \leq 0 \quad \text{for } x \in \Omega_c := \{x : V(x) \leq c\}.$$

As a consequence of Lasalle's principle, the trajectory enters the largest invariant set in $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\} = \Omega_c \cap \{x_1, 0\}$. To obtain the largest invariant set in this region, note that

$$x_2(t) \equiv 0 \quad \Longrightarrow \quad x_1(t) \equiv x_{10} \quad \Longrightarrow \quad \dot{x}_2(t) = 0 = -f(0) - g(x_{10}),$$

where x_{10} is some constant. Consequently, we have that

$$g(x_{10}) = 0 \quad \Longrightarrow \quad x_{10} = 0.$$

Thus, the largest invariant set inside $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\}$ is the origin and, by Lasalle's principle, the origin is locally asymptotically stable.

There is a version of Lasalle's theorem which holds for periodic systems as well. However, there are no significant generalizations for non-periodic systems and this restricts the utility of Lasalle's principle in applications.

5 Position Control and Trajectory Tracking

In this section, we consider the position control problem for robot manipulators: given a desired trajectory, how should the joint torques be chosen so that the manipulator follows that trajectory. We would like to choose a control strategy which is robust with respect to initial condition errors, sensor noise, and modeling errors. We ignore the problems of actuator dynamics, and assume that we can command arbitrary torques which are exerted at the joints.

5.1 Problem description

We are given a description of the dynamics of a robot manipulator in the form of the equation

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau, \tag{4.47}$$

where $\theta \in \mathbb{R}^n$ is the set of configuration variables for the robot and $\tau \in \mathbb{R}^n$ denotes the torques applied at the joints. We are also given a joint trajectory $\theta_d(\cdot)$ which we wish to track. For simplicity, we assume that θ_d is specified for all time and that it is at least twice differentiable.

If we have a perfect model of the robot and $\theta(0) = \theta_d(0), \dot{\theta}(0) = \dot{\theta}_d(0)$, then we may solve our problem by choosing

$$\tau = M(\theta_d)\ddot{\theta}_d + C(\theta_d, \dot{\theta}_d)\dot{\theta}_d + N(\theta_d, \dot{\theta}_d).$$

Since both θ and θ_d satisfy the same differential equation and have the same initial conditions, it follows from the uniqueness of the solutions of differential equations that $\theta(t) = \theta_d(t)$ for all $t \geq 0$. This an example of an *open-loop* control law: the current state of the robot is not used in choosing the control inputs.

Unfortunately, this strategy is not very robust. If $\theta(0) \neq \theta_d(0)$, then the open-loop control law will never correct for this error. This is clearly undesirable, since we almost never know the current position of a robot *exactly*. Furthermore, we have no guarantee that if our starting configuration is near the desired initial configuration that the trajectory of the robot will stay near the desired trajectory for all time. For this reason, we introduce feedback into our control law. This feedback must be chosen such that the actual robot trajectory converges to the desired trajectory. In particular, if our trajectory is a single setpoint, the closed-loop system should be asymptotically stable about the desired setpoint.

There are several approaches for designing stable robot control laws. Using the structural properties of robot dynamics, we will be able to prove stability of these control laws for *all* robots having those properties.

Hence, we do not need to design control laws for a specific robot; as long as we show that stability of a particular control algorithm requires only those properties given in Lemma 4.2 on page 171, then our control law will work for general open-chain robot manipulators. Of course, the performance of a given control law depends heavily on the particular manipulator, and hence the control laws presented here should only be used as a starting point for synthesizing a feedback compensator.

5.2 Computed torque

Consider the following refinement of the open-loop control law presented above: given the current position and velocity of the manipulator, cancel all nonlinearities and apply exactly the torque needed to overcome the inertia of the actuator,

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

Substituting this control law into the dynamic equations of the manipulator, we see that

$$M(\theta)\ddot{\theta} = M(\theta)\ddot{\theta}_d,$$

and since $M(\theta)$ is uniformly positive definite in θ, we have

$$\ddot{\theta} = \ddot{\theta}_d. \tag{4.48}$$

Hence, if the initial position and velocity of the manipulator matches the desired position and velocity, the manipulator will follow the desired trajectory. As before, this control law will not correct for any initial condition errors which are present.

The tracking properties of the control law can be improved by adding state feedback. The linearity of equation (4.48) suggests the following control law:

$$\boxed{\tau = M(\theta)\left(\ddot{\theta}_d - K_v\dot{e} - K_p e\right) + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta})} \tag{4.49}$$

where $e = \theta - \theta_d$, and K_v and K_p are constant gain matrices. When substituted into equation (4.47), the error dynamics can be written as:

$$M(\theta)\left(\ddot{e} + K_v\dot{e} + K_p e\right) = 0.$$

Since $M(\theta)$ is always positive definite, we have

$$\ddot{e} + K_v\dot{e} + K_p e = 0. \tag{4.50}$$

This is a linear differential equation which governs the error between the actual and desired trajectories. Equation (4.49) is called the *computed torque* control law.

The computed torque control law consists of two components. We can write equation (4.49) as

$$\tau = \underbrace{M(\theta)\ddot{\theta}_d + C\dot{\theta} + N}_{\tau_{\text{ff}}} + \underbrace{M(\theta)\left(-K_v\dot{e} - K_p e\right)}_{\tau_{\text{fb}}}.$$

The term τ_{ff} is the *feedforward* component. It provides the amount of torque necessary to drive the system along its nominal path. The term τ_{fb} is the *feedback* component. It provides correction torques to reduce any errors in the trajectory of the manipulator.

Since the error equation (4.50) is linear, it is easy to choose K_v and K_p so that the overall system is stable and $e \to 0$ exponentially as $t \to \infty$. Moreover, we can choose K_v and K_p such that we get independent exponentially stable systems (by choosing K_p and K_v diagonal). The following proposition gives one set of conditions under which the computed torque control law (4.49) results in exponential tracking.

Proposition 4.8. Stability of the computed torque control law *If $K_p, K_v \in \mathbb{R}^{n\times n}$ are positive definite, symmetric matrices, then the control law* (4.49) *applied to the system* (4.47) *results in exponential trajectory tracking.*

Proof. The error dynamics can be written as a first-order linear system:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}}_{A} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}.$$

It suffices to show that each of the eigenvalues of A has negative real part. Let $\lambda \in \mathbb{C}$ be an eigenvalue of A with corresponding eigenvector $v = (v_1, v_2) \in \mathbb{C}^{2n}$, $v \neq 0$. Then,

$$\lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_2 \\ -K_p v_1 - K_v v_2 \end{bmatrix}.$$

It follows that if $\lambda = 0$ then $v = 0$, and hence $\lambda = 0$ is not an eigenvalue of A. Further, if $\lambda \neq 0$, then $v_2 = 0$ implies that $v_1 = 0$. Thus, $v_1, v_2 \neq 0$ and we may assume without loss of generality that $\|v_1\| = 1$. Using this, we write

$$\begin{aligned} \lambda^2 &= v_1^* \lambda^2 v_1 = v_1^* \lambda v_2 \\ &= v_1^*(-K_p v_1 - K_v v_2) = -v_1^* K_p v_1 - \lambda v_1^* K_v v_1, \end{aligned}$$

where $*$ denotes complex conjugate transpose. Since $\alpha = v_1^* K_p v_1 > 0$ and $\beta = v_1^* K_v v_1 > 0$, we have

$$\lambda^2 + \alpha\lambda + \beta = 0 \qquad \alpha, \beta > 0$$

and hence the real part of λ is negative. □

The power of the computed torque control law is that it converts a nonlinear dynamical system into a linear one, allowing the use of any of a number of linear control synthesis tools. This is an example of a more general technique known as *feedback linearization*, where a nonlinear system is rendered linear via full-state nonlinear feedback. One disadvantage of using feedback linearization is that it can be demanding (in terms of computation time and input magnitudes) to use feedback to globally convert a nonlinear system into a single linear system. For robot manipulators, unboundedness of the inputs is rarely a problem since the inertia matrix of the system is bounded and hence the control torques which must be exerted always remain bounded. In addition, experimental results show that the computed torque controller has very good performance characteristics and it is becoming increasingly popular.

5.3 PD control

Another approach to controller synthesis for nonlinear systems is to design a linear controller based on the linearization of the system about an operating point. Since the linearization of a system locally determines the stability of the full system, this class of controllers is guaranteed to be locally stable. In many situations, it is possible to prove global stability for a linear controller by explicit construction of a Lyapunov function.

An example of this design methodology is a proportional plus derivative (PD) control law for a robot manipulator. In its simplest form, a PD control law has the form

$$\tau = -K_v\dot{e} - K_p e, \tag{4.51}$$

where K_v and K_p are positive definite matrices and $e = \theta - \theta_d$. Since this control law has no feedforward term, it can never achieve exact tracking for non-trivial trajectories. A common modification is to add an integral term to eliminate steady-state errors. This introduces additional complications since care must be taken to maintain stability and avoid integrator windup.

Before adding a feedforward term, we first show that the PD controller gives asymptotic setpoint stabilization.

Proposition 4.9. *If $\dot{\theta}_d \equiv 0$ and $K_v, K_p > 0$, the control law* (4.51) *applied to the system* (4.47) *renders the equilibrium point $\theta = \theta_d$ globally asymptotically stable.*

Proof. For $\theta_d \equiv 0$, the closed-loop system is

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + K_v\dot{\theta} + K_p(\theta - \theta_d) = 0. \tag{4.52}$$

Without loss of generality, we assume that $\theta_d = 0$ (if not, redefine $\theta' = \theta - \theta_d$). We choose the total energy of the system as our Lyapunov function,

$$V(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \frac{1}{2}\theta^T K_p \theta.$$

The function V is (globally) positive definite and decresent. Evaluating $\dot{V}$ along trajectories of (4.52),

$$\begin{aligned}\dot{V}(\theta, \dot{\theta}) &= \dot{\theta}^T M\ddot{\theta} + \frac{1}{2}\dot{\theta}^T \dot{M}\dot{\theta} + \dot{\theta}^T K_p \theta \\ &= -\dot{\theta}^T K_v \dot{\theta} + \frac{1}{2}\dot{\theta}^T (\dot{M} - 2C)\dot{\theta},\end{aligned}$$

and since $\dot{M} - 2C$ is skew-symmetric, we have

$$\dot{V} = -\dot{\theta}^T K_v \dot{\theta}.$$

Although K_v is positive definite, the function $\dot{V}$ is only negative *semi*-definite, since $\dot{V} = 0$ for $\dot{\theta} = 0$ and $\theta \neq 0$. Hence from Lyapunov's basic theorem, we can concluded only stability of the equilibrium point.

To check for asymptotic stability, we appeal to Lasalle's principle. The set S for which $\dot{V} \equiv 0$ is given by

$$S = \{(\theta, \dot{\theta}) : \dot{\theta} \equiv 0\}.$$

To find the largest invariant set contained in S, we substitute $\dot{\theta} \equiv 0$ into the closed loop equations 4.52. This gives

$$K_p \theta = 0$$

(recalling that $\theta_d = 0$) and since K_p is positive definite, it follows that the largest invariant set contained within S is the single point $\theta = 0$. Hence, the equilibrium point $\theta = 0$ is asymptotically stable. □

Since we are primarily interested in tracking, we consider a modified version of the PD control law:

$$\boxed{\tau = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta}_d + N(\theta, \dot{\theta}) - K_v \dot{e} - K_p e} \qquad (4.53)$$

We call this controller the *augmented PD control law.* Note that the second term in equation (4.53) is different from the Coriolis term $C(\theta, \dot{\theta})\dot{\theta}$. The reason for this difference is found in the proof of the following theorem.

Proposition 4.10. Stability of the PD control law
The control law (4.53) *applied to the system* (4.47) *results in exponential trajectory tracking if* $K_v, K_p > 0$.

Proof. The closed-loop system is

$$M(\theta)\ddot{e} + C(\theta, \dot{\theta})\dot{e} + K_v\dot{e} + K_p e = 0. \tag{4.54}$$

As in the proof of the previous proposition, using the energy of the system as a Lyapunov function does not allow us to conclude exponential stability because $\dot{V}$ is only negative semi-definite. Furthermore, since the system is time-varying (due to the $\theta_d(\cdot)$ terms), we cannot apply Lasalle's principle.

To show exponential stability, we adopt the same approach as with the spring mass system of the previous section. Namely, we skew the level sets of the energy function by choosing the Lyapunov function candidate

$$V(e, \dot{e}, t) = \frac{1}{2}\dot{e}^T M(\theta)\dot{e} + \frac{1}{2}e^T K_p e + \epsilon e^T M(\theta)\dot{e},$$

which is positive definite for ϵ sufficiently small since $M(\theta) > 0$ and $K_p > 0$. Evaluating $\dot{V}$ along trajectories of (4.54):

$$\begin{aligned}
\dot{V} &= \dot{e}^T M\ddot{e} + \frac{1}{2}\dot{e}^T\dot{M}\dot{e} + \dot{e}^T K_p e + \epsilon\dot{e}^T M\dot{e} + \epsilon e^T\left(M\ddot{e} + \dot{M}\dot{e}\right) \\
&= -\dot{e}^T(K_v - \epsilon M)\dot{e} + \frac{1}{2}\dot{e}^T(\dot{M} - 2C)\dot{e} + \epsilon e^T\left(-K_p e - K_v\dot{e} - C\dot{e} + \dot{M}\dot{e}\right) \\
&= -\dot{e}^T(K_v - \epsilon M)\dot{e} - \epsilon e^T K_p e + \epsilon e^T(-K_v + \frac{1}{2}\dot{M})\dot{e}
\end{aligned}$$

Choosing $\epsilon > 0$ sufficiently small insures that $\dot{V}$ is negative definite (see Exercise 11) and hence the system is exponentially stable using Theorem 4.5. □

If $\dot{\theta}_d \equiv 0$, i.e., we wish to stabilize a point, the control law (4.53) simplifies to the original PD control law (4.51). We also note that asymptotic tracking requires exact cancellation of friction and gravity forces and relies on accurate models of these quantities as well as the manipulator inertia matrix. In practice, errors in modeling will result in errors in tracking.

A further difficulty in using the PD control law is choosing the gains K_p and K_v. The linearization of the system about a given operating point θ_0 gives error dynamics of the form

$$M(\theta_0)\ddot{e} + K_v\dot{e} + K_p e = 0.$$

Since this is a linear system, it is possible to choose K_v and K_p to achieve a given performance specification using linear control theory. However, if we are tracking a trajectory, then there is no guarantee that we will remain near θ_0 and the chosen gains may not be appropriate. In practice, one can usually get reasonable results by choosing the gains based on the linearization about an equilibrium point in the middle of the robot's workspace.

5.4 Workspace control

Suppose we are given a path $g_d(t) \in SE(3)$ which represents the desired configuration of the end-effector as a function of time. One way to move the manipulator along this path is to solve the inverse kinematics problem at each instant in time and generate a desired joint angle trajectory $\theta_d(t) \in Q$ such that $g(\theta_d(t)) = g_d(t)$. The methods of the previous sections can then be used to generate a feedback controller which follows this path.

There are several disadvantages to solving the feedback control problem in this manner. Since solving the inverse kinematics problem is a time-consuming task, systems in which g_d is specified in real-time must use powerful computers to compute θ_d at a rate suitable for control. Furthermore, it may be difficult to choose the feedback gains in joint space in a meaningful way, since the original task was given in terms of the end-effector trajectory. For example, a joint-space, computed torque controller with diagonal gain matrices (K_p and K_v) will generate a decoupled response in joint space, resulting in straight line trajectories in θ if the setpoint of the manipulator is changed. However, due to the nonlinear nature of the kinematics, this will *not* generate a straight line trajectory in $SE(3)$. For many tasks, this type of behavior is undesirable.

To overcome these disadvantages, we consider formulating the problem directly in end-effector coordinates. In doing so, we will eliminate the need to solve the inverse kinematics and also generate controllers whose gains have a more direct connection with the task performance. However, in order to use the tools developed in Section 4, we must choose a set of local coordinates for $SE(3)$, such as parameterizing orientation via Euler angles. This limits the usefulness of the technique somewhat, although for many practical applications this limitation is of no consequence. This approach to writing controllers is referred to as *workspace control*, since x represents the configuration of the end-effector in the workspace of the manipulator.

Let $f : Q \to \mathbb{R}^p$ be a smooth and invertible mapping between the joint variables $\theta \in Q$ and the workspace variables $x \in \mathbb{R}^p$. In particular, this requires that $n = p$ so that the number of degrees of freedom of the robot equals the number of workspace variables x. We allow for the possibility that $p < 6$, in which case the workspace variables may only give a partial parameterization of $SE(3)$. An example of this situation is the SCARA robot, for which the position of the end-effector and its orientation with respect to the z-axis form a natural set of coordinates for specifying a task.

The dynamics of the manipulator in joint space has the form

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

where τ is the vector of joint torques and M, C, and N describe the

dynamic parameters of the system, as before.

We can rewrite the dynamics in terms of $x \in \mathbb{R}^p$ by using the Jacobian of the mapping $f : \theta \mapsto x$,

$$\dot{x} = J(\theta)\dot{\theta} \qquad J(\theta) = \frac{\partial f}{\partial \theta}.$$

Note that J is the Jacobian of the *mapping* $f : Q \to \mathbb{R}^p$ and not the manipulator Jacobian. Under the assumption that f is smooth and invertible, we can write

$$\dot{\theta} = J^{-1}\dot{x} \qquad \text{and} \qquad \ddot{\theta} = J^{-1}\ddot{x} + \frac{d}{dt}(J^{-1})\dot{x}.$$

We can now substitute these expressions into the manipulator dynamics and pre-multiply by $J^{-T} := (J^{-1})^T$ to obtain

$$J^{-T}M(\theta)J^{-1}\ddot{x} + \left(J^{-T}C(\theta,\dot{\theta})J^{-1} + J^{-T}M(\theta)\frac{d}{dt}(J^{-1})\right)\dot{x} + J^{-T}N(\theta,\dot{\theta}) = J^{-T}\tau.$$

We can write this in a more familiar form by defining

$$\begin{aligned}
\tilde{M} &= J^{-T}MJ^{-1} \\
\tilde{C} &= J^{-T}\left(CJ^{-1} + M\frac{d}{dt}\left(J^{-1}\right)\right) \\
\tilde{N} &= J^{-T}N \\
F &= J^{-T}\tau,
\end{aligned}$$

in which case the dynamics become

$$\tilde{M}(\theta)\ddot{x} + \tilde{C}(\theta,\dot{\theta})\dot{x} + \tilde{N}(\theta,\dot{\theta}) = F. \tag{4.55}$$

This equation represents the dynamics in terms of the workspace coordinates x and the robot configuration θ. We call $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ the *effective* parameters of the system. They represent the dynamics of the system as viewed from the workspace variables. Since f is locally invertible, we can in fact eliminate θ from these equations, and we see that equation (4.55) is nothing more than Lagrange's equations relative to the generalized coordinates x. However, since for most robots we measure θ directly and compute x via the forward kinematics, it is convenient to leave the θ dependence explicit.

Equation (4.55) has the same basic structure as the dynamics for an open-chain manipulator written in joint coordinates. In order to exploit this structure in our control laws, we must verify that some of the properties which we used in proving stability of controllers are also satisfied. The following lemma verifies that this is indeed the case.

Lemma 4.11. Structural properties of the workspace dynamics *Equation* (4.55) *satisfies the following properties:*

1. $\tilde{M}(\theta)$ *is symmetric and positive definite.*

2. $\dot{\tilde{M}} - 2\tilde{C} \in \mathbb{R}^{n \times n}$ *is a skew-symmetric matrix.*

Proof. Since J is an invertible matrix, property 1 follows from its definition. To show property 2, we calculate the $\dot{\tilde{M}} - 2\tilde{C}$:

$$\dot{\tilde{M}} - 2\tilde{C} = J^{-T}(\dot{\tilde{M}} - 2\tilde{C})J^{-1} + \frac{d}{dt}(J^{-T})\tilde{M}J^{-1} - J^{-T}\tilde{M}\frac{d}{dt}(J^{-1}).$$

A direct calculation shows that this matrix is indeed skew-symmetric. □

These two properties allow us to immediately extend the control laws in the previous section to workspace coordinates. For example, the computed torque control law becomes

$$F = \tilde{M}(\theta)\left(\ddot{x}_d - K_v\dot{e} - K_p e\right) + C(\theta, \dot{\theta})\dot{x} + N(\theta, \dot{\theta})$$
$$\tau = J^T F,$$

where x_d is the desired workspace trajectory and $e = x - x_d$ is the workspace error. The proof of stability for this control law is identical to that given previously. Namely, using the fact that $M(\theta)$ is positive definite, we can write the workspace error dynamics as

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

which is again a linear differential equation whose stability can be verified directly. The PD control law can be similarly extended to workspace coordinates.

The advantage of writing the control law in this fashion is that the matrices K_v and K_p now specify the gains directly in workspace coordinates. This simplifies the task of choosing the gains that are needed to accomplish a specific task. Furthermore, it eliminates the need to solve for the inverse mapping f^{-1} in order to control the robot. Instead, we only have to calculate the Jacobian matrix for f and its (matrix) inverse.

Notice that when the manipulator approaches a singular configuration relative to the coordinates x, the effective inertia $\tilde{M}$ gets very large. This is an indication that it is difficult to move in some directions and hence large forces produce very little motion. It is important to note that this singularity is strictly a function of our choice of parameterization. Such singularities never appear in the joint space of the robot.

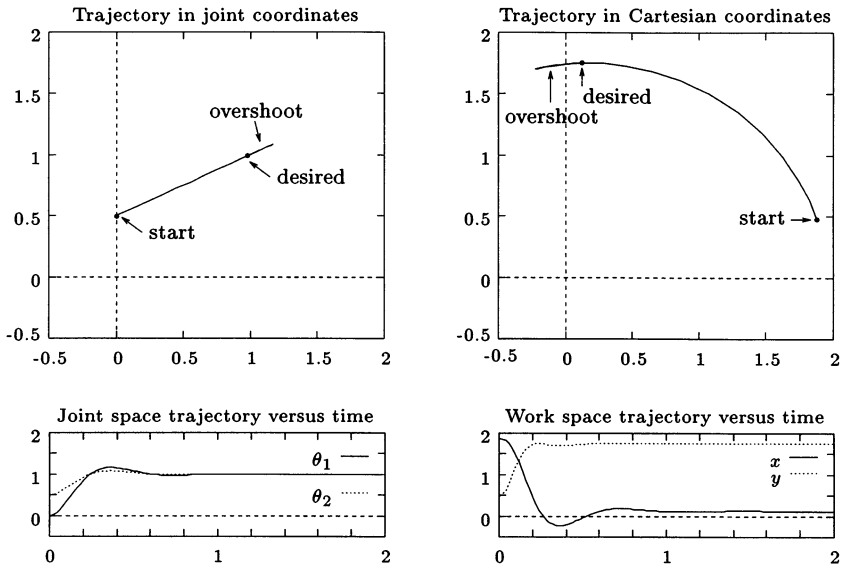


Figure 4.10: Step response of a joint space, computed torque controller.

Example 4.8. Comparison of joint space and workspace controllers

To illustrate some of the differences between implementing a controller in joint space versus workspace, we consider the control of a planar two degree of freedom robot. We take as our workspace variables the xy position of the end-effector.

Figure 4.10 shows the step response of a computed torque control law written in joint coordinates. Note that the trajectory of the end-effector, shown on the right, follows a curved path. The time response of the joint trajectories is a classical linear response for an underdamped mechanical system.

Figure 4.11 shows the step response of a computed torque control law written in workspace coordinates. Now the trajectory of the end-effector, including the overshoot, follows a straight line in the workspace and a curved line in the joint space.

6 Control of Constrained Manipulators

In this short section, we provide a brief treatment of the control of constrained manipulators. A more thorough development is given in Chapter 6.

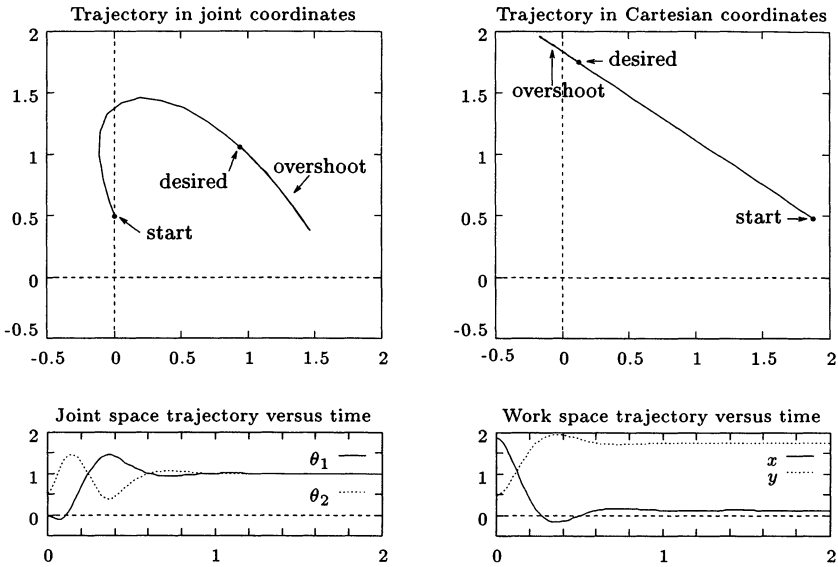


Figure 4.11: Step response of a workspace, computed torque controller.

6.1 Dynamics of constrained systems

Consider a problem in which we wish to move the tip of a robot along a surface and apply a force against that surface. For simplicity, we assume the surface is frictionless, although the analysis presented here can be readily extended to the more general case. We suppose that the surface we wish to move along can be described by a set of independent, smooth constraints

$$h_j(\theta_1, \ldots, \theta_n) = 0 \qquad j = 1, \ldots, k, \tag{4.56}$$

and that there exists a smooth, injective map $f : \mathbb{R}^{n-k} \to \mathbb{R}^n$ such that

$$h_j(f_1(\phi), \cdots, f_n(\phi)) = 0. \tag{4.57}$$

That is, $\phi \in \mathbb{R}^{n-k}$ parameterizes the allowable motion on the surface and $\theta = f(\phi)$ corresponds to a configuration in which the robot is in contact with the surface.

The control task is to follow a given trajectory $\phi_d(t)$ while applying a force against the surface. Since the surface is represented in joint space as the level set of the map $h(\theta) = 0$, the normal vectors to this surface are given by the span of the gradients of ∇h_i. (Since the surface is $n - k$ dimensional, the dimension of the space of normal vectors is k.) Any

torques of the form

$$\tau_N = \sum \lambda_j \nabla h_j(\theta) \tag{4.58}$$

correspond to normal forces applied against the surface. In the absence of friction, the work done by these torques is given by

$$\begin{aligned}\tau_N \cdot \dot{\theta} &= \sum \lambda_i \nabla h_i \cdot \dot{\theta} = \sum \lambda_i \left(\frac{\partial h_i}{\partial \theta} \dot{\theta} \right) \\ &= \sum \lambda_i \frac{d}{dt} (h(\theta)) = 0.\end{aligned}$$

Hence the normal forces do no work on the system and therefore cause no motion in the system. We assume that a desired normal force, specified by $\lambda_1(t), \ldots, \lambda_k(t)$, is given as part of the task description.

If the robot remains in contact with the surface, as desired, then the dynamics of the manipulator can be written in terms of ϕ. Differentiating $\theta = f(\phi)$, we have

$$\begin{aligned}\dot{\theta} &= \frac{\partial f}{\partial \phi} \dot{\phi} \\ \ddot{\theta} &= \frac{\partial f}{\partial \phi} \ddot{\phi} + \frac{d}{dt} \left(\frac{\partial f}{\partial \phi} \right) \dot{\phi}.\end{aligned} \tag{4.59}$$

These equations can be substituted into the robot equations of motion,

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

to yield

$$M(\theta) \frac{\partial f}{\partial \phi} \ddot{\phi} + \left(C(\theta, \dot{\theta}) \frac{\partial f}{\partial \phi} + M(\theta) \frac{d}{dt} \left(\frac{\partial f}{\partial \phi} \right) \right) \dot{\phi} + N(\theta, \dot{\theta}) = \tau, \tag{4.60}$$

where we have left M, C, and N in terms of θ to simplify notation.

Equation (4.60) can be made symmetric by multiplying both sides by $\frac{\partial f}{\partial \phi}^T$. Letting $J = \frac{\partial f}{\partial \phi}(\phi)$, we define

$$\begin{aligned}\tilde{M}(\phi) &= J^T M(f(\phi)) J \\ \tilde{C}(\phi, \dot{\phi}) &= J^T \left(C(f(\phi), J\dot{\phi}) J + M(f(\phi)) \dot{J} \right) \\ \tilde{N}(\phi, \dot{\phi}) &= J^T N(f(\phi), J\dot{\phi}) \\ F &= J^T \tau.\end{aligned} \tag{4.61}$$

Using these definitions, the projected equations of motion can be written as

$$\tilde{M}(\phi)\ddot{\phi} + \tilde{C}(\phi, \dot{\phi})\dot{\phi} + \tilde{N}(\phi, \dot{\phi}) = F. \tag{4.62}$$

This equation has the same form as the equation for an unconstrained manipulator. We shall show in Chapter 6 that equation (4.62) also satisfies the properties in Lemma 4.2. This is not particularly surprising since the coordinates ϕ were chosen to be a set of generalized coordinates under the assumption that the robot maintains contact with the surface.

It is important to keep in mind that equation (4.62) represents the dynamics of the system only along the surface given by the level sets $h(\theta) = 0$. By pre-multiplying by J^T, we have eliminated the information about the forces of constraint. For many applications, we are interested in regulating the forces of constraint and hence we must use the full equations of motion given in equation (4.60).

6.2 Control of constrained manipulators

The control task for a constrained robot system is to simultaneously regulate the position of the system along the constraint surface and regulate the forces of the system applied against this surface. In terms of analyzing stability, it is enough to analyze only the motion along the surface, since no movement occurs perpendicular to the surface. Of course, implicit in this point of view is that we maintain contact with the surface. If the manipulator is not physically constrained, this may require that we regulate the forces so as to insure that we are always pushing against the surface and never pulling away from it.

In this section we show how to extend the computed torque formalism presented earlier to regulate the position and force of the manipulator. We give only a sketch of the approach, leaving a more detailed discussion until Chapter 6, where we shall see that hybrid position/force control is just one example of the more general problem of controlling single and multiple robots interacting with each other and their environment.

We take as given a path on the constraint surface, specified by $\phi_d(t)$, and a normal force to be applied against the surface, specified by the Lagrange multipliers $\lambda_1(t), \ldots, \lambda_k(t)$ as in equation (4.58). Since we are interested in regulating the force applied against the constraint, it is important to insure that the position portion of the controller does not push against the constraint. Define

$$\tau_\phi = M(\theta)\frac{\partial f}{\partial \phi}(\ddot{\phi}_d - K_v\dot{e}_\phi - K_p e_\phi) + \left(C(\theta,\dot{\theta})\frac{\partial f}{\partial \phi} + M(\theta)\frac{d}{dt}\left(\frac{\partial f}{\partial \phi}\right)\right)\dot{\phi} + N(\theta,\dot{\theta}),$$

where $e_\phi = \phi - \phi_d$. This is the torque required to move the manipulator along the surface while applying no force against the surface. In other words, if we apply $\tau = \tau_\phi$ and remove the constraint completely, the manipulator will follow the correct path, as if the constraint were present.

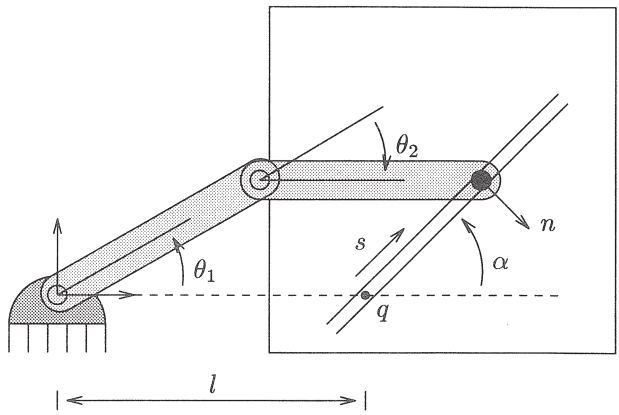


Figure 4.12: Planar manipulator moving in a slot.

To apply the appropriate normal forces, we simply add τ_N as defined in equation (4.58) to τ_ϕ. Since τ_N is in the normal direction to the constraint, it does not affect the position portion of the controller. Of course, this requires that the constraint surface actually be present to resist the normal forces applied to it. The complete control law is given by

$$\tau = \tau_\phi + \sum_{i=1}^{k} \lambda_i(t) \nabla h_i \tag{4.63}$$

where τ_ϕ is given above. We defer the analysis and proof of convergence for this control law until Chapter 6.

As in the previous control laws, the force control law presented here relies on accurate models of the robot and the surface. In particular, we note that the applied normal force does not use feedback to correct for model error, sensor noise, or other non-ideal situations.

6.3 Example: A planar manipulator moving in a slot

As a simple example of a constrained manipulator, consider the control of a two degree of freedom, planar manipulator whose end-effector is forced to lie in a slot, as shown in Figure 4.12. This system resembles a slider-crank mechanism, except that we are allowed to apply torques on both revolute joints, allowing us to control both the motion of the slider as well as the force applied against the slot. This example is easily adapted to a robot pushing against a wall, in which case the forces against the slot must always be pointed in a preferred direction.

We take the slot to be a straight line passing through the point $q = (l, 0)$ and making an angle α with respect to the x-axis of the base frame. The vector normal to the direction of the slot is given by

$$n = \begin{bmatrix} \sin\alpha \\ -\cos\alpha \end{bmatrix},$$

and the slot can be described as the set of all points $p \in \mathbb{R}^2$ such that $(p - q) \cdot n = 0$.

The constraint on the manipulator is obtained by requiring that the position of the end-effector remain in the slot. Letting $p(\theta) \in \mathbb{R}^2$ represent the position of the tool frame, this constraint becomes

$$h(\theta) = \left(p(\theta) - \begin{bmatrix} l \\ 0 \end{bmatrix} \right) \cdot \begin{bmatrix} \sin\alpha \\ -\cos\alpha \end{bmatrix} = 0.$$

Substituting the forward kinematics of the manipulator yields

$$\begin{aligned} h(\theta) &= (l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) - l) \sin\alpha \\ &\quad - (l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)) \cos\alpha \\ &= -l_1 \sin(\theta_1 - \alpha) - l_2 \sin(\theta_1 + \theta_2 - \alpha) - l \sin\alpha. \end{aligned}$$

The gradient of the constraint, which gives the direction of the normal force, is given by

$$\nabla h(\theta) = \begin{bmatrix} -l_1 \cos(\theta_1 - \alpha) - l_2 \cos(\theta_1 + \theta_2 - \alpha) \\ -l_2 \cos(\theta_1 + \theta_2 - \alpha) \end{bmatrix}.$$

Note that this is the direction of the normal force in *joint coordinates*. That is, joint torques applied in this direction will cause no motion, only forces against the side of the slot.

To parameterize the allowable motion along the slot, we let $s \in \mathbb{R}$ represent the position along the slot, with $s = 0$ denoting the point $q = (l, 0)$. Finding a function $f(s)$ such that $h(f(s)) = 0$ involves solving the inverse kinematics of the manipulator: given the position along the slot, we must find joint angles which achieve that position.

If the end of the manipulator is at a position s along the slot, then the xy coordinates of the end-effector are

$$\begin{aligned} x(s) &= l + s\cos\alpha \\ y(s) &= s\sin\alpha. \end{aligned}$$

Solving the inverse kinematics (see Chapter 3, Section 3) and assuming the elbow down solution, we have

$$f(s) = \begin{bmatrix} \theta_1(s) \\ \theta_2(s) \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{s\sin\alpha}{l + s\cos\alpha}\right) + \cos^{-1}\left(\frac{s^2 + 2ls\cos\alpha + l^2 + l_1^2 - l_2^2}{2l_1\sqrt{s^2 + 2ls\cos\alpha + l^2}}\right) \\ \pi + \cos^{-1}\left(\frac{l_1^2 + l_2^2 - s^2 - 2ls\cos\alpha - l^2}{2l_1 l_2}\right) \end{bmatrix}.$$

The Jacobian of the mapping is given by

$$J = \begin{bmatrix} \frac{-(s+l\cos\alpha)(s^2+2ls\cos\alpha+l^2-l_1^2+l_2^2)}{2l_1(s^2+2ls\cos\alpha+l^2)^{\frac{3}{2}}\sqrt{1-\frac{(s^2+2ls\cos\alpha+l^2+l_1^2-l_2^2)}{4l_1^2(s^2+2ls\cos\alpha+l^2)}}} + \frac{l\sin\alpha}{s^2+l^2+2ls\cos\alpha} \\ \frac{2(s+l\cos\alpha)}{\sqrt{4l_1^2l_2^2-(s^2+2ls\cos\alpha+l^2-l_1^2-l_2^2)^2}} \end{bmatrix}$$

(after some simplification).

This matrix can now be used to compute the equations of motion and derive an appropriate control law. In particular, the computed torque controller has the form

$$\tau = M(\theta)J\left(\ddot{s}_d - K_v\dot{e}_s - K_pe_s\right) + \left(C(\theta,\dot{\theta})J + M(\theta)\dot{J}\right)\dot{s} + \lambda n,$$

where $e_s = s - s_d$; λ is the desired force against the slot; $K_v, K_p \in \mathbb{R}$ are the gain and damping factors; and M and C are the generalized inertial and Coriolis matrices. The inertial parameters were calculated in Section 2.3 and are given by

$$M(\theta) = \begin{bmatrix} \alpha+\beta c_2 & \delta+\frac{1}{2}\beta c_2 \\ \delta+\frac{1}{2}\beta c_2 & \delta \end{bmatrix} \quad C(\theta,\dot{\theta}) = \begin{bmatrix} -\frac{1}{2}\beta s_2\dot{\theta}_2 & -\frac{1}{2}\beta s_2(\dot{\theta}_1+\dot{\theta}_2) \\ \frac{1}{2}\beta s_2\dot{\theta}_1 & 0 \end{bmatrix},$$

where

$$\begin{aligned} \alpha &= \mathcal{I}_{z1} + \mathcal{I}_{z2} + m_1r_1^2 + m_2(l_1^2 + r_2^2) \\ \beta &= m_2l_1l_2 \\ \delta &= \mathcal{I}_{z2} + m_2r_2^2. \end{aligned}$$

It is perhaps surprising that such a simple problem can have such an unwieldy solution. The difficulty is that we have cast the entire problem into the joint space of the manipulator, where the constraint $\theta = f(s)$ is a very complex looking curve.

A better way of deriving the equations of motion for this system is to rewrite the dynamics of the system in terms of workspace variables which describe the position of the end-effector (see Exercise 12). Once written in this way, the constraint that the end of the manipulator remain in the slot is a very simple one. This is the basic approach used in Chapter 6, where we present a general framework which incorporates this example and many other constrained manipulation systems.

7 Summary

The following are the key concepts covered in this chapter:

1. The equations of motion for a mechanical system with Lagrangian $L = T(q, \dot{q}) - V(q)$ satisfies *Lagrange's equations*:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i,$$

   where $q \in \mathbb{R}^n$ is a set of generalized coordinates for the system and $\Upsilon \in \mathbb{R}^n$ represents the vector of generalized external forces.

2. The equations of motion for a rigid body with configuration $g(t) \in SE(3)$ are given by the *Newton-Euler equations*:

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b,$$

   where m is the mass of the body, $\mathcal{I}$ is the inertia tensor, and $V^b = (v^b, \omega^b)$ and F^b represent the instantaneous body velocity and applied body wrench.

3. The equations of motion for an open-chain robot manipulator can be written as

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

   where $\theta \in \mathbb{R}^n$ is the set of joint variables for the robot and $\tau \in \mathbb{R}^n$ is the set of actuator forces applied at the joints. The dynamics of a robot manipulator satisfy the following properties:

   (a) $M(\theta)$ is symmetric and positive definite.

   (b) $\dot{M} - 2C \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix.

4. An equilibrium point x^* for the system $\dot{x} = f(x, t)$ is *locally asymptotically stable* if all solutions which start near x^* approach x^* as $t \to \infty$. Stability can be checked using the *direct method of Lyapunov*, by finding a locally positive definite function $V(x, t) \geq 0$ such that $-\dot{V}(x, t)$ is a locally positive definite function along trajectories of the system. In situations in which $-\dot{V}$ is only positive semi-definite, *Lasalle's invariance principle* can be used to check asymptotic stability. Alternatively, the *indirect method of Lyapunov* can be employed by examining the linearization of the system, if it exists. Global exponential stability of the linearization implies local exponential stability of the full nonlinear system.

5. Using the form and structure of the robot dynamics, several control laws can be shown to track arbitrary trajectories. Two of the most common are the *computed torque control law*,

$$\tau = M(\theta)(\ddot{\theta}_d + K_v\dot{e} + K_p e) + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}),$$

and an *augmented PD control law*,

$$\tau = M(\theta)\ddot{\theta}_d + C(\theta,\dot{\theta})\dot{\theta}_d + N(\theta,\dot{\theta}) + K_v\dot{e} + K_p e.$$

Both of these controllers result in exponential trajectory tracking of a given joint space trajectory. Workspace versions of these control laws can also be derived, allowing end-effector trajectories to be tracked without solving the inverse kinematics problem. Stability of these controllers can be verified using Lyapunov stability.

8 Bibliography

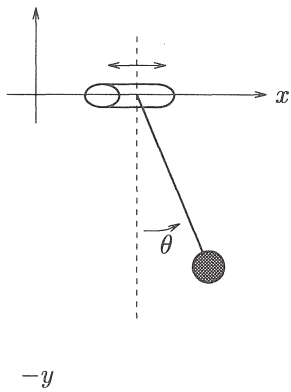
The Lagrangian formulation of dynamics is classical; a good treatment can be found in Rosenberg [100] or Pars [90]. Its application to the dynamics of a robot manipulator can be found in most standard textbooks on robotics, for example [2, 21, 35, 53, 111].

The geometric formulation of the of the equations of motion for kinematic chains presented in Section 3.3 is based on the recent work of Brockett, Stokes, and Park [15, 88]. This is closely related to the spatial operator algebra formulation of Rodriguez, Jain, and Kreutz-Delgado [46, 99], in which the tree-like nature of the system is more fully exploited in computing inertial properties of the system.

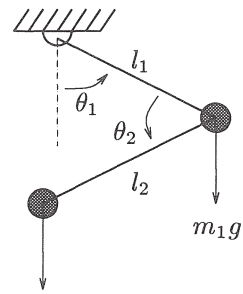
The literature on control of robot manipulators is vast. An excellent treatment, covering many of the different approaches to robot control, is given by Spong and Vidyasagar [111]. The collection [110] also provides a good survey of recent research in this area. The modified PD control law presented in Section 5 was originally formulated by Koditschek [52]. For a survey of manipulator control using exact linearization techniques, see Kreutz [54]. The use of skew terms in Lyapunov functions to prove exponential stability for PD controllers has been pointed out, for example, by Wen and Bayard [120].

9 Exercises

1. Derive the equations of motion for the systems shown below.



(a)

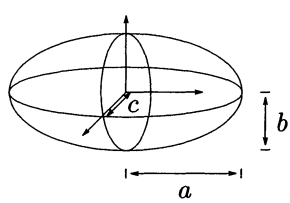


(b)

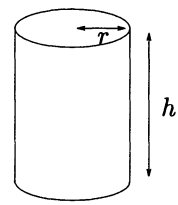
   (a) Pendulum on a wire: an idealized planar pendulum whose pivot is free to slide along a horizontal wire. Assume that the top of the pendulum can move freely on the wire (no friction).

   (b) Double pendulum: two masses connected together by massless links and revolute joints.

2. Compute the inertia tensor for the objects shown below.



(a) Ellipsoid



(b) Cylinder

3. *Transformation of the generalized inertia matrix*
   Show that under a change of body coordinate frame from B to C, the generalized inertia matrix for a rigid body is given by

$$\mathcal{M}_c = \mathrm{Ad}_{g_{bc}}^T \, \mathcal{M}_b \, \mathrm{Ad}_{g_{bc}} = \begin{bmatrix} mI & mR_{bc}^T \widehat{p}_{bc} R_{bc} \\ -mR_{bc}^T \widehat{p}_{bc} R_{bc} & R_{bc}^T (\mathcal{I} - m\widehat{p}_{bc}^2) R_{bc} \end{bmatrix},$$

   where g_{bc} denotes the rigid motion taking C to B, and $\mathcal{M}_b$ and $\mathcal{M}_c$ are the generalized inertia matrices expressed in frame B and frame C.

4. Show that Euler's equation written in spatial coordinates is given by

$$\mathcal{I}'\dot{\omega}^s + \omega^s \times \mathcal{I}'\omega^s = \tau,$$

where $\mathcal{I}' = R\mathcal{I}R^T$ and τ is the torque applied to the center of mass of the rigid body, written in spatial coordinates.

5. Calculate the Newton-Euler equations in spatial coordinates.

6. Show that it is possible to choose M and C such that the Newton-Euler equations can be written as

$$M\dot{V}^b + C(g, \dot{g})V^b = F^b,$$

where $M > 0$ and $\dot{M} - 2C$ is a skew-symmetric matrix.

7. Verify that the equations of motion for a planar, two-link manipulator, as given in equation (4.11), satisfy the properties of Lemma 4.2.

8. *Passivity of robot dynamics*
Let $H = T + V$ be the total energy for a rigid robot. Show that if $\dot{M} - 2C$ is skew-symmetric, then energy is conserved, i.e., $\dot{H} = \dot{\theta} \cdot \tau$.

9. Show that the workspace version of the PD control law results in exponential trajectory tracking.

10. Show that the control law

$$\tau = M(\theta)(\ddot{\theta}_d + \lambda\dot{e}) + C(\theta, \dot{\theta})(\dot{\theta}_d + \lambda e) + N(\theta, \dot{\theta}) + K_v\dot{e} + K_p e$$

results in exponential trajectory tracking when $\lambda \in \mathbb{R}$ is positive and $K_v, K_p \in \mathbb{R}^{n \times n}$ are positive definite [108].

11. Show that the matrix

$$\begin{bmatrix} \epsilon A & \epsilon B \\ \epsilon B^T & C + \epsilon D \end{bmatrix}$$

is positive definite if A and C are symmetric, positive definite, and $\epsilon > 0$ is chosen sufficiently small.

12. *Hybrid control using workspace coordinates*
Consider the constrained manipulation problem described in Section 6.3. Let $p_{st}(\theta) \in \mathbb{R}^2$ be the coordinates of the end-effector and let $w = p(\theta)$ represent a set of workspace coordinates for the system.

    (a) Compute the matrix $J(\theta)$ which is used to convert the joint space dynamics into workspace dynamics (as in Section 5.4).

(b) Compute the constraint function in terms of the workspace variables and find a parameterization $f : \mathbb{R} \to \mathbb{R}^2$ which maps the slot position to the workspace coordinates. Let $K(s)$ represent the Jacobian of the mapping $w = f(s)$.

(c) Write the dynamics of the constrained system in terms of ω and its derivatives, the dynamic parameters of the unconstrained system, and the matrices $J(\theta)$ and $K(s)$.

(d) Verify that the equations of motion derived in step (c) are the same as the equations of motion derived in Section 6.3. In particular, show that τ_N and the inertia matrix $\tilde{M}(s)$ are the same in both cases.

Chapter 5

Multifingered Hand Kinematics

DOI: 10.1201/9781315136370-5
This chapter has been made available under a CC-BY 4.0 license.

In this chapter, we study the kinematics of a multifingered robot hand grasping an object. Given a description of the fingers and the object, we derive the relationships between finger and object velocities and forces, and study conditions under which a grasp can be used to manipulate an object. In addition to the usual fixed contact case, we also include a complete derivation of the kinematics of grasp when the fingers are allowed to roll or slide along the object.

1 Introduction to Grasping

Traditional robot manipulators used in industry are composed of a large arm with a simple gripper attached as an end-effector. This type of robot is effective for tasks in which large motions of the payload are required, but it cannot accurately perform precise movements of the payload, such as those that might be required in an assembly task. With a traditional manipulator, fine motions of a grasped object require precise movements of the joints of the robot arm. Due to the size of the links in a typical robot manipulator, moving the entire arm is rarely an effective means of achieving accurate motions of a grasped object. The situation is analogous to a person trying to write with a pencil by moving his or her entire arm.

A second disadvantage with traditional robot manipulators is that for a given gripper, only a small class of objects can be grasped. A parallel jaw gripper, for example, is very effective at grasping objects which have parallel faces. It cannot, however, be used to "stably" grasp a tetrahedron. This limitation is sometimes overcome by equipping the robot arm with a tool changer, which allows different grippers to be

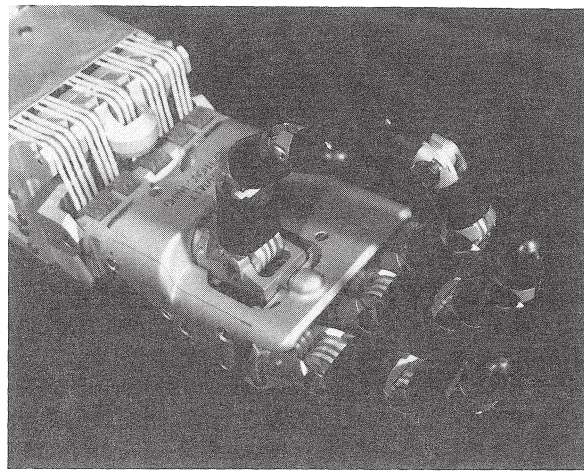


Figure 5.1: The Utah/MIT hand. (Photo courtesy of Sarcos, Inc.)

attached to a robot in an efficient fashion. While this effectively extends the class of objects which can be lifted, it does not specifically address the fine motion problem.

In this chapter and the next, we investigate the use of multifingered robot hands as an alternative method for overcoming these difficulties. A multifingered hand is a set of robots which is attached to the end of a larger robot arm for the purpose of manipulation. Since a robot hand is typically smaller than the robot to which it is attached, it is able to improve the overall accuracy of the robot. Further, the extra degrees of freedom in a multifingered hand make it possible to grasp a large class of objects with a single "end-effector."

The price of using a multifingered hand is the complexity of the overall system. A robot arm with a multifingered end-effector has many degrees of freedom, complicating both the kinematic and dynamic analyses of the system. In particular, since the hand is in contact with the object being manipulated, we must study the kinematics and dynamics of mechanical systems with contact constraints. Additionally, the increased degrees of freedom of the system increase the difficulty of planning a feasible grasp to perform a given task.

Because of the added complexity inherent in the use of robot hands, it is important to realize that a multifingered robot hand is not the answer to every manipulation problem. The use of custom end-effectors can solve a large number of problems in such areas as manufacturing and materials

transport. Furthermore, custom end-effectors are capable of generating stable and rigid grasps by design; a similar grasp using an articulated hand requires the use of feedback control and results in an overall decrease in the rigidity of the grasp. Nonetheless, for many situations, a multifingered robot hand attached to a robot arm is an attractive solution to a difficult problem. The principles involved in the study of robot hands are applicable to a number of other areas, most notably coordinated manipulation between robots. Indeed, if we view two (large) robots as fingers of a hand, then the problem of coordinated lifting reduces to a problem in grasping. With this point of view in mind, we will present the study of multifingered hands in a framework which is applicable to a large class of robot systems.

Overview

We break the study of multifingered robot hands into two basic parts: kinematics and planning, and dynamics and control. In each of these areas, the techniques presented in previous chapters are extended to account for the additional complexity of a robot hand. We will also see many new techniques which are unique to the study of multifingered hands, such as grasp planning and the kinematics of rolling contact.

We begin with a detailed study of the kinematics of a multifingered robot hand. Given models of an object, a set of robot fingers, and the contact between the fingers and the object, we wish to find the relationship between forces and motions of the object and fingers. We will be interested primarily with the kinematics of this system when the fingers do not slip on the object. An implicit assumption in studying the kinematics of the hand is that the contact locations are known (or can be measured). Under this assumption, we compute the fundamental grasping constraint which governs the motion of the hand.

The grasp planning problem is to determine a set of contact locations for the object and the fingers. To do so, we first characterize desirable properties of a grasp. These properties include:

1. *The ability to resist external forces.* Given a wrench applied to an object, we wish to apply contact forces which generate an opposing wrench. We will refer to a grasp in which the fingers can resist arbitrary external forces as a *force-closure* grasp.

2. *The ability to dextrously manipulate the object.* In order to manipulate an object, we must be able to move the object in a way compatible with the desired task. Depending on the task, this may require independent motion in all directions or only some directions. We refer to a grasp in which the fingers can accommodate arbitrary object motions as a *manipulable* grasp.

Given an object, we seek to choose contacts so that these properties hold wherever possible. We will briefly present some procedures for choosing the locations of the contacts for some simple cases which illustrate the techniques available.

Chapter 6 studies the dynamics and control problem for multifingered robot hands. We extend the dynamic formulation presented in Chapter 4 to include robotic systems with contact constraints. In fact, it is possible to do so in such a way that all of the control laws which can be applied to robot manipulators can be immediately extended and applied to multifingered robot hands. We also present extensions for dynamics and control of redundant robots, as these are often present in multifingered robot systems.

Throughout this chapter and the next, we make two assumptions to allow precise analysis:

1. The object is a rigid body in contact with a rigid link robot.

2. Accurate models of the fingers and object are given.

The relaxation of these conditions is a topic of current research (see, for example, [48, 74]).

2 Grasp Statics

We begin by studying a particularly simple case in which all contacts between the fingers and the object are idealized as point contacts at a fixed location. This case allows one to ignore the possibility that a finger rolls or slides along the surface of the object (a possibility which we shall study in some detail later). We also begin by ignoring the kinematics of the fingers which make up the hand: we consider only the transmission of forces between a set of contacts and the object.

2.1 Contact models

A *contact* between a finger and an object can be described as a mapping between forces exerted by the finger at the point of contact and the resultant wrenches at some reference point on the object. In order to simplify the formulation of the dynamics in Chapter 6, we always choose the object reference point to be the center of mass of the object. We represent the forces at the contacts and on the object in terms of a set of coordinate frames attached at each contact location and the object reference point. We assume that the location of the contact point on the object is fixed.

For convenience, we shall always choose the contact coordinate frame, C_i, such that the z-axis of this frame points in the direction of the inward

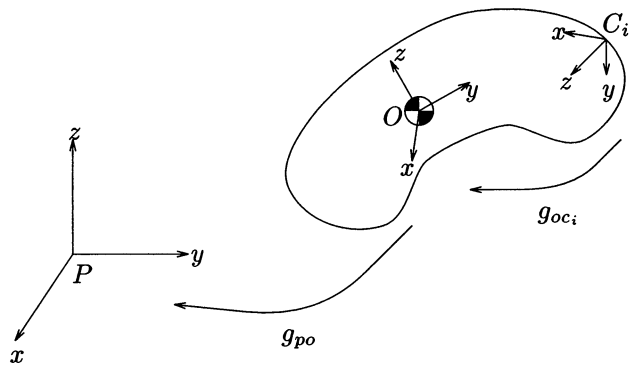


Figure 5.2: Coordinate frames for contact and object forces.

surface normal at the point of contact (as shown in Figure 5.2). We describe the contact location by its relative position and orientation with respect to the object reference frame, O. That is,

$$g_{oc_i} = (p_{oc_i}, R_{oc_i}) \in SE(3)$$

is the location of the contact with respect to the object. The action of g_{oc_i} is to take the coordinates of a point given in the contact coordinate frame and return the coordinates of the same point in the object coordinate frame. The configuration of the object relative to a fixed palm frame is given by $x_o := g_{po} \in SE(3)$. The force applied by a contact is modeled as a wrench F_{c_i} applied at the origin of the contact frame, C_i.

Typically, a finger will not be able to exert forces in every direction; several simple contact models are used to classify common contact configurations. We begin by studying the simplest of these contact types: one in which the finger is only allowed to apply normal forces to the object at the contact location. We then extend this analysis to include some simple models of friction.

A *frictionless point contact* is obtained when there is no friction between the fingertip and the object. In this case, forces can only be applied in the direction normal to the surface of the object and hence we can represent the applied wrench as

$$F_{c_i} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} f_{c_i} \quad f_{c_i} \geq 0, \tag{5.1}$$

where $f_{c_i} \in \mathbb{R}$ is the magnitude of the force applied by the finger in the normal direction. The requirement that f_{c_i} be positive models the fact that a contact of this type can push on an object, but it cannot pull on the object.

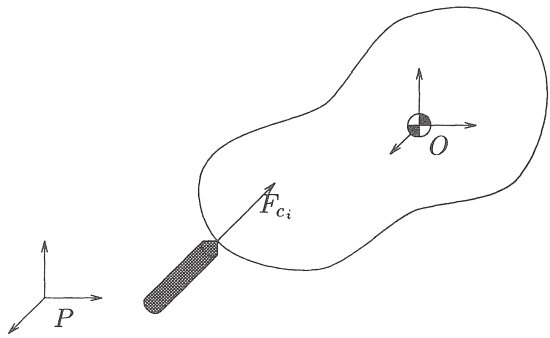


Figure 5.3: Frictionless point contact.

Frictionless point contacts almost never occur in a practical situation, but they can serve as a useful *model* for contacts in which the friction between the finger and the object is low or unknown. Since a frictionless contact cannot exert forces except in the normal direction, modeling a contact as frictionless insures that we do not rely on frictional forces when we manipulate the object.

For grasps in which we do wish to make use of frictional forces, we must provide a model for friction. We will use a simple model which is often referred to as the *Coulomb friction model.* We would like to describe how much force a contact can apply in the tangent directions to a surface as a function of the applied normal force. The Coulomb friction model is an empirical model which asserts that the allowed tangential force is proportional to the applied normal force, and the constant of proportionality is a function of the materials which are in contact.

If we let $f^t \in \mathbb{R}$ denote the magnitude of the tangential force and $f^n \in \mathbb{R}$ denote the magnitude of the normal force, Coulomb's law states that slipping begins when

$$|f^t| > \mu f^n,$$

where $\mu > 0$ is the (static) coefficient of friction. This implies that the range of tangential forces which can be applied at a contact is given by

$$|f^t| \leq \mu f^n. \tag{5.2}$$

In particular, we see that f^n must be positive in order for this relationship to hold for at least some non-zero f^t.

Equation (5.2) can be represented geometrically, as shown in Figure 5.4. The set of forces which can be applied at a contact must lie in a cone centered about the surface normal. This cone is called the *friction cone*; the angle of the cone with respect to the normal is given by

$$\alpha = \tan^{-1} \mu.$$

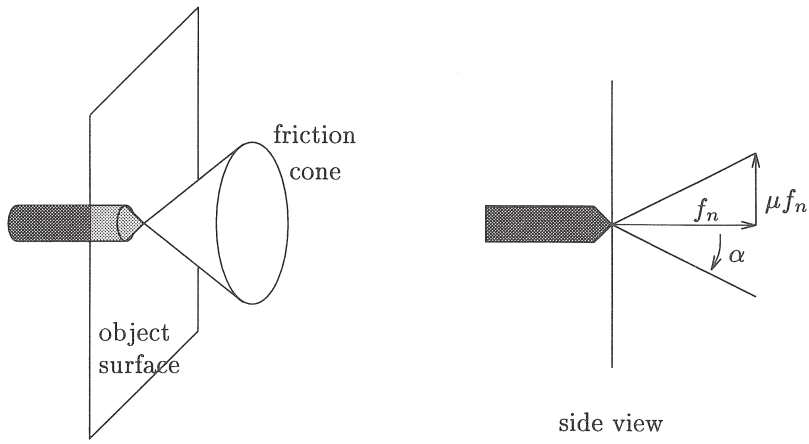


Figure 5.4: Geometric interpretation of the Coulomb friction model.

A short table of friction coefficients for common materials is given in Table 5.1. Typical values of μ are less than 1, and hence the friction cone angle is typically less than 45°.

A *point contact with friction* model is used when friction exists between the fingertip and the object, in which case forces can be exerted in any direction that is within the friction cone for the contact. We represent the wrench applied to the object with respect to a basis of directions which are consistent with the friction model:

$$F_{c_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} f_{c_i} \qquad f_{c_i} \in FC_{c_i}, \tag{5.3}$$

where

$$FC_{c_i} = \{f \in \mathbb{R}^3 : \sqrt{f_1^2 + f_2^2} \leq \mu f_3,\ f_3 \geq 0\}.$$

A more realistic contact model is the *soft-finger* contact. Here we allow not only forces to be applied in a cone about the surface normal, but also torques about that normal. For simplicity, we model the torques as being limited by a torsional friction coefficient. The applied contact wrench is

$$F_{c_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} f_{c_i} \qquad f_{c_i} \in FC_{c_i} \tag{5.4}$$

and the friction cone becomes

$$FC_{c_i} = \{f \in \mathbb{R}^4 : \sqrt{f_1^2 + f_2^2} \leq \mu f_3,\ f_3 \geq 0,\ |f_4| \leq \gamma f_3\},$$

Table 5.1: Static friction coefficients for some common materials. (Source: CRC Handbook of Chemistry and Physics)

Steel on steel	0.58	Wood on wood	0.25-0.5
Polyethylene on steel	0.3–0.35	Wood on metals	0.2-0.6
Polyethylene on self	0.5	Wood on leather	0.3–0.4
Rubber on solids	1–4	Leather on metal	0.6

where $\gamma > 0$ is the coefficient of torsional friction.

In general, we model a contact using a *wrench basis*, $B_{c_i} \in \mathbb{R}^{p \times m_i}$, and a friction cone, FC_{c_i}. In all of our examples, we chose $p = 6$, the dimension of the space of generalized forces that can be applied in $SE(3)$. Other choices are possible, the most common being $p = 3$, which is used for planar grasping. The dimension of the wrench basis, m_i, indicates the number of independent forces that can be applied by the contact. We require that FC_{c_i} satisfy the following properties:

1. FC_{c_i} is a closed subset of $\mathbb{R}^{m_i}$ with non-empty interior.
2. $f_1, f_2 \in FC_{c_i} \implies \alpha f_1 + \beta f_2 \in FC_{c_i}$ for $\alpha, \beta > 0$.

The set of allowable contact forces applied by a given contact is:

$$F_{c_i} = B_{c_i} f_{c_i} \qquad f_{c_i} \in FC_{c_i}. \tag{5.5}$$

Several common contact types are summarized in Table 5.2. Other contacts, such as line and face contacts, are explored in the exercises.

2.2 The grasp map

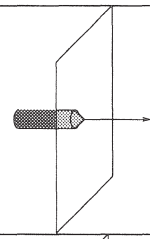
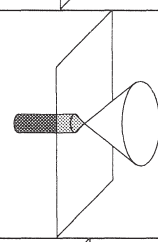
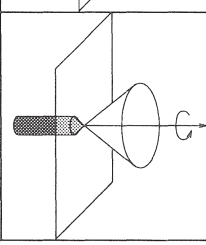
To determine the effect of the contact forces on the object, we transform the forces to the object coordinate frame. Let (p_{oc_i}, R_{oc_i}) be the configuration of the i^{th} contact frame relative to the object frame. Then, the force exerted by a single contact can be written in object coordinates as

$$F_o = \mathrm{Ad}^T_{g^{-1}_{oc_i}} F_{c_i} = \begin{bmatrix} R_{oc_i} & 0 \\ \widehat{p}_{oc_i} R_{oc_i} & R_{oc_i} \end{bmatrix} B_{c_i} f_{c_i}, \qquad f_{c_i} \in FC_{c_i}.$$

The matrix $\mathrm{Ad}^T_{g^{-1}_{oc_i}}$ is the wrench transformation matrix which maps contact wrenches to object wrenches. For brevity we shall often write (p_{c_i}, R_{c_i}) for the configuration of the i^{th} contact frame, dropping the o subscript. We define the contact map, $G_i \in \mathbb{R}^{p \times m_i}$, to be the linear map between contact forces, represented with respect to B_{c_i}, and the object wrench:

$$G_i := \mathrm{Ad}^T_{g^{-1}_{oc_i}} B_{c_i}.$$

Table 5.2: Common contact types.

Contact type	Picture	Wrench basis	FC
Frictionless point contact		$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$f_1 \geq 0$
Point contact with friction		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\sqrt{f_1^2 + f_2^2} \leq \mu f_3$ $f_3 \geq 0$
Soft-finger		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\sqrt{f_1^2 + f_2^2} \leq \mu f_3$ $f_3 \geq 0$ $\lvert f_4 \rvert \leq \gamma f_3$

If we have k fingers contacting an object, the total wrench on the object is the sum of the object wrenches due to each finger. The map between the contact forces and the total object force is called the *grasp map*, $G\colon \mathbb{R}^m \to \mathbb{R}^p$, $m = m_1 + \cdots + m_k$. Since each contact map is linear and wrenches can be superposed (as long as they are all written in the same coordinate frame), the net object wrench is

$$F_o = G_1 f_{c_1} + \cdots + G_k f_{c_k} = \begin{bmatrix} G_1 & \cdots & G_k \end{bmatrix} \begin{bmatrix} f_{c_1} \\ \vdots \\ f_{c_k} \end{bmatrix},$$

and the grasp map is

$$\boxed{G = \begin{bmatrix} \mathrm{Ad}^T_{g_{oc_1}^{-1}} B_{c_1} & \cdots & \mathrm{Ad}^T_{g_{oc_k}^{-1}} B_{c_k} \end{bmatrix}} \tag{5.6}$$

With this definition, the object wrench can be written

$$F_o = G f_c \qquad f_c \in FC, \tag{5.7}$$

where

$$f_c = (f_{c_1}, \ldots, f_{c_k}) \in \mathbb{R}^m$$
$$FC = FC_{c_1} \times \cdots \times FC_{c_k} \subset \mathbb{R}^m$$
$$m = m_1 + \cdots + m_k.$$

Thus, a grasp is completely described by the grasp map G and the friction cone FC.

Definition 5.1. Representation of a grasp
A complete description of a *grasp* consists of a matrix $G \in \mathbb{R}^{p \times m}$ and a set $FC \subset \mathbb{R}^m$ which satisfies the following properties:

1. FC is a closed subset of $\mathbb{R}^m$ with non-empty interior.
2. $f_1, f_2 \in FC \implies \alpha f_1 + \beta f_2 \in FC$ for $\alpha, \beta > 0$.

The set of wrenches that can be applied to the object by the contacts has the form

$$F_o = G f_c \qquad f_c \in FC.$$

From now on, we will represent a grasp by the quantities G and FC. We often omit explicit mention of the friction cone and refer to G as a grasp.

Example 5.1. Grasp map for frictionless point contacts
Consider first the case of several point contacts touching an object. Then, each contact wrench can be written as

$$F_o = \begin{bmatrix} R_{c_i} & 0 \\ \widehat{p}_{c_i} R_{c_i} & R_{c_i} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} f_{c_i} \quad f_{c_i} \geq 0.$$

Performing the matrix multiplication, we see that the object wrench has the form

$$F_o = \begin{bmatrix} n_{c_i} \\ p_{c_i} \times n_{c_i} \end{bmatrix} f_{c_i}$$

where n_{c_i} is the direction of the inward surface normal written in the object coordinate frame. Combining the effect of each of the fingers yields:

$$F_o = \begin{bmatrix} n_{c_1} & \cdots & n_{c_k} \\ p_{c_1} \times n_{c_1} & \cdots & p_{c_k} \times n_{c_k} \end{bmatrix} \begin{bmatrix} f_{c_1} \\ \vdots \\ f_{c_k} \end{bmatrix} = G f_c, \qquad \begin{matrix} F_o \in \mathbb{R}^6 \\ f_{c_i} \geq 0. \end{matrix}$$

Example 5.2. Soft-finger grasp of a box
Consider a box grasped by two soft-finger contacts, as shown in Figure 5.5. The position of the contact frame with respect to the object

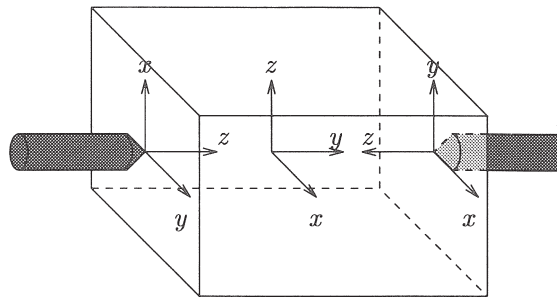


Figure 5.5: Box grasped by two fingers.

frame is given by

$$R_{c_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad p_{c_1} = \begin{bmatrix} 0 \\ -r \\ 0 \end{bmatrix} \qquad R_{c_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad p_{c_2} = \begin{bmatrix} 0 \\ r \\ 0 \end{bmatrix}.$$

The grasp map for each finger is obtained by transforming the standard wrench basis into the object coordinate frame:

$$G_i = \begin{bmatrix} R_{c_i} & 0 \\ \widehat{p}_{c_i} R_{c_i} & R_{c_i} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Calculating and combining these gives the grasp map,

$$G = \left[\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline -r & 0 & 0 & 0 & 0 & +r & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & +r & 0 & 0 & -r & 0 & 0 & 0 \end{array}\right],$$

with contact forces

$$f_c = (f_{c_1}^1, f_{c_1}^2, f_{c_1}^3, f_{c_1}^4,\ f_{c_2}^1, f_{c_2}^2, f_{c_2}^3, f_{c_2}^4) \in \mathbb{R}^8.$$

The friction cone in this coordinate frame is given by:

$$\begin{aligned} FC &= FC_{c_1} \times FC_{c_2} \\ FC_{c_1} &= \left\{ f_c : \sqrt{(f_{c_1}^1)^2 + (f_{c_1}^2)^2} \le \mu f_{c_1}^3,\ |f_{c_1}^4| \le \gamma f_{c_1}^3,\ f_{c_1}^3 \ge 0 \right\} \\ FC_{c_2} &= \left\{ f_c : \sqrt{(f_{c_2}^1)^2 + (f_{c_2}^2)^2} \le \mu f_{c_2}^3,\ |f_{c_2}^4| \le \gamma f_{c_2}^3,\ f_{c_2}^3 \ge 0 \right\}. \end{aligned}$$

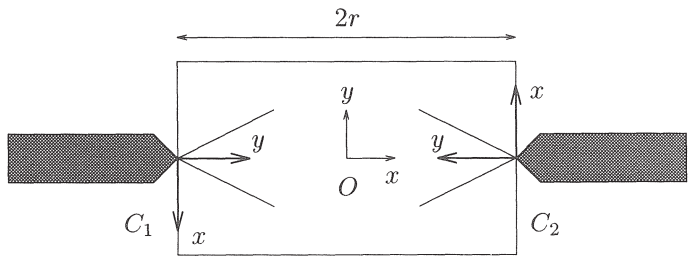


Figure 5.6: Planar grasping.

Example 5.3. Planar grasp of a rectangle
Consider the planar grasp shown in Figure 5.6. Rather than analyze this grasp in $SE(3)$, we specialize our results to $SE(2)$. A wrench in $SE(2)$ is represented by a linear component $f \in \mathbb{R}^2$ and an angular component $\tau \in \mathbb{R}$. f corresponds to the forces in the plane and τ to the torque about the normal to the plane (see Exercise 2.11). Wrenches in $SE(2)$ are transformed by the rule

$$\begin{bmatrix} f_o \\ \tau_o \end{bmatrix} = \begin{bmatrix} R_{c_i} & 0 \\ [-p_y \quad p_x]\, R_{c_i} & 1 \end{bmatrix} \begin{bmatrix} f_{c_i} \\ \tau_{c_i} \end{bmatrix},$$

where $R_{c_i} \in SO(2)$ and $p_{c_i} = (p_x, p_y) \in \mathbb{R}^2$ represents the location of c_i relative to the object reference frame.

Given these definitions, we can proceed to analyze Figure 5.6. In $SE(2)$, we shall choose contact coordinate frames such that the y-axis points in the direction of the inward surface normal. With this choice, the contact locations are

$$R_{c_1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad R_{c_2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$
$$p_{c_1} = \begin{bmatrix} -r \\ 0 \end{bmatrix} \qquad p_{c_2} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

A wrench basis for a point contact with friction is

$$B_{c_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

with the friction cone constraint written as

$$f_{c_i} \in FC_{c_i}$$
$$FC_{c_i} = \{f \in \mathbb{R}^2 : |f_1| \leq \mu f_2,\ f_2 \geq 0\}.$$

Finally, the grasp map is given by

$$G = \begin{bmatrix} \mathrm{Ad}^T_{g_{oc_1}^{-1}} B_{c_1} & \mathrm{Ad}^T_{g_{oc_2}^{-1}} B_{c_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ r & 0 & r & 0 \end{bmatrix}.$$

3 Force-Closure

An important property of a grasp is the ability to balance external object wrenches by applying appropriate finger wrenches at the contact points. For example, if we are using a multifingered hand to lift an object from a table, we must be able to exert forces on the object which act in the opposite direction to gravity. Depending on the task, we may also need to resist wrenches in other directions. Doing so is complicated by the fact that we must insure that the applied finger forces remain in the friction cone at all times so as to avoid slippage of the fingers on the surface of the object.

3.1 Formal definition

If a grasp can resist any applied wrench, we say that such a grasp is *force-closure.*[1] Formally, we make the following definition:

Definition 5.2. Force-closure grasp
A grasp is a *force-closure* grasp if given any external wrench $F_e \in \mathbb{R}^p$ applied to the object, there exist contact forces $f_c \in FC$ such that

$$Gf_c = -F_e.$$

The following proposition follows directly from the definition.

Proposition 5.1. Characterization of all force-closure grasps
A grasp is force-closure if and only if $G(FC) = \mathbb{R}^p$.

A key feature of a force-closure grasp is the existence of *internal forces.* An internal force is a set of contact forces which result in no net force on the object. In the previous (planar) example, we see that

$$f_N = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \Longrightarrow \quad Gf_N = 0.$$

This motivates the following definition. Let $\text{int}(FC)$ denote the interior of the friction cone.

Definition 5.3. Internal forces
If $f_N \in \mathcal{N}(G) \cap FC$, then f_N is an *internal force.* If $f_N \in \mathcal{N}(G)$ and $f_N \in \text{int}(FC)$, then it is called a *strictly internal force.*

[1] To be consistent with the literature, we use the term force-closure instead of wrench-closure.

Internal forces can be used to insure that contact forces satisfy friction cone constraints. Note that an internal force is a set of *contact forces*, represented with respect to the wrench basis at each contact. The following proposition shows that the existence of a strictly internal force is a necessary condition for a grasp to be force-closure.

Proposition 5.2. Necessity of internal forces
A grasp is force-closure if and only if G is surjective and there exists a vector of contact forces $f_N \in \mathcal{N}(G)$ such that $f_N \in \text{int}(FC)$.

Proof. (Sufficiency) Choose $F_o \in \mathbb{R}^p$ and let f_c' be any vector such that $F_o = Gf_c'$. Since G is surjective, such an f_c' must exist. We will show that there exists an α such that $f_c' + \alpha f_N \in \text{int}(FC)$. Notice that

$$\lim_{\alpha \to \infty} \frac{f_c' + \alpha f_N}{\alpha} = f_N \in \text{int}(FC);$$

therefore, there exists α' sufficiently large such that

$$\frac{f_c' + \alpha' f_N}{\alpha'} \in \text{int}(FC) \subset FC.$$

From the properties of the friction cone, it follows that

$$f_c := f_c' + \alpha' f_N \in \text{int}(FC)$$

and $Gf_c = Gf_c' = F_o$.

(Necessity) Suppose that a grasp is force-closure. Choose $f_1 \in \text{int}(FC)$ such that $F_o = Gf_1 \neq 0$. Choose $f_2 \in FC$ such that $Gf_2 = -F_o$ and define $f_N = f_1 + f_2$. Then, we see that $Gf_N = F_o - F_o = 0$ and $f_N \in \text{int}(FC)$ from the properties of a cone. □

3.2 Constructive force-closure conditions

Verifying that a grasp is force-closure can be difficult due to the form of the friction cone. In certain simple cases, however, it is possible to verify that a grasp is force-closure directly from the definition. For example, if a grasp consists only of frictionless point contacts, then we saw that the grasp map had the form

$$G = \begin{bmatrix} n_{c_1} & \cdots & n_{c_k} \\ p_{c_1} \times n_{c_1} & \cdots & p_{c_k} \times n_{c_k} \end{bmatrix}$$
$$FC = \{f \in \mathbb{R}^k : f_i \geq 0\}.$$

Thus, $G(FC) = \mathbb{R}^6$ if and only if positive linear combinations of the columns of G span $\mathbb{R}^6$.

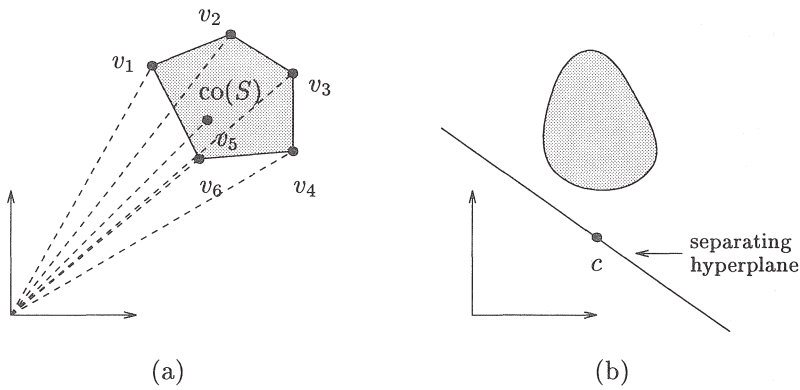


Figure 5.7: Convex hull and separating hyperplane.

This example shows that the force-closure problem can be reformulated as a problem in positive linear spaces. This motivates the following set of definitions; see [98] for a more detailed introduction.

A set of vectors $\{v_1, \cdots, v_k\}$ with $v_i \in \mathbb{R}^p$ is *positively dependent* if there exist $\alpha_i > 0$ such that

$$\sum_{i=1}^{k} \alpha_i v_i = 0.$$

A set of vectors $\{v_1, \cdots, v_k\}$ *positively spans* $\mathbb{R}^n$ if for every $x \in \mathbb{R}^n$ there exists $\alpha_i > 0$, $i = 1, \cdots, k$ such that

$$\sum_{i=1}^{k} \alpha_i v_i = x.$$

Any set of vectors which positively spans $\mathbb{R}^n$ is positively dependent (by choosing $x = 0$). Note that positively dependent vectors must have *strictly* positive coefficients: $\alpha_i > 0$.

A set K is said to be *convex* if for every $x, y \in K$, $\lambda x + (1 - \lambda)y \in K$, $\lambda \in [0, 1]$. Thus any line connecting two points in a convex set lies in that set. The *convex hull* of a set S, denoted $\text{co}(S)$, is the smallest convex set K containing S (i.e., it is the intersection of all convex sets containing S). Using these definitions, it can be shown that given a set $S = \{v_1, \cdots, v_k\}$, the convex hull of S is

$$\text{co}(S) = \{v = \textstyle\sum \alpha_i v_i \colon \sum \alpha_i = 1,\ \alpha_i \geq 0\}.$$

This concept is illustrated in Figure 5.7a.

A useful object in the study of convex sets is the separating hyperplane. Given a point $c \in \mathbb{R}^p$, we define a hyperplane through c with

normal $v \in \mathbb{R}^p$ as

$$H_v(c) = \{x \in \mathbb{R}^p : v^T(x - c) = 0\}.$$

It is a basic theorem in convex analysis that given any convex set K and a point $c \notin K$, there exists a hyperplane $H_v(c)$ such that for all $x \in K$, $v^T(x - c) > 0$. That is, all points in K lie on the "same side" of $H_v(c)$. Such a hyperplane is called a *separating hyperplane* between K and c (see Figure 5.7b). If c lies on the boundary of K, then we replace the condition $v^T(x-c) > 0$ by $v^T(x-c) \geq 0$ and we say $H_v(c)$ is a *supporting hyperplane.*

The following proposition shows the utility of these concepts in determining if a grasp is force-closure.

Proposition 5.3. Convexity conditions for force-closure grasps *Consider a fixed contact grasp which contains only frictionless point contacts. Let $G \in \mathbb{R}^{p \times m}$ be the associated grasp matrix and let $\{G_i\}$ denote the columns of G. The following statements are equivalent:*

1. *The grasp is force-closure.*
2. *The columns of G positively span $\mathbb{R}^p$.*
3. *The convex hull of $\{G_i\}$ contains a neighborhood of the origin.*
4. *There does not exist a vector $v \in \mathbb{R}^p$, $v \neq 0$, such that for $i = 1, \ldots, m$, $v \cdot G_i \geq 0$.*

The equivalence of conditions (1) and (2) follows from the definition of force-closure for a grasp with frictionless point contacts. The remaining conditions rely on tools from convexity theory which are beyond the scope of this book. See [98] for details.

The fourth condition of Proposition 5.3 is computationally attractive since candidates for v can be constructed from the columns of G. Namely, given any set of $p-1$ independent columns of G, let $v \in \mathbb{R}^p$ be a nonzero vector which is perpendicular to the $p-1$ columns. To verify condition 4, we only need to check that the dot products of v and the remaining columns of G do not all have the same sign. It is left as an exercise to show that only values of v defined in this manner need to be considered. Note that the vector v, if it exists, defines the normal to a supporting hyperplane through the origin.

Example 5.4. Using convexity to determine force-closure Consider the planar point contact grasps shown in Figure 5.8. In the first grasp, the contact locations are clustered near the corners of the rectangular object. The grasp map for these contacts is given by

$$G = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ -a & b & -a & b \end{bmatrix},$$

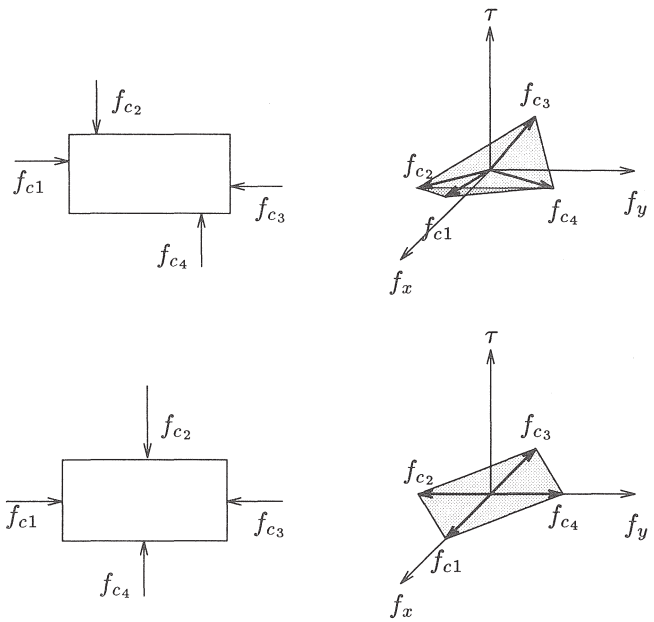


Figure 5.8: Determining whether a planar grasp with point contacts is a force-closure grasp.

where $a, b > 0$. A supporting hyperplane candidate can be formed by choosing any two columns of G. Let v_{ij} represent the normal to the plane formed by G_i and G_j. Enumerating all possibilities, we have

$$
\begin{aligned}
v_{12}^T &= \begin{bmatrix} a & b & 1 \end{bmatrix} & v_{12}^T G &= \begin{bmatrix} 0 & 0 & -2a & 2b \end{bmatrix} \\
v_{13}^T &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & v_{13}^T G &= \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix} \\
v_{14}^T &= \begin{bmatrix} a & -b & 1 \end{bmatrix} & v_{14}^T G &= \begin{bmatrix} 0 & 2b & -2a & 0 \end{bmatrix} \\
v_{23}^T &= \begin{bmatrix} a & -b & -1 \end{bmatrix} & v_{23}^T G &= \begin{bmatrix} 2a & 0 & 0 & -2b \end{bmatrix} \\
v_{24}^T &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} & v_{24}^T G &= \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix} \\
v_{34}^T &= \begin{bmatrix} a & b & -1 \end{bmatrix} & v_{34}^T G &= \begin{bmatrix} 2a & -2b & 0 & 0 \end{bmatrix}.
\end{aligned}
$$

Since none of these possibilities satisfy condition 4 of Proposition 5.3, we conclude the grasp is force-closure. The convex hull for this grasp is shown in the upper right of Figure 5.8.

For the second grasp, the contact points have been chosen such that $a, b = 0$, and hence

$$
G = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.
$$

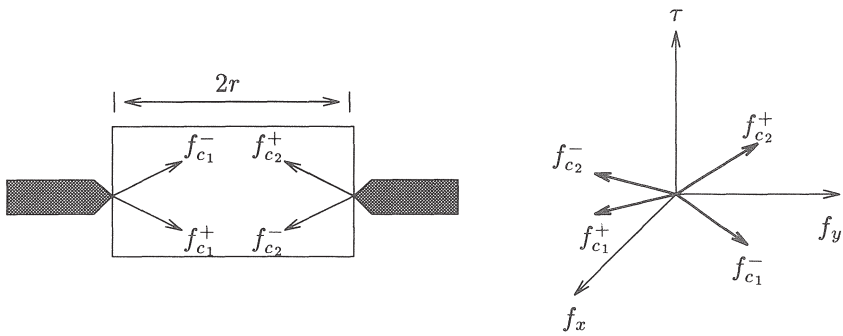


Figure 5.9: Force-closure for point contact with friction.

This grasp is not force-closure since it is not possible to resist torques using positive linear combinations of the columns of G (condition 2). The convex hull of the columns of G lies completely in the f_x, f_y plane, and does not include a neighborhood of the origin (condition 3). Finally, choosing $v = (0, 0, 1)$,

$$v^T G = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix},$$

and hence condition 4 also fails.

In some cases, the results of Proposition 5.3 can be extended to grasps which include frictional contacts. We motivate this with an example:

Example 5.5. Using convexity for grasps with friction
Consider the planar grasp shown in Figure 5.9. Rather than describe this grasp using our usual contact wrench basis, we note that all contact forces which lie in the friction cone can be written as positive linear combinations of the forces which describe the edges of the cone. Hence, we can determine if a grasp is force-closure by checking to see if the object wrenches corresponding to the edges of the contact friction cone positively span the object wrench space.

In the case of planar contacts, this is equivalent to defining a contact map

$$G_i' = \begin{bmatrix} 1 & 1 \\ \mu & -\mu \\ \mu r & -\mu r \end{bmatrix}$$

and a friction cone

$$FC_{c_i}' = \{f \in \mathbb{R}^2 : f_1, f_2 \geq 0\},$$

which has the same form as two independent, frictionless point contacts. Hence, by appropriate choice of basis, we can use Proposition 5.3 to check force-closure of planar grasps with friction.

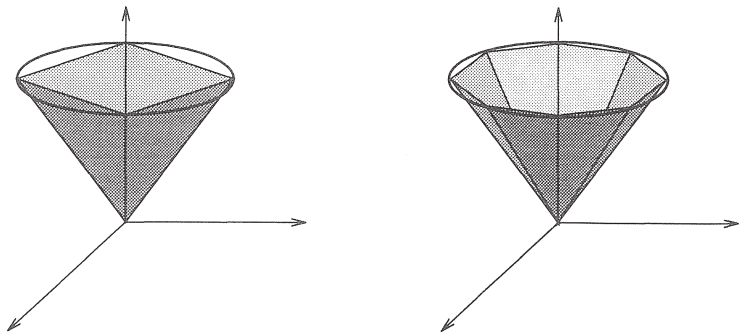


Figure 5.10: Approximations of a spatial friction cone.

For spatial (3-D) grasps, the extension of Proposition 5.3 to contact models with friction is not so straightforward. Since the friction cone *cannot* be represented as the sum of a finite set of vectors, we cannot use the convex hull of a finite basis to determine if a grasp is force-closure. In practice, the circular friction cone can be approximated by a cone generated by a finite set of vectors. This situation is shown in Figure 5.10. The force-closure condition can then be checked by evaluating the convex hull of the (conservative) approximation. If the convex hull contains a neighborhood of the origin, then the grasp is force-closure. For a finite-dimensional approximation, this is a sufficient but not necessary condition.

Additional methods of checking force-closure for spatial grasps are given in the next section.

4 Grasp Planning

In this section, we briefly present methods for choosing contact locations which result in a force-closure grasp.

4.1 Bounds on number of required contacts

Suppose that we are given the task of designing a multifingered robot hand for use in a given set of tasks. One of the first decisions that must be made is the number and type of the fingers. In this section, we consider the following question: given a contact model, how many fingers are necessary to grasp *any* object? This places a lower bound on the number of fingers which we must include on our hand, given a choice of contact type.

Given an object $O \subset \mathbb{R}^3$ (or $O \subset \mathbb{R}^2$ for the planar case), let $\Sigma = \partial O$ denote the boundary of the object. We assume that Σ is a connected,

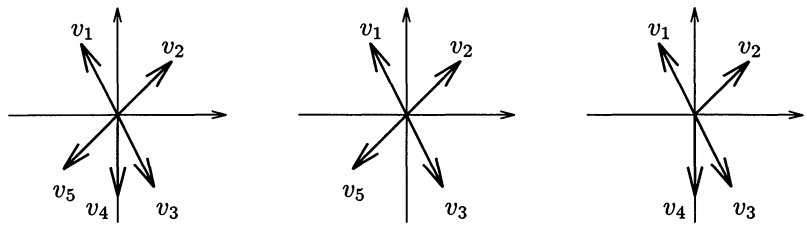


Figure 5.11: Sets of vectors which positively span $\mathbb{R}^2$.

piecewise smooth surface. Let $\Lambda(\Sigma)$ denote the set of all wrenches that can be applied to an object using frictionless point contacts:

$$\Lambda(\Sigma) = \{ \begin{bmatrix} n_{c_i} \\ p_{c_i} \times n_{c_i} \end{bmatrix} : c_i \in \partial O \},$$

where n_{c_i} is the inward pointing unit normal to Σ at the point c_i, and p_{c_i} is the contact point relative to the object frame. A surface Σ is an *exceptional surface* if the convex hull of $\Lambda(\Sigma)$ does not contain a neighborhood of the origin. An object with an exceptional surface can never be grasped using only frictionless point contacts. Examples of exceptional surfaces are the sphere in $\mathbb{R}^3$ and the circle in $\mathbb{R}^2$.

For non-exceptional surfaces, upper and lower bounds for the number of contacts required for a force-closure grasp can be obtained by using two classical theorems in convex analysis.

Theorem 5.4 (Caratheodory). *If a set $X = \{v_1, \ldots, v_k\}$ positively spans $\mathbb{R}^p$, then $k \geq p + 1$.*

Theorem 5.5 (Steinitz). *If $S \subset \mathbb{R}^p$ and $q \in \text{int}(\text{co}\, S)$, then there exists $X = \{v_1, \ldots, v_k\} \subset S$ such that $q \in \text{int}(\text{co}\, X)$ and $k \leq 2p$.*

Caratheodory's theorem implies that if a surface can be grasped with a force-closure grasp, then it must have at least $p + 1$ contacts. This is easy to visualize in the plane, as shown in Figure 5.11. In this case $p = 2$, and hence Caratheodory's theorem asserts that any set of vectors which positively spans $\mathbb{R}^2$ must have at least 3 elements. Suppose we choose any two of the vectors shown in the figure. If we label the two vectors as v_i and v_j, then $-(v_i + v_j)$ never lies in the positive span of v_i and v_j. A similar argument holds for $p > 2$. Of course, a set of vectors may have more than $p + 1$ elements and still not positively span $\mathbb{R}^p$. For example, the set $\{v_1, v_2, v_3\}$ in Figure 5.11a does not positively span $\mathbb{R}^2$.

Steinitz's theorem places an upper bound on the minimal number of contacts which are needed. Let $S = \Lambda(\Sigma)$ and choose $q = 0$. If the

Table 5.3: Lower bounds on the number of fingers required to grasp an object.

Space	Object type	Lower	Upper	FPC	PCWF	SF
Planar	Exceptional	4	6	n/a	3	3
($p = 3$)	Non-exceptional			4	3	3
Spatial	Exceptional	7	12	n/a	4	4
($p = 6$)	Non-exceptional			12	4	4
	Polyhedral			7	4	4

grasp is force-closure, then $0 \in \text{int}(S)$ and by Steinitz's theorem there exist $2p$ vectors $v_1, \ldots, v_{2p} \in \Lambda(\Sigma)$ which generate a force-closure grasp. Hence, Steinitz's theorem guarantees that we need at most $2p$ contacts to grasp *any* non-exceptional object. These $2p$ vectors may not be unique; Figures 5.11b and 5.11c show two choices of 4 vectors which span $\mathbb{R}^2$. Note that in Figure 5.11c, v_3 is redundant, but in Figure 5.11b all of the vectors are needed to positively span $\mathbb{R}^2$.

Caratheodory's and Steinitz's theorems allow us to bound the number of contacts required for a force-closure grasp using frictionless point contacts. If we let p be the dimension of the wrench space and k be the number of contacts, then

$$\begin{aligned} p = 3 \text{ (planar)} &\implies 4 \leq k \leq 6 \\ p = 6 \text{ (spatial)} &\implies 7 \leq k \leq 12. \end{aligned}$$

It can be shown that for the planar case ($p = 3$) the lower bound is achievable for all non-exceptional surfaces. For the spatial case ($p = 6$), proofs exist for special classes of objects showing that the lower bound can be achieved, but the general case is not completely solved. Among the more useful classes of objects which can be grasped with at most seven contacts are the set of polyhedral objects and the set of everywhere smooth objects.

For more general contact types, bounds on the number of contacts are also available. A detailed description of the these bounds is beyond the scope of this book, but can be found in [67]. We note that for the point contact with friction case, the lower bounds can be exhibited using a regular tetrahedron (or equilateral triangle in the plane) and choosing $\mu < \tan 19.5°$. This issue is explored further in the exercises and in the next section.

Table 5.3 summarizes the number of contacts required for various contact models. These bounds are lower bounds on the number of contacts required for a force-closure grasp of any object in the indicated class. That is, in order to be able to grasp *every* object in the class, at least that many contacts must be available.

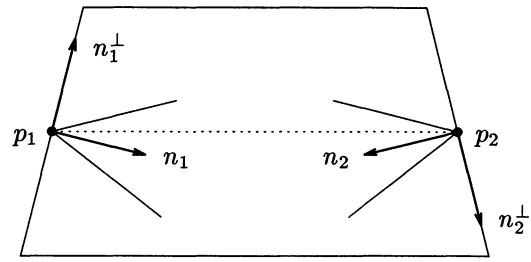


Figure 5.12: Geometric relationships for 2-D force-closure grasp using two point contacts with friction.

4.2 Constructing force-closure grasps

In this section, we show one particularly simple method of computing and constructing force-closure grasps for the case of two point contacts with friction and a planar object. The algorithm is based on the following theorem:

Theorem 5.6. Planar antipodal grasps [83]
A planar grasp with two point contacts with friction is force-closure if and only if the line connecting the contact point lies inside both friction cones.

A grasp satisfying the conditions of this theorem is called an *antipodal grasp*. The proof of the theorem is left as an exercise.

To convert this theorem into an algorithm for planning grasps, we note that the conditions of the theorem can be written as a set of linear inequality constraints. Let n_1 denote the inward pointing normal at the contact point p_1 and n_2 denote the inward point normal at the contact point p_2 (see Figure 5.12). Given a unit vector n, define $n^\perp$ to be the unit vector perpendicular to n which satisfies

$$n \otimes n^\perp = +1,$$

where $\otimes$ is the 2-D cross product:

$$a \otimes b = \det[a\ b] \qquad a, b \in \mathbb{R}^2.$$

Suppose that we are given a polygonal object and we wish to find all possible force-closure grasps. The constraints that the line between contacts lie in the relevant friction cones become:

$$\begin{aligned} A: &\quad (n_1 - \mu n_1^\perp) \otimes (p_2 - p_1) > 0 \\ B: &\quad (n_1 + \mu n_1^\perp) \otimes (p_2 - p_1) < 0 \\ C: &\quad (n_2 - \mu n_2^\perp) \otimes (p_1 - p_2) > 0 \\ D: &\quad (n_2 + \mu n_2^\perp) \otimes (p_1 - p_2) < 0. \end{aligned}$$

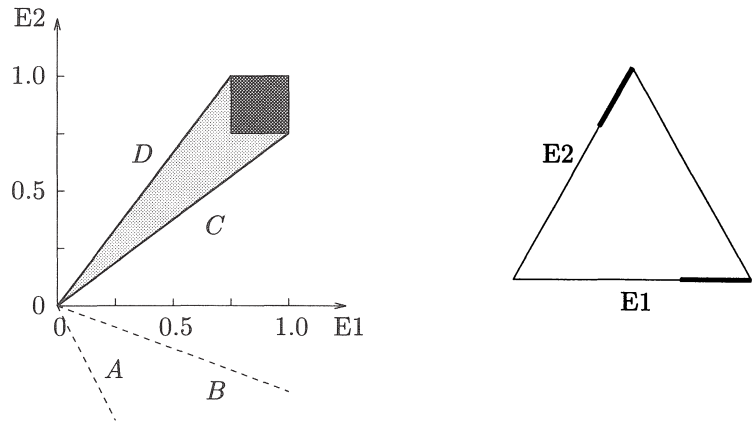


Figure 5.13: Force-closure grasps for an equilateral triangle. The lightly shaded region corresponds to the set of all force-closure grasps and the dark square (and corresponding darkened regions on the triangle) indicates maximally independent regions of contact.

Given any two edges, we can represent these inequalities on a graph, by drawing the lines which correspond to the inequalities becoming identically zero. Figure 5.13 gives an example of such a graph.

This graph gives a complete description of the force-closure grasps between any two edges. By enumerating all pairs of edges, we can generate all possible force-closure grasps for the object. There are many methods which might be used to choose among the grasps. One is to assign a quality measure to a grasp and choose the grasp which maximizes this quality measure and is also force-closure. Another possibility is to choose a grasp such that it is maximally distant from the edges of the force-closure region. Such a grasp is equivalent to choosing maximal independent regions of contact on each edge. That is, we can attempt to find regions on each edge such that if either contact is within the respective region, the grasp is force-closure. Such a region corresponds to a square in the force-closure grasp.

The algorithm presented above can be extended to the spatial case using the following variant of Theorem 5.6:

Theorem 5.7. Spatial antipodal grasps [83]
A spatial grasp with two soft-finger contacts is force-closure if and only if the line connecting the contact point lies inside both friction cones.

It is also possible to extend the algorithm sketched above to work for objects with curved surfaces. In this case, the constraints become nonlinear functions of the contact locations (suitably parameterized).

However, the algorithm can still be applied if there is a way to find the zeros of the given expressions.

5 Grasp Constraints

In the previous sections, we analyzed the grasp kinematics ignoring the kinematics of the fingers. That is, we assumed that forces could be applied at contact points irrespective of how those forces were generated. In this section, we extend our analysis to include the kinematics and statics of the robotic fingers.

We shall view a robot hand grasping an object as a kinematically constrained system. In correspondence with the preceding sections, we assume that the contact locations are fixed on the object. In this case, the constraints between the object and a finger can be formulated by requiring that certain velocities are equal. For example, at a given contact point, the velocity of the fingertip and the velocity of the contact point (on the object) must agree in the direction normal to the surface. For simplicity, we assume throughout the remainder of this section that all contacts are either point contacts with friction or soft-finger contacts. This justifies our assumption that the contact location is fixed; the more general case is considered in the next section.

5.1 Finger kinematics

In Chapter 3 we derived the forward kinematics for an open-chain manipulator using the product of exponentials formula. Recall that the spatial velocity of the end-effector of the robot can be written as

$$V_{st}^s = J_{st}^s(\theta)\dot{\theta},$$

where $J_{st}^s(\theta) \in \mathbb{R}^{p \times n}$ is the spatial Jacobian of the forward kinematics function. The twist V_{st}^s is the generalized velocity of the tool frame, written with respect to a fixed base frame. The body velocity of the tool frame is given by

$$V_{st}^b = \mathrm{Ad}_{g_{st}(\theta)}^{-1} J_{st}^s(\theta)\dot{\theta}.$$

It represents the instantaneous velocity of the tool frame, written in tool coordinates.

We model a robot hand as a collection of robots (fingers) attached to a common base (palm). For each finger in the hand, attach a frame S_i to the base of the finger and a frame F_i to the fingertip at the point of contact, as shown in Figure 5.14. Note that the frame F_i moves with the fingertip, while the frame C_i, also located at the contact point, moves with the object. Let $J_{s_i f_i}^s$ be the Jacobian for F_i relative to the

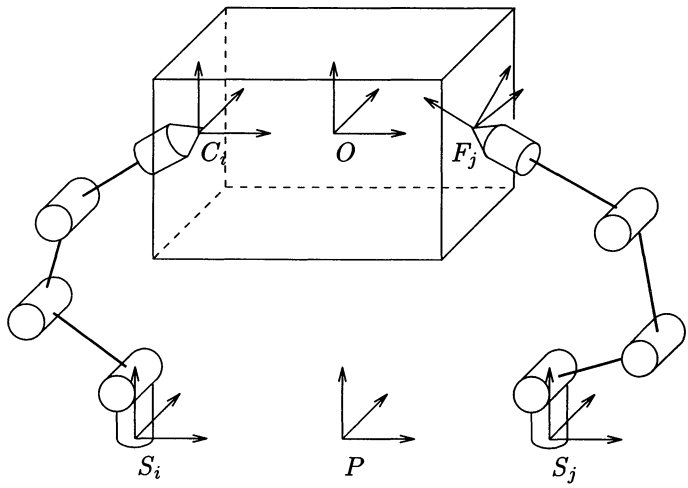


Figure 5.14: Grasp coordinate frames.

fixed frame S_i so that

$$V^b_{s_i f_i} = \mathrm{Ad}^{-1}_{g_{s_i f_i}(\theta_{f_i})} J^s_{s_i f_i}(\theta_{f_i})\dot{\theta}_{f_i}, \tag{5.8}$$

where $\theta_{f_i} \in \mathbb{R}^{n_i}$ is the vector of joint angles for the i^{th} finger.

The grasping constraint for a finger gives the directions in which motion is not allowed. Relative to the contact frame C_i, these constraints can be written as constraints on the velocity of the finger frame F_i. For example, a frictionless point contact restricts the velocities of the finger and the object such that the relative velocity between these two frames is zero in the direction normal to the surface. By convention, this is the z-axis direction in the contact frame, and hence we can write the constraint as

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} V^b_{f_i c_i} = 0.$$

Note that the matrix multiplying $V^b_{f_i c_i}$ is precisely the transpose of the wrench basis for a frictionless point contact.

In general, the directions in which motion is constrained are precisely those in which forces can be exerted. Hence, for a contact with wrench basis B_{c_i}, we require that

$$B^T_{c_i} V^b_{f_i c_i} = 0. \tag{5.9}$$

Equation (5.9) restricts the relative velocity between the finger and the object to be perpendicular to the directions in which forces can be applied.

We can now proceed to write equation (5.9) in terms of a set of known quantities. We perform all computations using body velocities, and hence

we temporarily drop the superscript b. Using the velocity relations in Chapter 2 Section 4, we can write $V_{f_ic_i}$ as

$$\begin{aligned} V_{f_ic_i} &= \mathrm{Ad}^{-1}_{g_{pc_i}} V_{f_ip} + V_{pc_i} \\ &= -\mathrm{Ad}^{-1}_{g_{pc_i}} \mathrm{Ad}_{g_{pf_i}} V_{pf_i} + V_{pc_i}. \end{aligned} \tag{5.10}$$

V_{pf_i} is the velocity of the fingertip relative to the palm frame. Since the finger base frames are fixed relative to the palm frame, it follows that

$$V_{pf_i} = \mathrm{Ad}^{-1}_{g_{s_if_i}} V_{ps_i} + V_{s_if_i} = V_{s_if_i}. \tag{5.11}$$

We can write V_{pc_i} by adding the velocity between the palm and object to the velocity between the object and contact. Since the contact frame is fixed relative to the object frame (by the fixed contact assumption), we have

$$V_{pc_i} = \mathrm{Ad}^{-1}_{g_{oc_i}} V_{po} + V_{oc_i} = \mathrm{Ad}^{-1}_{g_{oc_i}} V_{po}. \tag{5.12}$$

Substituting equations (5.11) and (5.12) into equation (5.10) gives

$$V_{f_ic_i} = -\mathrm{Ad}^{-1}_{g_{pc_i}} \mathrm{Ad}_{g_{pf_i}} V_{s_if_i} + \mathrm{Ad}^{-1}_{g_{oc_i}} V_{po}.$$

We can now substitute this into the constraint equation and use the finger kinematics to yield

$$B^T_{c_i} \left(\mathrm{Ad}^{-1}_{g_{pc_i}} \mathrm{Ad}_{g_{pf_i}} \mathrm{Ad}^{-1}_{g_{s_if_i}} \right) J^s_{s_if_i} \dot{\theta}_{f_i} = B^T_{c_i} \mathrm{Ad}^{-1}_{oc_i} V_{po}.$$

This equation represents the i^{th} contact constraint in terms the finger joint angles θ and the object configuration g_{po}.

The contact constraint can be simplified by making use of the adjoint equation

$$\mathrm{Ad}^{-1}_{g_{pc_i}} \mathrm{Ad}_{g_{pf_i}} \mathrm{Ad}^{-1}_{g_{s_if_i}} = \left(\mathrm{Ad}_{g_{s_ip}} \mathrm{Ad}_{g_{po}} \mathrm{Ad}_{g_{oc_i}} \right)^{-1} = \mathrm{Ad}^{-1}_{g_{s_ic_i}}.$$

Making this substitution, the constraint for the i^{th} finger becomes

$$B^T_{c_i} \mathrm{Ad}^{-1}_{g_{s_ic_i}} J^s_{s_if_i}(\theta_{f_i}) \dot{\theta}_{f_i} = (\mathrm{Ad}^T_{g^{-1}_{oc_i}} B_{c_i})^T V^b_{po} = G^T_i V^b_{po}. \tag{5.13}$$

Stacking equation (5.13) for each finger, we can write the constraint in matrix form. Let k be the number of fingers, n_i the number of degrees of freedom for the i^{th} finger, and set $n = n_1 + \cdots + n_k$. Define the *hand Jacobian* as the matrix $J_h(\theta, x_o) : \mathbb{R}^n \to \mathbb{R}^m$ given by

$$J_h(\theta, x_o) = \begin{bmatrix} B^T_{c_1} \mathrm{Ad}^{-1}_{g_{s_1c_1}} J^s_{s_1f_1}(\theta_{f_1}) & & 0 \\ & \ddots & \\ 0 & & B^T_{c_k} \mathrm{Ad}^{-1}_{g_{s_kc_k}} J^s_{s_kf_k}(\theta_{f_k}) \end{bmatrix}, \tag{5.14}$$

where $\theta = (\theta_{f_1}, \ldots, \theta_{f_k}) \in \mathbb{R}^n$. Then, the constraints in equation (5.13) have the form

$$\boxed{J_h(\theta, x_o)\dot{\theta} = G^T V_{po}^b} \tag{5.15}$$

where $x_o := g_{po}$ is the configuration of the object and θ is the vector of all finger joint angles. Equation (5.15) is the *fundamental grasping constraint*; it relates velocities of the fingers to the velocity of the object. The quantities in the constraint plus a description of the grasp friction cone provide all of the necessary information for modeling a fixed contact grasp.

Although we have derived the grasp constraint in a very mechanical fashion, it is possible to interpret equation (5.15) in a simple and meaningful way. Equation (5.15) equates the velocity of the object and the velocity of the fingertip at the point of contact between the two. Since motion in some directions may be allowed (for example, rotation about the contact point), we only constrain those directions specified by the columns of B_{c_i}. Equation (5.15) is merely a restatment of equation (5.9) written in a more useful form.

The quantities J_h, G, and FC completely characterize the properties of a set of fingers grasping an object. This motivates the following definition.

Definition 5.4. Representation of a multifingered grasp
A *multifingered grasp* is described by the hand Jacobian $J_h(\theta, x_o) : \mathbb{R}^n \to \mathbb{R}^m$, the grasp map $G : \mathbb{R}^m \to \mathbb{R}^p$, and the friction cone $FC \subset \mathbb{R}^m$ which satisfies the properties in Definition 5.1.

Formally, we distinguish between a grasp, which includes specification only of the object and the contact locations, and a multifingered grasp, which includes a description of the finger kinematics. When the usage is clear from context, we use the term grasp.

5.2 Properties of a multifingered grasp

We are now in a position to study the mathematical properties of the grasp constraint in equation (5.15) and interpret them. In the beginning of this chapter, we saw that the force-closure properties of a grasp are characterized by the grasp map G and the friction cone FC. In studying force-closure, we assumed that all allowable contact forces could be applied by the fingers; we can now detail under what conditions this is true and what happens when this condition fails.

A fundamental property of a multifingered grasp is the ability of the robot fingers to accommodate an object motion. If a set of fingers can accommodate any motion of the object without losing contact, we say that such a grasp is manipulable:

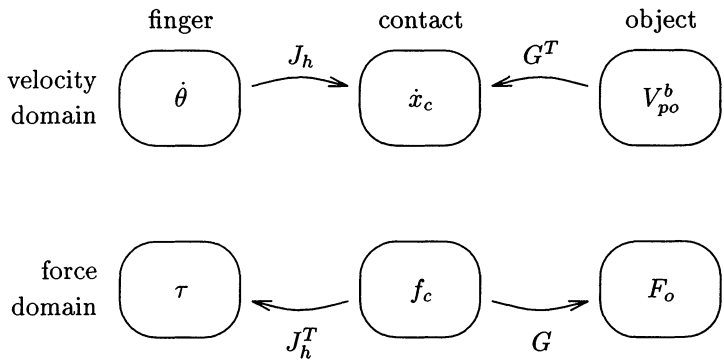


Figure 5.15: Diagram of relationships for a multifingered grasp. The contact force must satisfy $f_c \in FC$ for these relationships to hold.

Definition 5.5. Manipulable grasp
A multifingered grasp is *manipulable* at a configuration (θ, x_o) if for any object motion V_{po}^b there exists $\dot{\theta}$ which satisfies equation (5.15).

The following proposition follows from the definition.

Proposition 5.8. Characterization of all manipulable grasps
A grasp is manipulable at a configuration (θ, x_o) if and only if

$$\mathcal{R}(G^T) \subset \mathcal{R}(J_h(\theta, x_o)).$$

Manipulability does not require that the matrix J_h be injective (one-to-one); there may be many finger motions which accommodate a given object motion. This can happen precisely in the case where J_h is full (row) rank with more columns than rows ($n > m$), and hence has a non-trivial null space. Any joint velocities $\dot{\theta}_N \in \mathcal{N}(J_h)$ result in no motion of the contacts; we call $\dot{\theta}_N$ an *internal motion.* Internal motions can be added to a given finger velocity without changing the velocity of the grasped object and so the finger-object system becomes a redundant manipulator. Note that this kinematic redundancy may appear even if none of the individual fingers are redundant, due to the B_{c_i} terms in the definition of the hand Jacobian. These have the effect of masking out joint velocities in directions which do not violate the contact constraints, and hence allowing internal motions in those directions.

Equation (5.15) describes the velocity relationships between the object and fingers. We will also make use of the force relationships. As described in Section 2, the grasp map characterizes the relationship between contact forces and object wrenches:

$$F_o = G f_c.$$

Table 5.4: Grasp properties.

Property	Definition	Description
Force-closure	Can resist any applied wrench	$G(FC) = \mathbb{R}^p$
Manipulable	Can accommodate any object motion	$\mathcal{R}(G^T) \subset \mathcal{R}(J_h)$
Internal forces	Contact forces f_N which cause no net object wrench	$f_N \in \mathcal{N}(G) \cap \text{int}(FC)$
Internal motions	Finger motions $\dot{\theta}_N$ which cause no object motion	$\dot{\theta}_N \in \mathcal{N}(J_h)$
Structural forces	Object wrench F_I which causes no net joint torques	$G^+ F_I \in \mathcal{N}(J_h^T)$

The contact forces can be related to the joint torques by using the transpose of the hand Jacobian:

$$\tau = J_h^T f_c \tag{5.16}$$

(this follows by equating the work done by the joints and contacts). As in our derivation of the grasp map, the contact forces must lie in the friction cone FC in order to avoid slipping. The complete set of relationships between the velocities and forces in a multifingered grasp are summarized in Figure 5.15.

As in Chapter 3, some care must be taken in interpreting equation (5.16) in the case that $J_h \in \mathbb{R}^{m \times n}$ is not square. If the grasp is manipulable and force-closure, then it follows that J_h is surjective onto the range of G^T. In this case, we can always exert a given contact force, but the joint torques required may not be unique. If internal motions are present, the dynamics of the manipulator must be taken into account. This is discussed more fully in the next chapter.

If a grasp is not manipulable, then it may not be possible to exert arbitrary contact forces. In this case, J_h^T will have a non-trivial null space and hence there may exist contact forces f_c which give $\tau = 0$ in equation (5.16). This is completely analogous to the case in an open-chain manipulator with fewer joint degrees of freedom than the dimension of the workspace. In particular, we call contact forces which lie in the null space of J_h^T *structurally dependent forces*, since they generate forces in the mechanism that cannot be determined without more information about the elastic properties of the mechanism.

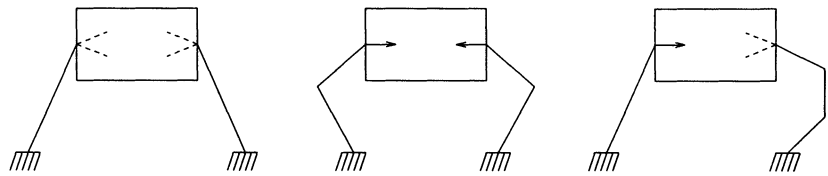


Figure 5.16: Some grasps illustrating manipulability and force-closure properties.

As we have seen above, the properties of a grasp can be completely described by the grasp map G, the hand Jacobian J_h, and the friction cone FC. These properties of a grasp are summarized in Table 5.4. Note that force-closure and manipulability are separate properties. It is possible for a grasp to be force-closure but not manipulable, manipulable and not force-closure, or neither force-closure or manipulable. A few of these possibilities are illustrated in Figure 5.16.

5.3 Example: Two SCARA fingers grasping a box

As an example of how these calculations proceed, consider the grasp shown in Figure 5.17. In this example, two SCARA fingers are used to grasp a box. Assume that the contact points are located at $p_{c_i} = (0, \pm r, 0)$ with respect to the object frame of reference, and that each contact is modeled as a soft-finger contact.

The majority of the quantities needed to describe the grasping constraint have been previously computed. The grasp map was derived in Example 5.2:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -r & 0 & 0 & 0 & 0 & +r & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & +r & 0 & 0 & -r & 0 & 0 & 0 \end{bmatrix},$$

using the contact frames shown in Figure 5.5. The wrench basis for each finger was given by

$$B_{c_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For simplicity, we take the length of the fingertips as $l_3 = 0$ and hence the contact occurs at the usual SCARA end-effector location. In this case, the Jacobian is that given in Example 3.8:

$$J^s_{s_i f_i} = \begin{bmatrix} 0 & l_1 \cos\theta_1 & l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & 0 \\ 0 & l_1 \sin\theta_1 & l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

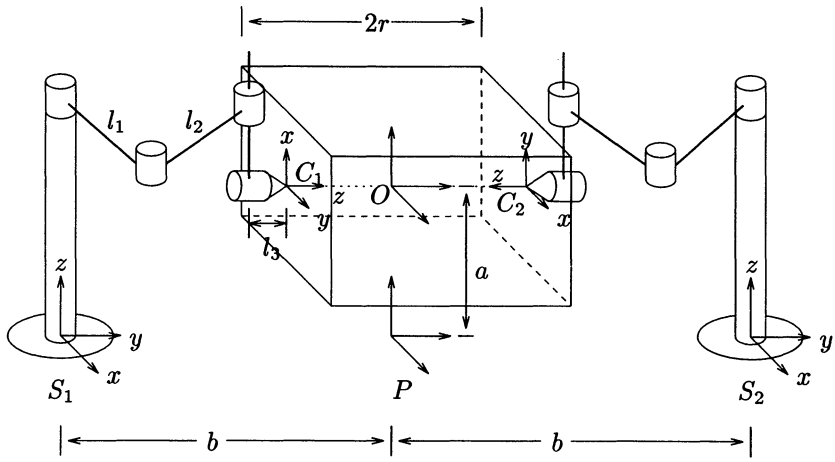


Figure 5.17: Two-fingered grasp using SCARA robots.

If $l_3 \neq 0$, then we must recompute the Jacobian to include the additional displacement (see Exercise 11).

The only remaining quantity to calculate is $\mathrm{Ad}^{-1}_{g_{s_ic_i}}$, which maps twists from the spatial frame into the contact frame for a given finger. We can construct $\mathrm{Ad}^{-1}_{g_{s_ic_i}}$ using the transformation

$$g^{-1}_{s_ic_i} = (g_{s_ip}\, g_{po}\, g_{oc_i})^{-1} = g^{-1}_{oc_i}\, g^{-1}_{po}\, g^{-1}_{s_ip}.$$

Since the transformations g_{oc_i} and g_{s_ip} are constant, we can write

$$\mathrm{Ad}^{-1}_{g_{s_ic_i}} = \mathrm{Ad}^{-1}_{g_{oc_i}}\, \mathrm{Ad}^{-1}_{g_{po}}\, \mathrm{Ad}^{-1}_{g_{s_ip}}.$$

Note that only $\mathrm{Ad}^{-1}_{g_{po}}$ depends on $x_o = g_{po}$, the current configuration of the object, and the other transformations are constant matrices.

To construct the grasp constraint, we compose the matrices given above in the appropriate fashion. In full generality, the contact constraints are quite complex and the calculations are more suited for automated rather than manual computation. We therefore restrict ourselves to analyzing the grasp constraints for the object configuration shown in Figure 5.17. That is, we set

$$R_{po} = I \qquad p_{po} = \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix}$$

and hence $\mathrm{Ad}^{-1}_{g_{s_ic_i}}$ becomes a constant matrix. Rather than compute it in pieces, we use the transformations $g_{s_ic_i}$, which can be written down

by inspection:

$$R_{s_1c_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad \Longrightarrow \qquad \mathrm{Ad}^{-1}_{g_{s_1c_1}} = \begin{bmatrix} R^T_{s_1c_1} & \begin{pmatrix} b-r & 0 & 0 \\ 0 & a & r-b \\ -a & 0 & 0 \end{pmatrix} \\ 0 & R^T_{s_1c_1} \end{bmatrix}$$

$$p_{s_1c_1} = \begin{bmatrix} 0 \\ b-r \\ a \end{bmatrix}$$

$$R_{s_2c_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \Longrightarrow \qquad \mathrm{Ad}^{-1}_{g_{s_2c_2}} = \begin{bmatrix} R^T_{s_2c_2} & \begin{pmatrix} 0 & a & b-r \\ r-b & 0 & 0 \\ a & 0 & 0 \end{pmatrix} \\ 0 & R^T_{s_2c_2} \end{bmatrix}.$$

$$p_{s_2c_2} = \begin{bmatrix} 0 \\ -b+r \\ a \end{bmatrix}$$

Finally, we calculate J_h:

$$J_h = \begin{bmatrix} B^T_{c_1} \mathrm{Ad}^{-1}_{g_{s_1c_1}} J^s_{s_1f_1}(\theta_{f_1}) & 0 \\ 0 & B^T_{c_2} \mathrm{Ad}^{-1}_{g_{s_2c_2}} J^s_{s_2f_2}(\theta_{f_2}) \end{bmatrix} = \begin{bmatrix} J_{11} & 0 \\ 0 & J_{22} \end{bmatrix}$$

$$J_{11} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -b+r & -b+r+l_1c_1 & -b+r+l_1c_1+l_2c_{12} & 0 \\ 0 & l_1s_1 & l_1s_1+l_2s_{12} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_{22} = \begin{bmatrix} b-r & b-r+l_3c_3 & b-r+l_3c_3+l_4c_{34} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -l_3s_3 & -l_3s_3-l_4s_{34} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(for the particular configuration shown in Figure 5.17).

We can now evaluate the properties of the grasp. From Example 5.2 we have that the grasp is force-closure (this does not depend on the fingers which actually exert the forces). However, the grasp is *not* manipulable, since rotation of the object about the y-axis yields

$$G^T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \notin \mathcal{R}(J_h).$$

The grasp contains internal motions which are in the vector space spanned by

$$\dot{\theta}_{N_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \dot{\theta}_{N_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

at the configuration shown. These motions correspond to rotating the last revolute joint of each finger, resulting in a rotation of the fingertip about contact point. In the more general case where $l_3 \neq 0$ and the hand is in an arbitrary configuration, the internal motion still exists but all three revolute joints will have nonzero instantaneous velocities.

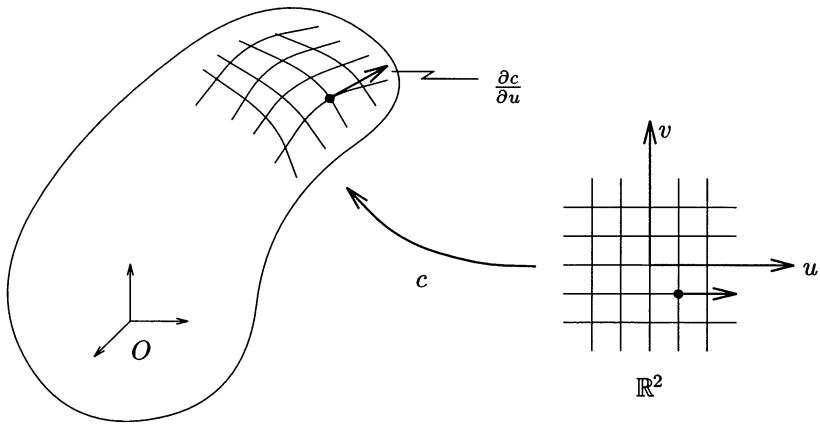


Figure 5.18: Surface chart for a two-dimensional object in $\mathbb{R}^3$.

6 Rolling Contact Kinematics

Most real-world grasping situations involve moving rather than fixed contacts. Human fingers and many robotic fingers are actually surfaces, and manipulation of an object by a set of fingers involves rolling of the fingers along the object surface. In this section, we derive the kinematic equations for one object rolling against another and extend the grasping formulation to moving contacts. The formulation here covers rolling between smooth fingers and smooth objects. It does not cover scenarios in which fingers roll on the edge of an object, but extensions to this case are possible. We begin with a brief description of surface parameterizations.

6.1 Surface models

Given an object in $\mathbb{R}^3$, we describe the surface of the object using a *local coordinate chart*, $c : U \subset \mathbb{R}^2 \to \mathbb{R}^3$, as shown in Figure 5.18. The map c takes a point $(u, v) \in \mathbb{R}^2$ to a point $x \in \mathbb{R}^3$ on the surface of the object, written in the object frame of reference, O. Thus, locally, we can describe a point on the surface by specifying the corresponding (u, v). In general, it may take several coordinate charts to completely describe the surface of the object. A surface S is *regular* if for each point $p \in S$ there exists a neighborhood $V \subset \mathbb{R}^3$, an open set $U \subset \mathbb{R}^2$, and a map $c : U \to V \cap S$ such that

1. c is differentiable.

2. c is a homeomorphism from U to $V \cap S$. That is, c is continuous, bijective (one-to-one and onto), and has a continuous inverse.

3. For every $\alpha = (u, v) \in U$, the map $\frac{\partial c}{\partial \alpha}(\alpha) : \mathbb{R}^2 \to \mathbb{R}^3$ is injective (one-to-one).

At any point on the object, we can define a tangent plane which consists of the space of all vectors which are tangent to the surface of the object at that point. The tangent plane is spanned by the vectors $c_u := \frac{\partial c}{\partial u}$ and $c_v := \frac{\partial c}{\partial v}$. That is, any vector which is tangent to the surface at a point $c(u, v)$ may be expressed as a linear combination of the vectors c_u and c_v, evaluated at (u, v). A coordinate chart is an *orthogonal coordinate chart* if c_u and c_v are orthogonal.

Theorem 5.9. *Locally, there exists an orthogonal chart for all regular surfaces.*

The proof can be found in any standard book on differential geometry.

In the sequel, we will make frequent use of some more detailed concepts from differential geometry. Although the application of the rolling equations does not require knowledge of these concepts, they do allow a deeper understanding of the material. We present here a brief review of the relevant topics; a full description can be found in DoCarmo [27].

In order to define the area of a surface, one needs to define the inner product between two tangent vectors on the surface. This defines the area of a parallelogram and the total area can then be calculated by integrating the infinitesimal areas generated by parallelogram-shaped patches on the surface. The *first fundamental form* for a surface describes how the inner product of two tangent vectors is related to the natural inner product on $\mathbb{R}^3$. In a local coordinate chart, it is represented by a quadratic form $\mathrm{I}_p : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ which takes two tangent vectors attached at a point $p = c(u, v)$ and gives their inner product. If c is a local parameterization, then the matrix representation of the quadratic form is given by

$$\mathrm{I}_p = \begin{bmatrix} c_u^T c_u & c_u^T c_v \\ c_v^T c_u & c_v^T c_v \end{bmatrix}. \tag{5.17}$$

We will use the symbol I_p to represent both the quadratic form and its matrix representation. Note that each element of I_p is an inner product between vectors in $\mathbb{R}^3$ and that I_p is symmetric and positive definite. If a parameterization is orthogonal, only the diagonal terms are nonzero.

The first fundamental form can be used to define the *metric tensor* for a surface. The metric tensor is given by the square root of the first fundamental form and is used to normalize tangent vectors. We define the matrix $M_p : \mathbb{R}^2 \to \mathbb{R}^2$ as the positive definite matrix which satisfies

$$\mathrm{I}_p = M_p \cdot M_p.$$

In the case that the parametrization is orthogonal, M_p has the form

$$M_p = \begin{bmatrix} \|c_u\| & 0 \\ 0 & \|c_v\| \end{bmatrix}. \tag{5.18}$$

(The metric tensor described here is the square root of the Riemannian metric used in differential geometry.)

At each point on a surface S, we can define an outward pointing unit normal by taking the cross product between the vectors that define the tangent space. We identify the set of all unit vectors in $\mathbb{R}^3$ with $\mathbb{S}^2$, the unit sphere in $\mathbb{R}^3$. The *Gauss map* $N : S \to \mathbb{S}^2$ gives the unit normal at each point on the surface S. In local coordinates,

$$N(u,v) = \frac{c_u \times c_v}{|c_u \times c_v|}. \tag{5.19}$$

For smooth, orientable surfaces, the Gauss map is a well defined, differentiable mapping. We write $n = N(u,v)$ for the unit normal at a point on the surface.

The directional derivative of the Gauss map defines the *second fundamental form* for a surface. The second fundamental form is a measure of the curvature of a surface. In a local coordinate chart, it is represented by a map $\mathrm{II}_p : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$: which has a matrix representation

$$\mathrm{II}_p = \begin{bmatrix} c_u^T n_u & c_u^T n_v \\ c_v^T n_u & c_v^T n_v \end{bmatrix}, \tag{5.20}$$

where $p = c(u,v)$, $n_u := \frac{\partial n}{\partial u}$, and $n_v := \frac{\partial n}{\partial v}$. This matrix describes the rate of change of the normal vector projected onto the tangent plane. It may be interpreted as follows: if $p(s) \in S$ is a curve lying on S that is parameterized by arc length and whose tangent vector is p', then $(p')^T \mathrm{II}_p p'$ gives the usual curvature of the spatial curve p.

For our purposes, it will be convenient to scale the second fundamental form and define the *curvature tensor* for a surface. For an orthogonal set of coordinates, the curvature tensor is a mapping $K : \mathbb{R}^2 \to \mathbb{R}^2$ defined as

$$K_p = M_p^{-T} \mathrm{II}_p M_p^{-1} = \begin{bmatrix} \frac{c_u^T n_u}{\|c_u\|^2} & \frac{c_u^T n_v}{\|c_u\|\|c_v\|} \\ \frac{c_v^T n_u}{\|c_u\|\|c_v\|} & \frac{c_v^T n_v}{\|c_v\|^2} \end{bmatrix}. \tag{5.21}$$

The factors of M_p^{-1} are normalization factors which account for scaling present in the local coordinate chart for the surface.

The curvature tensor can also be computed in terms of a special coordinate frame called the *normalized Gauss frame*. If $c(u,v)$ is an orthogonal chart, then we define the normalized Gauss frame as

$$\begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} \frac{c_u}{\|c_u\|} & \frac{c_v}{\|c_v\|} & n \end{bmatrix}. \tag{5.22}$$

The normalized Gauss frame provides an orthonormal frame at each point on the surface. In terms of this frame, the curvature tensor is given by

$$K_p = \begin{bmatrix} x^T \\ y^T \end{bmatrix} \begin{bmatrix} \frac{n_u}{\|c_u\|} & \frac{n_v}{\|c_v\|} \end{bmatrix}, \tag{5.23}$$

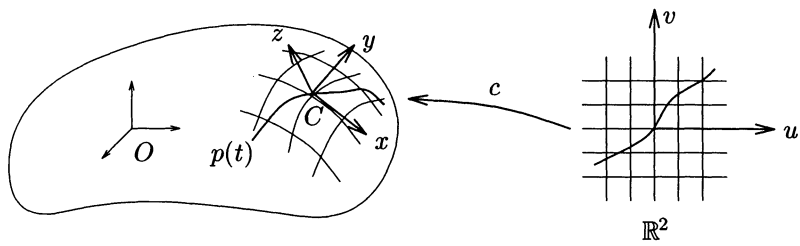


Figure 5.19: The contact frame is a moving frame along a curve $p(t)$.

which can be interpreted as a measure of how the unit normal varies across the surface, as projected on the tangent plane. Again, a normalization factor is present to account for scaling due to the parameterization. If the surface is flat, then $n_u = n_v = 0$ and $K_p = 0$.

Finally, we define the *torsion form.* For a curve, the torsion measures the rate of change of curvature along the curve. The torsion of a surface is a measure of how the Gauss frame twists as we move across the surface, again projected onto the tangent plane. To compute the torsion, we only need to keep track of how either x or y changes, since they are orthonormal. We define the torsion form $T : \mathbb{R}^2 \to \mathbb{R}$ as

$$T_p = y^T \begin{bmatrix} \frac{x_u}{\|c_u\|} & \frac{x_v}{\|c_v\|} \end{bmatrix} = \begin{bmatrix} \frac{c_v^T c_{uu}}{\|c_u\|^2 \|c_v\|} & \frac{c_v^T c_{uv}}{\|c_u\| \|c_v\|^2} \end{bmatrix}, \qquad (5.24)$$

where x_u and x_v denote the partial derivative of x with respect to u and v, $c_{uu} := \frac{\partial^2 c}{\partial u^2}$, and $c_{uv} := \frac{\partial^2 c}{\partial u \partial v}$. Note that the torsion form is represented as a row vector, not a matrix. The torsion form is related to the Christoffel symbols for a surface parameterization (see [27]). If a surface is flat, then $T_p = 0$.

Given a parameterization, (M_p, K_p, T_p) are collectively referred to as the *geometric parameters* of the surface. These parameters describe the local geometry of the surface and play an important role in the kinematics of contact which we will pursue in the section that follows.

Let $p(t) \in S$ be a curve on the surface of the object and define the *contact frame* along the curve to be the frame C which coincides with the Gauss frame at time t (see Figure 5.19).[2] We wish to determine the motion of the contact frame as a function of the geometric parameters and the velocity of the curve. If we fix a frame O in the object then the motion of the frame C is given by the rigid transformation $g_{oc}(t) \in SE(3)$. We assume $p(t)$ lies in a single coordinate chart $c : U \to \mathbb{R}^3$ and we let $\alpha(t) = c^{-1}(p(t))$ represent the local coordinates.

[2]This is different from our previous convention, where we choose the z-axis along the inward pointing normal.

Lemma 5.10. Induced velocity of the contact frame
The (body) velocity of the contact frame C relative to the reference frame O of the object is given by $V_{oc}^b = (v_{oc}, \omega_{oc})$ where

$$v_{oc} = \begin{bmatrix} M\dot{\alpha} \\ 0 \end{bmatrix} \tag{5.25}$$

$$\widehat{\omega}_{oc} = \left[\begin{array}{cc|c} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ \hline -\omega_y & \omega_x & 0 \end{array}\right] = \left[\begin{array}{cc|c} 0 & -TM\dot{\alpha} & KM\dot{\alpha} \\ TM\dot{\alpha} & 0 & \\ \hline & -(KM\dot{\alpha})^T & 0 \end{array}\right], \tag{5.26}$$

and M, K, and T are the geometric parameters of the surface relative to the coordinate chart (c, U).

Proof. The position and orientation of the contact frame relative to the reference frame are given by

$$p_{oc} = p(t) = c(\alpha(t))$$
$$R_{oc} = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = \begin{bmatrix} \frac{c_u}{\|c_u\|} & \frac{c_v}{\|c_v\|} & \frac{c_u \times c_v}{\|c_u \times c_v\|} \end{bmatrix}.$$

The translational component of the body velocity is given by $v_{oc} = R_{oc}^T \dot{p}_{oc}$ which yields

$$v_{oc} = \begin{bmatrix} x^T \\ y^T \\ z^T \end{bmatrix} \frac{\partial c}{\partial \alpha}\dot{\alpha} = \begin{bmatrix} x^T c_u & x^T c_v \\ y^T c_u & y^T c_v \\ z^T c_u & z^T c_v \end{bmatrix} \dot{\alpha} = \begin{bmatrix} M\dot{\alpha} \\ 0 \end{bmatrix}.$$

To show equation (5.26), we compute the body angular velocity:

$$\widehat{\omega}_{oc} = R_{oc}^T \dot{R}_{oc} = \begin{bmatrix} x^T \\ y^T \\ z^T \end{bmatrix} \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & x^T\dot{y} & x^T\dot{z} \\ y^T\dot{x} & 0 & y^T\dot{z} \\ z^T\dot{x} & z^T\dot{y} & 0 \end{bmatrix},$$

where the zero entries follow because x, y, and z are unit vectors. Now using the definitions of K, T, and M we have

$$\widehat{\omega}_{oc} = \left[\begin{array}{cc|c} 0 & -TM\dot{\alpha} & \\ TM\dot{\alpha} & 0 & KM\dot{\alpha} \\ \hline & -(KM\dot{\alpha})^T & 0 \end{array}\right].$$

□

Example 5.6. Geometric parameters of a sphere in $\mathbb{R}^3$
A coordinate chart for the sphere of radius ρ can be obtained by using spherical coordinates, as shown in Figure 5.20. For the hemisphere centered about the x-axis, we have

$$c(u, v) = \begin{bmatrix} \rho \cos u \cos v \\ \rho \cos u \sin v \\ \rho \sin u \end{bmatrix},$$

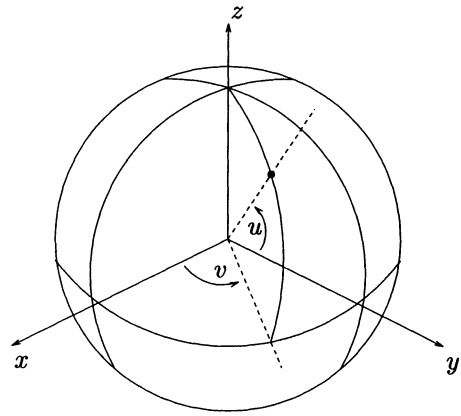


Figure 5.20: Spherical coordinate chart for a sphere.

with

$$U = \{(u, v) : -\pi/2 < u < \pi/2, -\pi < v < \pi\}.$$

The partial derivatives of c with respect to u and v are given by

$$c_u = \begin{bmatrix} -\rho \sin u \cos v \\ -\rho \sin u \sin v \\ \rho \cos u \end{bmatrix} \qquad c_v = \begin{bmatrix} -\rho \cos u \sin v \\ \rho \cos u \cos v \\ 0 \end{bmatrix}$$

and we see that $c_u^T c_v = 0$ and hence the chart is orthogonal. The curvature, torsion, and metric tensors are:

$$K = \begin{bmatrix} 1/\rho & 0 \\ 0 & 1/\rho \end{bmatrix} \qquad M = \begin{bmatrix} \rho & 0 \\ 0 & \rho \cos u \end{bmatrix} \qquad T = \begin{bmatrix} 0 & -1/\rho \tan u \end{bmatrix}.$$

6.2 Contact kinematics

Consider two objects with surfaces S_o and S_f which are touching at a point, as shown in Figure 5.21. We are interested in the motion of the points of contact across the surfaces of the objects in response to a relative motion of the objects.

Let $p_o(t) \in S_o$ and $p_f(t) \in S_f$ be the positions at time t of the point of contact relative to two body-fixed frames O and F, respectively. For simplicity, we will restrict ourselves to the case where motion is contained in a single coordinate chart for each object. Let (c_o, U_o) and (c_f, U_f) be charts for the two surfaces, and $\alpha_o = c_o^{-1}(p_o) \in U_o$ and $\alpha_f = c_f^{-1}(p_f) \in U_f$ be local coordinates. We will assume that c_o and c_f are orthogonal representations of the surfaces. Let ψ be the angle of contact, defined as the angle between the tangent vectors $\frac{\partial c_f}{\partial u_f}$ and $\frac{\partial c_o}{\partial u_o}$. We choose the sign of ψ so that a rotation of $\frac{\partial c_o}{\partial u_o}$ through an angle ψ around the outward normal

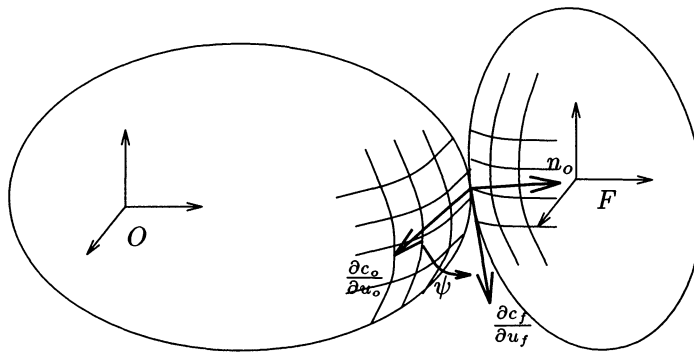


Figure 5.21: Motion of two objects in contact.

of S_o aligns $\frac{\partial c_o}{\partial u_o}$ with $\frac{\partial c_f}{\partial u_f}$ (see Figure 5.21). Collecting the quantities which describe the contact, we call $\eta = (\alpha_f, \alpha_o, \psi)$ the *contact coordinates* for S_f and S_o.

Let $g_{of} \in SE(3)$ describe the relative position and orientation of S_f with respect to S_o. We wish to study the relationship between g_{of} and the local contact coordinates. To do so, we assume that $g_{of} \in W \subset SE(3)$, where W is the set of all relative positions and orientations for which the two objects are in contact.

The coordinate charts (c_o, U_o) and (c_f, U_f) induce a normalized Gauss frame at all points in $c_o(U_o) \subset S_o$ and $c_f(U_f) \subset S_f$. We define the contact frames C_o and C_f as the coordinate frames that coincide with the normalized Gauss frame at $p_o(t)$ and $p_f(t)$, for all $t \in I$, where I is the interval of interest. We also define a continuous family of coordinate frames, two for each $\tau \in I$, as follows. Let the local frames $L_o(\tau)$ and $L_f(\tau)$, be the coordinate frames fixed relative to O and F, respectively, that coincide at time $t = \tau$ with the normalized Gauss frame at $p_o(t)$ and $p_f(t)$ (see Figure 5.22).

We describe the motion of O and F at time t using local coordinate frames $L_o(t)$ and $L_f(t)$. Let $v_{l_o l_f} = (v_x, v_y, v_z)$ be the components of the (body) translational velocity of $L_f(t)$ relative to $L_o(t)$ at time t. Similarly, let $\omega_{l_o l_f} = (\omega_x, \omega_y, \omega_z)$ be the (body) rotational velocity. Here (ω_x, ω_y) are the *rolling velocities* along the tangent plane at the point of contact, and ω_z is the rotational velocity about the contact normal. Likewise, (v_x, v_y) are the linear velocities along the tangent plane, i.e., the *sliding velocities*, and v_z is the linear velocity in the direction contact normal. As long as the two bodies remain in contact, $v_z = 0$. In addition, for pure rolling contact we have $v_x = v_y = 0$ and $\omega_z = 0$ and for pure sliding contact we have $\omega_x = \omega_y = \omega_z = 0$. Since the local frames are fixed relative to their respective frame of reference, according

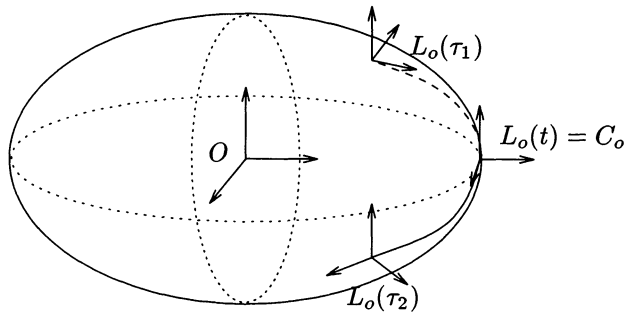


Figure 5.22: Relation between the one-parameter family of local frames, $L_f(t)$, and the contact frame, C_f. At time $t \in I$, $L_f(t)$ coincides with C_f.

to Proposition 2.15 we have $V_{l_o l_f} = Ad_{g_{fl_f}^{-1}} V_{of}$ or, in components,

$$V^b_{l_o l_f} = \left[\begin{array}{c} v_x \\ v_y \\ v_z \\ \hline \omega_x \\ \omega_y \\ \omega_z \end{array}\right] = \begin{bmatrix} R^T_{fl_f} v_{of} - R^T_{fl_f} \hat{p}_o \omega_{of} \\ R^T_{fl_f} \omega_{of} \end{bmatrix}. \tag{5.27}$$

We also let

$$R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ -\sin\psi & -\cos\psi \end{bmatrix} \qquad \text{and} \qquad \tilde{K}_o = R_\psi K_o R_\psi.$$

Note that R_ψ is the orientation of the x- and y-axes of C_f relative to the x- and y-axes of C_o. Thus, $\tilde{K}_o$ is the curvature of O at the point of contact relative to the x- and y-axes of C_f. We call $K_f + \tilde{K}_o$ the relative curvature form.

Theorem 5.11. Kinematic equations of contact [73]
The motion of the contact coordinates, $\dot{\eta}$, as a function of the relative motion is given by

$$\begin{aligned} \dot{\alpha}_f &= M_f^{-1}(K_f + \tilde{K}_o)^{-1} \left(\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} - \tilde{K}_o \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right) \\ \dot{\alpha}_o &= M_o^{-1} R_\psi (K_f + \tilde{K}_o)^{-1} \left(\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} + K_f \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right) \\ \dot{\psi} &= \omega_z + T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o \\ 0 &= v_z, \end{aligned} \tag{5.28}$$

where $(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z) = V^b_{l_o l_f} = Ad^{-1}_{g_{fl_f}} V^b_{of}$.

Before presenting the proof of the theorem, let us note that the equations of contact only make sense when the relative curvature is nonsingular. An example of a singular relative curvature occurs when one object is concave and the other is convex, and both have the same radius of curvature. In this case, small motions of the object can cause large motions of the contact and continuity is lost. To avoid this possibility, we shall assume that all manipulation occurs in an open set on which the relative curvature is invertible.

Proof of theorem. We perform all calculations using body velocities and hence temporarily drop the superscript b. Since the frame $L_f(t)$ is fixed relative to the frame F, the velocity of $L_f(t)$ relative to F is given by $V_{fl_f} = 0$. Therefore, using the velocity relationships from Chapter 2,

$$V_{fc_f} = Ad^{-1}_{g_{l_f c_f}} V_{fl_f} + V_{l_f c_f} = V_{l_f c_f}. \tag{5.29}$$

Similarly, we find that

$$V_{oc_o} = \mathrm{Ad}^{-1}_{g_{l_o c_o}} V_{ol_o} + V_{l_o c_o} = V_{l_o c_o}. \tag{5.30}$$

We now compute the velocity of C_f relative to $L_o(t)$ via two intermediate frames, namely $L_f(t)$ and C_o.

At time t, the position and orientation of C_f relative to $L_f(t)$ are $p_{l_f c_f} = 0$ and $R_{l_f c_f} = I$. Thus,

$$V_{l_o c_f} = V_{l_o l_f} + V_{l_f c_f} \tag{5.31}$$

and since $p_{c_o c_f} = 0$,

$$V_{l_o c_f} = \begin{bmatrix} R^T_{c_o c_f} & 0 \\ 0 & R^T_{c_o c_f} \end{bmatrix} V_{l_o c_o} + V_{c_o c_f}. \tag{5.32}$$

Combining equations (5.29) through (5.32) yields

$$V_{l_o l_f} + V_{f c_f} = \begin{bmatrix} R^T_{c_o c_f} & 0 \\ 0 & R^T_{c_o c_f} \end{bmatrix} V_{o c_o} + V_{c_o c_f}. \tag{5.33}$$

We now find the values of each of the quantities in equation (5.33) in terms of the geometric parameters and motion parameters. First, we observe that

$$\begin{aligned} p_{c_o c_f} &= 0 & & & v_{c_o c_f} &= 0 \\ R_{c_o c_f} &= \begin{bmatrix} R_\psi & 0 \\ 0 & 1 \end{bmatrix} & \Longrightarrow & & \omega_{c_o c_f} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{5.34}$$

and, by definition, $V_{l_o l_f} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$.

Second, according to Lemma 5.10,

$$v_{fc_f} = \begin{bmatrix} M_f\dot{\alpha}_f \\ 0 \end{bmatrix}$$
$$\widehat{\omega}_{fc_f} = \left[\begin{array}{cc|c} 0 & -T_f M_f\dot{\alpha}_f & K_f M_f\dot{\alpha}_f \\ T_f M_f\dot{\alpha}_f & 0 & \\ \hline -(K_f M_f\dot{\alpha}_f)^T & & 0 \end{array}\right] \tag{5.35}$$

and

$$v_{oc_o} = \begin{bmatrix} M_o\dot{\alpha}_o \\ 0 \end{bmatrix}$$
$$\widehat{\omega}_{oc_o} = \left[\begin{array}{cc|c} 0 & -T_o M_o\dot{\alpha}_o & K_o M_o\dot{\alpha}_o \\ T_o M_o\dot{\alpha}_o & 0 & \\ \hline -(K_o M_o\dot{\alpha}_o)^T & & 0 \end{array}\right]. \tag{5.36}$$

Substituting equations (5.34) through (5.36) into equation (5.33) and equating components, we get

$$\begin{bmatrix} M_f\dot{\alpha}_f \\ 0 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} R_\psi M_o\dot{\alpha}_o \\ 0 \end{bmatrix}$$
$$\left[\begin{array}{c} K_f M_f\dot{\alpha}_f \\ \hline T_f M_f\dot{\alpha}_f \end{array}\right] + \left[\begin{array}{c} \omega_y \\ -\omega_x \\ \hline \omega_z \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ \hline \dot{\psi} \end{array}\right] - \left[\begin{array}{c} R_\psi K_o M_o\dot{\alpha}_o \\ \hline T_o M_o\dot{\alpha}_o \end{array}\right]. \tag{5.37}$$

After some algebraic manipulation, we can write equation (5.37) in the form given by equation (5.28). □

A common situation in grasping is to assume that the fingers roll without slipping on the object. In this case we can simplify the contact kinematics.

Corollary 5.11.1. *The contact coordinates for* rolling contact *evolve according to*

$$\begin{aligned} \dot{\alpha}_f &= M_f^{-1}(K_f + \tilde{K}_o)^{-1}\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \\ \dot{\alpha}_o &= M_o^{-1}R_\psi(K_f + \tilde{K}_o)^{-1}\begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \\ \dot{\psi} &= T_f M_f\dot{\alpha}_f + T_o M_o\dot{\alpha}_o. \end{aligned} \tag{5.38}$$

Example 5.7. Sphere rolling on a plane
Consider a spherical finger of radius ρ rolling on a plane, as shown in Figure 5.23. The local coordinates of the plane are chosen to be $c_o(u, v) =$

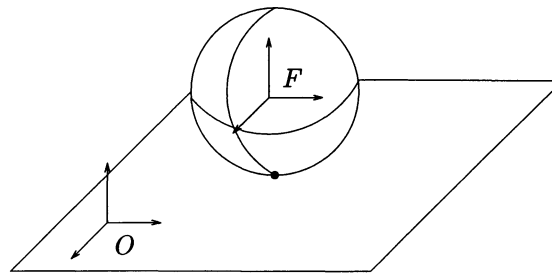


Figure 5.23: Spherical finger rolling on a plane. The finger is only allowed to roll on the object and not slip or twist.

$(u, v, 0)$. The sphere requires multiple coordinate charts to describe the entire surface, so we shall restrict ourselves to the chart

$$c_f(u,v) = \begin{bmatrix} \rho \cos u \cos v \\ -\rho \cos u \sin v \\ \rho \sin u \end{bmatrix},$$

where ρ is the radius of the sphere and $-\pi/2 < u < \pi/2$, $-\pi < v < \pi$. The curvature, torsion, and metric tensors are:

$$\begin{aligned} K_o &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & K_f &= \begin{bmatrix} 1/\rho & 0 \\ 0 & 1/\rho \end{bmatrix} \\ M_o &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & M_f &= \begin{bmatrix} \rho & 0 \\ 0 & \rho \cos u_f \end{bmatrix} \\ T_o &= \begin{bmatrix} 0 & 0 \end{bmatrix} & T_f &= \begin{bmatrix} 0 & -1/\rho \tan u_f \end{bmatrix}. \end{aligned}$$

The equations governing the evolution of the contact point are

$$\begin{bmatrix} \dot{u}_f \\ \dot{v}_f \\ \dot{u}_o \\ \dot{v}_o \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 \\ \sec u_f \\ -\rho \sin \psi \\ -\rho \cos \psi \\ -\tan u_f \end{bmatrix} \omega_x + \begin{bmatrix} -1 \\ 0 \\ -\rho \cos \psi \\ \rho \sin \psi \\ 0 \end{bmatrix} \omega_y. \tag{5.39}$$

6.3 Grasp kinematics with rolling

We are now in a position to describe the grasp kinematics when the fingers are allowed to roll and possibly slide on the object (depending on the contact model). Figure 5.24 shows the coordinate frames which we shall use. We assume that the finger and object shapes are completely known and that local, orthogonal, surface parameterizations are available.

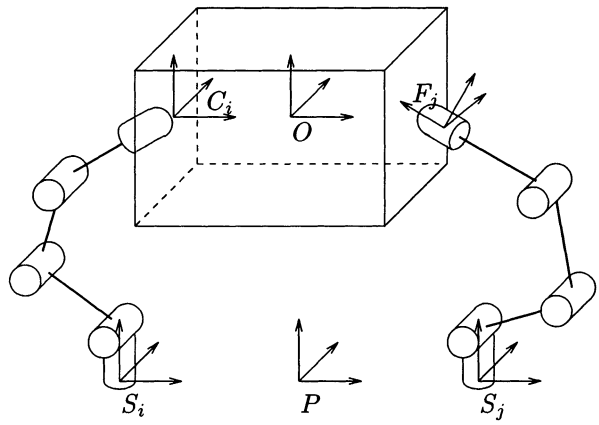


Figure 5.24: Grasp coordinate frames for moving contacts.

Abstractly, the derivation of the grasp kinematics is identical to the fixed contact case except that the grasp map and hand Jacobian depend on the instantaneous contact location, which in turn depends on η_i, the contact coordinates for each finger. Thus, the contact kinematics have the form

$$\begin{aligned} J_h(\theta, x_o, \eta)\dot{\theta} &= G^T(\eta)V_{po}^b \\ \dot{\eta}_i &= A_i(\eta_i)V_{of_i}^b, \end{aligned} \tag{5.40}$$

where $\eta = (\eta_1, \ldots, \eta_k) \in \mathbb{R}^{5k}$ is the vector of contact coordinates and $A_i(\eta_i) \in \mathbb{R}^{5\times 6}$ encodes the contact kinematics for each finger, as given in equation (5.28). One additional fact is needed to extend the previous analysis to the moving contact case: $B_{c_i}^T V_{oc_i}^b = 0$. This condition is a statement that the contact point is not allowed to move in any of the constrained directions. A complete derivation is left as an exercise.

Equation (5.40) describes the grasp kinematics in terms of a set of ordinary differential equations. To analyze the properties of the grasp, we make use of the following result. Let $W \subset SE(3)$ be the set of all configurations g_{of} for which the two objects are in contact.

Proposition 5.12. Smooth dependence of η on g_{of}
There is a smooth local bijection between $\eta \in \mathbb{R}^5$ and $g_{of} \in W \subset SE(3)$ if the matrix

$$K_f + R_\psi K_o R_\psi^T$$

is full rank.

The proof of this proposition requires application of the implicit function theorem to the mapping $\eta \mapsto g_{of} \in W$ and can be found in [78].

Since η_i is a smooth function of $g_{of} = g_{po}^{-1} g_{pf_i}$ and g_{pf_i} is a smooth function of θ, we can write this constraint as

$$J_h(\theta, x_o)\dot{\theta} = G^T(\theta, x_o) V_{po}^b. \tag{5.41}$$

where $x_o := g_{po}$. Equation (5.41) is a direct extension of equation (5.15) to the moving contact case. All of the definitions and properties which held for fixed contact grasps also hold for moving contact grasps.

For most applications, finding η as a function of g_{of} is not computationally feasible and we must use η directly. To make this explicit, let $\eta_i = (\alpha_{f_i}, \alpha_{o_i}, \psi_i)$ represent the contact coordinates for the i^{th} contact. In order to calculate J_h and G, we must compute the transformations $\mathrm{Ad}^{-1}_{g_{s_i c_i}}$ and $\mathrm{Ad}^{T}_{g^{-1}_{oc_i}}$, which are the only quantities that vary with η. In fact, since

$$\mathrm{Ad}^{-1}_{g_{s_i c_i}} = \left(\mathrm{Ad}_{g_{s_i p}} \mathrm{Ad}_{g_{po}} \mathrm{Ad}_{g_{oc_i}}\right)^{-1} = \mathrm{Ad}^{-1}_{g_{oc_i}} \mathrm{Ad}^{-1}_{g_{po}} \mathrm{Ad}^{-1}_{g_{s_i p}},$$

the only new quantity which really needs to be calculated is $\mathrm{Ad}^{-1}_{g_{oc_i}}$.

Without loss of generality, we assume that the contact frame is aligned with the normalized Gauss frame for the object and hence

$$R_{oc_i} = \left[\frac{\partial c_{o_i}}{\partial \alpha_{o_i}} M_{o_i}^{-1} \quad n_{o_i}\right],$$

where n_{o_i} is the outward pointing surface normal. Furthermore, p_{oc_i} is just the contact location $c_{o_i}(\alpha_i)$ and hence

$$g_{oc_i} = \begin{bmatrix} \left[\frac{\partial c_{o_i}}{\partial \alpha_{o_i}} M_{o_i}^{-1} \quad n_{o_i}\right] & c_{o_i}(\alpha_i) \\ 0 & 1 \end{bmatrix}. \tag{5.42}$$

Finally, although in principle the contact coordinates can be obtained by integrating the contact kinematics, in practice it is much more likely that one would need to measure each η_i directly in order to maintain accuracy (see, for example, [7]). These and other implementation details do not alter the overall structure and properties of the grasp, but can have a dramatic effect on the performance of the system.

7 Summary

The following are the key concepts covered in this chapter:

1. A *contact* is described by a mapping between forces exerted by a finger at a point on the object and the resultant wrenches in some object reference frame. The contact basis $B_{c_i} : \mathbb{R}^{m_i} \to \mathbb{R}^p$ describes the set of wrenches that can be exerted by the finger, written in the contact coordinate frame. For contacts with friction, the friction cone $FC_{c_i} \subset \mathbb{R}^{m_i}$ models the range of allowable contact forces that can be applied. The friction cone satisfies the following properties:

   (a) FC_{c_i} is a closed subset of $\mathbb{R}^{m_i}$ with non-empty interior.

   (b) $f_1, f_2 \in FC_{c_i} \implies \alpha f_1 + \beta f_2 \in FC_{c_i}$ for $\alpha, \beta > 0$.

2. A *grasp* is a collection of fingers which exert forces on an object. The net object wrench is determined from the individual contact forces by the relationship $F_o = G f_c$, where $G \in \mathbb{R}^{p \times m}$ is the *grasp map*:

$$G = \begin{bmatrix} \mathrm{Ad}^T_{g^{-1}_{oc_1}} B_{c_1} & \cdots & \mathrm{Ad}^T_{g^{-1}_{oc_k}} B_{c_k} \end{bmatrix}.$$

   $\mathrm{Ad}^T_{g^{-1}_{oc_i}} : \mathbb{R}^p \to \mathbb{R}^p$ is a the wrench transformation between the object and contact coordinate frames. The contact forces must all lie within the friction cone $FC = FC_{c_1} \times \cdots \times FC_{c_k}$.

3. A grasp is *force-closure* when finger forces lying in the friction cone span the space of object wrenches

$$G(FC) = \mathbb{R}^p.$$

   A grasp is force-closure if and only if the grasp map is surjective and there exists an *internal force* f_N which satisfies $G f_N = 0$ and $f_N \in \mathrm{int}(FC)$.

4. The *fundamental grasp constraint* describes the relationship between finger velocity and object velocity:

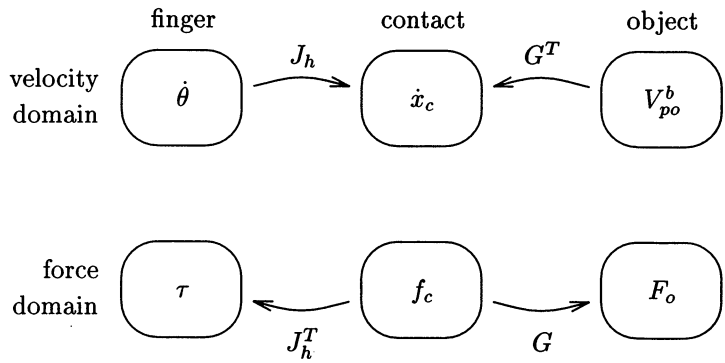
$$J_h(\theta, x_o)\dot{\theta} = G^T(\theta, x_o)\dot{x}_o,$$

   where $\theta \in \mathbb{R}^n$ is the vector of finger joint angles and $x_o := g_{po}$ is the configuration of the object frame relative to the palm frame. The *hand Jacobian* $J_h \in \mathbb{R}^{m \times n}$ is defined as

$$J_h = \begin{bmatrix} B^T_{c_1} \mathrm{Ad}^{-1}_{g_{s_1 c_1}} J^s_{s_1 f_1}(\theta_{f_1}) & & 0 \\ & \ddots & \\ 0 & & B^T_{c_k} \mathrm{Ad}^{-1}_{g_{s_k c_k}} J^s_{s_k f_k}(\theta_{f_k}) \end{bmatrix},$$

where $J^s_{s_i f_i}$ is the spatial Jacobian for the i^{th} finger and $\mathrm{Ad}^{-1}_{g_{s_i c_i}}$ is the twist transformation between the base and contact frames. For contacts in which rolling does not occur, G is a constant matrix.

5. The relationships between the forces and velocities in a multifingered grasp are summarized in the following diagram:



6. A grasp is *manipulable* when arbitrary motions can be generated by the fingers

$$\mathcal{R}(G^T) \subset \mathcal{R}(J_h).$$

A force-closure grasp is manipulable if and only if J_h is surjective.

7. The *contact kinematics* describe how the contact points move along the surface of the fingers and object. For an individual rolling contact, the contact kinematics are

$$\dot{\alpha}_f = M_f^{-1}(K_f + \tilde{K}_o)^{-1} \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix}$$

$$\dot{\alpha}_o = M_o^{-1} R_\psi (K_f + \tilde{K}_o)^{-1} \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix}$$

$$\dot{\psi} = T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o.$$

where (M_i, K_i, T_i) are the geometric parameters for a given coordinate chart on the surface. The contact kinematics allow G and J_h to be computed using $\eta = (\alpha_f, \alpha_o, \psi)$ rather than solving for η in terms of g_{po}.

8 Bibliography

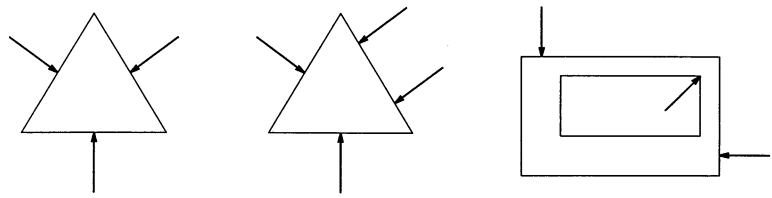
This chapter grew out of notes written for a workshop organized by Roger Brockett for the American Mathematical Society Annual Meeting in Liousville, Kentucky in January of 1989 [78]. Early treatments of the kinematics and statics of grasping can be found in the dissertations of Salisbury [102] and Kerr [49], on which much of Section 2 is based. See also the

work of Nakamura [81]. The material on force-closure and grasp planning is drawn from a number of sources, most notably Nguyen [83] and the seminal paper by Mishra, Schwartz, and Sharir [72]. Finger kinematics and issues of manipulability were described in Li et al. [63].

The kinematics of rolling contact were originally derived by Montana [73] and have been extended to the case of compliant contacts [74]. An alternate derivation of the equations of rolling contact can be found in [78].

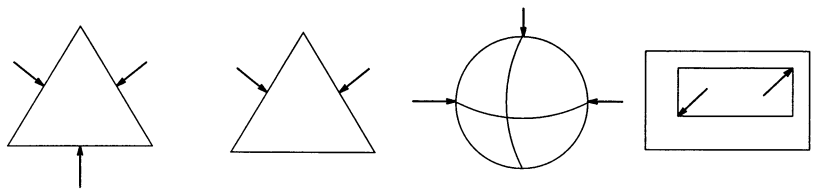
9 Exercises

1. Construct the grasp map for the grasps shown below and determine if the grasp is force-closure. Assume that all contacts are frictionless point contacts.



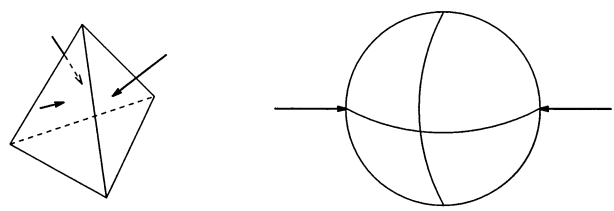
(a) (b) (c)

2. Construct the grasp map for the grasps shown below and write the friction cone conditions with respect to the contact basis you choose. Determine if the grasp is force-closure. Assume that all contacts are point contacts with friction.



$\mu = \tan 15°$ (a) $\mu = \tan 45°$ (b) $\mu > 0$ (c) $\mu > 0$ (d)

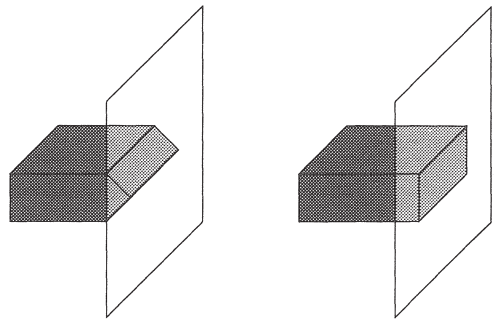
3. Construct the grasp map for the grasps shown below and write the friction cone conditions with respect to the contact basis you choose. Determine if the grasp is force-closure. Assume that all contacts are soft-finger contacts.



$\mu = \tan 15°$, $\gamma > 0$ (a) $\mu > 0$, $\gamma > 0$ (b)

4. *Additional contact models*
Derive the wrench basis and friction cone for the contact models

shown below. Assume the coefficient of sliding friction is $\mu > 0$.



Line contact (a) Plane contact (b)

5. Verify the lower bounds given in Table 5.3 by constructing the following objects. Assume that vertex contacts are not allowed.

   (a) Construct a 2-D object which cannot be grasped by two point-contact-with-friction fingers and $\mu = \tan 30°$. Prove explicitly that the object cannot be grasped.

   (b) Construct a 3-D object which cannot be grasped by three point-contact-with-friction fingers and $\mu = \tan 15°$. Prove explicitly that the object cannot be grasped.

   (c) True or false: For *any* convex 3-D object, a force-closure grasp can be constructed using at most three soft-finger contacts at smooth points (i.e., no vertex contacts). Prove or construct a counterexample.

   (d) (contributed by J. Canny). Prove that for any 3-D object with *smooth* boundary (not just piecewise smooth), a force-closure grasp can be constructed using at most three point contacts with friction or two soft-finger contacts.

6. When controlling a multifingered robot hand, it is important to insure that desired contact forces lie strictly inside the friction cone to avoid slip. That is, we require that for any F_o, there exists f_c such that $F_o = Gf_c$ and $f_c \in \text{int}(FC)$. Such a grasp is said to be *prehensile*. Let FC be the friction cone defined in Section 2. Show that if $G(FC) = \mathbb{R}^p$, then $G(\text{int}(FC)) = G(FC)$ using the following steps:

   (a) Show that $F \in \text{int}(FC) \implies \alpha F \in \text{int}(FC)$ for all $\alpha > 0$.

   (b) Show $F \in \text{int}(FC)$, $F' \in FC \implies F + F' \in \text{int}(FC)$.

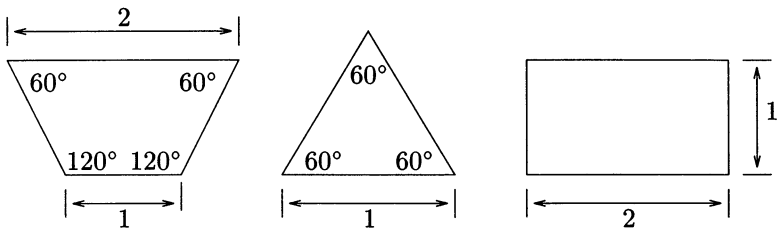
   (c) Show that there exists $f_N \in \text{int}(FC)$ such that $Gf_N = 0$.

(d) Complete the proof by showing that $G(FC) \subseteq G(\text{int}(FC))$.

Hence, a grasp is prehensile if and only if the grasp is force-closure.

7. Prove Theorem 5.6: Show that a planar grasp with two point contacts with friction is force-closure if and only if the line connecting the contact point lies inside both friction cones.

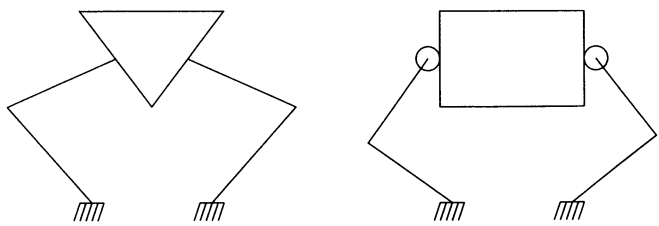
8. For the objects given below, find all force-closure grasps using two contacts with friction. On each the objects, draw the independent contact regions corresponding to a maximal square contained in a force-closure region. Assume that the coefficient of friction for all contacts is $\mu = \tan 45°$.



(a) (b) (c)

9. Find all the force-closure grasps for two point contacts with friction grasping a circle. Express your answer as a sketch of the force-closure regions versus the two finger locations on the circle. Indicate on the circle a set of independent contact regions corresponding to a maximal square contained in the force-closure region.

10. Derive the contact constraints for the hands shown below. Determine if the grasps are force-closure and/or manipulable at the configuration shown.



(a) (b)

11. Calculate the contact constraints for the grasp shown in Figure 5.17 with $l_3 > 0$.

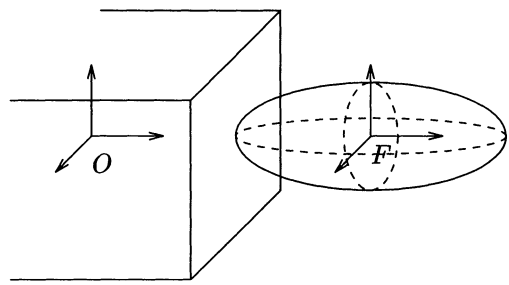
12. Consider the two-fingered grasp shown in Figure 5.17. Equate the locations of the fingertips with the contact locations on the box.

Differentiate this algebraic constraint and show that it is equivalent to the answer given in the example at the end of Section 5.

13. Give an example of two surfaces in contact which has singular relative curvature form.

14. Calculate the geometric parameters for an ellipsoid, a parabaloid, and a torus.

15. The figure below shows an elliptic fingertip in contact with a flat object. The principal axes of the ellipse have length a, b, and c, respectively.



(a) Give an orthonormal coordinate chart for the fingertip around the point of contact as shown in the figure.

(b) Assume the fingertip is in rolling contact with the object. Derive the equations of contact.

(c) Compute the velocity of the fingertip relative to the object which satisfies rolling constraint and produces a contact velocity of $\dot{\alpha}_o = (0, v)$, $v \in \mathbb{R}$.

16. Derive the equations of contact for a unit sphere in rolling contact with a sphere of radius ρ.

17. *Kinematics of planar contact*
The kinematics of contact for two planar objects can be obtained by restricting equation (5.28) to the plane. Let $g_{of} = (p, R) \in SE(2)$ be the relative configuration of the objects.

(a) Let $s_o \in \mathbb{R}$ be the point of contact on the object and $s_f \in \mathbb{R}$ be the point of contact on the finger. Assume that the surface is parameterized by arc length, so that $\| \frac{\partial c_i}{\partial s_i} \| = 1$ where $c_i : \mathbb{R} \to \mathbb{R}^2$ is a coordinate chart. Show that the contact coordinates

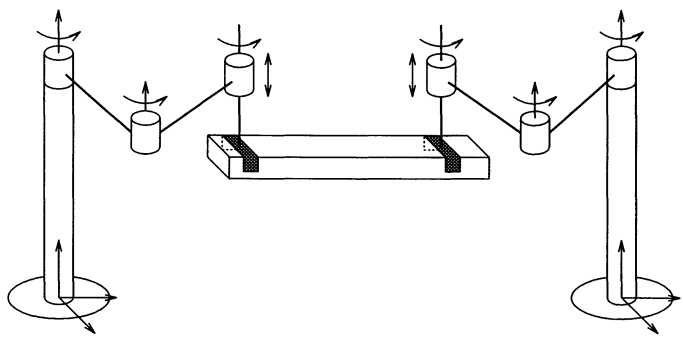
$\eta = (s_f, s_o)$ evolve according to

$$\dot{s}_f = (K_o + K_f)^{-1} \left(\frac{\partial c_o}{\partial s_o}^T \dot{R} n_f + K_o \frac{\partial c_o}{\partial s_o}^T (\dot{R} c_f + \dot{p}) \right)$$

$$\dot{s}_o = (K_o + K_f)^{-1} \left(-\frac{\partial c_o}{\partial s_o}^T \dot{R} n_f + K_f \frac{\partial c_o}{\partial s_o}^T (\dot{R} c_f + \dot{p}) \right)$$

(b) Derive the equations of contact for a planar, elliptical finger rolling against a flat object.

18. Consider the coordinated lifting problem shown below. Derive the constraints between the beam velocity and the robot joint velocities. Assume that each finger firmly grips the end of the beam and that the forces are transmitted to the beam at the center of the gripper.



19. Rederive the contact kinematics when the finger and object surfaces are given as the level sets of functions $h_f : \mathbb{R}^3 \to \mathbb{R}$ and $h_o : \mathbb{R}^3 \to \mathbb{R}$ (for example, a sphere of radius ρ satisfies $h_f(x) = x_1^2 + x_2^2 + x_3^2 - \rho^2 = 0$). Test your solution using an ellipsoid rolling on a plane.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 6

Hand Dynamics and Control

DOI: 10.1201/9781315136370-6
This chapter has been made available under a CC-BY 4.0 license.

In this chapter, we study the dynamics and control of a set of robots performing a coordinated task. Our primary example will be that of a multifingered robot hand manipulating an object, but the formalism is considerably broader. It allows a unified treatment of dynamics and control of robot systems subject to a set of velocity constraints, generalizing the treatment given in Chapter 4.

1 Lagrange's Equations with Constraints

For an open-chain manipulator, the equations of motion can be derived using Lagrange's equations or other similar methods. This involves finding a set of generalized coordinates which completely and minimally parameterize the configuration space of the system, and then writing the dynamics in terms of these coordinates and the corresponding generalized forces.

For a multifingered robot hand, the configuration of the system depends on the joint angles for the fingers as well as the position and orientation of the object. These quantities are not independent, however, since their velocities are related by the grasping constraint. Thus, we cannot apply Lagrange's equations directly. Finding a set of generalized coordinates in such examples is non-trivial and, in some cases, impossible.

To overcome this difficulty, we rederive the equations of motion for a mechanical system in the presence of constraints. That is, rather than attempting to eliminate the constraints by an appropriate choice of coordinates, we seek to incorporate the constraints directly into the equations of motion. The remainder of this section contains a sketch of this derivation. A more complete derivation can be found in Rosenberg [100] or Pars [90].

For simplicity, we assume throughout this section that the configuration space Q is an open subset of $\mathbb{R}^n$ with coordinates $q = (q_1, \ldots, q_n)$. More general configuration spaces can be handled by an appropriate choice of local coordinates.

1.1 Pfaffian constraints

A *constraint* on a mechanical system restricts the motion of the system by limiting the set of paths which the system can follow. A simple example is the case of two particles attached by an inextensible, massless rod. The configuration of each particle is described as a point $p_i \in \mathbb{R}^3$, but all trajectories of the particles must satisfy the algebraic constraint

$$\|p_1 - p_2\|^2 = L^2, \tag{6.1}$$

where L is the length of the rod. The constraint acts through the application of *constraint forces*, which modify the motion of the system to insure that the constraint is always satisfied. In the case of the two interconnected particles, the constraint force corresponds to the tension in the rod, which transmits forces applied on one particle to the other particle, and causes the distance between the particles to remain fixed.

The constraint in equation (6.1) is an example of a *holonomic* constraint. More generally, a constraint is said to be holonomic if it restricts the motion of the system to a smooth hypersurface in the (unconstrained) configuration space Q. Holonomic constraints can be represented locally as algebraic constraints on the configuration space,

$$h_i(q) = 0, \qquad i = 1, \ldots, k. \tag{6.2}$$

Each h_i is a mapping from Q to $\mathbb{R}$ which restricts the motion of the system. We assume that the constraints are linearly independent and hence the matrix

$$\frac{\partial h}{\partial q} = \begin{bmatrix} \frac{\partial h_1}{\partial q_1} & \cdots & \frac{\partial h_1}{\partial q_n} \\ & \ddots & \\ \frac{\partial h_k}{\partial q_1} & \cdots & \frac{\partial h_k}{\partial q_n} \end{bmatrix}$$

is full row rank. (In the classical mechanics literature, constraints of the form in equation (6.2) are sometimes referred to as *scleronomic* constraints. Holonomic and scleronomic come from Greek and mean respectively "all together lawful" and "rigid" respectively. Time-varying constraints on q are called *rheonomic*, for "flowing." We will not use the terms scleronomic and rheonomic in this book, only the term holonomic.)

Since holonomic constraints define a smooth hypersurface in the configuration space, it is possible to "eliminate" the constraints by choosing a set of coordinates for this surface. These new coordinates parameterize

all allowable motions of the system and are not subject to any further constraints. In fact, this is precisely the technique which we use when writing down the motion of a rigid body in terms of the position and orientation of a single coordinate frame rather than the (constrained) motion of the individual points of the rigid body.

The constraint forces for a set of holonomic constraints of the form in equation (6.2) are linear combinations of the gradients of the constraint functions $h_i : Q \to \mathbb{R}$. Letting $h : Q \to \mathbb{R}^k$ represent the vector-valued constraint function, we can thus write the constraint force as

$$\Gamma = \frac{\partial h}{\partial q}^T \lambda,$$

where $\lambda \in \mathbb{R}^k$ is the vector of relative magnitudes of the constraint forces. These constraint forces can be viewed as acting normal to the constraint surface, with the magnitude of the forces chosen to insure that the system remains on the constraint surface defined by equation (6.2). Note that no work is done by the constraints when the system is moved along feasible trajectories since

$$\Gamma \cdot \dot{q} = \lambda^T \frac{\partial h}{\partial q} \dot{q} = \lambda^T \frac{d}{dt}\big(h(q)\big) = 0.$$

A fundamentally different type of constraint occurs in the context of multifingered grasping, where the allowable motions of the system are restricted by the *velocity* constraint

$$J_h(\theta, x_o)\dot{\theta} = G^T(\theta, x_o)V_{po}^b.$$

More generally, for a system with configuration space Q, we consider velocity constraints of the form

$$A(q)\dot{q} = 0,$$

where $A(q) \in \mathbb{R}^{k \times n}$ represents a set of k velocity constraints. A constraint of this form is called a *Pfaffian constraint.* We assume that the constraints are pointwise linearly independent and hence that $A(q)$ is full row rank at $q \in Q$. For a multifingered hand, the matrix A has the form

$$A(q) = \begin{bmatrix} J_h(q) & -G^T(q) \end{bmatrix},$$

where $q = (\theta, x)$ and x represents a choice of local coordinates for the object position and orientation.

Since a Pfaffian constraint restricts the allowable velocities of the system but not necessarily the configurations, we cannot always represent it as an algebraic constraint on the configuration space. A Pfaffian constraint is said to be *integrable* if there exists a vector-valued function

$h : Q \to \mathbb{R}^k$ such that

$$A(q)\dot{q} = 0 \qquad \Longleftrightarrow \qquad \frac{\partial h}{\partial q}\dot{q} = 0.$$

Thus, an integrable Pfaffian constraint is equivalent to a holonomic constraint. It is important to note that we do not require that $A = \frac{\partial h}{\partial q}$, but only that they define the same subspace of the allowable velocities at every $q \in Q$.

A Pfaffian constraint which is not integrable is an example of a *nonholonomic* constraint. Nonholonomic constraints of this type occur when the instantaneous velocities of the system are constrained to an $n - k$ dimensional subspace, but the set of reachable configurations is not restricted to some $n-k$ dimensional hypersurface in the configuration space. As we shall see in the next chapter, not all Pfaffian constraints are integrable, and hence we must extend our derivation of the equations of motion to account for this case.

Despite the possibility that a constraint may be nonholonomic, it is still possible to speak of the forces of constraint. They are the forces which are generated by a set of Pfaffian constraints so as to insure that the system does not move in the directions given by the rows of the constraint matrix $A(q)$. The constraint forces at a configuration $q \in Q$ have the form

$$\Gamma = A^T(q)\lambda,$$

where $\lambda \in \mathbb{R}^k$ is the vector of relative magnitudes of the constraint forces. If the constraint happens to be integrable, then this is identical to the holonomic case since $A^T(q)$ and $\frac{\partial h}{\partial q}^T$ will have the same range space.

The constraint forces for a set of Pfaffian constraints prevent motion of the system in directions which would violate the constraints. In order to include these forces in the dynamics, we must add one additional assumption about the nature of the constraints. Namely, we assume that the forces which are generated by the constraints do no work on the system, and hence conserve energy. This assumption is often referred to as *d'Alembert's principle.*

The assumption that the constraints do no work is easy to justify in many situations. In particular, if a system is subject to rolling, either about a point or along a surface, then the frictional forces due to this rolling are very small and can usually be ignored. However, if some sliding occurs in addition to rolling, the work done depends upon the magnitude of the normal force, and the constraints cease to be workless. In cases such as these, friction and other nonconservative forces can be incorporated by ignoring them initially, and then adding them as external forces once the dynamics have been derived.

1.2 Lagrange multipliers

We can now proceed to derive the equations of motion for a mechanical system with configuration $q \in \mathbb{R}^n$ subject to a set of Pfaffian constraints. Let $L(q, \dot{q})$ represent the Lagrangian for the unconstrained system and let the constraints have the form

$$A(q)\dot{q} = 0 \qquad A(q) \in \mathbb{R}^{k \times n}. \tag{6.3}$$

We assume that constraints are everywhere smooth and linearly independent and that the forces of constraint do no work on the system.

The equations of motion are formed by considering the constraint forces as an additional force which affects the motion of the system. Hence, the dynamics can be written in vector form as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda - \Upsilon = 0, \tag{6.4}$$

where the columns of A^T form a non-normalized basis for the constraint forces and $\lambda \in \mathbb{R}^k$ gives the relative magnitudes of the forces of constraint. As before, Υ represents nonconservative and externally applied forces.

The scalars $\lambda_i, \ldots, \lambda_k$ are called *Lagrange multipliers*. They are determined by solving equations (6.3) and (6.4) for the $n + k$ variables q and λ, insuring that no motion occurs in the constrained directions, and hence equation (6.3) holds for all time. In general, each λ_i will be a function of q, $\dot{q}$, and Υ, since the constraint forces vary with the configuration, velocity, and applied force. Solving for these multipliers and substituting them back into the equations of motion gives a description of the dynamics of the system.

In the case that the Lagrangian has the form $L(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} - V(q)$ (kinetic minus potential), we can derive an explicit formula for the Lagrange multipliers. Using the notation from Chapter 4, the equations of motion can be written as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q, \dot{q}) + A^T(q)\lambda = F, \tag{6.5}$$

where F corresponds to the vector of external forces and $N(q, \dot{q})$ includes nonconservative forces as well as potential forces. Differentiating the constraint equation (6.3) yields

$$A(q)\ddot{q} + \dot{A}(q)\dot{q} = 0$$

and, solving for $\ddot{q}$ from equation (6.5), we obtain

$$(AM^{-1}A^T)\lambda = AM^{-1}(F - C\dot{q} - N) + \dot{A}\dot{q},$$

where we suppress the dependence on q and $\dot{q}$. The configuration dependent matrix $AM^{-1}A$ is full rank if the constraints are independent, and

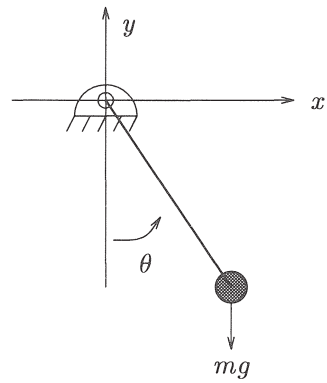


Figure 6.1: Idealized planar pendulum.

hence the Lagrange multipliers are

$$\lambda = (AM^{-1}A^T)^{-1}\left(AM^{-1}(F - C\dot{q} - N) + \dot{A}\dot{q}\right). \tag{6.6}$$

Using this equation, the Lagrange multipliers can be computed as a function of the current state, q and $\dot{q}$, and the applied forces, F. The equations of motion are now given by equation (6.5) with λ defined as in equation (6.6).

Example 6.1. Dynamics of an idealized planar pendulum
Consider an idealized pendulum, with its mass concentrated at the tip, as shown in Figure 6.1. The mass is a particle with two degrees of freedom; the pendulum mechanism acts as a constraint which restricts the motion to a single degree of freedom. Rather than parameterize the system in terms of the angular variable θ, we instead derive the equations using the method of Lagrange multipliers.

The configuration of the system is given by $q = (x, y) \in \mathbb{R}^2$. The constraint that the length of the pendulum remain fixed can be written as

$$x^2 + y^2 = l^2.$$

Differentiating this constraint and dividing by two, we obtain the Pfaffian constraint

$$\underbrace{\begin{bmatrix} x & y \end{bmatrix}}_{A(q)} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 0.$$

The (unconstrained) Lagrangian is simply $L(q, \dot{q}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy$.

Substituting into equations (6.5), Lagrange's equations for the constrained system become

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \lambda = 0.$$

The Lagrange multiplier λ is determined from equation (6.6), which gives

$$\begin{aligned}\lambda &= (AM^{-1}A^T)^{-1}\left(AM^{-1}(Q - C\dot{q} - N) - \dot{A}\dot{q}\right) \\ &= \frac{m}{x^2 + y^2}(-gy - \dot{x}^2 - \dot{y}^2) = -\frac{m}{l^2}(gy + \dot{x}^2 + \dot{y}^2),\end{aligned}$$

and hence the equations of motion are

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} - \frac{1}{l^2} \begin{bmatrix} x \\ y \end{bmatrix} \left(mgy + m(\dot{x}^2 + \dot{y}^2)\right) = 0.$$

Note that this is a second-order differential equation in two variables, x and y, even though the actual system only has one degree of freedom. Thus, we have increased the number of variables required to represent the motion of the system. On the other hand, we have an explicit measure of the tension in the rod supporting the mass, namely

$$\text{Tension} = \left\| \begin{bmatrix} x \\ y \end{bmatrix} \lambda \right\| = \frac{mg}{l} y + \frac{m}{l}(\dot{x}^2 + \dot{y}^2).$$

1.3 Lagrange-d'Alembert formulation

It is convenient and useful to rederive the equations of motion without explicitly solving for the instantaneous constraint forces present in the system. In essence, this proceeds by projecting the motion of the system onto the feasible directions and ignoring the forces in the constrained directions. In doing so, we will be able to get a more concise description of the dynamics which is in a form well suited for closed-loop control.

At a given configuration $q \in \mathbb{R}^n$, the instantaneous set of directions in which the system is allowed to move is given by the null space of the constraint matrix, $A(q)$. We adopt the classical notation and call a vector $\delta q \in \mathbb{R}^n$ which satisfies $A(q)\delta q = 0$ a *virtual displacement.* If F is a generalized force applied to the system, then we call $\delta W = F \cdot \delta q$ the *virtual work* due to a force F acting along a virtual displacement δq. D'Alembert's principle states that the forces of constraint do no virtual work. Hence,

$$(A^T(q)\lambda) \cdot \delta q = 0 \qquad \text{for } A(q)\delta q = 0.$$

It is important to keep in mind that δq is not the same as $\dot{q}$. The generalized velocity $\dot{q}$ satisfies both the velocity constraints *and* the equations of motion. The virtual displacement only satisfies the constraints. Hence, d'Alembert's principle asserts that constraint forces do no work for *any* instantaneous motion which satisfies the constraints, not just for the motion which the system actually follows.

To eliminate the constraint forces from equation (6.4), we project the equations of motion onto the linear subspace generated by the null space of $A(q)$. Since $(A^T\lambda)\cdot\delta q = 0$, equation (6.4) becomes

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \Upsilon\right)\cdot\delta q = 0, \tag{6.7}$$

where $\delta q \in \mathbb{R}^n$ satisfies

$$A(q)\delta q = 0. \tag{6.8}$$

We call equations (6.7) and (6.8) the *Lagrange-d'Alembert equations.* Note that in the case where there are no constraints on the system, δq is free and equation (6.7) reduces to the usual form of Lagrange's equations.

To get a more explicit description of the dynamics, we assume that $A(q)$ has the form

$$A(q) = \begin{bmatrix} A_1(q) & A_2(q) \end{bmatrix},$$

where $A_2(q) \in \mathbb{R}^{k\times k}$ is invertible. This can always be achieved locally by reordering the configuration space variables. We now relabel the configuration as $q = (q_1, q_2) \in \mathbb{R}^{n-k}\times\mathbb{R}^k$ so that

$$A(q)\cdot\delta q = 0 \qquad \Longleftrightarrow \qquad \delta q_2 = -A_2^{-1}(q)A_1(q)\delta q_1,$$

where δq_1 is free (unconstrained). Making use of this same notation in the Lagrange-d'Alembert equations yields

$$\begin{aligned}
&\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \Upsilon\right)\cdot\delta q \\
&\quad = \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right)\cdot\delta q_1 + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right)\cdot\delta q_2 \\
&\quad = \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right)\cdot\delta q_1 + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right)\cdot\left(-A_2^{-1}A_1\right)\delta q_1,
\end{aligned}$$

and since δq_1 is free, the equations of motion become

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right) - A_1^T A_2^{-T}\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right) = 0. \tag{6.9}$$

Equation (6.9) is a second-order differential equation in terms of $q = (q_1, q_2)$. We can further simplify this equation using the constraint $\dot{q}_2 = -A_2^{-1}A_1\dot{q}_1$ to eliminate $\dot{q}_2$ and $\ddot{q}_2$. The evolution of q_2 can be retrieved by reapplication of the constraint equations. This procedure is illustrated in the next example.

Example 6.2. Dynamics of a rolling disk

Consider the example of a disk rolling on the plane, as shown in Figure 6.2. The configuration of the disk is given by the xy position in the

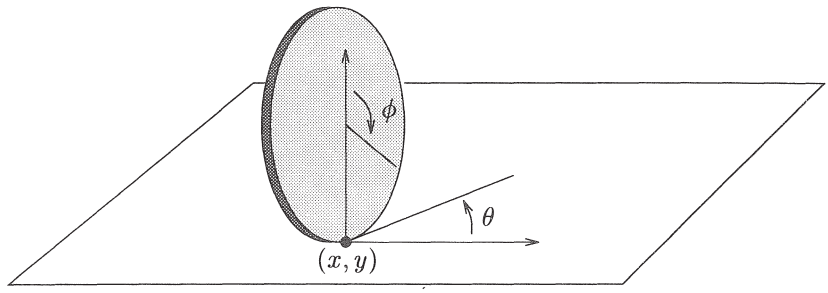


Figure 6.2: Disk rolling on a plane.

plane, the heading angle θ, and the orientation of the disk with respect to the vertical, ϕ. We write $q = (x, y, \theta, \phi)$ and let ρ denote the radius of the disk. We take as inputs the driving torque on the wheel, τ_θ, and the steering torque (about the vertical axis), τ_ϕ.

We make the assumption that the disk rolls without slipping, just as in the case of a fingertip rolling on an object. This condition can be written as a set of velocity constraints

$$\begin{aligned} \dot{x} - \rho\cos\theta\dot{\phi} &= 0 \\ \dot{y} - \rho\sin\theta\dot{\phi} &= 0 \end{aligned} \qquad \text{or} \qquad A(q)\dot{q} = \begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix} \dot{q} = 0. \tag{6.10}$$

These constraints require that the disk roll in the direction in which it is heading and that the velocity of the disk match the rate at which it is rolling on the plane. The constraints are everywhere linearly independent.

The Lagrangian for this system is the kinetic energy associated with the disk, ignoring the constraints. Let m be the mass of the disk, $\mathcal{I}_1$ its inertia about the horizontal (rolling) axis, and $\mathcal{I}_2$ its inertia about the vertical axis. The Lagrangian is

$$L(q, \dot{q}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}\mathcal{I}_1\dot{\theta}^2 + \frac{1}{2}\mathcal{I}_2\dot{\phi}^2.$$

We can now proceed to derive the equations of motion for the system. Let $\delta q = (\delta x, \delta y, \delta\theta, \delta\phi)$ represent a virtual displacement of the system. The Lagrange-d'Alembert equations are given by

$$\left(\begin{bmatrix} m & & & \\ & m & & \\ & & \mathcal{I}_1 & \\ & & & \mathcal{I}_2 \end{bmatrix} \ddot{q} - \begin{bmatrix} 0 \\ 0 \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \right) \cdot \delta q = 0 \quad \text{where} \quad \begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix} \delta q = 0.$$

From the form of the constraints, we can solve for δx and δy to obtain

$$\begin{aligned} \delta x &= \rho\cos\theta\delta\phi \\ \delta y &= \rho\sin\theta\delta\phi. \end{aligned} \tag{6.11}$$

The equations of motion can now be rewritten as

$$\left(\begin{bmatrix} 0 & 0 \\ m\rho\cos\theta & m\rho\sin\theta \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \mathcal{I}_1 & 0 \\ 0 & \mathcal{I}_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} - \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix} \right) \cdot \begin{bmatrix} \delta\theta \\ \delta\phi \end{bmatrix} = 0,$$

and since $\delta\theta$ and $\delta\phi$ are free, the dynamics become

$$\begin{bmatrix} 0 & 0 \\ m\rho\cos\theta & m\rho\sin\theta \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \mathcal{I}_1 & 0 \\ 0 & \mathcal{I}_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix}. \tag{6.12}$$

We can further simplify the equations by reusing the constraints to eliminate $\dot{x}$, $\dot{y}$ and $\ddot{x}$, $\ddot{y}$. Differentiating the constraints, we have

$$\begin{aligned} \ddot{x} &= \rho\cos\theta\ddot{\phi} - \rho\sin\theta\dot{\theta}\dot{\phi} \\ \ddot{y} &= \rho\sin\theta\ddot{\phi} + \rho\cos\theta\dot{\theta}\dot{\phi}, \end{aligned} \tag{6.13}$$

and substituting into equation (6.12) gives

$$\begin{bmatrix} \mathcal{I}_1 & 0 \\ 0 & \mathcal{I}_2 + m\rho^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} \tau_\theta \\ \tau_\phi \end{bmatrix}, \tag{6.14}$$

which describes the motion of the system as a second-order differential equation in θ and ϕ. Note that for this system the equations of motion for θ and ϕ do not depend on x and y, but in general this is not the case.

The motion of the x and y positions of the disk can be retrieved from the first-order differential equations

$$\begin{aligned} \dot{x} &= \rho\cos\theta\dot{\phi} \\ \dot{y} &= \rho\sin\theta\dot{\phi}. \end{aligned}$$

Thus, given the trajectory of θ and ϕ, we can determine the trajectory of the disk as it rolls along the plane. The splitting of the equations of motion into a set of second-order equations in a reduced set of variables plus a set of first-order equations representing the constraints is characteristic of nonholonomic systems.

1.4 The nature of nonholonomic constraints

The machinery that we have developed in this section allows us to calculate the dynamics of a mechanical system subject to a set of Pfaffian constraints without trying to integrate the constraints and find a set of generalized coordinates which completely (and minimally) parameterize the configuration of the system. In the case that the constraints *are* integrable, these equations are identical to those obtained by finding a set of appropriate generalized coordinates. When the constraints are nonintegrable, it is very important to incorporate the constraints in the proper

way. In this section, we try to shed some light on how nonintegrable constraints affect the mechanics of the system and indicate what can go wrong if one is not careful.

A common mistake when deriving the equations of motion for a mechanical system with nonholonomic constraints is to substitute the constraints into the Lagrangian and then apply Lagrange's equations. This would seem to eliminate the dependent variables and minimize computations. As we shall see in a moment, however, this gives the wrong equations of motion for the system. We have been very careful in this section to always compute the *unconstrained* Lagrangian, substitute into the Lagrange-d'Alembert equations, and *then* reapply the constraints to eliminate the dependent variables.

To see what happens when the constraints are used at the wrong time, consider a three-dimensional, unforced Pfaffian system with configuration $q = (r, s) \in \mathbb{R}^2 \times \mathbb{R}$, constraint

$$\dot{s} + a^T(r)\dot{r} = 0 \qquad a(r) \in \mathbb{R}^2,$$

and Lagrangian $L(r, \dot{r}, \dot{s})$. For simplicity, we assume that both the Lagrangian and the constraints do not depend on the variable s. Define the *constrained Lagrangian* as

$$L_c(r, \dot{r}) = L(r, \dot{r}, -a^T(r)\dot{r}).$$

The constrained Lagrangian is the kinetic minus potential energy of the system, but evaluated with the constraints taken into account.

Suppose we substitute the constrained Lagrangian into the Lagrange-d'Alembert equations. Since there is no dependence on either s or $\dot{s}$, these equations yield

$$\frac{d}{dt}\frac{\partial L_c}{\partial \dot{r}_i} - \frac{\partial L_c}{\partial r_i} = 0 \qquad i = 1, 2.$$

Expanding the equations using the definition of L_c, we obtain

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}_i} - a_i(r)\frac{\partial L}{\partial \dot{s}}\right) - \left(\frac{\partial L}{\partial r_i} - \frac{\partial L}{\partial \dot{s}}\sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_j\right) = 0$$

and, rearranging terms, we can write this equation as

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{r}_i} - \frac{\partial L}{\partial r_i}\right) - a_i(r)\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s}\right) = \frac{\partial L}{\partial \dot{s}}\left(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i\right). \tag{6.15}$$

Notice that the left-hand side of this equation is exactly the Lagrange-d'Alembert equation for the system. The right-hand side of the equation represents spurious terms that arise from substituting the constraints

into the Lagrangian. This shows that, in general, substituting a set of Pfaffian constraints into the Lagrangian and then applying the Lagrange-d'Alembert equations gives the *wrong equations.*

Suppose now that the constraint is actually holonomic. For simplicity, we assume that the constraint is explicitly integrable, so that there exists a function $h(r)$ such that

$$a_i(r) = \frac{\partial h}{\partial r_i}.$$

In this case, the right-hand side of equation (6.15) is

$$\frac{\partial L}{\partial \dot{s}}(\dot{a}_i(r) - \sum_j \frac{\partial a_j}{\partial r_i}\dot{r}_i) = \frac{\partial L}{\partial \dot{s}}\left(\sum \frac{\partial^2 h}{\partial r_i \partial r_j}\dot{r}_j - \sum \frac{\partial^2 h}{\partial r_j \partial r_i}\dot{r}_i\right),$$

which is identically zero since mixed partials commute. Hence, if the constraints *happen* to be holonomic, then substituting the constraints into the Lagrangian will give the correct equations of motion.

2 Robot Hand Dynamics

Using the results from the previous section, we now derive the equations of motion for a system of robots performing a coordinated task. We begin with a multifingered hand manipulating an object and then show how the same formalism can be applied to other problems.

2.1 Derivation and properties

The overall dynamics for a multifingered hand grasping an object are obtained by combining the dynamics of the fingers and the object via the grasp constraint. Adopting the notation of the previous chapters, the finger dynamics have the form

$$M_f(\theta)\ddot{\theta} + C_f(\theta, \dot{\theta})\dot{\theta} + N_f(\theta, \dot{\theta}) = \tau, \tag{6.16}$$

where $\theta = (\theta_{f_1}, \ldots, \theta_{f_k}) \in \mathbb{R}^n$ is the vector of joint angles for all of the fingers in the hand and $\tau \in \mathbb{R}^n$ is the corresponding vector of joint torques. M_f, C_f, and N_f are formed by appropriately stacking the quantities obtained for the individual fingers:

$$M_f = \begin{bmatrix} M_{f_1} & & 0 \\ & \ddots & \\ 0 & & M_{f_k} \end{bmatrix} \quad C_f = \begin{bmatrix} C_{f_1} & & 0 \\ & \ddots & \\ 0 & & C_{f_k} \end{bmatrix} \quad N_f = \begin{bmatrix} N_{f_1} \\ \vdots \\ N_{f_k} \end{bmatrix}.$$

The object dynamics are given by the Newton-Euler equations, derived in Chapter 4. These equations have the form

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b,$$

where $V^b = (v^b, \omega^b)$ is the body velocity of the object, F^b is the body wrench, m is the mass, and $\mathcal{I}$ is the moment of inertia matrix, all with respect to a frame attached at the center of mass. We write $x_o = (p, R)$ for the configuration of the object. If the object is subject to gravity alone, the dynamics can be written as

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \dot{V}^b + \begin{bmatrix} m\widehat{\omega}^b & 0 \\ 0 & \frac{1}{2}(\widehat{\omega}^b\mathcal{I} - \mathcal{I}\widehat{\omega}^b) \end{bmatrix} V^b + \begin{bmatrix} R^T(m\vec{g}) \\ 0 \end{bmatrix} = 0, \quad (6.17)$$

where $\vec{g}$ is a vector in the direction of gravity, usually taken to be $\vec{g} = (0, 0, 1)$.

In order to apply the Lagrange-d'Alembert equations, it is necessary to rewrite the Newton-Euler equations in local coordinates. Letting $x \in \mathbb{R}^6$ be a local parameterization for $x_o \in SE(3)$, the object dynamics can be written as

$$M_o(x)\ddot{x} + C_o(x, \dot{x})\dot{x} + N_o(x, \dot{x}) = 0,$$

which has the same form as the robot dynamics in equation (6.16). It can be shown that $M_o(x) > 0$ and $\dot{M}_o - 2C_o$ is skew-symmetric (see Exercise 4).

The fingers and object are connected by the grasp constraints

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}, \quad (6.18)$$

written here in local coordinates. We make three assumptions about the structure of the grasp:

1. *The grasp is force-closure and manipulable.* Recall that this is equivalent to the conditions

$$G(FC) = \mathbb{R}^p \qquad \text{and} \qquad \mathcal{R}(G^T) \subset \mathcal{R}(J_h)$$

for all possible configurations.

2. *The hand Jacobian is invertible.* This condition insures that there are no redundant motions of the fingers and hence the hand has exactly the number of degrees of freedom required to complete the task.

3. *The contact forces remain in the friction cone at all times.* This condition is necessary in order to insure that the grasp constraints in equation (6.18) hold at all times. It is the task of the control law to insure that this condition holds, since otherwise the object can slip or drop from the grasp.

The first two assumptions can be relaxed; we discuss those cases in the next section.

The Lagrangian for the composite system is given by

$$L = \frac{1}{2}\dot{\theta}^T M_f \dot{\theta} + \frac{1}{2}\dot{x}^T M_o \dot{x} - V_f(\theta) - V_o(x), \tag{6.19}$$

where V_f and V_o are the potential energy terms due to gravity. Let $q = (\theta, x)$ represent the overall configuration of the system.

We can now apply the results of the previous section to derive the equations of motion. The velocity constraint in equation (6.18) generates a constraint on the virtual displacements $\delta\theta$ and δx, namely $\delta\theta = J_h^{-1} G^T \delta x$. Making use of this relationship, we can rewrite the Lagrange-d'Alembert equations as

$$\begin{aligned}
&\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} - \begin{bmatrix}\tau \\ 0\end{bmatrix}\right) \cdot \delta q \\
&\quad = \begin{bmatrix} \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} - \tau \\ \frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} \end{bmatrix} \cdot \begin{bmatrix}\delta\theta \\ \delta x\end{bmatrix} \\
&\quad = \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} - \tau\right) \cdot (J_h^{-1} G^T \delta x) + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) \cdot \delta x \\
&\quad = G J_h^{-T} \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} - \tau\right) \cdot \delta x + \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) \cdot \delta x,
\end{aligned}$$

where we have used the properties of the inner product in the final step to isolate δx. Since δx is free,

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x}\right) + G J_h^{-T} \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta}\right) = G J_h^{-T} \tau. \tag{6.20}$$

Equation (6.20) together with the velocity constraints given in equation (6.18) describe the system completely. Note that equation (6.20) is a second-order vector equation with $n - m$ rows and equation (6.18) is a first-order vector equation with m rows.

Since the Lagrangian for the system splits into two separate parts, the two terms on the left-hand side of equation (6.20) simplify into the object dynamics and the finger dynamics, respectively. Using this fact and eliminating $\dot{\theta}$ and $\ddot{\theta}$ via the constraints, the equations of motion become

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q, \dot{q})\dot{x} + \tilde{N}(q, \dot{q}) = F, \tag{6.21}$$

where $q = (\theta, x)$ and

$$\begin{aligned}
\tilde{M} &= M_o + G J_h^{-T} M_f J_h^{-1} G^T \\
\tilde{C} &= C_o + G J_h^{-T} \left(C_f J_h^{-1} G^T + M_f \frac{d}{dt}\left(J_h^{-1} G^T\right)\right) \\
\tilde{N} &= N_o + G J_h^{-T} N_f \\
F &= G J_h^{-T} \tau.
\end{aligned} \tag{6.22}$$

These equations have the same form as the equations for a single open-chain manipulator. $\tilde{M}$ is called the the effective mass of the object, $\tilde{C}$ the effective Coriolis matrix, and $\tilde{N}$ the effective gravitational and nonconservative forces. These quantities include the dynamics of the fingers; however, the details of the finger kinematics and dynamics are effectively hidden in $\tilde{M}$, $\tilde{C}$, $\tilde{N}$. The following lemma verifies that these equations also satisfy the same structural properties as the unconstrained case.

Lemma 6.1. Structural properties of the equations of motion *Equation* (6.21) *satisfies the following properties:*

1. $\tilde{M}(q)$ *is symmetric and positive definite.*
2. $\dot{\tilde{M}}(q) - 2\tilde{C}$ *is a skew-symmetric matrix.*

Proof. Since the grasp is assumed to be force-closure and manipulable and J_h is assumed injective, property 1 follows from its definition. To show property 2,

$$\dot{\tilde{M}} - 2\tilde{C} = (\dot{M}_o - 2C_o) + GJ^{-T}(\dot{M}_f - 2C_f)J^{-1}G^T + \frac{d}{dt}\left(GJ^{-T}\right)M_f J^{-1}G^T - GJ^{-T}M_f\frac{d}{dt}\left(J^{-1}G^T\right)$$

The first line is the sum of skew-symmetric pieces. Taking transposes and using symmetry of M_f inverts the sign of the last line, and hence it too is skew-symmetric. □

Equation (6.21) was derived in terms of a local parameterization of $SE(3)$. This was necessary since the Lagrange-d'Alembert equations only make sense for $q \in \mathbb{R}^n$. Since $SE(3)$ is not locally Euclidean, we cannot apply the Lagrange-d'Alembert equations directly in terms of $x_o \in SE(3)$. However, having derived the equations of motion in local coordinates, we can now revert to $SE(3)$ and write the dynamics as

$$\tilde{M}(q)\dot{V}_o^b + \tilde{C}(q,\dot{q})V_o^b + \tilde{N}(q,\dot{q}) = F^b, \tag{6.23}$$

where $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ are described precisely as before except using the global versions of the inertial parameters. The proof of this fact is left as an exercise. We make use of the local parameterization of the system for the remainder of this section since it simplifies the exposition.

2.2 Internal forces

In Chapter 5, we saw that if a grasp is force-closure, then there exist contact forces which produce no net wrench on the object. These forces are called *internal forces.* Their presence in the dynamics is seen in the

forcing term $F = GJ_h^{-T}\tau$, which maps joint torques into object forces. If $J_h^{-T}\tau \in \mathcal{N}(G)$, then no net wrench is generated.

However, even if $J_h^{-T}\tau$ has no component in $\mathcal{N}(G)$, internal forces may still be present in the system. These internal forces can arise due to the constraint forces which the Lagrange-d'Alembert equations eliminated. Defining $q = (\theta, x)$, the full equations of motion can be written as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda - \Upsilon = 0,$$

where $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers and the columns of $A^T(q)$ correspond to the constraint directions. For a multifingered grasp we can choose

$$A(q) = \begin{bmatrix} -J_h(\theta, x) & G^T(\theta, x) \end{bmatrix}$$

and the full equations of motion have the form

$$\begin{bmatrix} M_f & 0 \\ 0 & M_o \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} C_f & 0 \\ 0 & C_o \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} N_f \\ N_o \end{bmatrix} + \begin{bmatrix} -J_h^T \\ G \end{bmatrix} \lambda = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \tag{6.24}$$

We see immediately from equation (6.24) that the Lagrange multipliers $\lambda \in \mathbb{R}^m$ can be interpreted as contact forces. The net constraint forces act as external forces applied at the tip of the fingers and at the contact points on the object. If λ has a component in the null space of G, then internal forces will be present.

To solve for the instantaneous forces during motion, we solve for the Lagrange multipliers using the results of Section 1.2. Letting $\bar{M}$, $\bar{C}$, and $\bar{N}$ represent the block matrices in equation (6.24), we have

$$\lambda = (A\bar{M}^{-1}A^T)^{-1}\left(A\bar{M}^{-1}\left(\begin{bmatrix} \tau \\ 0 \end{bmatrix} - \bar{C}\dot{q} - \bar{N}\right) + \dot{A}\dot{q}\right). \tag{6.25}$$

We can compute the internal forces as a function of the current configuration, velocity, and applied torques by projecting λ onto the null space of G. The computation of λ and its projection is evidently extremely messy. In practice, the internal forces generated by non-quasistatic motion are either ignored or measured directly (via force sensors at the contacts).

An alternative way to find the constraint forces is to solve for them in terms of the joint accelerations. If J_h is invertible, the contact forces satisfy

$$\lambda = J_h^{-T}\left(\tau - M_f\ddot{\theta} - C_f\dot{\theta} - N_f\right).$$

From this equation it is clear that if the system is in motion, then internal forces may exist even if $J_h^{-T}\tau$ has no component in the null space of G: the dynamic terms may generate internal forces. Notice also that $\lambda = f_c := J_h^{-T}\tau$ only when the system is at rest.

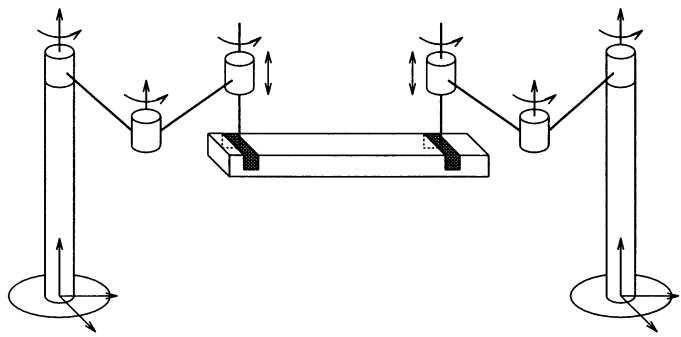


Figure 6.3: Coordinated lifting.

All of the analysis presented here relies on the assumption that the fundamental grasping constraint holds at all times. Even though we are solving for the contact forces, it is important to remember that the contact forces are generated by the constraints. If the contact forces fall outside of the friction cone, then the velocity constraints on the system no longer hold and the equations of motion are no longer given by equation (6.24). It is one of the tasks of control system to keep the contact forces inside the friction cone so that the models we use here remain valid.

2.3 Other robot systems

Although the analysis so far has been motivated by grasping, the basic formulation holds for any robot system subject to constraints of the form

$$J(q)\dot{\theta} = G^T(q)\dot{x}$$

with $q = (\theta, x)$. Any such system will automatically have dynamics with the same form and structure as those of an open-chain manipulator. In this section, we describe several such examples.

Coordinated lifting

Consider a robotic system which consists of several individual robots lifting a single object, such as the example shown in Figure 6.3. Assume that each robot firmly grasps the object and can exert arbitrary wrenches without losing contact.

To derive the kinematics of this system we can treat each robot as a single finger. The contact model for a robot firmly grasping an object is given by $B_{c_i} = I \in \mathbb{R}^{p \times p}$ and $FC = \mathbb{R}^p$. This is just the mathematical statement that the robot can exert arbitrary forces and torques. Substituting these relations into the grasp constraint and identifying the

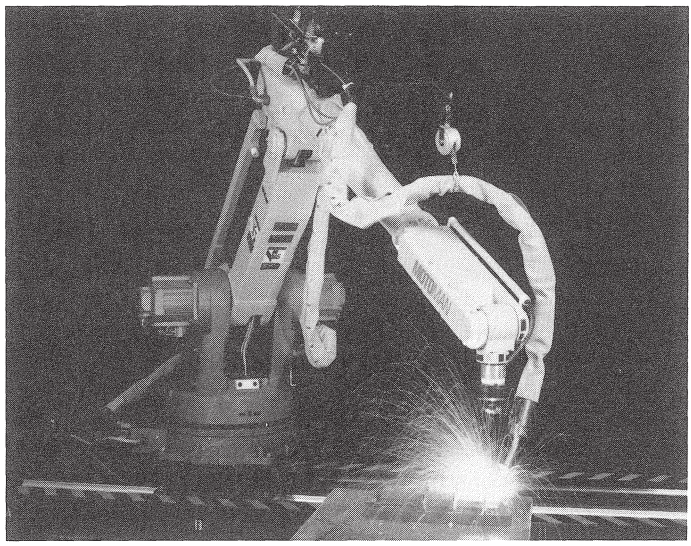


Figure 6.4: The Motoman K10MSB robot performing an arc welding task. (Photograph courtesy of Motoman)

contact and tool frames, we have

$$\underbrace{\begin{bmatrix} \mathrm{Ad}^{-1}_{g_{s_1t_1}} J^s_{s_1t_1} & & 0 \\ & \ddots & \\ 0 & & \mathrm{Ad}^{-1}_{g_{s_kt_k}} J^s_{s_kt_k} \end{bmatrix}}_{J} \dot{\theta} = \underbrace{\begin{bmatrix} \mathrm{Ad}^{-1}_{g_{ot_1}} \\ \vdots \\ \mathrm{Ad}^{-1}_{g_{ot_k}} \end{bmatrix}}_{G^T} V^b_{po}.$$

The dynamics of the system follow exactly as in the grasping case.

Workspace dynamics

Many robot systems perform tasks which are most naturally described in workspace coordinates rather than joint space coordinates. For example, in the welding application depicted in Figure 6.4, a natural description of the task would be in terms of the position and orientation of the welding tip. Rather than solve an inverse kinematics problem to generate the corresponding path in joint space, it is possible to directly specify the dynamics in terms of the workspace coordinates ($SE(3)$ for this example).

For simplicity, we work in local coordinates. Let $g : Q \to \mathbb{R}^p$ represent the forward kinematics of the manipulator and $J(\theta) = \frac{\partial g}{\partial \theta}$ the Jacobian. We assume that J is square and nonsingular. We take dynamics of the

object held by the robot (e.g., the arc welder) as

$$M_o(x)\ddot{x} + C_o(x,\dot{x})\dot{x} + N_o(x,\dot{x}) = 0,$$

where $x \in \mathbb{R}^p$ represents the workspace coordinates and the usual structural properties are satisfied by M_o and C_o.

The kinematics of the mechanism is given by

$$J(\theta)\dot{\theta} = \dot{x},$$

which has the form of our canonical constraint with $G = I \in \mathbb{R}^{p\times p}$. Thus, we can write the dynamics as

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q,\dot{q})\dot{x} + \tilde{N}(q,\dot{q}) = F, \tag{6.26}$$

where $q = (\theta, x)$ and

$$\begin{aligned}
\tilde{M} &= M_o + J^{-T}M_f J^{-1} \\
\tilde{C} &= C_o + J^{-T}\left(C_f J^{-1} + M_f \frac{d}{dt}\left(J^{-1}\right)\right) \\
\tilde{N} &= N_o + J^{-T}N_f \\
F &= J^{-T}\tau
\end{aligned}$$

(quantities with the subscript f refer to the robot dynamics). It follows immediately that $\tilde{M} > 0$ and $\dot{\tilde{M}} - 2\tilde{C}$ is skew-symmetric.

The dynamics given in equation (6.26) represent the equations of motion relative to the workspace coordinates $x \in \mathbb{R}^p$. Thus, $\tilde{M}(q)$ represents the inertia of the system as viewed from the object frame of reference. As in the grasping case, $\tilde{M}(q)$ incorporates both the object inertia and the inertia of the robot (at its current configuration). If the robot approaches a singular configuration, the inertia matrix becomes unbounded. This is because large workspace forces produce small object accelerations, and hence the effective inertia appears very large. However, this singularity is strictly due to the parameterization of the dynamics in terms of the workspace coordinates. The dynamics of the mechanism in joint coordinates are never singular.

There are several variations on this problem. The dynamics can be written in terms of $x_o \in SE(3)$ by replacing x with x_o, $\dot{x}$ with V_o^b, and $\ddot{x}$ with $\dot{V}_o^b$ (see the comments at the end of Section 2.1). We can also write the dynamics even if no object is present, by setting M_o, C_o, and N_o to zero. This is useful if we simply wish to command the trajectory of a robot in end-effector coordinates. Finally, we can in certain cases relax the assumption that $J(\theta)$ be square. This is discussed in more detail in Section 3.

The primary difference between this class of examples and grasping is the lack of any "internal forces" (since $G = I$ never has a null space).

Hybrid position/force dynamics

Another common manipulation task is one which consists of moving the robot in certain directions while pushing in other directions. An elementary example is writing on a chalkboard: the task specification consists of a desired motion in the plane of the chalkboard and a desired force against the chalkboard. A preliminary discussion of this topic is contained in Chapter 4. We now use the tools developed in this chapter to describe this situation more completely.

To analyze the kinematics of this system, we assume that the end-effector is required to satisfy a holonomic constraint of the form

$$h(\theta, x) = 0, \tag{6.27}$$

where $x \in \mathbb{R}^p$ parameterizes the allowable motions of the manipulator. For example, when writing on a chalkboard, x might specify the location of the chalk on the board as well as its orientation (in some suitable set of local coordinates). More generally, $h(\theta, x)$ specifies a p-dimensional surface in the configuration space of the manipulator. The task description consists of motion along this surface and generalized forces against this surface.

Equation (6.27) can be converted into the standard problem by differentiating with respect to time:

$$\underbrace{\frac{\partial h}{\partial \theta}}_{J} \dot{\theta} = \underbrace{-\frac{\partial h}{\partial x}}_{G^T} \dot{x} \tag{6.28}$$

As before, we assume that J is square and nonsingular, indicating that no internal motions are present and that the manipulator is not at a singular configuration relative to the task. Internal forces correspond to joint torques τ such that $GJ^{-T}\tau = 0$. These are precisely the torques which generate forces against the constraint surface.

To formulate the dynamics of the mechanism, we assume that the object held by the robot is accounted for in the robot dynamics, and hence

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q, \dot{q})\dot{x} + \tilde{N}(q, \dot{q}) = F,$$

where

$$\begin{aligned}
\tilde{M} &= GJ^{-T} M_f J^{-1} G^T \\
\tilde{C} &= GJ^{-T} \left(C_f J^{-1} G^T + M_f \frac{d}{dt} \left(J^{-1} G^T \right) \right) \\
\tilde{N} &= N_o + GJ^{-T} N_f \\
F &= GJ^{-T} \tau.
\end{aligned}$$

The reason for combining the object dynamics with those of the robot is that $x \in \mathbb{R}^p$ may not actually correspond to the configuration of an

object in many applications. For instance, in the chalkboard example, only two of the linear velocities are specified by $\dot{x}$. If the chalkboard were curved instead of flat, correctly specifying the object dynamics in terms of x becomes much more involved.

As with grasping, internal forces do not affect the equations of motion and hence they can be ignored if only the free motion is to be studied. If internal forces are to be controlled or regulated, they can be found by solving for the Lagrange multipliers.

Several variations of this problem are possible. The use of local coordinates for motion which is constrained to a subgroup of $SE(3)$ can be relaxed by appropriate interpretation of velocities and accelerations. In addition, the specifications of the task need not be in the form of a holonomic constraint. For some problems, it may be more natural to specify the kinematics directly in terms of J and G.

3 Redundant and Nonmanipulable Robot Systems

In order to perform a given task, a robot must have enough degrees of freedom to accomplish that task. In the analysis presented so far, we assumed that our robots had exactly the number of degrees of freedom required to complete the task. That is to say, we assumed that each finger could follow the object through any allowable trajectory, but that the fingers had only the number of degrees of freedom required by the contact type (i.e., one degree of freedom for a frictionless point contact, three for a point contact with friction, and four for a soft-finger contact). This assumption manifested itself in our requirement that J_h be an invertible square matrix when we derived the dynamics. We now relax that requirement and discuss its consequences.

There are two situations in which J_h can fail to be square and invertible:

1. *The manipulator has too many degrees of freedom.* In this case, J_h will have two or more columns which are linearly dependent, allowing internal motions which leave the contact locations fixed.

2. *The manipulator has too few degrees of freedom.* If J_h is not full row rank, it is not possible for the fingers to follow arbitrary motions of the contact points. This potentially limits the motion of the object, though it is possible for this situation to occur even if the grasp is manipulable.

These cases are not distinct; it is possible for a manipulator to have both internal motions and fail to be manipulable at the same time. In any case, we seek to cast the problem into the general framework developed in the

previous section by augmenting or decreasing the number of variables used to describe the task.

As before, we note that the material contained herein applies not only to multifingered hands, but to many other constrained systems as well. A few of these variations are described in the exercises.

3.1 Dynamics of redundant manipulators

Unlike conventional robot manipulators, constrained robot manipulators do not need to have more than six degrees of freedom in order to be redundant. The constraints themselves can introduce kinematic redundancy into a system. For example, if we attach a six degree of freedom finger to an object using a soft-finger contact, we have introduced two redundant degrees of freedom: the finger is free to roll in either of two directions without affecting the position of the object. Thus, it is absolutely essential that we include redundant mechanisms in our formulation.

It is interesting to note that there are actually two sources of redundancy introduced by our constraints: kinematic redundancy and actuator redundancy. *Kinematic redundancy* refers to *motions* of the fingers which do not affect the motion of the object. *Actuator redundancy* refers to *forces* applied by the fingers which do not affect the object motion, i.e., internal forces. The grasping constraint

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}$$

contains all the information necessary to determine these redundancies. Namely, the null space of J_h describes the set of joint motions which do not affect the motion of the object and the null space of G is precisely the space of internal forces. Since we have already discussed internal forces, we restrict our discussion to kinematic redundancy.

Consider first the kinematics of a single redundant manipulator, with no constraints. If we are willing to control the manipulator in joint space, the dynamics formulation presented above holds without modification. However, in order to perform a task specified in terms of the configuration of the end-effector, we must first choose a joint trajectory which accomplishes this task. Suppose instead that we wish to write our controllers directly in end-effector coordinates. We represent the kinematics as a function $g : \mathbb{R}^n \to SE(3)$. In this case, the manipulator Jacobian $J(\theta) := J^s_{st}(\theta) \in \mathbb{R}^{p \times n}$ is not square, so J^{-1} is not well defined and we cannot write $\dot{\theta}$ in terms of V^s_{st} as we did previously.

It is still possible to write the dynamics of redundant manipulators in a form consistent with that derived earlier. To do so, we define a matrix $K(\theta) \in \mathbb{R}^{(n-p)\times n}$ whose rows span the null space of $J(\theta)$. As before, we assume that $J(\theta)$ is full row rank and hence $K(\theta)$ has constant rank $n - p$. The rows of $K(\theta)$ are basis elements for the space of velocities

which cause no motion of the end-effector; we can thus define an *internal motion*, $v_N \in \mathbb{R}^{n-p}$, using the equation

$$\begin{bmatrix} \dot{x} \\ v_N \end{bmatrix} = \begin{bmatrix} J \\ K \end{bmatrix} \dot{\theta} =: \bar{J}\dot{\theta}.$$

By definition, $\bar{J}$ is invertible and it follows from our previous derivation that

$$\tilde{M}(q) \begin{bmatrix} \ddot{x} \\ \dot{v}_N \end{bmatrix} + \tilde{C}(q,\dot{q}) \begin{bmatrix} \dot{x} \\ v_N \end{bmatrix} + \tilde{N}(q,\dot{q}) = \bar{J}^{-T}\tau,$$

where $\tilde{M}$ and $\tilde{C}$ are obtained as in the nonredundant case, but replacing J with $\bar{J}$ and G with I:

$$\begin{aligned} \tilde{M} &= \bar{J}^{-T} M \bar{J}^{-1} \\ \tilde{C} &= \bar{J}^{-T}\left(C\bar{J}^{-1} + M\frac{d}{dt}(\bar{J}^{-1})\right) \\ \tilde{N} &= N_o + \bar{J}^{-T} N_f. \end{aligned}$$

If we choose K such that its rows are orthonormal and also perpendicular to the rows of J, then $\bar{J}^{-1} = \begin{bmatrix} J^+ & K^T \end{bmatrix}$ where $J^+ = J^T(JJ^T)^{-1}$ is the least-squares right (pseudo-) inverse of J.

Note that we have parameterized the internal motion of the system by a *velocity* and not a new variable y. We do this out of necessity: since we chose K only to span the null space of J, there may not exist a function h such that $y = h(\theta)$ and $\frac{\partial h}{\partial \theta} = K$. A necessary and sufficient condition for such an h to exist is that each row of K satisfy $\frac{\partial K_{ij}}{\partial \theta_k} = \frac{\partial K_{ik}}{\partial \theta_j}$. This is merely the statement that mixed partials of h must commute. A more thorough treatment of this point is given in Chapter 7, and is illustrated briefly in the next example.

In general, it may not always be easy to choose $K(\theta)$ such that it is the differential of some function. For this reason, we shall not generally assume that an explicit coordinatization of the internal motion manifold is available. Thus, in the same way as we were forced to use velocity relationships when modeling constraints, we also use velocity relationships for redundant manipulators. The Lagrange-d'Alembert formalism lets us treat this case without difficulty.

Example 6.3. Three-link planar manipulator
Consider a three-link planar manipulator with unit-length links, as shown in Figure 6.5. If we let (x, y) be the location of the end-effector, then

$$\begin{aligned} x &= \cos\theta_1 + \cos\theta_2 + \cos\theta_3 \\ y &= \sin\theta_1 + \sin\theta_2 + \sin\theta_3, \end{aligned}$$

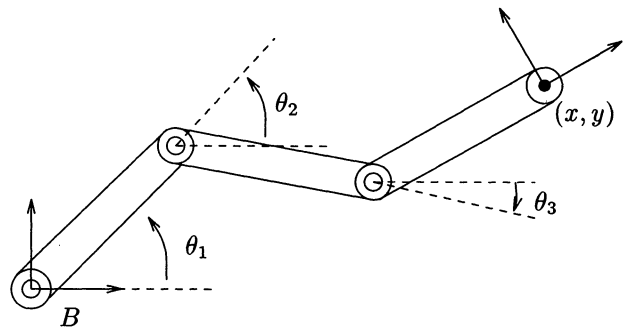


Figure 6.5: Three-link planar manipulator with joint angles measured relative to the horizontal axis.

where the link angles are all with respect to a fixed (inertial) axis. The Jacobian of the mapping $\theta \mapsto (x, y)$ is

$$J(\theta) = \begin{bmatrix} -\sin\theta_1 & -\sin\theta_2 & -\sin\theta_3 \\ \cos\theta_1 & \cos\theta_2 & \cos\theta_3 \end{bmatrix}.$$

There are many choices of $K(\theta)$ to complete $J(\theta)$. If we choose

$$K(\theta) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

then $K = \frac{\partial h}{\partial \theta}$ with $h(\theta) = \theta_3$. This corresponds to choosing the angle of the end-effector to parametrize internal motions. This choice of $K(\theta)$ is valid as long as $\theta_1 \neq \theta_2$ (i.e., when the first two links are not aligned). If, on the other hand, we choose

$$K(\theta) = \begin{bmatrix} \sin(\theta_2 - \theta_3) & -\sin(\theta_1 - \theta_3) & \sin(\theta_1 - \theta_2) \end{bmatrix},$$

which is valid as long as all three links are not aligned, then

$$\frac{\partial K_1(\theta)}{\partial \theta_2} = \cos(\theta_2 - \theta_3) \neq -\cos(\theta_1 - \theta_3) = \frac{\partial K_2(\theta)}{\partial \theta_1}.$$

Hence, there is no h such that $\frac{\partial h}{\partial \theta} = K$ and the velocity $v_N = K(\theta)\dot{\theta}$ is not the derivative of a variable y.

We now derive the equations of motion for the system in terms of x, y, and v_N. Let $M(\theta)$ be the inertia matrix for the manipulator in joint space with $C(\theta, \dot{\theta})$, the corresponding Coriolis matrix. For brevity, we ignore the potential and nonconservative forces. The dynamics of the mechanism in end-effector coordinates are given by

$$(\bar{J}^{-T} M \bar{J}^{-1}) \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \dot{v}_N \end{bmatrix} + \left(\bar{J}^{-T} C \bar{J}^{-1} + \bar{J}^{-T} M \frac{d}{dt}(\bar{J}^{-1}) \right) \begin{bmatrix} \dot{x} \\ \dot{y} \\ v_N \end{bmatrix} = \bar{J}^{-T} \tau \tag{6.29}$$

where

$$\bar{J} = \begin{bmatrix} J(\theta) \\ K(\theta) \end{bmatrix} = \begin{bmatrix} -\sin\theta_1 & -\sin\theta_2 & -\sin\theta_3 \\ \cos\theta_1 & \cos\theta_2 & \cos\theta_3 \\ \sin(\theta_2 - \theta_3) & \sin(\theta_3 - \theta_2) & \sin(\theta_1 - \theta_2) \end{bmatrix}.$$

We now return to the general case and extend our treatment to include the full grasp constraints. Consider a force-closure and manipulable grasp with velocity constraints

$$J_h\dot{\theta} = G^T\dot{x},$$

where $\mathcal{N}(J_h) \neq 0$. As before, we augment the constraint by choosing a matrix $K_h(\theta)$ whose rows span the null space of $J_h(\theta)$. The grasp constraint can now be written as

$$\underbrace{\begin{bmatrix} J_h \\ K_h \end{bmatrix}}_{\bar{J}_h}\dot{\theta} = \underbrace{\begin{bmatrix} G^T & 0 \\ 0 & I \end{bmatrix}}_{\bar{G}^T}\begin{bmatrix} \dot{x} \\ v_N \end{bmatrix},$$

where $\bar{J}_h$ and $\bar{G}$ represent the augmented hand Jacobian and grasp matrix. This constraint has the same form as the standard (nonredundant) grasp constraint and $\bar{J}_h$ is now invertible by construction. Hence, we can write the dynamics as

$$\tilde{M}(q)\begin{bmatrix} \ddot{x} \\ \dot{v}_N \end{bmatrix} + \tilde{C}(q,\dot{q})\begin{bmatrix} \dot{x} \\ v_N \end{bmatrix} = \bar{G}\bar{J}_h^{-T}\tau, \tag{6.30}$$

where $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ are

$$\tilde{M} = \bar{G}\bar{J}_h^{-T}M\bar{J}_h^{-1}\bar{G}^T$$
$$\tilde{C} = \bar{G}\bar{J}_h^{-T}\left(C\bar{J}_h^{-1}\bar{G}^T + M\frac{d}{dt}\left(\bar{J}_h^{-1}\bar{G}^T\right)\right).$$

We now see that redundant manipulators can be incorporated into the same general framework as other robot systems. The necessity of augmenting the description of the system stems from our use of the task variables, parameterized by x, to specify the motion of the system. Since the mechanism is redundant, the x variables alone do not provide sufficient information to determine the motion of the system. Augmenting the task description with the variables v_N gives a complete description of the motion of the system.

One final comment is in order regarding the relationship between the joint torques and the object wrench for a redundant grasp. In Chapter 5, we derived the static relationships between the joint torques, the contact forces, and the object wrench. These relationships were used to determine how to push on an object, via the fingers, in order to resist applied forces.

In the redundant case, a bit of care must be taken in interpreting these results.

Consider the general grasping situation described above. Reverting to twists, the kinematic constraints have the form

$$\begin{bmatrix} J_h \\ K_h \end{bmatrix} \dot{\theta} = \begin{bmatrix} G^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V_{po}^b \\ v_N \end{bmatrix},$$

where $V_{po}^b \in \mathbb{R}^p$ is the object's body velocity and $v_N \in \mathbb{R}^{n-p}$ is the internal velocity. The associated quasistatic forces satisfy

$$\tau = \begin{bmatrix} J_h^T & K_h^T \end{bmatrix} \begin{bmatrix} f_c \\ f_N \end{bmatrix} \qquad F_{po}^b = G f_c. \tag{6.31}$$

The forces f_N parameterize the forces which correspond to internal motions v_N. If these forces are chosen to be zero, then we retrieve the usual force relationships between joint torques and object wrenches.

If the forces f_N are not chosen to be zero, then the manipulator will begin to exhibit internal motions. This motion can cause accelerations of the manipulator joints and we can no longer use equation (6.31) to represent the force relationships in the system. Instead, we must consider the full equations of motion as given in equation (6.30). In particular, the internal motions of the system may generate constraint forces and hence the relationships in equation (6.31) are no longer correct. This situation does not occur in nonredundant systems since if we keep the end-effector fixed, then all joint angles also remain fixed and hence no dynamic terms are present.

3.2 Nonmanipulable grasps

We now consider the situation in which the hand Jacobian is not full row rank. In this case, there are some motions of the individual contacts which cannot be tracked by the fingers. We assume that the hand Jacobian is full column rank. If not, the methods of the preceding subsection can be used to augment the grasp with appropriate internal velocities.

In most situations, if the hand Jacobian is not full row rank, then the grasp fails to be manipulable. However, in certain special situations, it is possible that a multifingered grasp is both manipulable and nonredundant but J_h is not square. This can occur when the structure of the grasp is such that J_h is bijective onto the range of G^T but is not surjective as a map from $\mathbb{R}^n \to \mathbb{R}^m$. This situation almost never occurs in practice, and hence we concentrate here only on the case where J_h is nonmanipulable.

To treat the nonmanipulable case, we must restrict the motions of the object to those which can be accommodated by the fingers. This restriction is enforced by structural forces within the hand, which resist motion of the system in directions in which the fingers are unable to move.

As usual, our formulation assumes that the contacts are maintained and hence the contact forces must remain inside the friction cone at all times. It is the responsibility of the control law to insure that this condition holds at all times.

Consider a nonmanipulable grasp with grasp constraint

$$J_h\dot{\theta} = G^T\dot{x}.$$

Let $W(\theta, x)$ represent the space of allowable object velocities,

$$W(\theta, x) = \{\dot{x} \in \mathbb{R}^p : \exists\ \dot{\theta} \in \mathbb{R}^m \text{ with } J_h\dot{\theta} = G^T\dot{x}\}.$$

We assume that $W(\theta, x)$ has constant dimension $l > 0$ and that W varies smoothly as a function of its arguments. Choosing a matrix $H(\theta, x) \in \mathbb{R}^{p\times l}$ whose columns span $W(\theta, x)$, we can write the grasp constraints as

$$\begin{aligned} J_h\dot{\theta} &= G^T H w \\ \dot{x} &= Hw, \end{aligned} \tag{6.32}$$

where $w \in \mathbb{R}^l$ represents the object velocity in terms of the basis formed by the columns of H.

To formulate the equations of motion, we write the dynamics in terms of the velocities $w \in \mathbb{R}^l$. By construction, J_h is surjective onto the range of $\bar{G}^T := G^T H$ and hence we can solve for $\dot{\theta}$ given w. However, J_h is not necessarily square so we must use the pseudo-inverse $J_h^+ = (J_h^T J_h)^{-1} J_h^T$ in place of J_h^{-1}. The resulting dynamics are given by

$$\tilde{M}(q)\dot{w} + \tilde{C}(q, \dot{q})w + \tilde{N}(q, \dot{q}) = F, \tag{6.33}$$

where $q = (\theta, x)$,

$$\begin{aligned} \tilde{M} &= M_o + \bar{G}J_h^{+T} M_f J_h^+ \bar{G}^T \\ \tilde{C} &= C_o + \bar{G}J_h^{+T}\left(C_f J_h^+ \bar{G}^T + M_f \frac{d}{dt}\left(J_h^+ \bar{G}^T\right)\right) \\ \tilde{N} &= N_o + \bar{G}J_h^{+T} N_f \\ F &= G\bar{J}_h^{+T}\tau, \end{aligned}$$

and J_h^{+T} is the transpose of J_h^+. The second-order dynamics in equation (6.33) combined with the first-order constraints in equation (6.32) give a complete description of the motion of the system.

3.3 Example: Two-fingered SCARA grasp

To illustrate the results of this section, we consider the dynamics of two SCARA fingers grasping a box, as shown in Figure 6.6. Each finger is

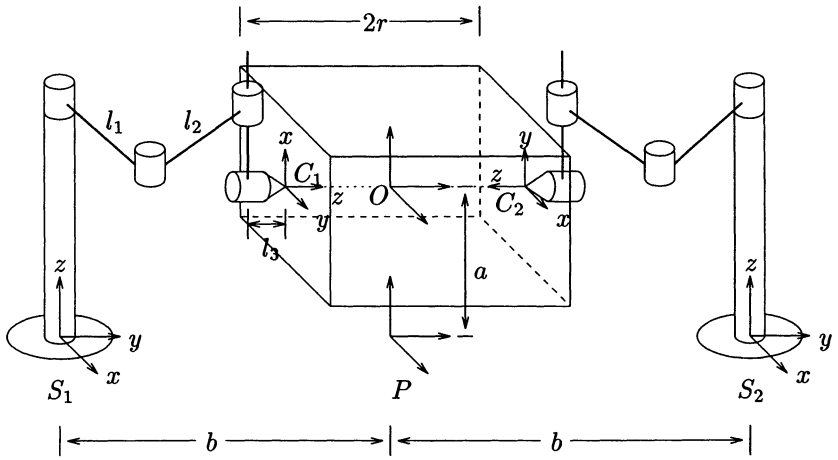


Figure 6.6: Two-fingered grasp using SCARA robots.

modeled as a soft-finger finger contact. The fundamental grasp constraint for the system has the general form

$$8\left\{\begin{bmatrix} J_{h1} & 0 \\ 0 & J_{h2} \end{bmatrix}\dot{\theta} = \begin{bmatrix} G_1^T \\ G_2^T \end{bmatrix} V_{po}^b.\right.$$

$$\underbrace{\qquad\qquad}_{8} \qquad \underbrace{\qquad}_{8}$$

Although $J_h(\theta) \in \mathbb{R}^{8\times 8}$ is a square matrix, it is not invertible. It was shown in Section 5.3 of Chapter 5 that the grasp is not manipulable and also contains internal motions. The lack of manipulability comes from the fact that rotations about the line connecting the contacts violate the soft-finger contact constraints. The internal motions correspond to motions of the first three joints of each finger which leave the xy positions of the fingertips fixed.

To parameterize the internal motion of the system, we augment the system using the angles of the last joint of the fingers, as in Example 6.3. Letting $y = (\theta_{11} + \theta_{12} + \theta_{13}, \theta_{21} + \theta_{22} + \theta_{23})$, the contact constraints become

$$10\left\{\left[\begin{array}{c|c} J_{h1} & 0 \\ 0 & J_{h2} \\ \hline \begin{matrix}1&1&1&0\\0&0&0&0\end{matrix} & \begin{matrix}0&0&0&0\\1&1&1&0\end{matrix} \end{array}\right]\dot{\theta} = \left[\begin{array}{c|c} G_1^T & 0 \\ G_2^T & 0 \\ \hline 0 & I \end{array}\right]\begin{bmatrix} V_{po}^b \\ \dot{y} \end{bmatrix}.\right.$$

$$\underbrace{\qquad\qquad}_{8} \qquad \underbrace{\qquad}_{8}$$

Note that for this example we were able to choose actual variables to parameterize the internal motions and not just velocities.

The addition of the internal variables does not alter the nonmanipulable nature of the grasp since J_h still does not span the range of G^T. Because rotations of the object about the line connecting the contacts are the source of the difficulty, we eliminate these directions from the allowable velocities of the system. Choosing

$$H = \left[\begin{array}{ccccc|cc} 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ \hline 0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1 \end{array}\right] = \left[\begin{array}{c|c} H' & 0 \\ \hline 0 & I \end{array}\right]$$

the resulting contact becomes

$$10 \left[\begin{array}{c|c} J_{h1} & 0 \\ \hline 0 & J_{h2} \\ \hline K_1 & K_2 \end{array}\right] \dot{\theta} = \left[\begin{array}{c|c} G_1^T H' & 0 \\ \hline G_2^T H' & 0 \\ \hline 0 & I \end{array}\right] \begin{bmatrix} w' \\ \\ \dot{y} \end{bmatrix},$$

$\leftarrow 8 \rightarrow$ $\leftarrow 7 \rightarrow$

where $V_{po}^b = H'w'$ is in the set of allowable object velocities. A detailed calculation verifies that this constraint is manipulable and that no internal motions exist. We can now solve for the dynamics of the system in terms of the workspace variables $w = (w', y) \in \mathbb{R}^7$, keeping in mind that the pseudo-inverse of $\bar{J}_h$ must be used since $\bar{J}_h$ is surjective onto the range of $\bar{G}^T$ but not square.

4 Kinematics and Statics of Tendon Actuation

In many robot applications, it is difficult to control the torques on the joints directly, due to the size of the actuators required to exert reasonably sized forces. A more practical approach is to use a transmission network to carry forces from an actuator to the appropriate joint. Such a network typically consists of some combination of linkages, tendons, gears, and pulleys.

In this section, we consider one of the more common transmission systems, a network of tendons. Tendons offer advantages in terms of weight and flexibility; however, they can complicate the kinematics of the mechanism. The basic problem which we study is to describe how forces applied at the end of a tendon relate to the joint torques applied to the mechanism. We also examine a second, somewhat less-common situation in which elastic tendons are driven by position-controlled actuators, such as a stepper motor.

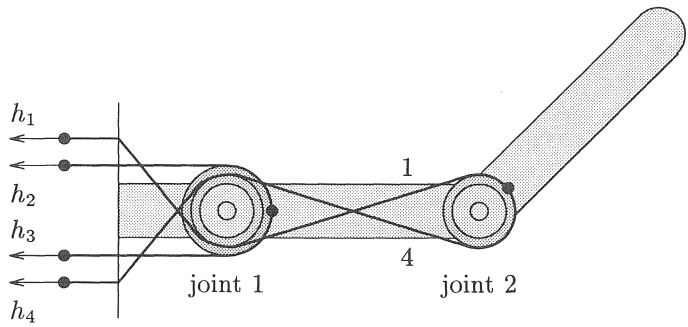


Figure 6.7: A tendon-driven finger.

4.1 Inelastic tendons

Consider a finger which is actuated by a set of inelastic tendons, such as the one shown in Figure 6.7. Each tendon consists of an inextensible cable connected to a force generator, such as a DC motor. For simplicity, we assume that each tendon/actuator pair is connected either between the base of the hand and a link of the finger, or between two links of the finger. Interconnections between tendons are not allowed. We wish to describe how forces applied at the end of the tendons are related to the torques applied at the joints.

Note that even though each tendon can be connected to only one link, pulling on a tendon may generate forces on many joints. This occurs because as we pull on a tendon, it exerts forces all along its length against whatever parts of the mechanism are holding it in place. This coupling is difficult to eliminate without awkward routing of the tendons.

We model the routing of each tendon by an *extension function*, $h_i : Q \to \mathbb{R}$. The extension function measures the displacement of the end of the tendon as a function of the joint angles of the finger. For simple tendon networks composed of pulleys, such as those shown in Figure 6.7, the tendon extension is a linear function of the joint angles

$$h_i(\theta) = l_i \pm r_{i1}\theta_1 \pm \cdots \pm r_{in}\theta_n,$$

where l_i is the nominal extension (at $\theta = 0$) and r_{ij} is the radius of the pulley at the j^{th} joint. The sign depends on whether the tendon path gets longer or shorter when the angle is changed in a positive sense.

More complicated tendon geometries may involve nonlinear functions of the joint angles. For example, for the joint pictured in Figure 6.8, the top tendon has an extension function of the form

$$h_1(\theta) = l_1 + 2\sqrt{a^2+b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta}{2}\right) - 2b \qquad \theta > 0,$$

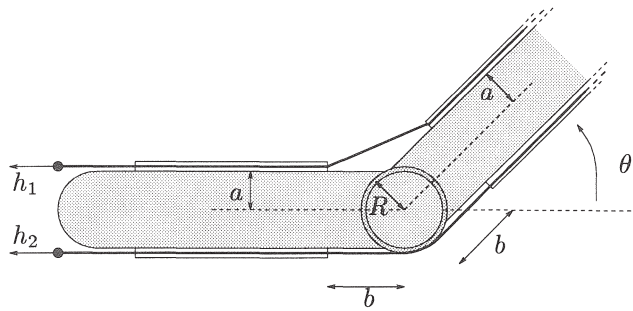


Figure 6.8: Example of tendon routing with nonlinear extension function.

while the bottom tendon satisfies

$$h_2(\theta) = l_2 + R\theta, \qquad \theta > 0.$$

When $\theta < 0$, these relations are reversed.

Once the tendon extension functions have been computed, we can determine the relationships between the tendon forces and the joint torques by applying conservation of energy. Let $e = h(\theta) \in \mathbb{R}^p$ represent the vector of tendon extensions for a system with p tendons and define the matrix $P(\theta) \in \mathbb{R}^{n \times p}$ as

$$P(\theta) = \frac{\partial h}{\partial \theta}^T(\theta).$$

Then,

$$\dot{e} = \frac{\partial h}{\partial \theta}(\theta)\dot{\theta} = P^T(\theta)\dot{\theta}.$$

Since the work done by the tendons must equal that done by the fingers (in the absence of friction or other losses), we can use conservation of energy to conclude

$$\tau = P(\theta)f,$$

where $f \in \mathbb{R}^p$ is the vector of forces applied to the ends of the tendons. The matrix $P(\theta)$ is called the *coupling matrix* and plays a role similar to that of the grasp map defined in Chapter 5.

The kinematics of the tendon network can be combined with the dynamics of the mechanism to yield

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = P(\theta)f.$$

The structure of this equation relies on the assumption that the actuator and tendon dynamics can be ignored, and hence a force applied at the end of the tendon is immediately transmitted to the joints through the coupling matrix.

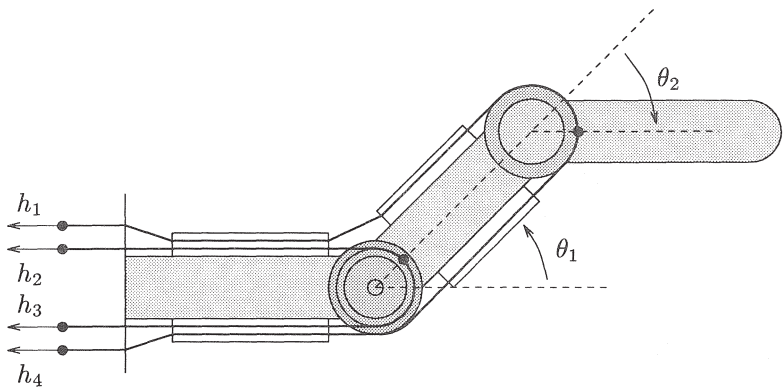


Figure 6.9: Planar tendon-driven finger.

Example 6.4. Two-link tendon-driven finger
As an example, we consider the planar finger shown in Figure 6.9. It consists of two revolute joints driven by four tendons. The tendons are routed through sheaths attached to the sides of the links.

The extension functions for the tendon network are calculated by adding the contribution from each joint. The two tendons attached to the first joint are routed across a pulley of radius R_1, and hence

$$h_2 = l_2 - R_1\theta_1$$
$$h_3 = l_3 + R_1\theta_1.$$

The tendons for the outer link have more complicated kinematics due to the routing through the tendon sheaths. Their extension functions are

$$h_1 = l_1 + 2\sqrt{a^2+b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) - 2b - R_2\theta_2 \qquad \theta_1 > 0.$$
$$h_4 = l_4 + R_1\theta_1 + R_2\theta_2$$

When $\theta_1 < 0$, these relations are reversed.

The coupling matrix for the finger is computed directly from the extension functions. When $\theta_1 > 0$,

$$P(\theta) = \frac{\partial h}{\partial \theta}^T = \begin{bmatrix} -2\sqrt{a^2+b^2}\sin\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) & -R_1 & R_1 & R_1 \\ 0 & 0 & -R_2 & R_2 \end{bmatrix}.$$

Note that pulling on the tendons routed to the outer joints (tendons 1 and 4) generates torques on the first joint as well as the second joint.

4.2 Elastic tendons

The preceding kinematic analysis can also be extended to elastic tendons. We assume that the tendons are completely free to slide along the fingers,

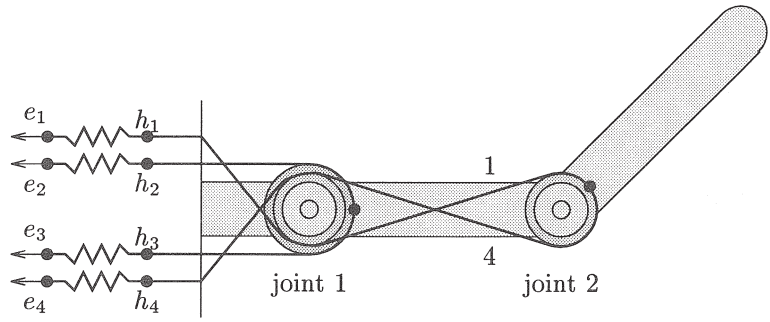


Figure 6.10: Planar finger with position-controlled elastic tendons.

and hence we can lump all elasticity into a single spring element at the base of the tendon, as shown in Figure 6.10. We further assume that the tendon is massless and hence has no dynamics. In practice, this is a good approximation since tendon networks are usually much faster than the dynamics of the underlying robot.

Since tendons are one-dimensional devices, the force relationships derived for inelastic tendons also hold in the case when the tendons are elastic. To see why this is so, consider the instantaneous effect of applying a force to the end of a tendon. Assuming the tendon has negligible mass, the tendon will immediately stretch until the force due to the displacement of the tendon balances the applied force. However, in this case, the tendon will be applying exactly the same amount of force to the mechanism, and hence our previous analysis holds.

When elastic tendons are used, it is also possible to control the *position* of the end of the tendon and use the elasticity of the tendon to convert this into a force. Let e_i be the extension of the tendon as commanded by the actuator and let $h_i(\theta)$ be the extension of the tendon due to the mechanism. We assume that when $\theta = 0$ and $e_i = 0$ the tendon is under zero tension. The net force applied to the tendons is given by

$$f_i = k_i(e_i + h_i(0) - h_i(\theta)),$$

where k_i is the stiffness of the tendon.

Letting K be the diagonal matrix of tendon stiffnesses, we have, for a completely elastic network of tendons with extension e,

$$f = K(e + h(0) - h(\theta))$$

and the dynamics become

$$M(\theta) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) + PK(h(\theta) - h(0)) = PKe$$

The function $S(\theta) := PK(h(\theta) - h(0))$ models the stiffness of the tendon network while $Q := PK$ becomes the new coupling matrix between the

tendon extension and the equivalent joint torques. If the input positions are constant, then $S(\theta)$ gives the restoring force generated as a result of bending the finger away from the equilibrium configuration.

Force- and position-controlled tendons can also be combined, as illustrated in the exercises.

Example 6.5. Coupling matrix for a finger with elastic tendons Consider the example shown in Figure 6.10. The extension functions are given by

$$\begin{aligned} h_1 &= l_1 + r_{11}\theta_1 - r_{12}\theta_2 \\ h_2 &= l_2 - r_{21}\theta_1 \\ h_3 &= l_3 + r_{31}\theta_1 \\ h_4 &= l_4 - r_{41}\theta_1 + r_{42}\theta_2, \end{aligned}$$

where r_{ij} is the radius of the pulley for the i^{th} tendon on the j^{th} joint. The coupling matrix is

$$P(\theta) = \frac{\partial h}{\partial \theta}^T = \begin{bmatrix} r_{11} & -r_{21} & r_{31} & -r_{41} \\ -r_{12} & 0 & 0 & r_{42} \end{bmatrix}.$$

Since all of the extension functions are linear, the coupling matrix is constant.

To compute the relationship between the actuator position and the joint torques, we use the stiffness matrix

$$K = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix},$$

where $k_i > 0$ is the stiffness of the i^{th} tendon. The overall stiffness is given by

$$\begin{aligned} S(\theta) &= PK(h(\theta) - h(0)) \\ &= \begin{bmatrix} k_1 r_{11}^2 + k_2 r_{21}^2 + k_3 r_{31}^2 + k_4 r_{41}^2 & -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} \\ -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} & k_1 r_{12}^2 + k_4 r_{42}^2 \end{bmatrix} \theta, \end{aligned}$$

and the coupling matrix between the actuator extension and the joint torques is

$$Q = PK = \begin{bmatrix} k_1 r_{11} & -k_2 r_{21} & k_3 r_{31} & -k_4 r_{41} \\ -k_1 r_{12} & 0 & 0 & k_4 r_{42} \end{bmatrix}.$$

4.3 Analysis and control of tendon-driven fingers

One peculiarity of using tendon networks is that all tendon tensions must be strictly positive. Hence, the set of all torques which can be applied

is given by the positive span of the columns of $P(\theta)$. This is analogous to grasping using frictionless point contacts, and the same tools can be applied to analyze the kinematic properties of the tendon network.

A control law for a tendon-driven robot computes the joint torques $\tau \in \mathbb{R}^n$ which must be generated by applying forces to the tendons. We say that a tendon network is *force-closure* if for any $\tau \in \mathbb{R}^n$ there exists a set of forces $f \in \mathbb{R}^p$ such that

$$P(\theta)f = \tau \qquad \text{and} \qquad f_i > 0,\ i = 1, \ldots, p. \tag{6.34}$$

As in the grasping case, a necessary and sufficient condition for a tendon network to be force-closure is that $P(\theta)$ be surjective and there exist a strictly positive vector of internal forces $f_N \in \mathbb{R}^p$, $f_{N,i} > 0$ such that $P(\theta)f_N = 0$.

Limits on the number of tendons necessary to construct a force-closure tendon network are given by Caratheodory's and Steinitz's theorems, which were given in Chapter 5, Section 4. Caratheodory's theorem asserts that for a robot with n links, at least $n + 1$ tendons are required to actuate it, while Steinitz's theorem proves that any more than $2n$ tendons are redundant. In fact, these two bounds correspond to the two most common types of tendon networks, referred to as "$N + 1$" and "$2N$" tendon configurations. The $N+1$ configuration usually consists of a single tendon which pulls on all of the joints in one direction, together with n additional tendons which generate torques in the opposite direction. The $2N$ configuration is the one used in all of the examples here, where we attach two tendons to each joint, acting in opposite directions.

For tendon networks which are actuated by force-controlled devices, the tendon forces chosen to exert a given vector of joint torques have the form

$$f = P^+(\theta)\tau + f_N,$$

where $P^+ = P^T(PP^T)^{-1} \in \mathbb{R}^{m \times n}$ is the pseudo-inverse of the coupling matrix and $f_N \in \mathcal{N}(P) \cap \mathbb{R}^p_+$ is an internal force that insures that all tendon tensions are positive. In most situations, f_N will be chosen as small as possible, so that the tendons remain taut but are not subjected to excessive internal forces.

The case of elastic tendons with position-controlled actuators is handled exactly the same way, except that we must solve

$$P(\theta)Ke = \tau \quad \text{and} \quad e_i + h_i(0) - h_i(\theta) > 0,\ i = 1, \ldots, p. \tag{6.35}$$

Since $K \in \mathbb{R}^{p \times p}$ is an invertible stiffness matrix, if the tendon network is force-closure, then there exists a vector of extensions $e_N \in \mathbb{R}^p$ such that $e_{N,i} > 0$ and $PKe_N = 0$. By choosing

$$e = (PK)^+\tau + e_N$$

with e_N sufficiently large in magnitude, we can insure that the constraints in equation (6.35) are satisfied.

The tools presented here can be extended to the case of mixed networks of rigid and elastic tendons and also to tendons which extend between two links in the robot. These cases are explored in the exercises.

5 Control of Robot Hands

In this section, we concentrate on the control aspects of multifingered robot hands and show how to extend previous controllers (presented in Chapter 4) to apply to grasping and other coordinated manipulation tasks. In addition, we include some thoughts on organization of complex controllers motivated in part by the type of control mechanisms found in biological motor control systems.

5.1 Extending controllers

For a constrained manipulation problem, we can break the control problem into two main parts:

1. Tracking a given object (or workspace) trajectory
2. Maintaining a desired internal force

Under the assumption that all objects and links are rigid and their geometries completely known, these two problems can be partially decoupled. We first find joint torques which satisfy the tracking requirement and then add sufficient internal forces to keep the contact forces inside the appropriate friction cones or satisfy some other force objective.

More specifically, suppose that we have a constrained robotic system with dynamics of the form

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + N(q,\dot{q}) = F = GJ^{-T}\tau,$$

with $q = (\theta, x) \in \mathbb{R}^n \times \mathbb{R}^p$. As we saw in the first part of this chapter, a large class of systems can be modeled by equations of this form with $GJ^{-T} : \mathbb{R}^n \to \mathbb{R}^p$ a surjective map. In this framework, $x \in \mathbb{R}^p$ represents the position variable and the null space of G corresponds to the internal force directions (assuming J is invertible). In addition to the general form of the dynamics, we also assume that $M(q) > 0$ for all q and that $\dot{M} - 2C$ is a skew-symmetric matrix. These properties hold for all of the systems given in this chapter with the proper definition of J and G.

The tracking problem is to find joint torques which cause the system to asymptotically track a given workspace trajectory $x_d(\cdot)$. To solve this problem, we begin by treating F as a direct input to the system. Since GJ^{-T} is surjective, it is always possible to find a set of torques τ

which realizes this virtual input. Furthermore, since M and C satisfy the properties of Lemma 6.1, we can use any of the controllers derived for open-chain manipulators to asymptotically track a trajectory $x_d(t)$.

As an example, the computed torque controller for a constrained robot system becomes

$$F = M(q)\left(\ddot{x}_d - K_v\dot{e} - K_p e\right) + C(q,\dot{q})\dot{x} + N(q,\dot{q}),$$

where $e := x - x_d$. Note that in this controller $M(q)$ depends on both θ and x. In the special case that the system constraints are holonomic, the dependence on θ can be removed, but this usually involves inverting a nonlinear map. However, since the only essential properties required by the proofs of convergence are that $M(q)$ be positive definite and $\dot{M} - 2C$ skew-symmetric, all of the previous proofs apply directly.

Once F has been determined, the joint torques τ are chosen so that $GJ^{-T}\tau = F$. It is always possible to find some such τ, since by assumption GJ^{-T} is surjective. However, GJ^{-T} is not necessarily injective and hence there may be many values of τ which generate F. In fact, since J is taken to be invertible, the extra freedom in τ corresponds to the existence of internal forces in the system. The general solution to $GJ^{-T}\tau = F$ has the form

$$\tau = J^T G^+ F + J^T f_N, \tag{6.36}$$

where $G^+ = G^T(GG^T)^{-1}$ is the pseudo-inverse of G and $f_N \in \mathcal{N}(G)$. Since $Gf_N = 0$, f_N can be chosen arbitrarily without affecting the trajectory tracking characteristics of the controller.

The extra freedom in τ is used to satisfy the second part of the control problem, regulating internal forces. For a grasping problem, f_N must be chosen such that the net contact force lies in the friction cone FC. This is an extremely important condition, since our entire problem formulation assumed that the fingers remained in contact with the object (in the directions specified by the contact model). For other types of problems, such as coordinated lifting, the regulation of the internal forces is not quite so critical, since the mechanical structure will act to enforce the constraints at all times. In this case we often choose $f_N = 0$ to minimize application of internal forces.

In situations in which regulation of internal forces is desired, a further complication arises from the fact that the net contact force is not given simply by $G^+F + f_N$. Rather, this is the contact force due to the *actuators*. Additional contact forces may be generated by the dynamics of the system, as discussed in Section 2.2. To truly insure that the contact forces remain inside the friction cone, the full dynamics must be taken into account. This can be done either by computing the forces of constraint or measuring them.

Fortunately, for many problems a detailed analysis of the internal forces is not necessary. If the forces of constraint due to non-quasistatic motion are small compared to the desired internal force, they can often be ignored. We make that assumption here and assume that we are given a desired internal force $f_{N,d}(\cdot)$ which we wish to regulate and that the dynamic forces of constraint can be ignored. In this case the final control law has the form

$$\tau = J_h^T G^+ F + J_h^T f_{N,d}$$

where F is the virtual force generated to satisfy the trajectory tracking requirement.

A second possibility for controlling internal forces is to measure the applied internal forces and adjust f_N using a second feedback control law. This must be done carefully since, for a rigid robot system, forces are transmitted instantaneously and have no associated dynamics. This can lead to problems in which the control problem is ill-posed due to the existence of algebraic loops. For example, consider a "robot" which only applies forces. Let f be the applied force and f_d the desired force. A proportional controller regulating the force is given by

$$f = \alpha(f - f_d).$$

For $\alpha \neq 0$, this controller only satisfies $f = f_d$ if $f = f_d = 0$. For any other f_d, we have an algebraic loop which is never satisfied.

One common way to overcome this problem is to use an integrator. Again we let f_d be the desired force and set

$$f = f_d + \alpha \int (f - f_d)\, dt.$$

This controller consists of a feedforward piece, f_d, and an integrating compensator. Setting $f = f_d$ generates no contradictions, and hence this controller is well-posed. It is important to note that adding an integrator must be done carefully to avoid introducing unstable behavior into the system due to measurement noise and "integrator-windup." Details of these problems can be found in standard undergraduate texts on control engineering, such as [3] and [34].

5.2 Hierarchical control structures

A multifingered robot hand can be modeled as a set of robots which are coupled to each other and an object by a set of velocity constraints. The analysis presented in the beginning of this chapter allows us to model this interconnection and create a new dynamical system which incorporates the constraints. In fact, this procedure is sufficiently straightforward that it may be automated: by specifying the contact constraint between the

robots and the object, the new equations of motion for the composite robot can be derived using a symbolic manipulation program.

A significantly more difficult problem is that of constructing *controllers* for robot systems. Although conceptually simple, a controller for a multifingered robot hand must be able to control a very complex system with many degrees of freedom, large amounts of sensory data, and multiple control objectives. A typical hand might have 10–20 actuators, 10–15 constraints, and a state-space of dimension 30 or higher. A control law for such a system might need to run at a control frequency of 500 Hz or more in order to yield acceptable performance. Computing the control torques for such a system in under 2 milliseconds is often impossible if the system is modeled as a single, complex dynamical system.

The difficulties in controlling systems with many degrees of freedom have also been noted in the biomechanics literature. The study of human biological motor control mechanisms led the Russian psychologist Bernstein to question how the brain could control a system with so many different degrees of freedom interacting in such a complex fashion [41]. Many of these same complexities are also present in robotic systems and limit our ability to use multifingered hands and other robotic systems to their full advantage.

In the remainder of this section, we describe one possible way of structuring controllers which attempts to address some of the difficulties inherent in the control of constrained robot systems with many degrees of freedom. We present a set of primitive operations that allow a complex robot controller to be built up in a hierarchical fashion and discuss some of the issues involved in the resulting control structure. The material contained in this section was originally presented in [76], where a more detailed description is given.

Defining robots

We wish to build up complex control laws by utilizing the geometric constraints between the mechanisms which make up the overall system. We will model all mechanisms as a generalized object which we label as a *robot*. A robot consists of a dynamical system whose equations of motion have the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q,\dot{q}) = F.$$

The quantities M, C, and N completely parameterize the dynamics of the mechanism.

In addition to the parameters (M, C, N), a robot also has a set of inputs and outputs. The inputs consist of the desired position of the robot, x_d, and the forces to be applied to the robot, F_d. The outputs are the actual position of the mechanism, x, and the measured force F. Some types of robots may not use or define all of these inputs and outputs.

The relationship between the various parameters describing a robot depends on the robot itself. For example, we model an actuated, open-chain mechanism using the relationships

$$F = F_d$$
$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q,\dot{q}) = F.$$

Thus, given a desired force to be applied to the robot, the robot will move according to the equations of motion. The desired trajectory input for an uncontrolled robot is ignored. (We will make use of this input later when we attach controllers to robots.)

In addition to modeling actuated mechanisms, a robot can also describe an inanimate object. In this case, all inputs to the robot are ignored and the outputs from the robot provide information about the current position of the object and the forces acting on it, if available. The inertial parameters (M, C, N) are used as before to model the dynamics of the object.

The utility of defining a generalized object called a robot is that we may define operations which take one or more generalized robots and yield a new generalized robot. We define two such operations below. In order to define the new object, we must define the inertial properties as well as a description of the inputs and outputs. These are typically defined recursively, so that a composite robot queries and commands its children in response to requests for inputs and outputs.

Attaching robots

The first operation which we define is the *attach* operation, which reflects geometrical constraints between two or more robots. It creates a new robot object from the attributes of its children. Its definition (and name) is motivated by the attachment of a set of fingers to an object, but its use is much more general.

The arguments to the attach operation are a list of robots, which we refer to as fingers, together with a single object which we refer to as the payload. In addition, we are given a constraint between the configuration variables of the fingers and those of the object. For simplicity, we take this constraint to be of the form $h(\theta, x) = 0$ where $\theta = (\theta_{f_1}, \ldots, \theta_{f_k}) \in \mathbb{R}^n$ is the vector of finger joint angles and $x \in \mathbb{R}^p$ is the configuration of the payload.

To construct a new robot, we use the Lagrange-d'Alembert equations to write the dynamics in terms of the payload variables $x \in \mathbb{R}^p$. Let (M_f, C_f, N_f) be the (block-diagonal) parameters for the fingers and (M_p, C_p, N_p) be those of the payload. The dynamic parameters for the

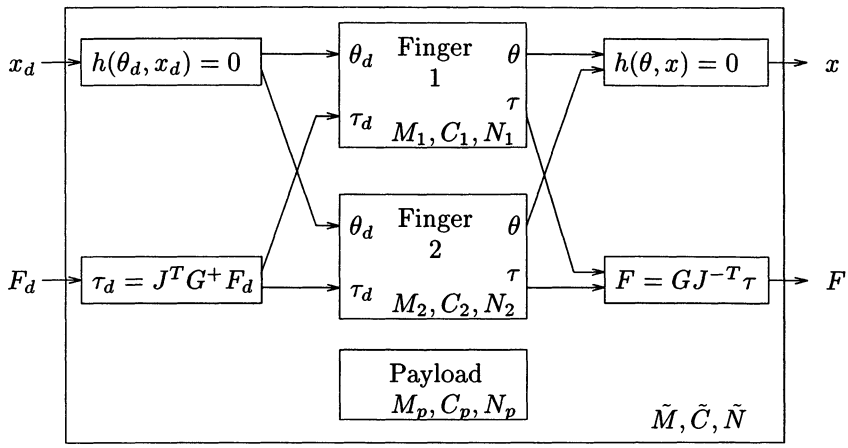


Figure 6.11: Data flow between two robots which have been attached to a payload.

constrained robot are given by

$$\tilde{M} := M_p + GJ^{-T}M_f J^{-1}G$$
$$\tilde{C} := C_p + GJ^{-T}C_f J^{-1}G + GJ^{-T}M_f \frac{d}{dt}\left(J^{-1}G\right)$$
$$\tilde{N} := N_p + GJ^{-T}N_f,$$

where $J = \frac{\partial h}{\partial \theta}$, $G^T = -\frac{\partial h}{\partial x}$, and we assume that J is invertible for the purposes of exposition. Note that in order to evaluate $\tilde{M}$ at the current configuration, we can query the payload and each of the fingers for their current inertia matrices and then combine these using the constraints.

To read the configuration of the composite robot, we query the state of all the robots in the list of daughter robots and then solve the (holonomic) constraint $h(\theta, x) = 0$ to find the current payload configuration. Alternatively, if the payload is equipped with sensors (perhaps an external camera which tracks the payload), this data can be used instead. A similar computation or measurement can be used to determine the net force on the object, which will consist of contact forces applied by the fingers and external forces applied by the environment.

Commanding the desired position and force on the robot also uses the constraint equations to distribute information to the fingers and the payload. If all fingers are uncontrolled, actuated mechanisms, then the desired forces will be applied to the actuators and the desired position will be ignored.

A diagram illustrating the data flow in a robot constructed by the attach operation is shown in Figure 6.11. In addition to modeling grasp

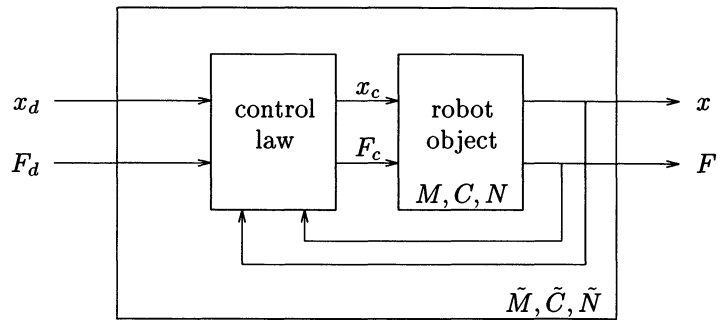


Figure 6.12: Data flow in a typical controlled robot.

constraints, the attach operation can be used to model other situations, as described in Section 2.3. For example, we can change from joint space to workspace coordinates or add variables parameterizing the internal motion of a redundant robot.

Controlling robots

The *control* operation is responsible for assigning a controller to a robot. It is also responsible for creating a new robot with attributes that properly represent the controlled robot. The attributes of the created robot are completely determined by the individual controller. For most controllers, the current state of the controlled robot is equivalent to the current state of the uncontrolled robot. Sending a desired trajectory to a controlled robot would cause the controller to buffer the data and attempt to follow that trajectory. A controlled robot is illustrated in Figure 6.12.

The dynamic attributes $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ for the newly created robot are determined by the controller. At one extreme, a controller which compensates for the inertia of the robot would set the dynamic attributes of the controlled robot to zero. This does not imply that the robot is no longer a dynamic object, but rather that controllers at higher levels can ignore the dynamic properties of the robot, since they are being compensated for at a lower level. At the other end of the spectrum, a controller may make no attempt to compensate for the inertia of a robot, in which case it should pass the dynamic attributes on to the next higher level. Controllers which lie in the middle of this range may partially decouple the dynamics of the manipulator without actually completely compensating for them.

Building hierarchical controllers

The operation described above allow us to build complex hierarchical control laws for robot systems. Since a controller accepts a robot object and

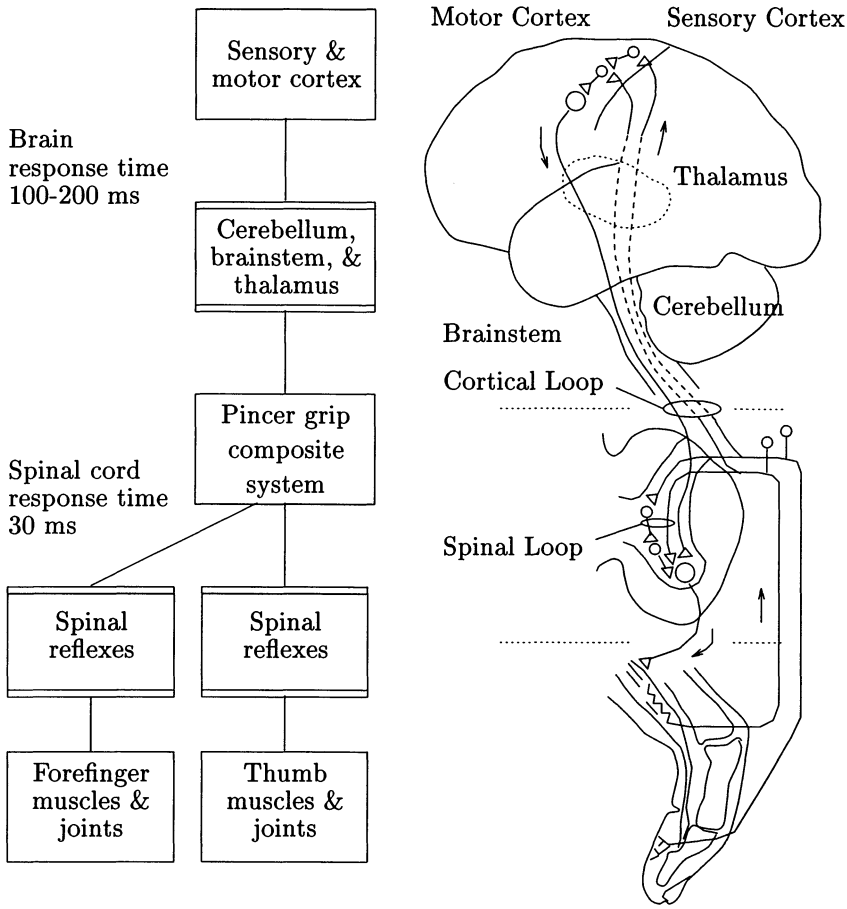


Figure 6.13: Hierarchical control scheme for a human finger. (Figure courtesy of D. Curtis Deno)

creates a robot object, it can be inserted at any level in the description of a constrained robot system. Thus, we can easily define hierarchical controllers whose structure mirrors the geometric structure of the system. We illustrate this in the following examples.

Example 6.6. Biological motor control
Figure 6.13 shows a hierarchical control scheme for a human finger. At the highest level, the brain is represented as sensory and motor cortex (where sensory information is perceived and conscious motor commands originate) and brainstem and cerebellar structures (where motor commands are coordinated and sent down the spinal cord). A pair of fingers forms a composite system for grasping which is shown integrated at the

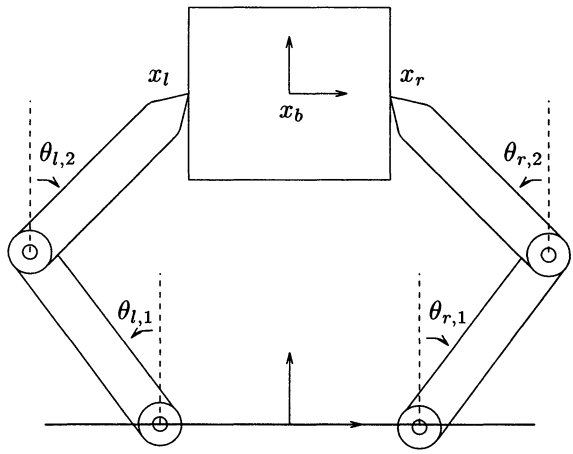


Figure 6.14: Planar two-fingered hand and a hierarchical control law.

level of the spinal cord. The muscles and sensory organs of each finger form low-level spinal reflex loops. These low-level loops respond more quickly to disturbances than sensory motor pathways which travel to the brain and back. Brain and spinal feedback controllers are represented by double-lined boxes.

The block diagram portion of Figure 6.13 is a (biological) example of a robot system built using the operations described above. Starting from the bottom: two fingers (robots) are defined; each finger is controlled by muscle tension/stiffness and spinal reflexes; the fingers are attached to form a composite hand; the brainstem and cerebellum help control and coordinate motor commands and sensory information; and finally, at the level of the cortex, the fingers are thought of as a pincer which engages in high-level tasks such as picking.

Example 6.7. Hierarchical control of a two-fingered planar hand As a second, more practical example, consider the planar hand shown in Figure 6.14. The parameters describing the dynamics of the fingers and the object, as well as the constraints between them, are easily computed and are given in earlier examples. We can build a hierarchical controller which is similar to the biological controller described in the previous example. This control structure is shown graphically in Figure 6.15.

At the lowest level, we use simple PD control laws attached directly to the individual fingers. These PD controllers mimic the stiffness provided by muscle coactivation in a biological system. Additionally, controllers at this level might be used to represent spinal reflex actions. At a somewhat higher level, the fingers are attached and considered as a single unit with relatively complicated dynamic attributes and Cartesian configuration.

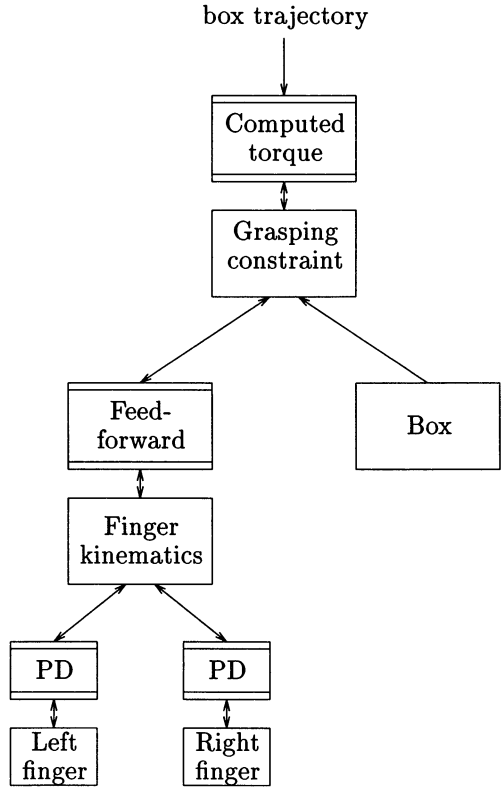


Figure 6.15: A hierarchical controller for multifingered grasping.

At this point, we employ a feedforward controller (computed torque with no error correction) to simplify these dynamic properties, as viewed by higher levels of the brain. With respect to these higher levels, the two fingers appear to be two Cartesian force generators, represented as a single composite robot.

Up to this point, the representation and control strategies do not explicitly involve the box, a payload object. The force generators are next attached to the box, yielding a robot with the dynamic properties of the box, but capable of motion due to the actuation in the fingers. Finally, we use a computed torque controller at the very highest level to allow us to command motions of the box without worrying about the details of muscle actuation. By this controller, we simulate the actions of the cerebellum and brainstem to coordinate motion and correct for errors.

It is helpful to illustrate the flow of information to the highest level control law. In the evaluation of the current box configuration and tra-

jectory, x_b and $\dot{x}_b$, the following sequence of actions occurs:

Hand: asks for current state, x_b and $\dot{x}_b$
  Finger: ask for current state, x_f and $\dot{x}_f$
    Left: read current state, θ_l and $\dot{\theta}_l$
    Right: read current state, θ_r and $\dot{\theta}_r$
  Finger: $x_f, \dot{x}_f \leftarrow f(\theta_l, \theta_r), J(\dot{\theta}_l, \dot{\theta}_r)$
Hand: $x_b, \dot{x}_b \leftarrow g(x_f), G^{+T}\dot{x}_f$.

When we write a set of hand forces, a similar chain of events occurs.

The structure in Figure 6.15 also has interesting properties from a more traditional control viewpoint. The low-level PD controllers can be run at high servo rates (due to their simplicity) and allow us to tune the response of the system to reject high-frequency disturbances. The Cartesian feedforward controller permits a distribution of the calculation of nonlinear compensation terms at various levels, lending itself to multiprocessor implementation. Finally, using a computed torque controller at the highest level gives the flexibility of performing the controller design in the task space and results in a system with linear error dynamics.

6 Summary

The following are the key concepts covered in this chapter:

1. The dynamics of a mechanical system with Lagrangian $L(q, \dot{q})$, subject to a set of *Pfaffian constraints* of the form

$$A(q)\dot{q} = 0 \qquad A(q) \in \mathbb{R}^{k \times n},$$

can be written as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} + A^T(q)\lambda - \Upsilon = 0,$$

where $\lambda \in \mathbb{R}^k$ is the vector of *Lagrange multipliers*. The values of the Lagrange multipliers are given by

$$\lambda = (AM^{-1}A^T)^{-1}\left(AM^{-1}(F - C\dot{q} - N) + \dot{A}\dot{q}\right).$$

2. The *Lagrange-d'Alembert formulation* of the dynamics represents the motion of the system by projecting the equations of motion onto the subspace of allowable motions. If $q = (q_1, q_2) \in \mathbb{R}^{(n-k)\times k}$ and the constraints have the form

$$\dot{q}_2 = \mathcal{A}(q)\dot{q}_1,$$

then the equations of motion can be written as

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \Upsilon_1\right) + \mathcal{A}^T\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_2} - \Upsilon_2\right) = 0.$$

In the special case that the constraint is integrable, these equations agree with those obtained by substituting the constraint into the Lagrangian and then using the unconstrained version of Lagrange's equations.

3. The dynamics for a *multifingered robot hand* with joint variables $\theta \in \mathbb{R}^n$ and (local) object variables $x \in \mathbb{R}^p$, subject to the grasp constraint

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x},$$

is given by

$$\tilde{M}(q)\ddot{x} + \tilde{C}(q, \dot{q})\dot{x} + \tilde{N}(q, \dot{q}) = F,$$

where $q = (\theta, x)$ and

$$\begin{aligned}
\tilde{M} &= M_o + GJ_h^{-T}M_f J_h^{-1}G^T \\
\tilde{C} &= C_o + GJ_h^{-T}\left(C_f J_h^{-1}G^T + M_f \frac{d}{dt}\left(J_h^{-1}G^T\right)\right) \\
\tilde{N} &= N_o + GJ_h^{-T}N_f \\
F &= GJ_h^{-T}\tau.
\end{aligned}$$

These same equations can be applied to a large number of other robotic systems by choosing G and J_h appropriately.

4. For *redundant* and/or *nonmanipulable* robot systems, the hand Jacobian is not invertible, resulting in a more complicated derivation of the equations of motion. For redundant systems, the constraints can be extended to the form

$$\underbrace{\begin{bmatrix} J_h \\ K_h \end{bmatrix}}_{\bar{J}_h} \dot{\theta} = \underbrace{\begin{bmatrix} G^T & 0 \\ 0 & I \end{bmatrix}}_{\bar{G}^T} \begin{bmatrix} \dot{x} \\ v_N \end{bmatrix},$$

where the rows of K_h span the null space of J_h, and v_N represents the *internal motions* of the system. For nonmanipulable systems, we choose a matrix H which spans the space of allowable object trajectories and write the constraints as

$$J_h \dot{\theta} = \underbrace{G^T H}_{\bar{G}^T} w,$$

where $\dot{x} = H(q)w$ represents the object velocity. In both the redundant and nonmanipulable cases, the augmented form of the constraints can be used to derive the equations of motion and put them into the standard form given above.

5. The kinematics of *tendon-driven systems* are described in terms of a set of *extension functions*, $h_i : Q \to \mathbb{R}$, which measures the displacement of the tendon as a function of the joint angles of the system. If a vector of tendon forces $f \in \mathbb{R}^k$ is applied at the end of the tendons, the resulting joint torques are given by

$$\tau = P(\theta) f,$$

where $P(\theta) \in \mathbb{R}^{n \times p}$ is the *coupling matrix*:

$$P(\theta) = \frac{\partial h}{\partial \theta}^T (\theta).$$

A tendon-system is said to be *force-closure* at a point θ if for every vector of joint torques, τ, there exists a set of tendon forces which will generate those torques.

6. The equations of motion for a constrained robot system are described in terms of the quantities $\tilde{M}(q)$, $\tilde{C}(q, \dot{q})$, and $\tilde{N}(q, \dot{q})$. When correctly defined, the quantities satisfy the following properties:

   (a) $\tilde{M}(q)$ is symmetric and positive definite.

   (b) $\dot{\tilde{M}}(q) - 2\tilde{C}$ is a skew-symmetric matrix.

Using these properties it is possible to extend the controllers presented in Chapter 4 to the more general class of systems considered in this chapter. For a multifingered hand, an extended control law has the general form

$$\tau = J^T G^+ F + J^T f_N,$$

where F is the generalized force in object coordinates (determined by the control law) and f_N is an internal force. The internal forces must be chosen so as to insure that all contact forces remain inside the appropriate friction cone so that the fingers satisfy the fundamental grasp constraint at all times.

7 Bibliography

The derivation of the equations of motion in the presence of nonholonomic constraints is classical, although many recent textbooks only present the case in which all constraints are holonomic. The derivation presented here is based in part on the much more detailed and instructional analysis presented by Rosenberg [100]. The splitting of the dynamics for nonholonomic systems into a first-order piece and second-order piece is based on the presentation by Bloch et al. [8].

Derivations of the dynamics of multifingered hands and extensions of standard robot control laws can be found in [63] and [76, 78]. The derivation presented here follows [78]. We also mention the work of Khatib on operational space methods [51], which is closely related and provided initial inspiration for our approach. The material on tendon kinematics is based on the formulation of Deno et al. [26], which also considers the more general case of actuator networks with compliance and internal loops.

8 Exercises

1. Calculate the dynamics of a spherical pendulum using the Lagrange-d'Alembert equations and check that they agree with the result derived in Example 6.1.

2. For the rolling disk in Example 6.2, show that substituting the constraints on $\dot{x}$ and $\dot{y}$ into the Lagrangian and then applying Lagrange's equations (without constraints) gives the wrong equations of motion.

3. Calculate the constraint forces for the rolling penny described in Example 6.2.

4. *Structural properties of the Newton-Euler equations*
   Consider the Newton-Euler equations for the motion of a rigid body:

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b.$$

   Let $x_o = (p, R) \in SE(3)$ denote the configuration of the rigid body and $V^b = (v^b, \omega^b) = (x_o^{-1}\dot{x}_o)^\vee \in \mathbb{R}^6$ the body velocity.

   (a) Show that the Newton-Euler equations can be written as

$$M(x_o)\dot{V}^b + C(x_o, \dot{x}_o)V^b = F^b,$$

   where $M(x_o) > 0$ and $\dot{M} - 2C$ is a skew-symmetric matrix.

   (b) Let $x_o = \phi(x)$, $x \in \mathbb{R}^6$ be a local parameterization of x_o (using Euler angles to represent rotation, for example). Assuming that the matrix

$$J(x) = \left[\left(\phi^{-1} \frac{\partial \phi}{\partial x_1} \right)^\vee \quad \cdots \quad \left(\phi^{-1} \frac{\partial \phi}{\partial x_6} \right)^\vee \right] \in \mathbb{R}^{6\times 6}$$

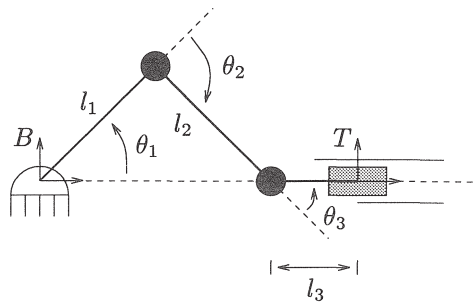
   is nonsingular, show that the equations of motion become

$$\bar{M}(x)\ddot{x} + +\bar{C}(x, \dot{x})\dot{x} = J^T(x)F^b,$$

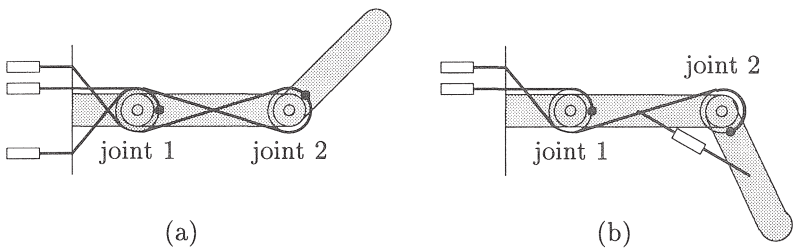
   where

$$\bar{M} = J^T M J$$
$$\bar{C} = J^T C J + J^T M \dot{J}.$$

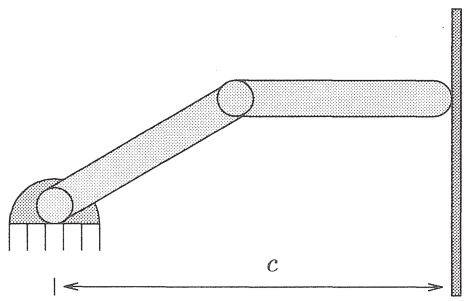
5. Derive the dynamics of the four-bar linkage shown below in terms of the crank angle $\theta_1 \in \mathbb{S}^1$.



6. Derive the actuator kinematics for the tendon networks shown below. Consider both the rigid and elastic cases.



7. Consider the two-link planar manipulator shown below.



(a) Calculate the kinematic constraints on the system.

(b) Calculate the dynamics of the manipulator assuming that the end-effector remains in contact with the wall at $x = c$. Model the inertia of each link as a point mass concentrated at the middle of the link.

(c) Calculate the internal forces due to the manipulator dynamics.

(d) Design a PD controller which gives asymptotic tracking of trajectories in the y direction while pushing with constant force $f_d > 0$ in the x direction.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 7

Nonholonomic Behavior in Robotic Systems

DOI: 10.1201/9781315136370-7
This chapter has been made available under a CC-BY 4.0 license.

In this chapter, we study the effect of nonholonomic constraints on the behavior of robotic systems. These constraints arise in systems such as multifingered robot hands and wheeled mobile robots, where rolling contact is involved, as well as in systems where angular momentum is conserved. We discuss the problem of determining when constraints on the velocities of the configuration variables of a robotic system are integrable, and illustrate the problem in a variety of different situations. The emphasis of this chapter is on the basic tools needed to analyze nonholonomic systems and the application of those tools to problems in robotic manipulation. These tools are drawn both from some basic theorems in differential geometry and from nonlinear control theory.

1 Introduction

In the preceding chapter, we derived the equations of motion for a robotic system with kinematic constraints. We restricted ourselves to Pfaffian constraints which had the general form

$$J(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}, \tag{7.1}$$

where $q = (\theta, x) \in \mathbb{R}^n$ is the configuration of the system. As we saw, equations of this form could be used to model a large number of robotic systems, including multifingered hands, robots in contact with their environment, and redundant manipulators.

By shifting our notation slightly, we can write the preceding constraints in the form

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k, \tag{7.2}$$

where the $\omega_i(q)$ are row vectors. We assume that the ω_i are linearly independent at each point $q \in \mathbb{R}^n$, since if they are not, the dependent constraints may be eliminated. Each ω_i describes one constraint on the directions in which $\dot{q}$ is permitted to take values.

Recall from Chapter 6 that a constraint is said to be *holonomic* if it restricts the motion of a system to a smooth hypersurface of the configuration space. It will be convenient to adopt some language and notation from differential geometry, so we call this smooth hypersurface a *manifold.* Locally, a holonomic constraint can be represented as a set of algebraic constraints on the configuration space,

$$h_i(q) = 0, \qquad i = 1, \ldots, k. \tag{7.3}$$

The dimension of the manifold on which the motion of the system evolves is $n - k$.

We say that a set of k Pfaffian constraints of the form in equation (7.2) is *integrable* if there exist functions $h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, k$ such that

$$h_i(q(t)) = 0 \quad \Longleftrightarrow \quad \omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k.$$

Thus, a set of Pfaffian constraints is integrable if it is equivalent to a set of holonomic constraints. We often call an integrable Pfaffian constraint a holonomic constraint, although strictly speaking the former is described by a set of velocity constraints and the latter by a set of functions. A set of Pfaffian constraints is said to be *nonholonomic* if it is not equivalent to a set of holonomic constraints.

As we saw in Chapter 6, the presence of nonholonomic constraints requires special care in deriving the equations of motion for the system. The point of view taken in this chapter is somewhat different. Here, we will try to understand when we can exploit the nonholonomy of the constraints to achieve motion between configurations. In particular, we will be interested in answering the following question: given two points q_0 and q_f, when does there exist a path $q(t)$ which satisfies the constraints in equation (7.2) at all times and connects q_0 to q_f? The set of all points which can be connected to q_0 via a path which satisfies the constraints is called the *reachable set* associated with q_0. Thus, we wish to understand under what conditions the reachable set will be the entire configuration space. This is intimately related to the nonholonomy of the constraints, since if the constraints are holonomic, then the motion of the system is restricted to the level sets given by $h_i(q) = h_i(q_0)$, $i = 1, \ldots, k$. Hence, for holonomic constraints the reachable set is some subset of the configuration space which lies in the level set $h_i(q) = h_i(q_0)$, and we cannot move freely between configurations on different level sets.

A good example of the type of behavior which we wish to exploit is that of an automobile. The kinematics of an automobile are constrained because the front and rear wheels are only allowed to roll and spin, but

not to slide sideways. As a consequence, the car itself is not capable of sliding sideways, or rotating in place. Despite this, we know from our own experience that we can park an automobile at any position and orientation. Thus, the constraints are not holonomic since the motion of the system is unrestricted. Finding an actual path between two given configurations is an example of a *nonholonomic motion planning problem* and is the subject of the next chapter.

Checking to see if a constraint is holonomic or nonholonomic is neither easy nor obvious. Consider first the case in which there is a single velocity constraint,

$$\omega(q)\dot{q} = \sum_{j=1}^{n} \omega_j(q)\dot{q}_j = 0.$$

This constraint is integrable if there exists a function $h : \mathbb{R}^n \to \mathbb{R}$ such that

$$\omega(q)\dot{q} = 0 \quad \Longleftrightarrow \quad h(q) = 0.$$

It follows by differentiating $h(q) = 0$ with respect to time that if the Pfaffian constraint is holonomic then

$$\sum_{j=1}^{n} \omega_j(q)\dot{q}_j = 0 \quad \Longrightarrow \quad \sum_{j=1}^{n} \frac{\partial h}{\partial q_j}\dot{q}_j = 0.$$

In turn, this implies that there exists some function $\alpha(q)$, called an *integrating factor*, such that

$$\alpha(q)\omega_j(q) = \frac{\partial h}{\partial q_j}(q) \qquad j = 1, \ldots, n. \tag{7.4}$$

Thus, a single Pfaffian constraint is holonomic if and only if there exists an integrating factor $\alpha(q)$ such that $\alpha(q)\omega(q)$ is the derivative of some function h.

Equation (7.4) is not very constructive from the point of view of checking integrability since it involves the unknown function $h(q)$. This situation may be remedied by using the fact that

$$\frac{\partial^2 h}{\partial q_i \partial q_j} = \frac{\partial^2 h}{\partial q_j \partial q_i}$$

to get

$$\frac{\partial(\alpha\omega_j)}{\partial q_i} = \frac{\partial(\alpha\omega_i)}{\partial q_j} \qquad i, j = 1, \ldots, n. \tag{7.5}$$

Equation (7.5) states that the constraint is equivalent to $h(q) = 0$ if there exists some integrating factor $\alpha(q)$ for which the equation (7.5) is true. This should really not be a surprise since

$$\omega(q)\dot{q} = 0 \quad \Longrightarrow \quad \alpha(q)\omega(q)\dot{q} = 0$$

for all choices of smooth functions $\alpha(q)$. However, one still has to find a function α which satisfies equation (7.5).

The question of integrability becomes much more difficult in the presence of multiple Pfaffian constraints. Given a set of k constraints of the form of equation (7.2), not only does one need to check whether each one of the k constraints is integrable, but also which independent linear combinations of these,

$$\sum_{i=1}^{k} \alpha_i(q)\omega_i(q)\dot{q},$$

are integrable. That is, even if the given constraints are not individually integrable, they may contain a set of integrable constraints. Thus, there may exist functions h_i for $i = 1, \ldots, p$ with $p \leq k$ such that

$$\text{span}\{\frac{\partial h_1}{\partial q}(q), \ldots, \frac{\partial h_p}{\partial q}(q)\} \quad \subset \quad \text{span}\{\omega_1(q), \ldots, \omega_k(q)\}$$

for all q. If it is possible to find these functions, the motion of the system is restricted to level surfaces of h, namely to sets of the form

$$\{q : h_1(q) = c_1, \ldots, h_p(q) = c_p\}.$$

If $p = k$, then the constraints are holonomic. In the case that $p < k$, the constraints are not holonomic (since they are not completely equivalent to a set of holonomic constraints) but the reachable points of the system are still restricted. Thus the constraints are "partially holonomic." We will be primarily interested in the case in which the constraints do not restrict the reachable configurations. We refer to this situation as being *completely nonholonomic.*

It will be convenient for us to convert problems with nonholonomic constraints into another form. Roughly speaking, we would like to examine the systems not from the point of view of the constraints (namely, the directions that we *cannot* move), but rather from the viewpoint of the directions in which we are *free* to move. We begin by choosing a basis for the right null space of the constraints, denoted by $g_j(q) \in \mathbb{R}^n$, $i = 1, \ldots, n - k =: m$. By construction, this basis satisfies

$$\omega_i(q)g_j(q) = 0 \qquad \begin{array}{l} i = 1, \ldots, k \\ j = 1, \ldots, n-k, \end{array}$$

and the allowable trajectories of the system can thus be written as the possible solutions of the control system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m. \tag{7.6}$$

That is, $q(t)$ is a feasible trajectory for the system if and only if $q(t)$ satisfies equation (7.6) for some choice of controls $u(t) \in \mathbb{R}^m$.

In this context, a constraint is completely nonholonomic if the corresponding control system can be steered between any two points. Thus the reachable configurations of the system are not restricted. Conversely, if a constraint is holonomic, then all motions of the system must lie on an appropriate constraint surface and the corresponding control system can only be steered between points on the given manifold. Hence, we can study the nature of Pfaffian constraints by studying the controllability properties of equation (7.6).

Nonholonomic constraints arise in a variety of applications. Besides rolling constraints on multifingered hands, nonholonomic constraints play an important role in the study of mobile robot systems and space-based robotic systems (in which conservation of angular momentum plays the role of a nonholonomic constraint). For these applications the primary question is that of reachability: when can we find a path between two arbitrary configurations and how do we go about computing such a path?

The outline of this chapter is as follows: in Section 2 we develop some tools from differential geometry and nonlinear control. Section 3 gives examples of systems with velocity constraints. In Section 4 the structure of nonholonomic systems is explored and the examples of Section 3 are analyzed. In the next chapter, we will develop methods for planning paths compatible with nonholonomic constraints.

Both this chapter and Chapter 8 are slightly more advanced in flavor than the previous chapters and represent some of the recent research in the robotics literature. Nonholonomic behavior also plays a strong role in many problems in geometric mechanics, which we touch on only briefly in the examples and exercises. In classical mechanics, nonholonomic behavior is closely related to the geometric phase associated with a group symmetry in a Hamiltonian or Lagrangian system. A good introduction to these concepts can be found in the lecture notes by Marsden [68].

2 Controllability and Frobenius' Theorem

In the previous section, we saw the difficulties in trying to determine whether or not constraints on a system were holonomic (or integrable). Further, if they are not holonomic, it is not completely clear as to when they are completely nonholonomic. In this section, we will develop the machinery needed for analyzing nonholonomic systems, in particular for answering the question of when a set of Pfaffian constraints is holonomic.

The tools we develop are based on a variety of results from differential geometry and nonlinear control theory, more specifically Frobenius' theorem and nonlinear controllability. To keep the mathematical prerequisites to a minimum, we do all the calculations in $\mathbb{R}^n$ and restrict ourselves to drift-free control systems (i.e., control systems whose state remains fixed when the input is turned off). Many of the proofs in this section rely on

some properties of manifolds which we have omitted from the discussion; they can be skipped without loss of continuity. A good introduction to nonlinear control theory which includes many of the necessary differential geometric concepts can be found in Isidori [44] or Nijmeijer and van der Schaft [84].

2.1 Vector fields and flows

We restrict our attention to $\mathbb{R}^n$. We choose to make a distinction, however, between the space and its tangent space at a given point. A point of contact with Chapter 2 is our insistence there on making a distinction between points and vectors in $\mathbb{R}^3$ and enforcing the distinction by augmenting points by 1 and vectors by 0. Denote by $T_q\mathbb{R}^n$ the tangent space to $\mathbb{R}^n$ at a point $q \in \mathbb{R}^n$. A *vector field* on $\mathbb{R}^n$ is a smooth map which assigns to each point $q \in \mathbb{R}^n$ a tangent vector $f(q) \in T_q\mathbb{R}^n$. In local coordinates, we represent f as a column vector whose elements depend on q,

$$f(q) = \begin{bmatrix} f_1(q) \\ \vdots \\ f_n(q) \end{bmatrix}.$$

A vector field is smooth if each $f_i(q)$ is smooth.

Vector fields are to be thought of as right-hand sides of differential equations:

$$\dot{q} = f(q). \tag{7.7}$$

The rate of change of a smooth function $V : \mathbb{R}^n \to \mathbb{R}$ along the flow of f is given by

$$\dot{V} = \frac{\partial V}{\partial q} f(q) = \sum_{i=1}^{n} \frac{\partial V}{\partial q_i} f_i.$$

The time derivative of V along the flow of f is referred to as the *Lie derivative* of V along f and is denoted $L_f V$:

$$L_f V := \frac{\partial V}{\partial q} f(q).$$

Associated with a vector field, we define the *flow* of a vector field to represent the solution of the differential equation (7.7). Specifically, $\phi_t^f(q)$ represents the state of the differential equation at time t starting from q at time 0. Thus $\phi_t^f : \mathbb{R}^n \to \mathbb{R}^n$ satisfies

$$\frac{d}{dt}\phi_t^f(q) = f(\phi_t^f(q)) \qquad q \in \mathbb{R}^n.$$

A vector field is said to be *complete* if its flow is defined for all t. By the existence and uniqueness theorem of ordinary differential equations, for

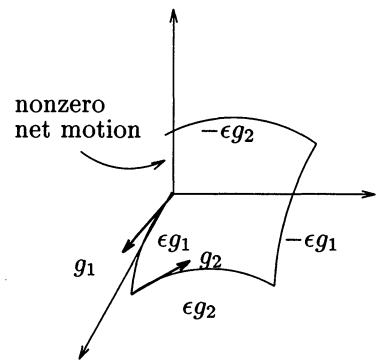


Figure 7.1: A Lie bracket motion.

each fixed t, ϕ_t^f is a local diffeomorphism of $\mathbb{R}^n$ onto itself. Further, it satisfies the following group property:

$$\phi_t^f \circ \phi_s^f = \phi_{t+s}^f,$$

for all t and s, where $\circ$ stands for the composition of the two flows, namely $\phi_t^f(\phi_s^f(q))$.

2.2 Lie brackets and Frobenius' theorem

Given two vector fields g_1 and g_2, the map $\phi_t^{g_1} \circ \phi_s^{g_2}$ stands for the composition of the flow of g_2 for s seconds with the flow of g_1 for t seconds. In general, this quantity is different from the map $\phi_s^{g_2} \circ \phi_t^{g_1}$, which stands for the composition in reverse order. Indeed, consider the flow depicted in Figure 7.1 starting from q_0. It consists of a flow along g_1 for ϵ seconds followed by a flow along g_2 for ϵ seconds, $-g_1$ for ϵ seconds, and $-g_2$ for ϵ seconds. For ϵ small, we may evaluate the Taylor series in ϵ for the state of the differential equation as

$$\begin{aligned} q(\epsilon) &= \phi_\epsilon^{g_1}(q(0)) \\ &= q(0) + \epsilon\dot{q}(0) + \frac{1}{2}\epsilon^2\ddot{q}(0) + O(\epsilon^3) \\ &= q_0 + \epsilon g_1(q_0) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial q} g_1(q_0) + O(\epsilon^3), \end{aligned}$$

where the notation $O(\epsilon^k)$ represents terms of order ϵ^k and the partial derivative of g_1 is evaluated at q_0.

Now evaluating at time 2ϵ,

$$\begin{aligned}
q(2\epsilon) &= \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(q_0) \\
&= \phi_\epsilon^{g_2}(q_0 + \epsilon g_1(q_0) + \frac{\epsilon^2}{2}\frac{\partial g_1}{\partial q} g_1(q_0) + O(\epsilon^3)) \\
&= q_0 + \epsilon g_1(q_0) + \frac{1}{2}\epsilon^2 \frac{\partial g_1}{\partial q} g_1(q_0) \\
&\quad + \epsilon g_2(q_0 + \epsilon g_1(q_0)) + \frac{\epsilon^2}{2}\frac{\partial g_2}{\partial q} g_2(q_0) + O(\epsilon^3) \\
&= q_0 + \epsilon(g_1(q_0) + g_2(q_0)) \\
&\quad + \frac{1}{2}\epsilon^2(\frac{\partial g_1}{\partial q} g_1(q_0) + \frac{\partial g_2}{\partial q} g_2(q_0) + 2\frac{\partial g_2}{\partial q} g_1(q_0)) + O(\epsilon^3).
\end{aligned}$$

Here, we have used the Taylor series expansion for $g_2(q_0 + \epsilon g_1(q)) = g_2(q_0) + \epsilon \frac{\partial g_2}{\partial q} g_1(q_0) + O(\epsilon^2)$. At the next step (we invite the reader to verify this), we get

$$\begin{aligned}
q(3\epsilon) &= \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(q_0) \\
&= q_0 + \epsilon g_2(q_0) \\
&\quad + \frac{\epsilon^2}{2}(\frac{\partial g_2}{\partial q} g_2(q_0) + 2\frac{\partial g_2}{\partial q} g_1(q_0) - 2\frac{\partial g_1}{\partial q} g_2(q_0)) + O(\epsilon^3).
\end{aligned}$$

Finally, we get

$$\begin{aligned}
q(4\epsilon) &= \phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(q_0) \\
&= q_0 + \epsilon^2(\frac{\partial g_2}{\partial q} g_1(q_0) - \frac{\partial g_1}{\partial q} g_2(q_0)) + O(\epsilon^3).
\end{aligned} \tag{7.8}$$

Motivated by this calculation, we define the *Lie bracket* of two vector fields f and g as

$$[f, g](q) = \frac{\partial g}{\partial q} f(q) - \frac{\partial f}{\partial q} g(q).$$

The Lie bracket is thus the infinitesimal motion (actually of order ϵ^2) that results from flowing around a square defined by two vector fields f and g. If $[f, g] = 0$ then it can be shown that the right hand side of equation (7.8) is identically equal to q_0 and f and g are said to *commute*. A *Lie product* is a nested set of Lie brackets, for example,

$$[[f, g], [f, [f, g]]].$$

Example 7.1. Lie brackets of linear vector fields
Consider two linear vector fields given by $f(q) = Aq$ and $g(q) = Bq$. Then the Lie bracket of the two linear vector fields is a linear vector field given by

$$[f, g](q) = (BA - AB)q,$$

that is, it is the *commutator* of the two matrices A, B.

The following properties of Lie brackets follow from the definition. Their proof is left as an exercise.

Proposition 7.1. Properties of Lie brackets
Given vector fields f, g, h on $\mathbb{R}^n$ and smooth functions $\alpha, \beta : \mathbb{R}^n \to \mathbb{R}$, the Lie bracket satisfies the following properties:

1. Skew-symmetry:
$$[f, g] = -[g, f]$$

2. Jacobi identity:
$$[f, [g, h]] + [h, [f, g]] + [g, [h, f]] = 0$$

3. Chain rule:
$$[\alpha f, \beta g] = \alpha\beta[f, g] + \alpha(L_f\beta)g - \beta(L_g\alpha)f,$$
where $L_f\beta$ and $L_g\alpha$ stand for the Lie derivatives of β and α along the vector fields f and g respectively.

An alternative method of defining the Lie bracket of two vector fields f and g is to require that it satisfies for all smooth functions $\alpha : \mathbb{R}^n \to \mathbb{R}$:

$$L_{[f,g]}\alpha = L_f(L_g\alpha) - L_g(L_f\alpha).$$

The reader should carefully parse the previous equation and convince herself of this fact.

A *distribution* assigns a subspace of the tangent space to each point in $\mathbb{R}^n$ in a smooth way. A special case is a distribution defined by a set of smooth vector fields, $g_1, \ldots, g_m$. In this case we define the distribution as

$$\Delta = \operatorname{span}\{g_1, \ldots, g_m\},$$

where we take the span over the set of smooth real-valued functions on $\mathbb{R}^n$. Evaluated at any point $q \in \mathbb{R}^n$, the distribution defines a linear subspace of the tangent space

$$\Delta_q = \operatorname{span}\{g_1(q), \ldots, g_m(q)\} \subset T_q\mathbb{R}^n.$$

The distribution is said to be *regular* if the dimension of the subspace Δ_q does not vary with q. A distribution is *involutive* if it is closed under the Lie bracket, i.e.,

$$\Delta \text{ involutive} \quad \Longleftrightarrow \quad \forall f, g \in \Delta, \quad [f, g] \in \Delta.$$

For a finite dimensional distribution it suffices to check that the Lie brackets of the basis elements are contained in the distribution. The *involutive closure* of a distribution, denoted $\overline{\Delta}$, is the closure of Δ under bracketing; that is, $\overline{\Delta}$ is the smallest distribution containing Δ such that if $f, g \in \overline{\Delta}$ then $[f, g] \in \overline{\Delta}$.

Definition 7.1. Lie algebra
A vector space V (over $\mathbb{R}$) is a *Lie algebra* if there exists a bilinear operation $V \times V \to V$, denoted $[\ ,\ ]$, satisfying (i) skew-symmetry and (ii) the Jacobi identity.

The set of smooth vector fields on $\mathbb{R}^n$ with the Lie bracket is a Lie algebra, and is denoted $\mathfrak{X}(\mathbb{R}^n)$. Let $g_1, \ldots, g_m$ be a set of smooth vector fields, Δ the distribution defined by $g_1, \ldots, g_m$ and, $\overline{\Delta}$ the involutive closure of Δ. Then, $\overline{\Delta}$ is a Lie algebra (in fact the smallest Lie algebra containing $g_1, \ldots, g_m$). It is called the Lie algebra generated by $g_1, \ldots, g_m$ and is often denoted $\mathcal{L}(g_1, \ldots, g_m)$. Elements of $\mathcal{L}(g_1, \ldots, g_m)$ are obtained by taking all linear combinations of elements of $g_1, \ldots, g_m$, taking Lie brackets of these, taking all linear combinations of these, and so on. We define the rank of $\mathcal{L}(g_1, \ldots, g_m)$ at a point $q \in \mathbb{R}^n$ to be the dimension of $\overline{\Delta}_q$ as a distribution.

A distribution Δ of constant dimension k is said to be *integrable* if for every point $q \in \mathbb{R}^n$, there exists a set of smooth functions $h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, n-k$ such that the row vectors $\frac{\partial h_i}{\partial q}$ are linearly independent at q, and for every $f \in \Delta$

$$L_f h_i = \frac{\partial h_i}{\partial q} f(q) = 0 \qquad i = 1, \ldots, n-k. \tag{7.9}$$

The hypersurfaces defined by the level sets

$$\{q : h_1(q) = c_1, \ldots, h_{n-k}(q) = c_{n-k}\}$$

are called *integral manifolds* for the distribution. If we regard an integral manifold as a smooth surface in $\mathbb{R}^n$, then equation (7.9) requires that the distribution be equal to the tangent space of that surface at the point q.

Integral manifolds are related to involutive distributions by the following celebrated theorem.

Theorem 7.2 (Frobenius). *A regular distribution is integrable if and only if it is involutive.*

Thus, if Δ is an k-dimensional involutive distribution, then locally there exist $n-k$ functions $h_i \colon \mathbb{R}^n \to \mathbb{R}$ such that integral manifolds of Δ are given by the level surfaces of $h = (h_1, \ldots, h_{n-k})$. These level surfaces form a *foliation* of $\mathbb{R}^n$. A single level surface is called a *leaf of the foliation.*

Associated with the tangent space $T_q\mathbb{R}^n$ is the dual space $T_q^*\mathbb{R}^n$, the set of linear functions on $T_q\mathbb{R}^n$. Just as we defined vector fields on $\mathbb{R}^n$, we define a *one-form* as a map which assigns to each point $q \in \mathbb{R}^n$ a *covector* $\omega(q) \in T_q^*\mathbb{R}^n$. In local coordinates we represent a smooth one-form as a row vector

$$\omega(q) = \begin{bmatrix} \omega_1(q) & \omega_2(q) & \cdots & \omega_n(q) \end{bmatrix}.$$

Differentials of smooth functions are good examples of one-forms. For example, if $\beta : \mathbb{R}^n \to \mathbb{R}$, then the one-form $d\beta$ is given by

$$d\beta = \begin{bmatrix} \frac{\partial \beta}{\partial q_1} & \frac{\partial \beta}{\partial q_2} & \cdots & \frac{\partial \beta}{\partial q_n} \end{bmatrix}.$$

Note, however, that all one-forms are not necessarily the differentials of smooth functions (a one-form which does happen to be the derivative of a function is said to be *exact*).

A one-form acts on a vector field to give a real-valued function on $\mathbb{R}^n$ by taking the inner product between the row vector ω and the column vector f:

$$\omega \cdot f = \sum_i \omega_i f_i.$$

A *codistribution* assigns a subspace of $T_q^*\mathbb{R}$ smoothly to each $q \in \mathbb{R}^n$. A special case is a codistribution obtained as a span of a set of one-forms,

$$\Omega = \text{span}\{\omega_1, \ldots, \omega_m\},$$

where the span is over the set of smooth functions. As before, the rank of the codistribution is the dimension of Ω_q. The codistribution Ω is said to be regular if its rank is constant.

To begin our study of motion planning for nonholonomic systems, our first task is to convert the specified constraints given as one-forms into an equivalent control system. To this end, consider the problem of constructing a path $q(t) \in \mathbb{R}^n$ between a given q_0 and q_f subject to the constraints

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k.$$

The ω_i's are linear functions on the tangent spaces of $\mathbb{R}^n$, i.e., one-forms. We assume that the ω_i's are smooth and linearly independent over the set of smooth functions. The following proposition is a formalization of the discussion of the introduction.

Proposition 7.3. Distribution annihilating constraints
Given a set of one-forms $\omega_i(q)$, $i = 1, \ldots, k$, there exist smooth, linearly independent vector fields $g_j(q), j = 1, \ldots, n-k$ such that $\omega_i(q) \cdot g_j(q) = 0$ for all i and j.

Proof. The ω_i's form a codistribution of dimension k in $\mathbb{R}^n$. We can choose local coordinates such that the set of one-forms is given by

$$\tilde{\omega}_i = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 & \alpha_{i,k+1} & \cdots & \alpha_{in} \end{bmatrix},$$

where the 1 in the preceding equation is in the i^{th} entry, and the functions $\alpha_{il}: \mathbb{R}^n \to \mathbb{R}$ are smooth functions. Define

$$g_j := \begin{bmatrix} -\alpha_{1,(j+k)} \\ \vdots \\ -\alpha_{k,(j+k)} \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix},$$

where the 1 is in the $j + k^{th}$ entry. The g_j's are linearly independent and annihilate the constraints since

$$\tilde{\omega}_i \cdot g_j = \alpha_{i(j+k)} - \alpha_{i(j+k)} = 0.$$

This shows that $\omega_i \cdot g_j = 0$ for $i = 1, \ldots, k$ and $j = 1, \ldots, n-k$. □

In the language of distributions and codistributions, the results of this proposition are expressed by defining the codistribution

$$\Omega = \text{span}\{\omega_1, \ldots, \omega_k\}$$

and the distribution

$$\Delta = \text{span}\{g_1, \ldots, g_{n-k}\}$$

and stating that

$$\Delta = \Omega^{\perp}.$$

We say that the distribution Δ annihilates the codistribution Ω. The control system associated with the distribution Δ is of the form

$$\dot{q} = g_1(q)u_1 + \cdots + g_{n-k}(q)u_{n-k},$$

with the controls u_i to be freely specified.

These results of this section can be used to determine if a set of Pfaffian constraints is holonomic:

Proposition 7.4. Integrability of Pfaffian constraints
A set of smooth Pfaffian constraints is integrable if and only if the distribution which annihilates the constraints is involutive.

2.3 Nonlinear controllability

In view of Proposition 7.3, which yields a set of vector fields orthogonal to a given set of one-forms, it is clear that the motion planning problem is equivalent to steering a control system. Thus, we will now restrict our attention to control systems of the form

$$\Sigma: \qquad \dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m \qquad \begin{matrix} q \in \mathbb{R}^n \\ u \in U \subset \mathbb{R}^m. \end{matrix} \tag{7.10}$$

This system is said to be *drift-free*, meaning to say that when the controls are set to zero the state of the system does not drift. We assume that the g_j are smooth, linearly independent vector fields on $\mathbb{R}^n$ and that their flows are defined for all time (i.e., the g_j are complete). We wish to determine conditions under which we can steer from $q_0 \in \mathbb{R}^n$ to an arbitrary $q_f \in \mathbb{R}^n$ by appropriate choice of $u(\cdot)$.

A system Σ is controllable if for any $q_0, q_f \in \mathbb{R}^n$ there exists a $T > 0$ and $u\colon [0,T] \to U$ such that Σ satisfies $q(0) = q_0$ and $q(T) = q_f$. A system is said to be *small-time locally controllable* at q_0 if we can reach nearby points in arbitrarily small amounts of time and stay near to q_0 at all times. Given an open set $V \subseteq \mathbb{R}^n$, define $\mathcal{R}^V(q_0, T)$ to be the set of states q such that there exists $u\colon [0,T] \to U$ that steers Σ from $q(0) = q_0$ to $q(T) = q_f$ and satisfies $q(t) \in V$ for $0 \le t \le T$. We also define

$$\mathcal{R}^V(q_0, \le T) = \bigcup_{0<\tau\le T} \mathcal{R}^V(q_0, \tau)$$

to be the set of states reachable up to time T. A system is small-time locally controllable (*locally controllable* for brevity) if $\mathcal{R}^V(q_0, \le T)$ contains a neighborhood of q_0 for all neighborhoods V of q_0 and $T > 0$.

Let $\overline{\Delta} = \mathcal{L}(g_1, \ldots, g_m)$ be the Lie algebra generated by $g_1, \ldots, g_m$. It is referred to as the the *controllability Lie algebra*. From the construction involved in the definition of the Lie bracket in the previous subsection, we saw that by using an input sequence of

$$\begin{array}{llll} u_1 = +1 & u_2 = 0 & \text{for } 0 \le t < \epsilon \\ u_1 = 0 & u_2 = +1 & \text{for } \epsilon \le t < 2\epsilon \\ u_1 = -1 & u_2 = 0 & \text{for } 2\epsilon \le t < 3\epsilon \\ u_1 = 0 & u_2 = -1 & \text{for } 3\epsilon \le t < 4\epsilon, \end{array}$$

we get motion in the direction of the Lie bracket $[g_1, g_2]$. If we were to iterate on this sequence, it should be possible to generate motion along directions given by all the other Lie products associated with the g_i. Thus, it is not surprising that it is possible to steer the system along all of the directions represented in $\mathcal{L}(g_1, \ldots, g_m)$. This is made precise by the following theorem, which was originally proved by W.-L. Chow (in somewhat different form) in the 1940s.

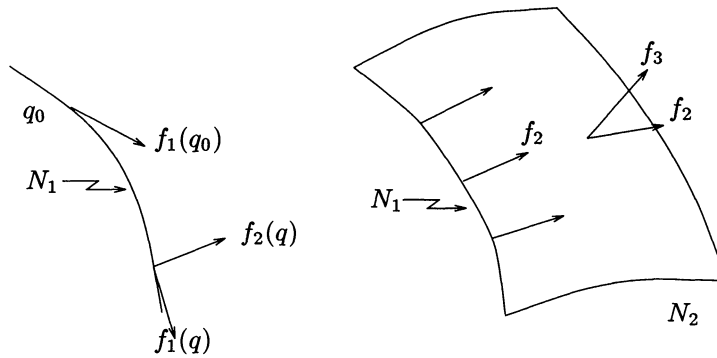


Figure 7.2: Proof of local controllability. At each step we can find a vector field which is not in N_k.

Theorem 7.5 (Chow). *The control system* (7.10) *is locally controllable at* $q \in \mathbb{R}^n$ *if* $\overline{\Delta}_q = T_q\mathbb{R}^n$.

This result asserts that the drift-free system Σ is controllable if the rank of the controllability Lie algebra is n. The condition of Chow's theorem consists of checking the rank of the controllability Lie algebra and is hence referred to as the *controllability rank condition.*

To prove Chow's theorem, we prove the following pair of implications for a given system Σ in a neighborhood of a point q:

$$\overline{\Delta}_q = T_q\mathbb{R}^n \implies \operatorname{int}\mathcal{R}^V(q, \leq T) \neq \{\} \iff \Sigma \text{ is locally controllable,}$$

where $\overline{\Delta} = \mathcal{L}(g_1, \ldots, g_m)$ and $\{\}$ stands for the empty set.

Proposition 7.6. Controllability rank condition
If $\overline{\Delta}_q = T_q\mathbb{R}^n$ *for all* q *in some neighborhood of* q_0, *then for any* $T > 0$ *and neighborhood* V *of* q_0, $\operatorname{int}\mathcal{R}^V(q_0, \leq T)$ *is non-empty.*

Proof. The proof is by recursion. Choose $f_1 \in \Delta$. For $\epsilon_1 > 0$ sufficiently small,

$$N_1 = \{\phi_{t_1}^{f_1}(q_0) : 0 < t_1 < \epsilon_1\}$$

is a smooth surface (manifold) of dimension one which contains points arbitrarily close to q_0. Without loss of generality, take $N_1 \subset V$. Assume $N_k \subset V$ is a k-dimensional manifold. If $k < n$, there exists $q \in N_k$ and $f_{k+1} \in \overline{\Delta}$ such that $f_{k+1} \notin T_qN_k$. If this were not so then $\overline{\Delta}_q \subset T_qN_k$ for any q in some open set $W \subset N_k$, which would imply $\overline{\Delta}|_W \subset TN_k$. This cannot be true since $\dim \overline{\Delta}_q = n > \dim N_k$. For ϵ_{k+1} sufficiently small

$$N_{k+1} = \{\phi_{t_{k+1}}^{f_{k+1}} \circ \cdots \circ \phi_{t_1}^{f_1}(q_0) : 0 < t_i < \epsilon_i,\ i = 1, \cdots, k+1\}$$

is a $k+1$ dimensional manifold. Since ϵ can be made arbitrarily small, we can assume $N_{k+1} \subset V$.

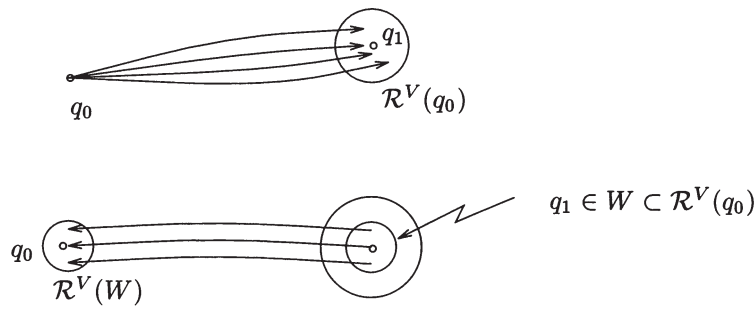


Figure 7.3: Proof of local controllability. To show $\mathcal{R}^V(q_0)$ contains a neighborhood of the origin, we move to any point q_f and map a neighborhood of q_f to a neighborhood of q_0 by reversing our original path.

If $k = n$, $N_k \subset V$ is an n-dimensional manifold and by construction $N_k \subset \mathcal{R}^V(q_0, \leq \epsilon_1 + \cdots + \epsilon_n)$. Hence $\mathcal{R}^V(q_0, \epsilon)$ contains an open set. By restricting each $\epsilon_i \leq T/n$, we can find such an open set for any $T > 0$. This proof is illustrated in Figure 7.2. □

Having established conditions under which the set $\operatorname{int} \mathcal{R}^V(q, \leq T)$ is not empty, we would like to determine if the set can be chosen so as to have q_0 in its interior. This is the subject of the next proposition:

Proposition 7.7. Local controllability
The interior of the set $\mathcal{R}^V(q_0, \leq T)$ is non-empty for all neighborhoods V of q_0 and $T > 0$ if and only if Σ is locally controllable at q_0.

Proof. The sufficiency follows from the definition of locally controllable. To prove necessity, we need to show that $\mathcal{R}^V(q_0, \leq T)$ contains a neighborhood of q_0. Choose a piecewise constant $u\colon [0, T/2] \to U$ such that u steers q_0 to some $q_f \in \mathcal{R}^V(q_0, \leq T/2)$ and $q(t) \in V$. Let ϕ_t^u be the flow corresponding to this input (as given in the proof of the previous theorem). Since Σ is drift-free, we can flow backwards from q_f to q_0 using $u'(t) = -u(T/2 - t)$, $t \in [0, T/2]$. The flow corresponding to u' is $(\phi_t^u)^{-1}$. By continuity of the flow, there exists $W \subset \mathcal{R}^V(q_0, T/2)$ such that $q_f \in W$ and $(\phi_t^u)^{-1}(W) \subset V$ for all t. Furthermore, $(\phi_{T/2}^u)^{-1}(W)$ is a neighborhood of q_0. It follows that $\mathcal{R}^V(q_0, \leq T)$ contains a neighborhood of q_0 since we can concatenate the inputs which steer q_0 to $q_f \in W$ with u' to obtain an open set containing q_0. This is illustrated in Figure 7.3. □

In principle, we now have a recipe for solving the motion planning problem for systems which meet the controllability rank condition. Given an initial point q_0 and final point q_f, find finitely many intermediate via

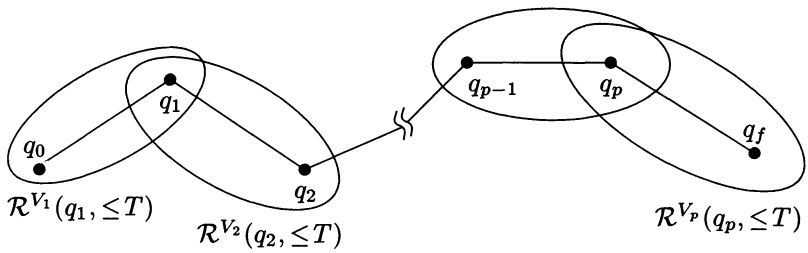


Figure 7.4: Steering between q_0 and q_f.

points $q_1, q_2, \ldots, q_p \in \mathbb{R}^n$ and neighborhoods V_i such that

$$\bigcup_{i=1}^{p} \mathcal{R}^{V_i}(q_i, \leq T)$$

contains the straight line segment connecting q_0 to q_f, as shown in Figure 7.4. Then there exists a control law of p segments which steers from q_0 to q_f. The difficulty with this procedure and the preceding theorems in this section is that they are *non-constructive.* It is in principle possible to solve the motion planning problem for a given set of constraints of the form

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k$$

for arbitrary given q_0 and q_f, provided that the associated control system

$$\dot{q} = g_1(q)u_1 + \cdots + g_{n-k}(q)u_{n-k}$$

has a full rank controllability Lie algebra. However, the preceding theorems do not give a constructive procedure for generating paths for the system joining q_0 and q_f. This *constructive controllability* is the goal of the next chapter.

3 Examples of Nonholonomic Systems

We now present a set of examples of systems with nonholonomic constraints which we will use repeatedly throughout this chapter and the next to illustrate the different concepts. Nonholonomic constraints arise in two kinds of situations:

1. Bodies in contact with each other which *roll without slipping*
2. *Conservation of angular momentum* in a multibody system

An example of the first kind can be found in the problem of dextrous manipulation with a multifingered robot hand. Here the nonholonomic

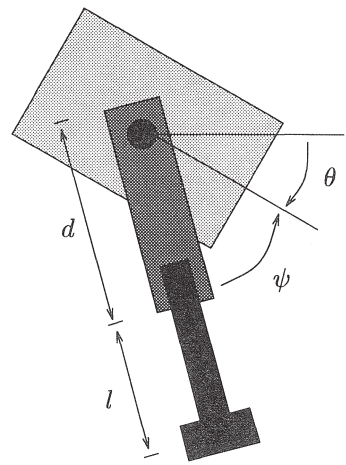


Figure 7.5: A simple hopping robot.

constraint arises from the fingers rolling without slipping on the surface of a grasped object. Other such examples arise in path planning problems for mobile robots or automobiles, where the wheels roll without slipping. For examples of the second kind, we have motion of a satellite with robotic appendages moving in space, where angular momentum is conserved, or a diver or gymnast in mid-air maneuvers.

In the sequel, we will give a description of several nonholonomic systems. The proof of their nonholonomy (that is the impossibility of finding functions of the configuration variables which are "integrals" of the constraints) is deferred to Section 4.

Example 7.2. Hopping robot in flight
As our first example, we consider the dynamics of a hopping robot in the flight phase, as shown in Figure 7.5. This robot consists of a body with an actuated leg that can rotate and extend; the "constraint" on the system is conservation of angular momentum.

The configuration $q = (\psi, l, \theta)$ consists of the leg angle, the leg extension, and the body angle of the robot. We denote the moment of inertia of the body by I and concentrate the mass of the leg, m, at the foot. The upper leg length is taken to be d, with l representing the extension of the leg past this point. The total angular momentum of the robot is given by

$$I\dot{\theta} + m(l+d)^2(\dot{\theta} + \dot{\psi}). \tag{7.11}$$

Assume that the angular momentum of the robot is initially zero. Equation (7.11) is a single Pfaffian constraint in the three velocities $\dot{\psi}$, $\dot{l}$, and $\dot{\theta}$. Thus, the associated control system has two inputs—three configuration

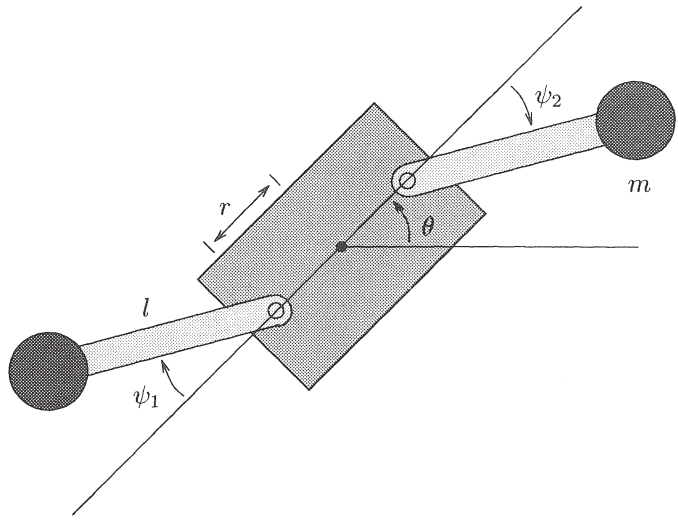


Figure 7.6: A simplified model of a planar space robot.

variables minus one constraint. As a basis for the 2-dimensional right null space of the constraint, we choose one vector field corresponding to controlling the leg angle ψ, and the other corresponding to controlling the leg extension l; i.e., set $\dot{\psi} = u_1$ and $\dot{l} = u_2$. Then, we have

$$g_1(q) = \begin{bmatrix} 1 \\ 0 \\ -\frac{m(l+d)^2}{I+m(l+d)^2} \end{bmatrix} \qquad g_2(q) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

and the equivalent control system is given by

$$\dot{q} = g_1(q)u_1 + g_2(q)u_2.$$

Example 7.3. Planar space robot
Figure 7.6 shows a simplified model of a planar robot consisting of two arms connected to a central body via revolute joints. If the robot is free floating, then the law of conservation of angular momentum implies that moving the arms causes the central body to rotate. In the case that the angular momentum is zero, this conservation law can be viewed as a Pfaffian constraint on the system.

Let M and I represent the mass and inertia of the central body and let m represent the mass of the arms, which we take to be concentrated at the tips. The revolute joints are located a distance r from the middle of the central body and the links attached to these joints have length l. We let (x_1, y_1) and (x_2, y_2) represent the position of the ends of each of the arms (in terms of θ, ψ_1, and ψ_2). Assuming that the body is

free floating in space and that friction is negligible, we can derive the constraints arising from conservation of angular momentum.

Let θ be the angle of the central body with respect to the horizontal, ψ_1 and ψ_2 the angles of the left arm and right arms with respect to the central body, and $p \in \mathbb{R}^2$ the location of a point on the central body (say the center of mass). The kinetic energy of the system has the form

$$\begin{aligned} K &= \frac{1}{2}(M+2m)\|\dot{p}\|^2 + \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}m(\dot{x}_1^2+\dot{y}_1^2) + \frac{1}{2}m(\dot{x}_2^2+\dot{y}_2^2) \\ &= \frac{1}{2}(M+2m)\|\dot{p}\|^2 + \frac{1}{2}\begin{bmatrix}\dot{\psi}_1\\ \dot{\psi}_2\\ \dot{\theta}\end{bmatrix}^T \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{12} & a_{22} & a_{23}\\ a_{13} & a_{23} & a_{33}\end{bmatrix}\begin{bmatrix}\dot{\psi}_1\\ \dot{\psi}_2\\ \dot{\theta}\end{bmatrix}, \end{aligned}$$

where a_{ij} can be calculated as

$$\begin{aligned} a_{11} &= a_{22} = ml^2 \\ a_{12} &= 0 \\ a_{13} &= ml^2 + mr\cos\psi_1 \\ a_{23} &= ml^2 + mr\cos\psi_2 \\ a_{33} &= I + 2ml^2 + 2mr^2 + 2mrl\cos\psi_1 + 2mrl\cos\psi_2. \end{aligned}$$

Note that the kinetic energy of the system is independent of the variable θ. It therefore follows from Lagrange's equations that in the absence of external forces,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \frac{\partial L}{\partial \theta} = 0.$$

Thus the quantity $\frac{\partial L}{\partial \dot{\theta}}$ is a constant of the motion. This is precisely the angular momentum, μ, of the system:

$$\mu = \frac{\partial L}{\partial \dot{\theta}} = a_{13}\dot{\psi}_1 + a_{23}\dot{\psi}_2 + a_{33}\dot{\theta}.$$

If the initial angular momentum is zero, then *conservation of angular momentum* ensures that the angular momentum stays zero, giving the following constraint equation

$$a_{13}(\psi)\dot{\psi}_1 + a_{23}(\psi)\dot{\psi}_2 + a_{33}(\psi)\dot{\theta} = 0. \tag{7.12}$$

Since the variables that are actuated are the hinge angles of the left and right arm, we choose as inputs $u_1 = \dot{\psi}_1$ and $u_2 = \dot{\psi}_2$. Using these in equation (7.12) and setting $q = (\psi_1, \psi_2, \theta)$, we get

$$\dot{q} = g_1(q)u_1 + g_2(q)u_2$$

where

$$g_1(q) = \begin{bmatrix}1\\ 0\\ \frac{-a_{13}}{a_{33}}\end{bmatrix} \qquad g_2(q) = \begin{bmatrix}0\\ 1\\ \frac{-a_{23}}{a_{33}}\end{bmatrix}.$$

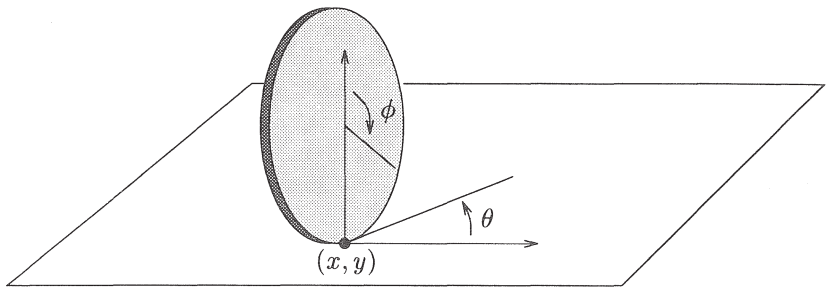


Figure 7.7: Disk rolling on a plane.

Example 7.4. Disk rolling on a plane
Consider the motion of a thin flat disk rolling on a plane shown in Figure 7.7. The configuration space of the system is parameterized by the xy location of the contact point of the disk with the plane, the angle θ that the disk makes with the horizontal line, and the angle ϕ of a fixed line on the disk with respect to the vertical axis. We assume that the disk rolls without slipping. As a consequence we have that

$$\dot{x} - \rho \cos\theta \, \dot{\phi} = 0$$
$$\dot{y} - \rho \sin\theta \, \dot{\phi} = 0,$$

where $\rho > 0$ is the radius of the disk. Writing these equations in the form of Pfaffian constraints with $q = (x, y, \theta, \phi)$ we have

$$\begin{bmatrix} 1 & 0 & 0 & -\rho\cos\theta \\ 0 & 1 & 0 & -\rho\sin\theta \end{bmatrix} \dot{q} = 0.$$

Choosing $\dot{\theta} = u_1$, the rate of rolling, and $\dot{\phi} = u_2$, the rate of turning about the vertical axis, we have the associated control system:

$$\dot{q} = \begin{bmatrix} \rho\cos\theta \\ \rho\sin\theta \\ 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_2. \tag{7.13}$$

Example 7.5. Kinematic car
Consider a simple kinematic model for an automobile with front and rear tires, as shown in Figure 7.8. The rear tires are aligned with the car, while the front tires are allowed to spin about the vertical axes. To simplify the derivation, we model the front and rear pairs of wheels as single wheels at the midpoints of the axles. The constraints on the system arise by allowing the wheels to roll and spin, but not slip.

Let (x, y, θ, ϕ) denote the configuration of the car, parameterized by the xy location of the rear wheel(s), the angle of the car body with

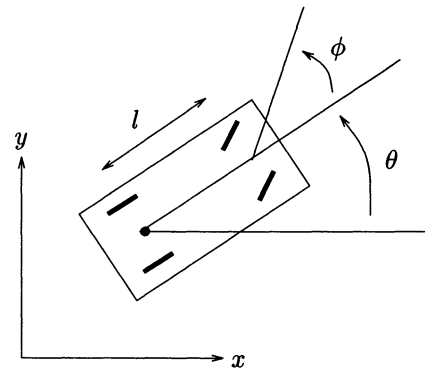


Figure 7.8: Kinematic model of an automobile.

respect to the horizontal, θ, and the steering angle with respect to the car body, ϕ. The constraints for the front and rear wheels are formed by setting the sideways velocity of the wheels to zero. In particular, the velocity of the back wheels perpendicular to their direction is $\sin\theta\dot{x} - \cos\theta\dot{y}$ and the velocity of the front wheels perpendicular to the direction they are pointing is $\sin(\theta+\phi)\dot{x} - \cos(\theta+\phi)\dot{y} - l\dot{\theta}\cos\phi$, so that the Pfaffian constraints on the automobile are:

$$\begin{aligned} \sin(\theta+\phi)\dot{x} - \cos(\theta+\phi)\dot{y} - l\cos\phi\,\dot{\theta} &= 0 \\ \sin\theta\,\dot{x} - \cos\theta\,\dot{y} &= 0. \end{aligned}$$

Converting this to a control system with the inputs chosen as the driving velocity u_1 and the steering velocity u_2 gives

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{l}\tan\phi \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2. \tag{7.14}$$

For this choice of vector fields, u_1 corresponds to the forward velocity of the rear wheels of the car and u_2 corresponds to the velocity of the angle of the steering wheel.

Example 7.6. Fingertip rolling on an object
Let us analyze the motion of a curved fingertip over a curved object. As we discussed in Section 6 of Chapter 5, we parameterize the object surface by $\alpha_o \in \mathbb{R}^2$, the fingertip surface by $\alpha_f \in \mathbb{R}^2$, and the angle of contact by $\psi \in \mathbb{S}^1$, giving a 5-dimensional configuration space. The kinematic

equations of contact are given by

$$\begin{aligned}
\dot{\alpha}_f &= M_f^{-1}(K_f + \tilde{K}_o)^{-1}\left(\begin{bmatrix}-\omega_y\\ \omega_x\end{bmatrix} - \tilde{K}_o\begin{bmatrix}v_x\\ v_y\end{bmatrix}\right)\\
\dot{\alpha}_o &= M_o^{-1}R_\psi(K_f + \tilde{K}_o)^{-1}\left(\begin{bmatrix}-\omega_y\\ \omega_x\end{bmatrix} + K_f\begin{bmatrix}v_x\\ v_y\end{bmatrix}\right)\\
\dot{\psi} &= \omega_z + T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o.
\end{aligned} \tag{7.15}$$

The rolling constraint is obtained by setting the sliding velocity and the velocity of rotation about the contact normal to zero:

$$\begin{bmatrix} v_x\\ v_y\end{bmatrix} = 0 \qquad \omega_z = 0. \tag{7.16}$$

Substituting (7.16) into equation (7.15) yields the following constraints:

$$\begin{aligned}
M_f\dot{\alpha}_f - R_\psi M_o \dot{\alpha}_o &= 0\\
T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o - \dot{\psi} &= 0.
\end{aligned} \tag{7.17}$$

If we set $q = (\alpha_f, \alpha_o, \psi) \in \mathbb{R}^5$, then the foregoing set of three constraints is of the form

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, 2, 3.$$

To obtain a control system associated with these constraints, we let $u_1 = \omega_x$ and $u_2 = \omega_y$ in the kinematic equations for rolling contact. After rearranging the results, we have

$$\dot{q} = \begin{bmatrix} M_f^{-1}\\ M_o^{-1}R_\psi\\ T_f + T_o R_\psi\end{bmatrix}(K_f + \tilde{K}_o)^{-1}\left(\begin{bmatrix}0\\1\end{bmatrix}u_1 + \begin{bmatrix}-1\\0\end{bmatrix}u_2\right). \tag{7.18}$$

We now specialize the example to the case that the object is flat and the fingertip is a sphere of radius one. The curvature forms, metric tensors, and torsions for the fingertip and the object have been derived in Example 5.7 and are reproduced here for convenience:

$$\begin{aligned}
K_o &= \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} \qquad K_f = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\\
M_o &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \qquad M_f = \begin{bmatrix}1 & 0\\ 0 & \cos q_1\end{bmatrix}\\
T_o &= \begin{bmatrix}0 & 0\end{bmatrix} \qquad T_f = \begin{bmatrix}0 & -\tan q_1\end{bmatrix}.
\end{aligned}$$

Substituting the above results into (7.17) gives

$$\begin{bmatrix}1 & 0 & -\cos q_5 & \sin q_5 & 0\\ 0 & \cos q_1 & \sin q_5 & \cos q_5 & 0\\ 0 & \sin q_1 & 0 & 0 & 1\end{bmatrix}\dot{q} = 0.$$

In this case, the formula (7.18) gives, with the inputs being the rates of rolling about the two tangential directions,

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \sec q_1 \\ -\sin q_5 \\ -\cos q_5 \\ -\tan q_1 \end{bmatrix} u_1 + \begin{bmatrix} -1 \\ 0 \\ -\cos q_5 \\ \sin q_5 \\ 0 \end{bmatrix} u_2. \tag{7.19}$$

4 Structure of Nonholonomic Systems

We return to the problem of motion planning for systems satisfying linear velocity constraints of the form

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k.$$

In Section 2 we showed how the problem of finding feasible trajectories in the configuration space could be dualized to one of finding trajectories of the control system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m, \tag{7.20}$$

with $m = n - k$ and $\omega_i(q)g_j(q) = 0$. From the controllability rank condition, it follows that one can find a trajectory joining an arbitrary starting point and end point if the rank of the Lie algebra generated by $g_1, \ldots, g_m$ is n. If $\overline{\Delta}_q \neq T_q\mathbb{R}^n$ and in addition $\overline{\Delta}_q$ has a constant rank $n - p$ which is less than n, then it follows from Frobenius' theorem that there exist functions $h_i(q) = c_i$, $i = 1, \ldots, p$ such that

$$\omega_i(q)\dot{q} = 0 \quad i = 1, \ldots, k \qquad \Longleftrightarrow \qquad h_j(q) = c_j \quad j = 1, \ldots, p.$$

Consider this a little further: since the dimension of $\overline{\Delta}$ is greater than or equal to the dimension of Δ, it follows that $p \leq k$. Thus, the number of functions whose level sets are tangential to the given distribution are fewer than the dimension of the distribution. The process of converting from the given constraints, specified as a codistribution, to an equivalent control system and then integrating the involutive closure of this distribution may seem to be convoluted. It is indeed possible to deal directly with a given codistribution and to find the maximal integrable codistribution contained within it, but this involves methods of exterior differential systems which are beyond the scope of this book. Of course, in the event that $\overline{\Delta} = T_q\mathbb{R}^n$ for all q, then $p = 0$, i.e., there are no nontrivial functions which integrate the given constraints. In this case the distribution is said to be *completely nonholonomic*, as was noted earlier.

In this section, we will try to make precise some notation that we will use in dealing with nonholonomic systems and apply it to the examples

that we considered in Section 3. Some additional machinery to study the growth of the controllability Lie algebra is discussed at the end of this section.

4.1 Classification of nonholonomic distributions

The complexity of the motion planning problem is related to the order of Lie brackets in its controllability Lie algebra. Here we develop some concepts which allow us to classify nonholonomic systems. Let $\Delta = \text{span}\{g_1, \ldots, g_m\}$ be the distribution associated with the control system (7.20). Define $\Delta_1 = \Delta$ and

$$\Delta_i = \Delta_{i-1} + [\Delta_1, \Delta_{i-1}],$$

where

$$[\Delta_1, \Delta_{i-1}] \doteq \text{span}\{[g, h] : g \in \Delta_1, h \in \Delta_{i-1}\}.$$

It is clear that $\Delta_i \subset \Delta_{i+1}$. The chain of the distributions Δ_i is defined as the *filtration* associated with the distribution $\Delta = \Delta_1$. Each Δ_i is defined to be spanned by the input vector fields plus the vector fields formed by taking up to $i-1$ Lie brackets of the generators, i.e., elements of Δ_1. The Jacobi identity (see Proposition 7.1, page 325) implies that $[\Delta_i, \Delta_j] \subset [\Delta_1, \Delta_{i+j-1}] \subset \Delta_{i+j}$. The proof of this fact is left as an exercise.

A filtration is said to be *regular* in a neighborhood U of q_0 if

$$\text{rank } \Delta_i(q) = \text{rank } \Delta_i(q_0) \qquad \forall q \in U.$$

We say the control system (7.20) is *regular* if the corresponding filtration is regular. If a filtration is regular, then at each step of its construction, Δ_i either gains dimension or $\Delta_{i+1} = \Delta_i$, so that the construction terminates. If rank Δ_{i+1} = rank Δ_i, then Δ_i is involutive and hence $\Delta_{i+j} = \Delta_i$ for all $j \geq 0$. Clearly, rank $\Delta_i \leq n$ and hence if a filtration is regular, then there exists an integer $\kappa < n$ such that $\Delta_i = \Delta_\kappa$ for all $i \geq \kappa$.

Definition 7.2. Degree of nonholonomy
Consider a regular filtration $\{\Delta_i\}$ associated with a distribution Δ. The smallest integer κ such that the rank of Δ_κ is equal to that of $\Delta_{\kappa+1}$ is called the *degree of nonholonomy* of the distribution.

We know that rank $\Delta_\kappa \leq n$. In general, let the rank of $\Delta_\kappa = n - p$. Then, by Frobenius' theorem there are p functions h_i for $i = 1, \ldots, p$ whose level surfaces are the integral manifolds of Δ_κ. Thus, the state q of the control system must be confined to the a level set of the h_i's. This, then, is the complete answer to the question we posed ourselves at the beginning of this chapter. The maximum number of functions h_i such that

$$\text{span}\{\frac{\partial h_1}{\partial q}, \ldots, \frac{\partial h_p}{\partial q}\} \quad \subset \quad \text{span}\{\omega_1, \ldots, \omega_k\}$$

is given to be the set of functions such that

$$\text{span}\{\frac{\partial h_1}{\partial q}, \ldots, \frac{\partial h_p}{\partial q}\} = (\overline{\Delta})^{\perp}$$

by Frobenius' theorem. If $p = 0$, that is rank Δ_κ is equal to n, then there are no nontrivial functions h_i and it is possible to steer between arbitrary given initial and final points. This is Chow's theorem, which was discussed in the previous section. Chow's theorem is actually also valid when the filtration Δ_i is not regular, as long as Δ is smooth and constant rank.

We now give a definition which serves to classify the growth of a filtration:

Definition 7.3. Growth vector, relative growth vector
Consider a regular filtration associated with a given distribution Δ and having degree of nonholonomy κ. For such a system, we define the *growth vector* $r \in \mathbb{Z}^\kappa$ as

$$r_i = \text{rank } \Delta_i.$$

We define the *relative growth vector* $\sigma \in \mathbb{Z}^\kappa$ as $\sigma_i = r_i - r_{i-1}$ and $r_0 := 0$.

The growth vector for a regular filtration is a convenient way to represent complexity information about the associated controllability Lie algebra.

4.2 Examples of nonholonomic systems, continued

In this subsection, we illustrate the classification of nonholonomic systems on the examples that were developed in Section 3.

Example 7.7. Hopping robot in flight
Recall from Section 3 that the configuration space for the hopping robot in flight is given by (ψ, l, θ): the leg angle, leg extension, and body angle of the robot. Since we control the leg angle and extension directly, we choose their velocities as our inputs and the control system associated with the hopper is given by

$$\begin{aligned}\dot{\psi} &= u_1 \\ \dot{l} &= u_2 \\ \dot{\theta} &= -\frac{m(l+d)^2}{I+(l+d)^2}u_1.\end{aligned}$$

The controllability Lie algebra is given by

$$g_1 = \begin{bmatrix} 1 \\ 0 \\ -\frac{m(l+d)^2}{I+(l+d)^2} \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad g_3 = [g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ \frac{2Im(l+d)}{(I+m(l+d)^2)^2} \end{bmatrix}.$$

In a neighborhood of $l = 0$, $\text{span}\{g_1, g_2, g_3\}$ is full rank and hence the hopping robot has degree of nonholonomy 2 with growth vector $(2, 3)$ and relative growth vector $(2, 1)$.

Example 7.8. Planar space robot
From Example 7.3, we have that the angular momentum conservation constraint yields

$$a_{13}(\psi)\dot{\theta}_1 + a_{23}(\psi)\dot{\theta}_2 + a_{33}(\psi)\dot{\theta} = 0,$$

where the vector of configuration variables is $q = (\psi_1, \psi_2, \theta)$. Using the control equations derived in Example 7.3, we have

$$g_1 = \begin{bmatrix} 1 \\ 0 \\ -\dfrac{ml^2 + mr\cos\psi_1}{I + 2ml^2 + 2mr^2 + 2mrl\cos\psi_1 + 2mrl\cos\psi_2} \end{bmatrix}$$

$$g_2 = \begin{bmatrix} 1 \\ 0 \\ -\dfrac{ml^2 + mr\cos\psi_2}{I + 2ml^2 + 2mr^2 + 2mrl\cos\psi_1 + 2mrl\cos\psi_2} \end{bmatrix}$$

and the Lie bracket is

$$g_3 = [g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ \dfrac{2m^2l^2r(-l\sin\psi_1 - r\sin(\psi_1 - \psi_2) + l\sin\psi_2)}{(I + 2ml^2 + 2mr^2 + 2mlr\cos\psi_1 + 2mlr\cos\psi_2)^2} \end{bmatrix}.$$

The vector field g_3 is zero when $\psi_1 = \psi_2$ and hence the filtration $\{\Delta_i\}$ is not regular. By computing higher order Lie brackets, however, it is possible to show that $\overline{\Delta}_q = T_q\mathbb{R}^3$ in a neighborhood of $q = 0$ and the system is controllable.

Example 7.9. Disk rolling on a plane
From Example 7.4, the control system which describes a disk rolling on a plane is described by the distribution spanned by

$$g_1 = \begin{bmatrix} \rho\cos\theta \\ \rho\sin\theta \\ 0 \\ 1 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

The control Lie algebra is constructed by computing the following vector fields:

$$g_3 = [g_1, g_2] = \begin{bmatrix} \rho\sin\theta \\ -\rho\cos\theta \\ 0 \\ 0 \end{bmatrix} \qquad g_4 = [g_2, g_3] = \begin{bmatrix} \rho\cos\theta \\ \rho\sin\theta \\ 0 \\ 0 \end{bmatrix}.$$

For all q, $\text{span}\{g_1, g_2, g_3, g_4\}$ is full rank and hence the rolling disk has degree of nonholonomy 3 with growth vector $(2, 3, 4)$. The relative growth vector for this system is $(2, 1, 1)$.

Example 7.10. Kinematic car
Recall that (x, y, θ, ϕ) denotes the configuration of the car, parameterized by the location of the rear wheel(s), the angle of the car body with respect to the horizontal, and the steering angle with respect to the car body. The constraints for the front and rear wheels to roll without slipping are given by the following equations:

$$\sin(\theta + \phi)\dot{x} - \cos(\theta + \phi)\dot{y} - l\cos\phi\,\dot{\theta} = 0$$
$$\sin\theta\,\dot{x} - \cos\theta\,\dot{y} = 0.$$

Converting this to a control system with the driving and steering velocity as inputs gives the control system of equation (7.14).

To calculate the growth vector, we build the filtration

$$g_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{l}\tan\phi \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$g_3 = [g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{l\cos^2\phi} \\ 0 \end{bmatrix} \qquad g_4 = [g_1, g_3] = \begin{bmatrix} -\frac{\sin\theta}{l\cos^2\phi} \\ \frac{\cos\theta}{\cos^2\phi} \\ 0 \\ 0 \end{bmatrix}.$$

The vector fields $\{g_1, g_2, g_3, g_4\}$ are linearly independent when $\phi \neq \pm\pi$. Thus the system has degree of nonholonomy 3 with growth vector $r = (2, 3, 4)$ and relative growth vector $\sigma = (2, 1, 1)$. The system is regular away from $\phi = \pm\pi/2$, at which point g_1 is undefined.

Example 7.11. Spherical finger rolling on a plane
Let the inputs be the two components of rolling velocities, i.e., $u_1 = \omega_x$ and $u_2 = \omega_y$. The associated control system is derived in (7.19), which in vector field form reads

$$g_1 = \begin{bmatrix} 0 \\ \sec q_1 \\ -\sin q_5 \\ -\cos q_5 \\ -\tan q_1 \end{bmatrix} \qquad g_2 = \begin{bmatrix} -1 \\ 0 \\ -\cos q_5 \\ \sin q_5 \\ 0 \end{bmatrix}.$$

Constructing the filtration, we have

$$g_3 = [g_1, g_2] = \begin{bmatrix} 0 \\ \tan q_1 \sec q_1 \\ -\tan q_1 \sin q_5 \\ -\tan q_1 \cos q_5 \\ -\sec^2 q_1 \end{bmatrix} \qquad g_4 = [g_1, g_3] = \begin{bmatrix} 0 \\ 0 \\ -\cos q_5 \\ \sin q_5 \\ 0 \end{bmatrix}$$

$$g_5 = [g_2, g_3] = \begin{bmatrix} 0 \\ -(1+\sin^2 q_1)\sec^3 q_1 \\ 2\sin q_5 \sec^2 q_1 \\ 2\cos q_5 \sec^2 q_1 \\ 2\tan q_1 \sec^2 q_1 \end{bmatrix}.$$

In a neighborhood of $q = 0$ (more specifically in a neighborhood not containing $q_1 = \frac{\pi}{2}$) the vector fields $\{g_1, g_2, g_3, g_4, g_5\}$ are linearly independent, thus establishing that the degree of nonholonomy is 3 and that the growth vector is $(2, 3, 5)$. The relative growth vector is $(2, 1, 2)$.

4.3 Philip Hall basis

Let $\mathcal{L}(g_1, \ldots, g_m)$ be the Lie algebra generated by a set of vector fields $g_1, \ldots, g_m$. One approach to equipping $\mathcal{L}(g_1, \ldots, g_m)$ with a basis is to list all the generators and all of their Lie products. The problem is that not all Lie products are linearly independent because of skew-symmetry and the Jacobi identity. The Philip Hall basis is a particular way to select a basis which takes into account skew-symmetry and the Jacobi identity.

Given a set of generators $\{g_1, \cdots, g_m\}$, we define the length of a Lie product recursively as

$$\begin{aligned} l(g_i) &= 1 \qquad\qquad i = 1, \cdots, m \\ l([A, B]) &= l(A) + l(B), \end{aligned}$$

where A and B are themselves Lie products. Alternatively, $l(A)$ is the number of generators in the expansion for A. A Lie algebra is *nilpotent* if there exists an integer k such that all Lie products of length greater than k are zero. The integer k is called the order of nilpotency.

A *Philip Hall basis* is an ordered set of Lie products $H = \{B_i\}$ satisfying:

1. $g_i \in H, i = 1, \ldots, m$
2. If $l(B_i) < l(B_j)$ then $B_i < B_j$
3. $[B_i, B_j] \in H$ if and only if
   (a) $B_i, B_j \in H$ and $B_i < B_j$ and

(b) either $B_j = g_k$ for some k or $B_j = [B_l, B_r]$ with $B_l, B_r \in H$ and $B_l \leq B_i$.

The proof that a Philip Hall basis is indeed a basis for the Lie algebra generated by $\{g_1, \ldots, g_m\}$ is beyond the scope of this book and may be found in [38] and [105]. A Philip Hall basis which is nilpotent of order k can be constructed from a set of generators using this definition. The simplest approach is to construct all possible Lie products with length less than k and use the definition to eliminate elements which fail to satisfy one of the properties. In practice, the basis can be built in such a way that only condition 3 need be checked.

Example 7.12. Philip Hall basis of order 3
A basis for the nilpotent Lie algebra of order 3 generated by g_1, g_2, g_3 is

g_1	g_2	g_3	
$[g_1, g_2]$	$[g_2, g_3]]$	$[g_3, g_1]$	
$[g_1, [g_1, g_2]]$	$[g_1, [g_1, g_3]]$	$[g_2, [g_1, g_2]]$	$[g_2, [g_1, g_3]]$
$[g_2, [g_2, g_3]]$	$[g_3, [g_1, g_2]]$	$[g_3, [g_1, g_3]]$	$[g_3, [g_2, g_3]]$

Note that $[g_1, [g_2, g_3]]$ does not appear since

$$[g_1, [g_2, g_3]] + [g_2, [g_3, g_1]] + [g_3, [g_1, g_2]] = 0$$

by the Jacobi identity and the second two terms in the formula are already present.

5 Summary

The following are the key concepts covered in this chapter:

1. *Nonholonomic constraints* are linear velocity constraints of the form

$$\omega_i(q)\dot{q} = 0 \quad i = 1, \ldots, k$$

which cannot be integrated to give constraints on the configuration variables q alone. By choosing $g_j(q), j = 1, \ldots, n-k =: m$ to be a basis for the null space of the linear velocity constraints, we get the associated control system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m.$$

The problem of *nonholonomic motion planning* consists of finding a trajectory $q(\cdot) : [0, T] \to \mathbb{R}^n$, given $q(0) = q_0$ and $q(T) = q_f$.

2. The *Lie bracket* between two vector fields f and g on $\mathbb{R}^n$ is a new vector field $[f, g]$ defined by

$$[f, g](q) = \frac{\partial g}{\partial q} f(q) - \frac{\partial f}{\partial q} g(q).$$

3. A *distribution* Δ is a smooth assignment of a subspace of the tangent space to each point $q \in \mathbb{R}^n$. One important way of generating it is as the span of a number of vector fields:

$$\Delta = \text{span}\{g_1, \ldots, g_m\}.$$

The distribution Δ is said to be *regular* if the dimension of Δ_q does not vary with q. The distribution Δ is said to be *involutive* if it is closed under the Lie bracket, that is if for all $f, g \in \Delta$, we have $[f, g] \in \Delta$.

4. A distribution Δ of dimension k is said to be *integrable* if there exist $n - k$ independent functions whose differentials annihilate the distribution. *Frobenius' theorem* asserts that a regular distribution is integrable if and only if it is involutive. A Pfaffian system or codistribution Ω

$$\Omega = \text{span}\{\omega_1, \ldots, \omega_k\}$$

is *completely nonholonomic* if the involutive closure of the distribution $\Delta = \Omega^{\perp}$ spans $\mathbb{R}^n$ for all q.

5. Consider the system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m.$$

The *controllability Lie algebra* is the Lie algebra generated by the vector fields $g_1, \ldots, g_m$. It is the smallest Lie algebra containing $g_1, \ldots, g_m$. *Chow's theorem* asserts that if the controllability Lie algebra is full rank, we can steer this system from any initial to any final point.

6. Given a distribution Δ, the *filtration* associated with Δ is defined by $\Delta_1 = \Delta$ and
$$\Delta_i = \Delta_{i-1} + [\Delta_1, \Delta_{i-1}],$$
where
$$[\Delta_1, \Delta_{i-1}] = \text{span}\{[g, h] : g \in \Delta_1, h \in \Delta_{i-1}\}.$$
The filtration is said to be *regular* if each of the Δ_i are regular. For a regular filtration, the smallest integer κ at which rank Δ_κ is equal to that of $\Delta_{\kappa+1}, \Delta_{\kappa+2}, \ldots$ is called the *degree of nonholonomy* of the distribution. The *growth vector* $r \in \mathbb{Z}^\kappa$ for a regular filtration is defined as $r_i := \text{rank } \Delta_i$. The *relative growth vector* $\sigma \in \mathbb{Z}^\kappa$ is defined as $\sigma_i = r_i - r_{i-1}$ with $r_0 = 0$.

7. Given $\Delta = \text{span}\{g_1, \ldots, g_m\}$, a *Lie product* is any nested set of Lie brackets of the generators g_i. A Lie algebra generated by Δ is said to be *nilpotent* if there exists an integer k such that all Lie products of length greater than k are zero. A *Philip Hall basis* is an ordered set of Lie products chosen by a set of rules so as to keep track of the restrictions imposed by the properties of the Lie bracket, namely skew-symmetry and the Jacobi identity.

6 Bibliography

The topic of holonomy and nonholonomy of Pfaffian constraints has captured the attention of many of the earliest writers on classical mechanics. A nice description of the mechanics point of view is given in [82]. Chapter 1 of Rosenberg [100] makes mention of the different kinds of constraints: holonomic, rheonomic, scleronomic. The examples in this chapter are drawn from our interest in fingers rolling on the surface of an object [61, 77], mobile robots and parking problems [79, 113], and space robots [119, 32]. A recent collection of papers on nonholonomic motion planning is [62].

Work on nonlinear controllability has a long history as well, with recognition of the connections between Chow's theorem and controllability in Hermann and Krener [40]. Good textbook presentations of the work on nonlinear control are available in [44], and [84]. The theory of nonholonomic distributions presented here was originally developed by Vershik and Gershkovic [117]. The notation we follow is theirs and is presented in [79].

A somewhat less obvious application of the methods of this chapter is in the analysis of control algorithms for redundant manipulators. In this application, one looks for an algorithm such that closed trajectories of the end-effector generate closed paths in the joint space of the manipulator. This is closely related to the integrability of a set of constraints. A good description of this is in the work of Shamir and Yomdin [106], Baillieul and Martin [5], Chiacchio and Siciliano [17], and De Luca and Oriolo [23].

7 Exercises

1. Show that the controllability rank condition is also a necessary condition for local controllability under the usual smoothness and regularity assumptions.

2. Show that the differential constraint in $\mathbb{R}^5$ given by

$$\begin{bmatrix} 0 & 1 & \rho \sin q_5 & \rho \cos q_3 & \cos q_5 \end{bmatrix} \dot{q} = 0$$

is nonholonomic.

3. Use the definition of the Lie bracket to prove the properties listed in Proposition 7.1.

4. Consider the system Σ,

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m.$$

Let $u : [0, T] \to \mathbb{R}^m$ be input which steers Σ from q_0 to q_f in T units of time.

(a) Show that the input $\tilde{u} : [0, 1] \to \mathbb{R}^m$ defined by

$$\tilde{u}(t) = u(t/T)$$

steers σ from q_0 to q_f in 1 unit of time.

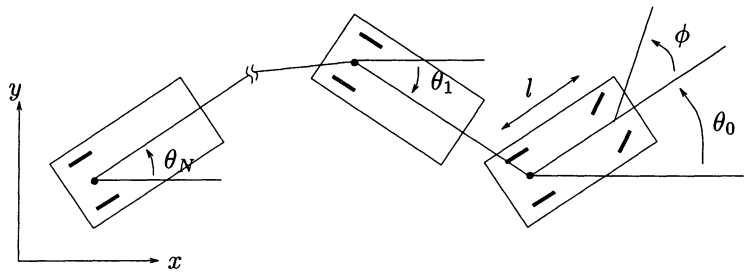
(b) Show that the input $\bar{u} : [0, 1] \longrightarrow \mathbb{R}^m$ defined by

$$\bar{u}(t) = -\tilde{u}(1 - t)$$

steers σ from q_f to q_0 in 1 unit of time.

5. *Spheres rolling on spheres*
Derive the control equation for a unit sphere in rolling contact with another sphere of radius ρ with the same inputs as in Example 7.6. Show that the system is controllable if and only if $\rho \neq 1$.

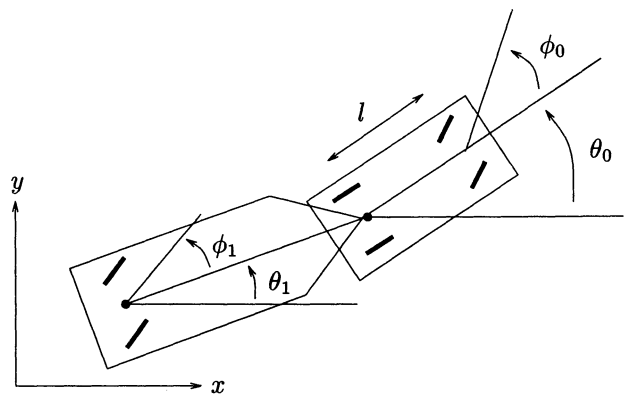
6. *Car with N trailers*
The figure below shows a car with N trailers attached. We attach the hitch of each trailer to the center of the rear axle of the previous trailer. The wheels of the individual trailers are aligned with the body of the trailer. The constraints are again based on allowing the wheels only to roll and spin, but not slip. The dimension of the state space is $N + 4$ with 2 controls.



Parameterize the configuration by the states of the automobile plus the angle of each of the trailers with respect to the horizontal. Show that the control equation for the system has the form

$$\begin{aligned}
\dot{x} &= \cos\theta_0\, u_1 \\
\dot{y} &= \sin\theta_0\, u_1 \\
\dot{\phi} &= u_2 \\
\dot{\theta}_0 &= \frac{1}{l}\tan\phi\, u_1 \\
\dot{\theta}_i &= \frac{1}{d_i}\left(\prod_{j=1}^{i-1}\cos(\theta_{j-1}-\theta_j)\right)\sin(\theta_{i-1}-\theta_i)u_1.
\end{aligned}$$

7. *Firetruck*
A firetruck can be modeled as a car with one trailer, with the difference that the trailer is steerable, as shown in the figure below.



The constraints on the system are similar to that of the car in Section 3, with the difference that back wheels are steerable. Derive the nonlinear control system for a firetruck corresponding to the control inputs for driving the cab and steering both the cab and the trailer, and show that it represents a controllable system.

8. Prove that a 1-dimensional distribution $\Delta_q = \text{span}\{f(q)\}$ is involutive. More specifically, show that for any two smooth functions α and β

$$[\alpha f, \beta f] \in \Delta.$$

9. Prove that the two definitions of Lie bracket given in this chapter, namely,

$$[f, g] = \frac{\partial g}{\partial q} f - \frac{\partial f}{\partial q} g,$$

and

$$L_{[f,g]}\alpha = L_f(L_g\alpha) - L_g(L_f\alpha) \qquad \forall \alpha : \mathbb{R}^n \to \mathbb{R},$$

are equivalent.

10. Use induction and Jacobi's identity to prove that

$$[\Delta_i, \Delta_j] \subset [\Delta_1, \Delta_{i+j-1}] \subset \Delta_{i+j},$$

where $\Delta = \Delta_1 \subset \Delta_2 \subset \cdots$ is a filtration associated with a distribution Δ.

11. Let $\Delta_i, i = 1, \ldots, \kappa$ be a regular filtration associated with a distribution. Show that if $\text{rank}(\Delta_{i+1}) = \text{rank}(\Delta_i)$ then Δ_i is involutive. (Hint: use Exercise 10).

12. *Satellite with 2 rotors*
Figure 7.9 shows a model of a satellite body with two symmetrically attached rotors, where the rotors' axes of rotation intersect at a point. The constraint on the system is conservation of angular momentum.

    (a) Assuming that the initial angular momentum of the system is zero, show that the (body) angular velocity, ω_1, of the satellite body is related to the rotor velocities (u_1, u_2) by

    $$\omega_1 = b_1 u_1 + b_2 u_2 \tag{7.21}$$

    where $b_1, b_2 \in \mathbb{R}^3$ are constant vectors.
    Equation (7.21) gives rise to a differential equation in the rotation group $SO(3)$ for the satellite body

    $$\dot{R}(t) = R(t)(\widehat{b}_1 u_1 + \widehat{b}_2 u_2). \tag{7.22}$$

    (b) Obtain a local coordinate description of (7.22) using the Euler parameters of $SO(3)$ (from Chapter 2) and show that the resulting system is controllable.

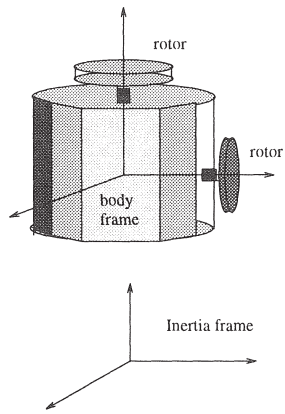
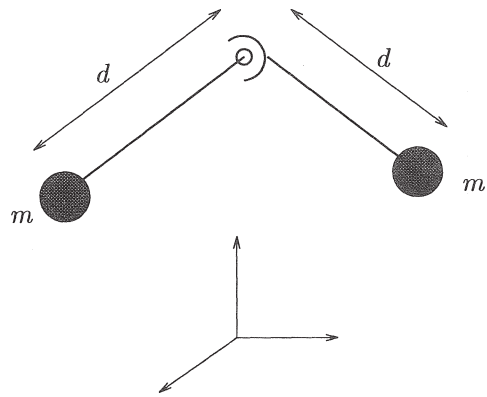


Figure 7.9: A model of a satellite body with two rotors. The satellite can be repositioned by controlling the rotor velocities. (Figure courtesy of Greg Walsh)

13. The figure below shows a simplified model of a *falling cat.* It consists of two pendulums coupled by a spherical joint. The configuration space of the system is $Q = \mathbb{S}^2 \times \mathbb{S}^2$, where $\mathbb{S}^2$ is the unit sphere in $\mathbb{R}^3$.



(a) Derive the Pfaffian constraints arising from conservation of angular momentum and dualize the results to obtain the control system for nonholonomic motion planning.

(b) Is the system in part (a) controllable?

14. Write a computer program to write a Philip Hall basis of given order for a set of m generators $g_1, \ldots, g_m$. Use your program to generate a Philip Hall basis of order 5 for a system with 2 generators.

15. Consider the system of Exercise 6. Write a computer program (or use *Mathematica* or any other symbolic manipulation software packages) to compute the filtration associated with the system. Show that the system is controllable, with degree of nonholonomy $N+2$ and relative growth vector $(2,1,\ldots,1)$.

16. In this chapter, we restricted ourselves to constraints of the form

$$\omega_i(q)\dot{q} = 0 \qquad i = 1,\ldots,k.$$

Consider what would happen if the constraints were of the form

$$\omega_i(q)\dot{q} = c_i \qquad i = 1,\ldots,k$$

for constants c_i. Dualize these constraints to get an associated control system of the form

$$\dot{q} = f(q) + \sum_{i=1}^{n-k} g_i(q)u_i.$$

What is the formula for $f(q)$? Apply this method to the space robot with nonzero angular momentum. What difficulties would one encounter in path planning for these examples? These systems are called systems *with drift.*

17. (Hard) Show that for a regular system the growth vector is bounded above by

$$\overline{\sigma}_i = \frac{1}{i}\left((\overline{\sigma}_1)^i - \sum_{j|i,\, j<i} j\overline{\sigma}_j\right) \qquad i > 1,$$

where $\overline{\sigma}_i$ is the maximum relative growth at the i^{th} stage and $j|i$ means all integers j such that j divides i. If $\sigma_i = \overline{\sigma}_i$ for all i, we say that the system has *maximum growth.*

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 8

Nonholonomic Motion Planning

DOI: 10.1201/9781315136370-8
This chapter has been made available under a CC-BY 4.0 license.

This chapter provides a survey of some of the techniques which have been used in planning paths for nonholonomic systems. This is an area of active research; here, we only provide an introduction to some of the recent techniques that may be applied. In addition to planning paths satisfying the constraints of the form of

$$\omega_i(q)\dot{q} = 0 \qquad i = 1, \ldots, k,$$

one frequently has to make sure that the trajectory $q(\cdot)$ does not intersect obstacles (which are modeled as infeasible regions in the state space). For example, in the case of fingers rolling on the surface of a grasped object, obstacle avoidance may consist of keeping the fingers from colliding with each other or with the grasped object. Conventional (holonomic) path planners implicitly assume that arbitrary motion in the configuration space is allowed as long as obstacles are avoided. If a system contains nonholonomic constraints, many of these path planners cannot be directly applied since they generate paths which violate the constraints. For this reason, it is important to understand how to efficiently compute paths for nonholonomic systems.

1 Introduction

In this section, we introduce the reader to some of the general approaches to nonholonomic motion planning and give an outline of the chapter.

Optimal control

Perhaps the most well-formulated method for finding trajectories of a general control system is the use of optimal control. By attaching a cost functional to each trajectory, we can limit our search to trajectories which minimize a cost function. Typical cost functions might be the length of the path (in some appropriate metric), the control cost, or the time required to execute the trajectory.

If the system has bounds on the magnitudes of the inputs, it makes sense to solve the motion planning problem in *minimum time.* It is well known that for many problems, when the set of allowable inputs is convex, then the time-optimal paths consist of saturating the inputs at all times (this is often referred to as *bang-bang* control). The inputs may change between one set of values and another at a possibly infinite number of *switching times.* Choosing an optimal trajectory is then equivalent to choosing the switching times and the values of the inputs between switching times.

Piecewise constant inputs

Related to the bang-bang trajectories of optimal control, it is also possible to steer nonholonomic systems using piecewise constant inputs. Perhaps the most naive way of using constant inputs is to pick a time interval and generate a graph by applying all possible sequences of inputs (discretized to finitely many values). Each node on the graph corresponds to a configuration, and branches indicate the choice of a fixed control over the time interval. The size of the graph grows as m^d, where m is the number of input combinations considered at each step and d is the number of steps. Since we do not know how long to search, the amount of computer memory required by such an algorithm can be very large. Also, we are likely not to hit our goal exactly, so some post-processing must be done if exact maneuvering is needed.

Recently, a very elegant and general motion planning method using piecewise constant inputs has also been developed by Lafferriere and Sussmann [55]. They consider the case of a *nilpotent* system. Recall from Chapter 7 that a distribution Δ spanned by $g_1, \ldots, g_m$ is nilpotent of order k if all the Lie products with more than k terms vanish. The advantage of nilpotent Lie distributions is that certain computations are greatly simplified, as we shall see in Section 3. The main tool in their method is the Baker-Campbell-Hausdorff formula. If the system is not nilpotent, it can be shown that if the initial and final points are close, then the algorithm of Lafferriere and Sussmann moves the original system closer to the goal by at least half. By breaking the path up into small pieces, we can move arbitrarily close to the goal with repeated applications of the algorithm.

Canonical paths

A third approach to solving the nonholonomic path planning problem is by choosing a family of paths which can be used to produce desired motions. For example, we might consider paths for a car that cause a net rotation of any angle, or a translation in the direction that the car is facing. We can then move to any configuration by reorienting the car, driving forward, and again reorienting the car. The path used to cause a net rotation might consist of a set of parameterized piecewise constant inputs or a heuristic trajectory. The set of canonical paths used for a given problem is usually specific to that problem. In some cases the paths may be derived from some unifying principle. For example, if we could solve the optimal control problem in closed form, these optimal paths would form a set of canonical paths. In the case of time-optimal control, we might consider paths corresponding to saturated inputs as canonical paths, but since it is not clear how to combine these paths to get a specific motion, we distinguish these lower level paths from canonical paths. Canonical paths have been used by Li and Canny to study the motion of a spherical fingertip on an object [61]. This method is discussed in Section 4.

Outline of the chapter

In remainder of this chapter, we will give a bit more detail about some of the various methods of nonholonomic motion planning being pursued in the literature. The sections are not organized according to the different approaches, but from a pedagogical point of view. In Section 2, we study the steering of "model" versions of our nonholonomic system,

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m. \tag{8.1}$$

By model systems we mean those which are in some sense canonical. We use some results from optimal control to explicitly generate optimal inputs for a class of systems. The class of systems which we consider, called first-order control systems, have sinusoids as optimal steering inputs. Motivated by this observation, we explore the use of sinusoidal input signals for steering second- and higher-order model control systems. This study takes us forward to a very important model class which we refer to as chained form systems.

In Section 3, we begin by applying the use of sinusoidal inputs to steering systems which are not in a model form. We do so by using some elementary Fourier analysis in some cases of systems that resemble chained form systems. The question of when a given system can be converted into a chained form system is touched upon as well. Then, we move on to the use of approximations of the optimal input class by the

Ritz approximation technique. Finally, we discuss the use of piecewise constant inputs to solve the general motion planning problem.

In Section 4, we apply the theory developed so far along with some geometric reasoning techniques involving "canonical trajectories" to study repositioning fingers on the surface of a grasped object.

2 Steering Model Control Systems Using Sinusoids

In this section, we study techniques for steering certain "model" control systems: that is, systems which are, in a certain sense, canonical. These systems appear to be superficially unrelated to the examples that we discussed in Chapter 7, but they are in fact quite relevant, as we shall see shortly. The use of sinusoids at integrally related frequencies is motivated by the results of Brockett in the context of optimally steering a class of systems. This section begins with a review of his results for a class of systems whose degree of nonholonomy is two and growth vector is $(m, m(m+1)/2)$. The technique is then extended to certain other classes of model control systems with specific attention being paid to a class of systems referred to as the so-called "chained form" systems. Techniques for steering using methods other than sinusoids and systems other than the model systems considered in this section are deferred to Section 3.

2.1 First-order controllable systems: Brockett's system

By a *first-order controllable* system, we mean a control system of the form

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m,$$

where the vector fields $g_i(q), i = 1, \ldots, m$ and their first-order Lie brackets $[g_j, g_k], j < k, k = 1, \ldots, m$ are linearly independent and furthermore, we have that

$$T_q\mathbb{R}^n = \text{span}\{g_i, [g_j, g_k] : i, j, k = 1, \ldots, m\}.$$

In particular, this implies that $n = m + m(m-1)/2 = m(m+1)/2$. A very important class of model control systems which satisfy this condition was proposed by Brockett [11]. We begin with a discussion of this class for the case that $m = 2$ and $n = m(m+1)/2 = 3$,

$$\begin{aligned} \dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= q_1u_2 - q_2u_1. \end{aligned} \tag{8.2}$$

For this system,

$$g_1 = \begin{bmatrix} 1 \\ 0 \\ -q_2 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 1 \\ q_1 \end{bmatrix} \qquad [g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}.$$

Thus the system is maximally nonholonomic with degree of nonholonomy 2 and growth vector $(2, 3)$. We will consider the problem of steering system (8.2) from $q_0 \in \mathbb{R}^3$ at $t = 0$ to $q_f \in \mathbb{R}^3$ at $t = 1$. In fact, we will do so as to minimize the least squares control cost given by

$$\int_0^1 \|u\|^2 \, dt.$$

Using the fact that $\dot{q}_i = u_i$ for $i = 1, 2$, an equivalent description of the last equation in (8.2) is as a constraint of the form

$$\dot{q}_3 = q_1 \dot{q}_2 - q_2 \dot{q}_1.$$

Similarly, the Lagrangian to be minimized can be written as $\dot{q}_1^2 + \dot{q}_2^2$. Using a Lagrange multiplier $\lambda(t)$ for the constraint, we augment the Lagrangian to be minimized as follows:

$$L(q, \dot{q}) = (\dot{q}_1^2 + \dot{q}_2^2) + \lambda(\dot{q}_3 - q_1 \dot{q}_2 + q_2 \dot{q}_1).$$

The method for minimizing this constrained Lagrangian is to use the classical calculus of variations for the Lagrangian $L(q, \dot{q})$ above, with the control system written as a constraint with a Lagrange multiplier (the reader wishing to learn more about optimal control may consult one of several nice books on the subject, such as [123]). There it is shown that stationary solutions satisfy the Euler-Lagrange equations. The Lagrange multiplier $\lambda(t)$ is determined using the form of the constraint. The fact that the equations are precisely the Euler-Lagrange equations of dynamics should come as no surprise when one considers that the dynamical equations may be derived from a least-action principle. The Euler-Lagrange equations for minimizing the Lagrangian of our optimal control problem are

$$\frac{d}{dt}\left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i}\right) - \frac{\partial L(q, \dot{q})}{\partial q_i} = 0,$$

or equivalently,

$$\begin{aligned} \ddot{q}_1 + \lambda \dot{q}_2 &= 0 \\ \ddot{q}_2 - \lambda \dot{q}_1 &= 0 \\ \dot{\lambda} &= 0. \end{aligned} \tag{8.3}$$

Equation (8.3) establishes that $\lambda(t)$ is constant and, in fact, that the optimal inputs satisfy the equations:

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\lambda \\ \lambda & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} := \Lambda \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

Note that the matrix Λ is skew-symmetric with λ constant, so that the optimal inputs are sinusoids at frequency λ; thus,

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \cos\lambda t & -\sin\lambda t \\ \sin\lambda t & \cos\lambda t \end{bmatrix} \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} := e^{\Lambda t} u(0).$$

Having established this functional form of the optimal controls, given values for q_0 and q_f one can solve for the $u(0)$ and λ required to steer the system optimally. However, from the form of the control system (8.2), it is clear that the states q_1 and q_2 may be steered directly. Thus, it is of greatest interest to steer from $q(0) = (0,0,0)$ to $q(1) = (0,0,a)$. By directly integrating for q_1 and q_2 we have that

$$\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = (e^{\Lambda t} - I)\Lambda^{-1} u(0).$$

Since $q_1(1) = q_2(1) = 0$, it follows that $e^{\Lambda} = I$ so that $\lambda = 2n\pi$, $n = 0, \pm 1, \pm 2, \ldots$. Integrating $\dot{q}_3$ yields

$$q_3(1) = \int_0^1 (q_1 u_2 - q_2 u_1)\, dt = -\frac{1}{\lambda}\left(u_1^2(0) + u_2^2(0)\right) = a.$$

Further, the total cost is

$$\int_0^1 \|u\|^2\, dt = \|u(0)\|^2 = -\lambda a.$$

Since $\lambda = 2n\pi$, it follows that the minimum cost is achieved for $n = -1$ and that $\|u(0)\|^2 = 2\pi a$. However, apart from its magnitude, the direction of $u(0) \in \mathbb{R}^2$ is arbitrary. Thus, the optimal input steering the system between the points $(0,0,0)$ and $(0,0,a)$ is a sum of sines and cosines at a frequency 2π (more generally $\frac{2\pi}{T}$ if the time period of the steering is T).

The generalization of the system (8.2) to an m-input system is the system

$$\begin{aligned} \dot{q}_i &= u_i & i &= 1, \ldots, m \\ \dot{q}_{ij} &= q_i u_j - q_j u_i & i &< j = 1, \ldots, m. \end{aligned} \tag{8.4}$$

A slightly more pleasing form of this equation is obtained by forming a skew-symmetric matrix $Y \in so(m)$ with the $-q_{ij}$ as the bottom lower half (below the diagonal) to give a control system in $\mathbb{R}^m \times so(m)$:

$$\begin{aligned} \dot{q} &= u \\ \dot{Y} &= qu^T - uq^T. \end{aligned} \tag{8.5}$$

The Euler-Lagrange equations for this system are an extension of those for the two input case, namely:

$$\ddot{q} - \Lambda\dot{q} = 0$$
$$\dot{\Lambda} = 0,$$

where $\Lambda \in so(m)$ is the skew-symmetric matrix of Lagrange multipliers associated with Y. As before, Λ is constant and the optimal input satisfies

$$\dot{u} = \Lambda u,$$

so that

$$u(t) = e^{\Lambda t}u(0).$$

It follows that $e^{\Lambda t} \in SO(m)$. It is of special interest to determine the nature of the input when $q(0) = q(1) = 0$, $Y(0) = 0$, and $Y(1)$ is a given matrix in $so(m)$. In this context, an amazing fact that has been shown by Brockett [11] is that when m is even and Y is non-singular, the input has $m/2$ sinusoids at frequencies

$$2\pi, 2 \cdot 2\pi, \ldots, m/2 \cdot 2\pi.$$

If m is odd, then Y is of necessity singular, but if it is of rank $m - 1$, then the input has $(m - 1)/2$ sinusoids at frequencies

$$2\pi, 2 \cdot 2\pi, \ldots, (m - 1)/2 \cdot 2\pi.$$

While the proof of this fact is somewhat involved and would take us afield from what we would like to highlight in this section, we may use the fact to propose the following algorithm for steering systems of the form of (8.4):

Algorithm 1. Steering first-order canonical systems

1. Steer the q_i to their desired values using any input and ignoring the evolution of the q_{ij}.

2. Using sinusoids at integrally related frequencies, find u_0 such that the input steers the q_{ij} to their desired values. By the choice of input, the q_i are unchanged.

The algorithm involves steering the states step by step. The states that are directly controlled (zeroth order) are steered first and then the first Lie bracket directions are steered.

2.2 Second-order controllable systems

Consider systems in which the first level of Lie bracketing is not enough to span $T_q\mathbb{R}^n$. We begin by trying to extend the previous canonical form to the next higher level of bracketing:

$$\begin{aligned} \dot{q}_i &= u_i && i = 1, \ldots, m \\ \dot{q}_{ij} &= q_i u_j && 1 \le i < j \le m \\ \dot{q}_{ijk} &= q_{ij} u_k && 1 \le i, j, k \le m \quad \text{(mod Jacobi identity).} \end{aligned} \tag{8.6}$$

Because the Jacobi identity imposes relationships between Lie brackets of the form

$$[g_i, [g_j, g_k]] + [g_k, [g_i, g_j]] + [g_j, [g_k, g_i]] = 0$$

for all i, j, k, it follows that not all state variables of the form of q_{ijk} are controllable. For this reason, we refer to the last of the preceding equations as "mod Jacobi identity". Indeed, a straightforward but somewhat laborious computation shows that

$$q_{231} - q_{132} = q_1 q_{23} - q_2 q_{13}.$$

From Exercise 17 of Chapter 7, it may be verified that the maximum number of controllable q_{ijk} is

$$\frac{(m+1)m(m-1)}{3}.$$

Constructing the Lagrangian with the same integral cost criterion as before and deriving the Euler-Lagrange equations does not, in general, result in a constant set of Lagrange multipliers. For the case of $m = 2$, Brockett and Dai [14] have shown that the optimal inputs are elliptic functions (see also the next section). However, we can extend Algorithm 1 to this case as follows:

Algorithm 2. Steering second-order model systems

1. Steer the q_i to their desired values. This causes drift in all the other states.

2. Steer the q_{ij} to their desired values using integrally related sinusoidal inputs. If the i^{th} input has frequency ω_i, then q_{ij} will have frequency components at $\omega_i \pm \omega_j$. By choosing inputs such that we get frequency components at zero, we can generate net motion in the desired states.

3. Use sinusoidal inputs a second time to move all the previously steered states in a closed loop and generate net motion only in the q_{ijk} direction. This requires careful selection of the input frequencies such that $\omega_i \pm \omega_j \neq 0$ but $\omega_i + \omega_j + \omega_k$ has zero frequency components.

The required calculations for Step 2 above are identical to those in Algorithm 1. A general calculation of the motion in Step 3 is quite cumbersome, although for specific systems of practical interest the calculations are quite straightforward. For example, if $m = 2$ equation (8.6) becomes

$$\begin{aligned}\dot{q}_1 &= u_1\\ \dot{q}_2 &= u_2\\ \dot{q}_{12} &= q_1 u_2\\ \dot{q}_{121} &= q_{12} u_1\\ \dot{q}_{122} &= q_{12} u_2.\end{aligned}$$

To steer q_1, q_2, and q_{12} to their desired locations, we apply Algorithm 1. To steer q_{121} independently of the other states, choose $u_1 = a \sin 2\pi t$, $u_2 = b \cos 4\pi t$ to obtain

$$q_{121}(1) = q_{121}(0) - \frac{a^2 b}{16\pi^2}.$$

Similarly, choosing $u_1 = b \cos 4\pi t, u_2 = a \sin 2\pi t$ gives

$$q_{122}(1) = q_{122}(0) + \frac{a^2 b}{32\pi^2}$$

and all the other states return to their original values.

Both the algorithms presented above require separate steps to steer in each of the q_{ijk} directions. It is also possible to generate net motion in multiple coordinates simultaneously by using a linear combination of sinusoids and by solving a polynomial equation for the necessary coefficients (see Exercise 4).

2.3 Higher-order systems: chained form systems

We now study more general examples of nonholonomic systems and investigate the use of sinusoids for steering such systems. As in the previous section, we may try to generate canonical classes of higher order systems, i.e., systems where more than one level of Lie brackets is needed to span the tangent space to the configuration space. Such a development is given by Grayson and Grossmann [37], and in [79] we showed that, in full generality, it is difficult to use sinusoids to steer such systems. This leads us to specialize to a smaller class of higher order systems, which we refer to as chained systems, that can be steered using sinusoids at integrally related frequencies. These systems are interesting in their own right as well, since they are duals of a classical construction in the literature on differential forms referred to as the Goursat normal form. Further, we can convert many other nonlinear systems into chained form systems as we discuss in the next section.

Consider a two-input system of the following form:

$$\begin{aligned} \dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= q_2 u_1 \\ \dot{q}_4 &= q_3 u_1 \\ &\vdots \\ \dot{q}_n &= q_{n-1} u_1. \end{aligned} \tag{8.7}$$

In vector field form, equation (8.7) becomes

$$\dot{q} = g_1 u_1 + g_2 u_2$$

with

$$g_1 = \begin{bmatrix} 1 \\ 0 \\ q_2 \\ q_3 \\ \vdots \\ q_{n-1} \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{8.8}$$

We define the system (8.7) as a *one-chain system.* The first item is to check the controllability of these systems. To this end, denote iterated Lie products as $\mathrm{ad}_{g_1}^k g_2$, defined by:

$$\mathrm{ad}_{g_1} g_2 = [g_1, g_2], \qquad \mathrm{ad}_{g_1}^k g_2 = [g_1, \mathrm{ad}_{g_1}^{k-1} g_2] = [g_1, [g_1, \ldots, [g_1, g_2] \ldots]]$$

Lemma 8.1. Lie bracket calculations
For the vector fields in equation (8.8)*, with* $k \geq 1$

$$ad_{g_1}^k g_2 = \begin{bmatrix} 0 \\ \vdots \\ (-1)^k \\ \vdots \\ 0 \end{bmatrix}.$$

(Here the only non-zero entry is in the $(k+2)^{th}$ *entry.)*

Proof. By induction. Since the first level of brackets is irregular, we begin

by expanding $[g_1, g_2]$ and $[g_1, [g_1, g_2]]$ to get

$$[g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad [g_1, [g_1, g_2]] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Now assume that the formula is true for k. Then

$$\mathrm{ad}_{g_1}^{k+1} g_2 = [g_1, \mathrm{ad}_{g_1}^{k} g_2] = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (-1)^{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

□

Proposition 8.2. Controllability of the one-chain system
The one-chain system (8.7) *is completely nonholonomic (controllable).*

Proof. There are n coordinates in (8.7) and the n Lie products

$$\{g_1, g_2, \mathrm{ad}_{g_1}^{i} g_2\} \qquad 1 \leq i \leq n-2$$

are independent using Lemma 8.1. □

To steer this system, we use sinusoids at integrally related frequencies. Roughly speaking, if we use $u_1 = \sin 2\pi t$ and $u_2 = \cos 2\pi k t$ then $\dot{q}_3$ will have components at frequency $2\pi(k-1)$, $\dot{q}_4$ at frequency $2\pi(k-2)$, etc. $\dot{q}_{k+2}$ will have a component at frequency zero and when integrated gives motion in q_{k+2} while all previous variables return to their starting values.

Algorithm 3. Steering chained form systems

1. Steer q_1 and q_2 to their desired values.
2. For each q_{k+2}, $k \geq 1$, steer q_k to its final value using $u_1 = a \sin 2\pi t$, $u_2 = b \cos 2\pi k t$, where a and b satisfy

$$q_{k+2}(1) - q_{k+2}(0) = \left(\frac{a}{4\pi}\right)^k \frac{b}{k!}.$$

Proposition 8.3. Chained form algorithm
Algorithm 3 can steer (8.7) *to an arbitrary configuration.*

Proof. The proof is constructive. We first show that using $u_1 = a \sin 2\pi t$, $u_2 = b \cos 2\pi kt$ produces motion only in q_{k+2} and not in q_j, $j < k+2$ after one period, by direct integration. If q_{k-1} has terms at frequency $2\pi n_i$, then q_k has corresponding terms at $2\pi(n_i \pm 1)$ (by expanding products of sinusoids as sums of sinusoids). Since the only way to have $q_i(1) \neq q_i(0)$ is to have q_i have a component at frequency zero, it suffices to keep track only of the lowest frequency component in each variable; higher components will integrate to zero. Direct computation starting from the origin yields

$$
\begin{aligned}
q_1 &= \frac{a}{2\pi}(1 - \cos 2\pi t), \\
q_2 &= \frac{b}{2\pi k} \sin 2\pi kt \\
q_3 &= \int \frac{ab}{2\pi k} \sin 2\pi kt \sin 2\pi t \, dt \\
&= \frac{1}{2} \frac{ab}{2\pi k} \left(\frac{\sin 2\pi(k-1)t}{2\pi(k-1)} - \frac{\sin 2\pi(k+1)t}{2\pi(k+1)} \right) \\
q_4 &= \frac{1}{2} \frac{a^2 b}{2\pi k \cdot 2\pi(k-1)} \int \sin 2\pi(k-1)t \cdot \sin 2\pi t \, dt + \cdots \\
&= \frac{1}{2^2} \frac{a^2 b}{2\pi k \cdot 2\pi(k-1) \cdot 2\pi(k-2)} \sin 2\pi(k-2)t + \cdots \\
&\vdots \\
q_{k+2} &= \int \frac{1}{2^{k-1}} \frac{a^k b}{2\pi k \cdot 2\pi(k-1) \cdots 2\pi} \sin^2 2\pi t \, dt + \cdots \\
&= \frac{1}{2^{k-1}} \frac{a^k b}{(2\pi)^k k!} \frac{t}{2} + \cdots
\end{aligned}
$$

It follows that $q_{k+2}(1) = q_{k+2}(0) + \left(\frac{a}{4\pi}\right)^k \frac{b}{k!}$ and all earlier q_i's are periodic and hence $q_i(1) = q_i(0)$, $i < k$. If the system does not start at the origin, the initial conditions generate extra terms of the form $q_{i-1}(0)u_1$ in the i^{th} derivative and this integrates to zero, giving no net contribution. □

For the case of systems with more than two inputs, and to the so-called *multi-chained* form systems, we refer the reader to [79].

3 General Methods for Steering

Model control systems of the kind that we discussed in the previous section will very seldom show up verbatim in applications. In this section, we consider some techniques in motion planning for more general nonholonomic systems.

3.1 Fourier techniques

The methods involving sinusoids at integrally related frequencies can be modified using some elementary Fourier analysis to steer systems which are not in any of the model classes that we have discussed. We will illustrate these notions on two examples that we studied in Chapter 7.

Example 8.1. Steering the hopping robot in flight
We saw in Chapter 7 that ψ was the angle of the hip of the hopping robot in the flight phase, l the length of the leg extension, and θ the angle of the body of the robot. The control equations are given by

$$\begin{aligned} \dot{\psi} &= u_1 \\ \dot{l} &= u_2 \\ \dot{\theta} &= -\frac{m(l+d)^2}{I+m(l+d)^2}u_1. \end{aligned} \tag{8.9}$$

Expanding the right-hand side of the last equation of (8.9) in a Taylor series about $l = 0$, we get

$$\dot{\theta} = -\frac{md^2}{md^2+I}\dot{\psi} - \frac{2mdI}{(md^2+I)^2}lu_1 + O(l^2)u_1$$

where $O(l^2)$ stands for quadratic and higher-order terms in l. This suggests a change of coordinates of the form $\alpha = \theta + \frac{md^2}{md^2+I}\psi$ to put the equations in the form

$$\begin{aligned} \dot{\psi} &= u_1 \\ \dot{l} &= u_2 \\ \dot{\alpha} &= -\frac{2mdI}{(md^2+I)^2}lu_1 + O(l^2)u_1 := f(l)u_1. \end{aligned}$$

If one neglects the higher order terms in the last equation, this equation has the form of a chained form system with 3 states. Using this as justification, we attempt to use the algorithm for steering chained form systems to steer the precise form of this system. We first steer the ψ and l variables to their desired values. Then, we use sinusoids

$$\begin{aligned} u_1 &= a_1 \sin 2\pi t \\ u_2 &= a_2 \cos 2\pi t \end{aligned}$$

to steer α. By choice, after one period (1 second), the last motion does not affect the final values of ψ and l. Since $l = \frac{a_2}{2\pi}\sin 2\pi t$ over this piece of the motion, we can expand $f(l)$ by its Fourier series as

$$f(\frac{a_2}{2\pi}\sin 2\pi t) = \beta_1 \sin 2\pi t + \beta_2 \sin 4\pi t + \cdots$$

Integrating $\dot{\alpha}$ over one period and noting that only the first term contributes to the net motion, yields

$$\begin{aligned}\alpha(1) - \alpha(0) &= \int_0^1 (a_1\beta_1 \sin^2 2\pi t + a_1\beta_2 \sin 2\pi t \sin 4\pi t + \cdots)\, dt \\ &= \frac{1}{2} a_1 \beta_1.\end{aligned}$$

Since β_1 is a function of a_2, one can solve for a_1, a_2 numerically to achieve a net change in α. Cartwheeling in mid-air consists of a net change in phase of 2π radians.

Example 8.2. Steering the kinematic car
The equations for the kinematic model of a front wheel drive car are given by (7.14), namely

$$\begin{aligned}\dot{x} &= \cos\theta\, u_1 \\ \dot{y} &= \sin\theta\, u_1 \\ \dot{\theta} &= \frac{1}{l} \tan\phi\, u_1 \\ \dot{\phi} &= u_2.\end{aligned}$$

In this form, u_1 does not control any state directly. We use a change of coordinates $z_1 = x$, $z_2 = \phi$, $z_3 = \sin\theta$, $z_4 = y$, and a change of inputs $v_1 = \cos\theta\, u_1$, $v_2 = u_2$ to put the equations in the form

$$\begin{aligned}\dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= \frac{1}{l} \tan z_2 v_1 \\ \dot{z}_4 &= \frac{z_3}{\sqrt{1 - z_3^2}} v_1.\end{aligned}$$

As in the previous example, the linear terms in the Taylor series expansions of the nonlinearities in the last two equations match the terms of the one-chain system, and we can include the effect of the nonlinear terms using Fourier analysis.

An example of the application of Algorithm 2 applied to the car is given in Figure 8.1. The first part of the path, labeled A, drives x and ϕ to their desired values using a constant input. The second portion labeled B, uses a sine and cosine to drive θ while bringing the other two states back to their desired values. The last step labeled C, involving the inputs

$$u_1 = a_1 \sin 4\pi t \qquad u_2 = a_2 \sin 2\pi t,$$

brings y to the desired value and returns the other states back to their correct values. The Lissajous figures obtained from the phase portraits of

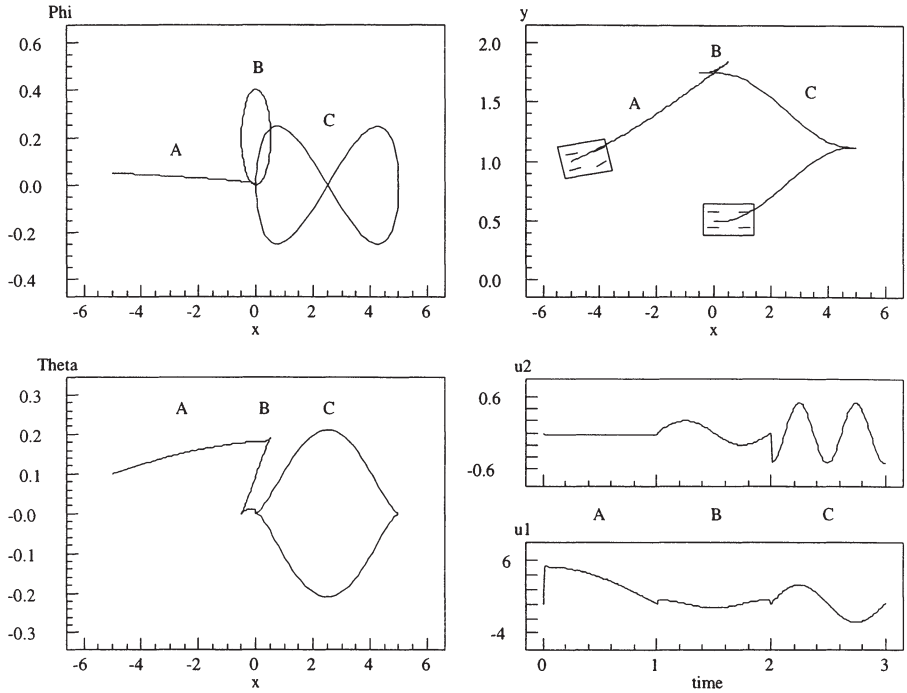


Figure 8.1: Sample trajectories for the motion of a car. The trajectory shown is a sample path which moves the car from $(x, y, \theta, \phi) = (-5, 1, 0.05, 1)$ to $(0, 0.5, 0, 0)$. The first three figures show the states versus x, the bottom right graphs show the inputs as functions of time.

the different variables are quite instructive. Consider the part of the curve labeled C. The upper left plot contains the Lissajous figure for x, ϕ (two loops); the lower left plot is the corresponding figure for x, θ (one loop); and the open curve in x, y shows the increment in the y variable. The interesting implication here is that the *Lie bracket motions correspond to rectification of harmonic periodic motions of the driving vector fields, and the harmonic relations are determined by the order of the Lie bracket corresponding to the desired direction of motion.*

3.2 Conversion to chained form

An interesting question to ask is whether it is possible using a change of input and nonlinear transformation of the coordinates to convert a given nonholonomic control system into one of the model forms discussed in the previous section. More precisely, given the system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m,$$

does there exist a matrix $\beta(q) \in \mathbb{R}^{m\times m}$ and a diffeomorphism $\phi : \mathbb{R}^n \to \mathbb{R}^n$ such that with

$$v = \beta(q)u \qquad z = \phi(q),$$

the system is in chained form in the z coordinates with inputs v? One can give necessary and sufficient conditions to solve this problem (see [75]), but the discussion of these conditions would take us into far too much detail for this section. In [79], we gave sufficient conditions for the two-input case, in which case the system was to be converted into the one-chain form. The conditions assume that g_1, g_2 are linearly independent. Now, the system can be converted into one-chain form if the following distributions are all of constant rank and involutive:

$$\begin{aligned}
\Delta_0 &= \operatorname{span}\{g_1, g_2, \operatorname{ad}_{g_1} g_2, \ldots, \operatorname{ad}_{g_1}^{n-2} g_1\} \\
\Delta_1 &= \operatorname{span}\{g_2, \operatorname{ad}_{g_1} g_2, \ldots, \operatorname{ad}_{g_1}^{n-2} g_2\} \\
\Delta_2 &= \operatorname{span}\{g_2, \operatorname{ad}_{g_1} g_2, \ldots, \operatorname{ad}_{g_1}^{n-3} g_2\}
\end{aligned}$$

and there exists a function $h_1(q)$ such that

$$dh_1 \cdot \Delta_1 = 0 \quad \text{and} \quad dh_1 \cdot g_1 = 1.$$

If these conditions are met, then a function h_2 independent of h_1 may be chosen so that

$$dh_2 \cdot \Delta_2 = 0.$$

The existence of independent h_1 and h_2 so that $dh_1 \cdot \Delta_1 = 0, dh_2 \cdot \Delta_2 = 0$ is guaranteed by Frobenius' theorem, since $\Delta_2 \subset \Delta_1$ are both involutive distributions. There is however an added condition on h_1, namely that $dh_1 \cdot g_1 = 1$. If we can find these functions h_1, h_2, then the map $\phi : q \mapsto z$ and input transformation given by

$$\begin{aligned}
z_1 &= h_1 & v_1 &:= u_1 \\
z_2 &= L_{g_1}^{n-2} h_2 & v_2 &:= (L_{g_1}^{n-1} h_2)u_1 + (L_{g_2} L_{g_1}^{n-2} h_2)u_2 \\
&\vdots \\
z_{n-1} &= L_{g_1} h_2 \\
z_n &= h_2
\end{aligned}$$

yields

$$\begin{aligned}
\dot{z}_1 &= v_1 \\
\dot{z}_2 &= v_2 \\
\dot{z}_3 &= z_2 v_1 \\
&\vdots \\
\dot{z}_n &= z_{n-1} v_1.
\end{aligned}$$

This procedure can be illustrated on the kinematic model of a car.

Example 8.3. Conversion of the kinematic car into chained form First, we rewrite the kinematic equations as

$$\begin{aligned}\dot{x} &= u_1\\ \dot{y} &= \tan\theta\, u_1\\ \dot{\theta} &= \frac{1}{l}\tan\phi\, u_1\\ \dot{\phi} &= u_2.\end{aligned}$$

Then, with $h_1 = x$ and $h_2 = y$, it is easy to verify that this system satisfies the conditions given above and the change of variables and input are given by

$$\begin{aligned} z_1 &= x & u_1 &= v_1\\ z_2 &= \frac{1}{l}\sec^2\theta\tan\phi & u_2 &= -\frac{2}{l}\sin^2\phi\tan\theta v_1 + l\cos^2\theta\cos^2\phi v_2\\ z_3 &= \tan\theta\\ z_4 &= y\end{aligned}$$

to give a one-chain system.

3.3 Optimal steering of nonholonomic systems

In this section, we discuss the least squares optimal control problem for steering a control system of the form

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m$$

from q_0 to q_f in 1 second. Thus, we minimize the cost function

$$\frac{1}{2}\int_0^1 \|u(t)\|^2\, dt.$$

Our treatment here is, of necessity, somewhat informal; to get all the smoothness hypotheses worked out would be far too large an excursion to make here. We will assume that the steering problem has a solution (by Chow's theorem, this is guaranteed when the controllability Lie algebra generated by $g_1, \ldots, g_m$ is of rank n for all q). We give a heuristic derivation from the calculus of variations of the necessary conditions for optimality. To do so, we incorporate the constraints into the cost function using a Lagrange multiplier function $p(t) \in \mathbb{R}^n$ to get

$$J(q,p,u) = \int_0^1 \left\{ \frac{1}{2}u^T(t)u(t) - p^T(\dot{q} - \sum_{i=1}^m g_i(q)u_i) \right\} dt. \quad (8.10)$$

Introduce the *Hamiltonian function*:

$$H(q,p,u) = \frac{1}{2}u^T u + p^T \sum_{i=1}^{m} g_i(q)u_i. \tag{8.11}$$

Using this definition and integrating the second term of (8.10) by parts yields

$$J(q,p,u) = -p^T(t)q(t)\Big|_0^1 + \int_0^1 (H(q,p,u) + \dot{p}^T q)\, dt.$$

Consider the variation in J caused by variations in the control input u,[1]

$$\delta J = -p^T(t)\delta q(t)\Big|_0^1 + \int_0^1 \left(\frac{\partial H}{\partial q}\delta q + \frac{\partial H}{\partial u}\delta u + \dot{p}^T \delta q\right) dt.$$

If the optimal input has been found, a necessary condition for stationarity is that the first variation above be zero for all variations δu and δq:

$$\dot{p} = -\frac{\partial H}{\partial q} \qquad \frac{\partial H}{\partial u} = 0. \tag{8.12}$$

From the second of these equations, it follows that the optimal inputs are given by

$$u_i = -p^T g_i(q), \quad i = 1, \ldots, m. \tag{8.13}$$

Using (8.13) in (8.11) yields the optimal Hamiltonian

$$H^*(q,p) = -\frac{1}{2}\sum_{i=1}^{m} \left(p^T g_i(q)\right)^2. \tag{8.14}$$

Thus, the optimal control system satisfies *Hamilton's equations*:

$$\begin{aligned} \dot{q} &= \frac{\partial H^*}{\partial p}(q,p) \\ \dot{p} &= -\frac{\partial H^*}{\partial q}(q,p) \end{aligned} \tag{8.15}$$

with boundary conditions $q(0) = q_0$ and $q(1) = q_f$. Using this result, we may derive the following proposition about the structure of the optimal controls:

Proposition 8.4. Constant norm of optimal controls
For the least squares optimal control problem for the control system

$$\dot{q} = \sum_{i=1}^{m} g_i(q)u_i$$

[1] In the calculus of variations, one makes a variation in u, namely δu, and calculates the changes in the quantities p and H as δp and δH. See for example [123].

which satisfies the controllability rank condition, the norm of the optimal input is constant, that is,

$$\|u(t)\|^2 = \|u(0)\|^2 \quad \forall t \in [0,1].$$

Proof. The formula for the optimal input is given in (8.13). Differentiating it yields

$$\dot{u}_i = -\dot{p}^T g_i(q) - p^T \frac{\partial g_i}{\partial q}\dot{q}.$$

Further, using the Hamiltonian equation for $\dot{p}_k$ given by

$$\dot{p}_k = -\sum_{i=1}^{m} p^T \frac{\partial g_i}{\partial q_k} u_i,$$

it may be verified that the formula for $\dot{u}_i$ is given by

$$\dot{u}_i = \sum_{j=1}^{m} p^T [g_i, g_j] u_j.$$

Collecting these in a matrix gives

$$\dot{u} = \Omega(q,p)u \tag{8.16}$$

where $\Omega(q,p)$ is a skew-symmetric matrix (i.e., it is in $so(m)$) given by

$$\Omega(q,p) = \begin{bmatrix} 0 & p^T[g_1,g_2] & \cdots & p^T[g_1,g_m] \\ -p^T[g_1,g_2] & 0 & \cdots & p^T[g_2,g_m] \\ \vdots & \vdots & & \vdots \\ -p^T[g_1,g_m] & -p^T[g_2,g_m] & \cdots & 0 \end{bmatrix}.$$

The solution of the linear time-varying equation (8.16) is of the form

$$u(t) = U(t)u(0) \tag{8.17}$$

for some $U(t) \in SO(m)$ (see Exercise 6). From this fact the statement of the proposition follows. □

This proposition provides an interesting formula (8.16) for the derivatives of the optimal input and establishes that the norm of the optimal input is constant. This fact can be used to establish that the same optimal input also solves other optimization problems which involve a monotone transformation of the integrand, such as

$$\int_0^1 \sqrt{u^T u}\, dt,$$

as well as some minimum-time problems (see Exercise 8). This proposition can be used to solve certain optimization problems, such as that for the so-called Engel's system:

Example 8.4. Optimal inputs for Engel's system
This system is of the form

$$\begin{aligned}\dot{q}_1 &= u_1\\ \dot{q}_2 &= u_2\\ \dot{q}_3 &= q_1u_2 - q_2u_1\\ \dot{q}_4 &= q_1^2u_2.\end{aligned}$$

This system has growth vector $(2,3,4)$. It may be verified that

$$[g_1, g_2] = \begin{bmatrix} 0\\ 0\\ 2\\ 2q_1 \end{bmatrix}$$

so that the optimal inputs satisfy the differential equation obtained by specializing (8.16), namely

$$\begin{aligned}\dot{u}_1 &= 2(p_3 + q_1p_4)u_2\\ \dot{u}_2 &= -2(p_3 + q_1p_4)u_1.\end{aligned} \tag{8.18}$$

The solution of this equation is of the form

$$\begin{aligned}u_1(t) &= r\sin\alpha(t)\\ u_2(t) &= r\cos\alpha(t),\end{aligned}$$

where $r^2 = u_1^2(0) + u_2^2(0)$. Further, since the optimal Hamiltonian given by

$$H^*(q,p) = -\frac{1}{2}(p_1 - q_2p_3)^2 - \frac{1}{2}(p_2 + q_1p_3 + q_1^2p_4)^2$$

is independent of q_3 and q_4, it follows that $\dot{p}_3 = \dot{p}_4 = 0$ so that p_3 and p_4 are constant. Using the functional form of u_1 and u_2 in (8.18) we get

$$\ddot{\alpha} = 2p_4r\sin\alpha.$$

To integrate this equation, multiply both sides by $2\dot{\alpha}$, integrate and define $\delta = 2p_4r$ to get

$$(\dot{\alpha})^2 = b - 2\delta\cos\alpha,$$

where b is a constant of the integration. If $b - 2|\delta| > 0$, this equation can be written as

$$\dot{\alpha} = \pm\sqrt{b - 2\delta\cos\alpha} = c\sqrt{1 - k\sin^2(\frac{\alpha}{2})},$$

for some constants c, k. This last equation may be integrated using elliptical integrals (see, for example, Lawden [57]) for α using

$$\int_0^{\alpha/2} \frac{d\sigma}{\sqrt{1 - k^2\sin^2\sigma}} = \frac{ct}{2} + d. \tag{8.19}$$

The left hand side of this equation is an elliptic integral. Hence the optimal inputs for the Engel's system come from elliptic functions.

In general, it is difficult to find the solution to the least squares optimal control problem for steering the system from an initial to a final point. However, it may be possible to find an approximate solution to the problem by using the so-called *Ritz approximation method.* To explain what this means, we begin by choosing an orthonormal basis for $L^2[0,1]$, the set of square integrable functions on $[0,1]$. One basis is a set of trigonometric functions $\{\psi_0(t), \psi_1(t), \dots\}$ given by $\psi_0(t) \equiv 1$ and for $k \geq 1$,

$$\psi_{2k-1}(t) = \sqrt{2}\cos 2k\pi t, \quad \psi_{2k}(t) = \sqrt{2}\sin 2k\pi t \quad t \in [0,1].$$

For a choice of integer N, a Ritz approximation to the optimal i^{th} input is assumed to be of the form

$$u_i(t) = \sum_{k=0}^{N} \alpha_{ik}\psi_k(t).$$

The plan now is to apply this input to steer the system

$$\dot{q} = g_1(q)u_1 + \cdots + g_m(q)u_m \tag{8.20}$$

from $q(0) = q_0$ to $q(1) = q_f$ and to determine the coefficient vectors, $\alpha_i = (\alpha_{i0}, \alpha_{i1}, \dots, \alpha_{iN}) \in \mathbb{R}^{N+1}$ for $i = 1, \dots, m$, so as to minimize the cost function

$$J = \frac{1}{2}(\sum_{i=1}^{m} \|\alpha_i\|^2 + \gamma\|q(1) - q_f\|^2). \tag{8.21}$$

In equation (8.21) the coefficient $\gamma > 0$ is a penalty term corresponding to reaching the final state. For large γ the cost function weights the reaching of the goal heavily. The heart of the Ritz approximation procedure is the hope that for large N and large γ, we will find a u that steers to the final point q_f with low cost.

At this point, we have a finite-dimensional optimization problem, which may be solved by a variety of methods. One method involving a modification of the Newton iteration for minimizing J has been explored in [32] and [33], where a software package called NMPack for this purpose is described. We refer the reader to these papers for details of its application to various examples.

3.4 Steering with piecewise constant inputs

In this section, we describe the rudiments of a method for motion planning for general nonholonomic systems due to Lafferriere and Sussmann [55].

The algorithm works for systems whose controllability Lie algebra is *nilpotent of order k*. By way of review, this means that for systems of the form

$$\dot{x} = g_1 u_1 + \cdots + g_m u_m$$

the Lie products between control vector fields of order greater than k are 0; i.e., $[g_{i_1}, \ldots, [g_{i_{p-1}}, g_{i_p}] \ldots]$ is zero when $p > k$. The method proposed in [55] is conceptually straightforward but the details of their method are somewhat involved. The first step is to derive a "canonical system equation" associated with the given control system. The chief new concept involved is that of formal power series in vector fields.

Recall from Section 2 of Chapter 7 that the flow associated with a vector field g_i was denoted $\phi_t^{g_i}(q)$, referring to the solution of the differential equation $\dot{q} = g_i(q)$ at time t starting from q at time 0. This flow is referred to as the *formal exponential* of g_i and is denoted

$$e^{tg_i}(q) := \phi_t^{g_i}(q).$$

The usage of the formal exponential is in that we will actually use identities of the form

$$e^{tg_i} = (I + tg_i + \frac{t^2}{2!} g_i^2 + \cdots),$$

where polynomials like g_i^2 and g_i^3 need to be carefully justified. We will defer this question for the moment and think formally of $e^{tg_i}(q)$ as a diffeomorphism from $\mathbb{R}^n$ to $\mathbb{R}^n$. Now, consider a nilpotent Lie algebra of order k generated by the vector fields $g_1, \ldots, g_m$. Recall from Section 4 of Chapter 7 that a Philip Hall basis is a basis for the controllability Lie algebra which has been constructed in such a way as to keep track of skew-symmetry and the Jacobi identity associated with the Lie bracket. We define the Philip Hall basis of the controllability Lie algebra generated by $g_1, \ldots, g_m$ to be

$$B_1, B_2, \ldots, B_s.$$

Thus, basis elements are Lie products of order less than or equal to k. In our language of formal power series, we will refer, for instance, to

$$[g_1, g_2] := g_1 g_2 - g_2 g_1$$
$$[g_1, [g_2, g_3]] := g_1 g_2 g_3 - g_1 g_3 g_2 - g_2 g_3 g_1 + g_3 g_2 g_1.$$

It is a basic result of nonlinear control, called the *Chen-Fliess series* formula, that all flows of the nonlinear control system (8.1), namely,

$$\dot{q} = \sum_{i=1}^{m} g_i(q) u_i \qquad q(0) = q$$

are of the form

$$S_t(q) = e^{h_s(t)B_s} e^{h_{s-1}(t)B_{s-1}} \cdots e^{h_2(t)B_2} e^{h_1(t)B_1}(q) \tag{8.22}$$

for some suitably chosen functions $h_1, h_2, \ldots, h_s$, known as the Philip Hall coordinates. The meaning of equation (8.22) is as follows: all flows that could possibly be generated by the control system of (8.1) may be obtained by composing flows along the Philip Hall basis elements $B_1, \ldots, B_s$. This result bears more than a passing resemblance to the product of exponentials formula for manipulator kinematics, but we will not digress to make this connection more explicit here. Furthermore, $S_t(q)$ satisfies a differential equation involving the basis elements, namely,

$$\dot{S}(t) = S(t)(B_1 v_1 + \cdots + B_s v_s) \qquad S(0) = 1, \tag{8.23}$$

where $S_t(q)$ has been replaced by $S(t)$ and the inputs $v_1, \ldots, v_s$ are the "fictitious inputs" corresponding to the directions of the Philip Hall basis elements $B_1, \ldots, B_s$. We say fictitious since only the first m of the Philip Hall basis elements correspond to $g_1, \ldots, g_m$. The other inputs correspond to Lie bracket elements and will eventually be dropped. Differentiating equation (8.22) yields

$$\begin{aligned}
\dot{S}(t) &= \sum_{j=1}^{s} e^{h_s B_s} \cdots e^{h_j B_j} \dot{h}_j B_j e^{h_{j-1} B_{j-1}} \cdots e^{h_1 B_1} \\
&= \sum_{j=1}^{s} S(t) e^{-h_1 B_1} \cdots e^{-h_{j-1} B_{j-1}} \dot{h}_j B_j e^{h_{j-1} B_{j-1}} \cdots e^{h_1 B_1} \\
&:= \sum_{j=1}^{s} S(t) \operatorname{Ad}_{e^{-h_1 B_1} \cdots e^{-h_{j-1} B_{j-1}}} \dot{h}_j B_j.
\end{aligned} \tag{8.24}$$

Here, in analogy with the formulas of Chapter 2, we have introduced the notation

$$\operatorname{Ad}_{e^{-h_i B_i}} B_j = e^{-h_i B_i} B_j e^{h_i B_i}.$$

Since the controllability Lie algebra is nilpotent of degree k, we can express each one of the elements on the right hand side in terms of the basis elements $B_1, \ldots, B_s$. More specifically, it may be verified that

$$\operatorname{Ad}_{e^{-h_1 B_1} \cdots e^{-h_{j-1} B_{j-1}}} = \operatorname{Ad}_{e^{-h_1 B_1}} \cdots \operatorname{Ad}_{e^{-h_{j-1} B_{j-1}}}. \tag{8.25}$$

Thus each element on the right hand side of (8.24) is a linear combination of $B_1, \ldots, B_s$ and we may express

$$\operatorname{Ad}_{e^{-h_1 B_1} \cdots e^{-h_{j-1} B_{j-1}}} \dot{h}_j B_j = \left(\sum_{k=1}^{s} p_{j,k}(h) B_k \right) \dot{h}_j$$

for some polynomials $p_{j,k}(h)$. Using this in the equation (8.23) and equating coefficients of the basis elements B_i yields

$$\sum_{j=1}^{s} p_{j,k}(h) \dot{h}_j = v_k \qquad k = 1, \ldots, s.$$

These equations are then solved to give the differential equation

$$\dot{h} = Q(h)v \quad h(0) = 0 \tag{8.26}$$

which is a control system in $\mathbb{R}^s$, called the Chen-Fliess-Sussmann equation, specifying the evolution of the Philip Hall coordinates in response to the "fictitious inputs" $v_1, \ldots, v_s$. It is important to realize that, in general, the dimension s of the Philip Hall coordinates is greater than n, the dimension of the state space of the control system. The initial conditions are $h_i(0) = 0$ corresponding to the identity diffeomorphism at $t = 0$. This equation is the canonical form associated with the nilpotent controllability Lie algebra associated with the given problem.

Example 8.5. Nilpotent system of degree three with two inputs Consider a two-input system which is nilpotent of degree three on $\mathbb{R}^4$. We will assume that g_1, g_2, $[g_1, g_2]$, $[g_1, [g_1, g_2]]$ are linearly independent. As the Philip Hall basis, we have

$$B_1 = g_1 \quad B_2 = g_2 \quad B_3 = [g_1, g_2]$$
$$B_4 = [g_1, [g_1, g_2]] \quad B_5 = [g_2, [g_1, g_2]].$$

Since $[g_2, [g_1, g_2]] = B_5$ is dependent on B_1, B_2, B_3, B_4 by hypothesis, we have in (8.23) that $v_5 \equiv 0$. An easy calculation shows that the coefficients of the $\dot{h}_j$ on the right hand side of (8.24) are given by

$$\begin{aligned}
\dot{h}_1 &: \quad B_1 \\
\dot{h}_2 &: \quad B_2 - h_1 B_3 + \frac{1}{2} h_1^2 B_4 \\
\dot{h}_3 &: \quad B_3 - h_2 B_5 - h_1 B_4 \\
\dot{h}_4 &: \quad B_4 \\
\dot{h}_5 &: \quad B_5.
\end{aligned}$$

For instance, the coefficient of $\dot{h}_2$ is calculated as

$$\begin{aligned}
\mathrm{Ad}_{e^{-h_1 B_1}} B_2 &= B_2 - h_1 [B_1, B_2] + \frac{1}{2} h_1^2 [B_1, [B_1, B_2]] \\
&= B_2 - h_1 B_3 + \frac{1}{2} h_1^2 B_4.
\end{aligned}$$

Equating the coefficients of the B_i to v_i with $v_5 = 0$, we get the Chen-Fliess-Sussmann equation

$$\begin{aligned}
\dot{h}_1 &= v_1 \\
\dot{h}_2 &= v_2 \\
\dot{h}_3 &= h_1 v_2 + v_3 \\
\dot{h}_4 &= \frac{1}{2} h_1^2 v_2 + h_1 v_3 + v_4 \\
\dot{h}_5 &= h_2 v_3 + h_1 h_2 v_2.
\end{aligned} \tag{8.27}$$

Note that this system is in $\mathbb{R}^5$, though the state space equations evolve on $\mathbb{R}^4$.

Example 8.6. Two-input, five-state chained system
Consider a chained system with two inputs and five states, where the input vector fields are

$$g_1 = \begin{bmatrix} 1 \\ 0 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The system is nilpotent of degree $k = 4$, and the Philip Hall basis vectors are

$$\begin{aligned} B_1, B_2 &: \quad g_1 \quad g_2 \\ B_3 &: \quad [g_1, g_2] \\ B_4, B_5 &: \quad [g_1, [g_1, g_2]] \quad [g_2, [g_1, g_2]] \\ B_6, B_7 &: \quad [g_1, [g_1, [g_1, g_2]]] \quad [g_2, [g_1, [g_1, g_2]]] \\ B_8 &: \quad [g_2, [g_2, [g_1, g_2]]] \end{aligned}$$

The vector fields g_1, g_2, $g_3 := \mathrm{ad}_{g_1} g_2$, $g_4 := \mathrm{ad}_{g_1}^2 g_2$, and $g_6 := \mathrm{ad}_{g_1}^3 g_2$ span the tangent space of $\mathbb{R}^5$.

Thus, for the Chen-Fliess-Sussmann equations, we have $v_5 = v_7 = v_8 = 0$, and the coefficient of $\dot{h}_j$ is given by

$$\mathrm{Ad}_{e^{-h_1 B_1}} \cdots \mathrm{Ad}_{e^{-h_{j-1} B_{j-1}}} B_j, \quad j = 1, \ldots .8.$$

We carry out the calculation for $\dot{h}_3$ in detail:

$$\begin{aligned} &\mathrm{Ad}_{e^{-h_1 B_1}} \mathrm{Ad}_{e^{-h_2 B_2}} B_3 \\ &\quad = \mathrm{Ad}_{e^{-h_1 B_1}} \left(B_3 - h_2 B_5 + \frac{1}{2} h_2^2 B_8 \right) \\ &\quad = B_3 - h_1 B_4 - h_2 B_5 + \frac{1}{2} h_1^2 B_6 + h_1 h_2 B_7 + \frac{1}{2} h_2^2 B_8. \end{aligned}$$

The calculations for the remaining terms are carried out in a similar fashion and the results are given below (we invite the reader to do the

calculation herself):

$$\begin{aligned}
\dot{h}_1: &\quad B_1 \\
\dot{h}_2: &\quad B_2 - h_1 B_3 + \frac{1}{2}h_1^2 B_4 - \frac{1}{6}h_1^3 B_6 \\
\dot{h}_3: &\quad B_3 - h_1 B_4 - h_2 B_5 + \frac{1}{2}h_1^2 B_6 + h_1 h_2 B_7 + \frac{1}{2}h_2^2 B_8 \\
\dot{h}_4: &\quad B_4 - h_1 B_6 - h_2 B_7 \\
\dot{h}_5: &\quad B_5 - h_1 B_7 - h_2 B_8 \\
\dot{h}_6: &\quad B_6 \\
\dot{h}_7: &\quad B_7 \\
\dot{h}_8: &\quad B_8.
\end{aligned}$$

Finally, with $v_5 = v_7 = v_8 = 0$ the differential equation for the $h \in \mathbb{R}^8$ is found to be

$$\begin{aligned}
\dot{h}_1 &= v_1 \\
\dot{h}_2 &= v_2 \\
\dot{h}_3 &= h_1 v_2 + v_3 \\
\dot{h}_4 &= \frac{1}{2}h_1^2 v_2 + h_1 v_3 + v_4 \\
\dot{h}_5 &= h_1 h_2 v_2 + h_2 v_3 \\
\dot{h}_6 &= \frac{1}{6}h_1^3 v_2 + \frac{1}{2}h_1^2 v_3 + h_1 v_4 + v_6 \\
\dot{h}_7 &= \frac{1}{2}h_1^2 h_2 v_2 + h_1 h_2 v_3 + h_2 v_4 \\
\dot{h}_8 &= \frac{1}{2}h_1 h_2^2 v_2 + \frac{1}{2}h_2^2 v_3
\end{aligned}$$

with initial condition $h(0) = 0$.

The task of steering the system from q_0 to q_f still remains to be done. This is accomplished by choosing *any* trajectory connecting q_0 to q_f indexed by time t. This is substituted into (8.23) to get the expression for the "fictitious inputs" $v_1, \ldots, v_n$, corresponding to the first n linearly independent vector fields in the Philip Hall basis of the control system. The inputs v_i are said to be fictitious, since they need to be generated by using the "real" inputs $u_i, i = 1, \ldots, m$. To do so involves a generalization of the definition of Lie brackets given in Section 2 of Chapter 7. For instance, to try to generate an input v_3 corresponding to $[g_1, g_2]$ in the previous example, one follows the definition and uses the concatenation of four segments, each for $\sqrt{\epsilon}$ seconds: $u_1 = 1, u_2 = 0; u_1 = 0, u_2 = 1; u_1 = -1, u_2 = 0; u_1 = 0, u_2 = -1$. The resulting flow is given by

$$e^{\sqrt{\epsilon}g_1} e^{\sqrt{\epsilon}g_2} e^{-\sqrt{\epsilon}g_1} e^{-\sqrt{\epsilon}g_2}. \tag{8.28}$$

A formula called the Campbell-Baker-Hausdorff formula gives an expression for this in terms of the Philip Hall basis elements. We will make a brief digression to state this formula, since it is important in its own right (the proof and a more exhaustive recursive formulation of the coefficients is given in [116]).

Theorem 8.5. Campbell-Baker-Hausdorff formula
Given two smooth vector fields g_1, g_2 the composition of their exponentials is given by

$$e^{g_1} e^{g_2} = e^{g_1 + g_2 + \frac{1}{2}[g_1, g_2] + \frac{1}{12}([g_1,[g_1,g_2]] - [g_2,[g_1,g_2]]) \cdots} \tag{8.29}$$

where the remaining terms may be found by equating terms in the (non-commutative) formal power series on the right- and left-hand sides.

Using the Campbell-Baker-Hausdorff formula for the flow in equation (8.28) gives the Philip Hall coordinates to be

$$h = (0, 0, \epsilon, h_4(\epsilon)),$$

where $h_4(\epsilon)$ is of higher order than one in ϵ. Thus, the strategy of cycling between u_1 and u_2 not only produces motion along the direction $[g_1, g_2]$, but also along $[g_1, [g_1, g_2]]$. Thus, both $v_3, v_4 \neq 0$. However both the h_1 and h_2 variables are unaffected. To generate motion along the v_4 direction, one concatenates an eight-segment input consisting of cycling between the u_1, u_2 for $\sqrt[3]{\epsilon}$ seconds. Since the controllability Lie algebra is nilpotent, this motion does not produce any motion in any of the other bracket directions (in the context of the Campbell-Baker-Hausdorff formula above, the series on the right hand side of (8.29) is finite). This example can be generalized to specify a constructive procedure for generating piecewise constant inputs for steering systems which have nilpotent controllability Lie algebras.

When the controllability Lie algebra is not nilpotent, the foregoing algorithm needs to be modified to steer the system only approximately. The two difficulties in this case are:

1. The Philip Hall basis is not finite.
2. The Campbell-Baker-Hausdorff formula does not terminate after finitely many terms.

One approach that has been suggested in this case is to try to change the input g_i using a transformation of the inputs, that is by choosing a matrix $\beta \in \mathbb{R}^{m \times m}$ so as to make the transformed $\tilde{g}_i$, defined by

$$\begin{bmatrix} \tilde{g}_1 & \tilde{g}_2 & \cdots & \tilde{g}_m \end{bmatrix} = \beta \begin{bmatrix} g_1 & g_2 & \cdots & g_M \end{bmatrix}$$

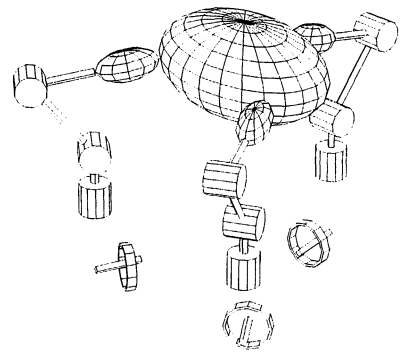


Figure 8.2: Three-fingered hand grasping an object. The small circles below each finger indicate the internal forces applied by each finger. (Figure courtesy of John Hauser)

a nilpotent control system. Other approaches involve "nilpotent approximations" to the given control system. For more details on the algorithm for steering nonholonomic systems using piecewise constant inputs, see the paper of Lafferriere and Sussmann in [62].

4 Dynamic Finger Repositioning

We return now to the problem of repositioning the fingers on a hand without actually lifting the fingers from the object. As we saw in the examples in Section 4 of Chapter 7, if we have spherical fingertips contacting a planar face, it is possible to move the contact point via pure rolling. This section works out this problem in detail and gives several different solutions for the dynamic finger repositioning problem.

4.1 Problem description

Consider the grasping control problem with rolling contacts, such as the system shown in Figure 8.2. In Chapter 5, we derived the kinematic equations of motion for a single body in contact with a set of fingers. In local coordinates, the overall constraints on the system have the form

$$J_h(\theta, x)\dot{\theta} = G^T(\theta, x)\dot{x}, \tag{8.30}$$

where θ is the vector of finger joint angles and x specifies the position and orientation of the grasped object.

If the grasp described by the constraints in equation (8.30) is manipulable, then any object velocity $\dot{x}$ can be accommodated by some finger

velocity vector $\dot{\theta}$. However, the vector $\dot{\theta}$ may not be unique in the case that the null space of J_h is nontrivial. This situation corresponds to the existence of internal motions of the fingers that do not affect the motion of the object. If we let u_1 be an input which controls the velocity of the object and let u_2 parameterize the internal motions, then equation (8.30) can be written as

$$\begin{aligned} \dot{x} &= u_1 \\ \dot{\theta} &= J_h^+ G^T u_1 + K u_2, \end{aligned} \tag{8.31}$$

where the columns of K span the null space of J_h.

Equation (8.31) describes the grasp kinematics as a control system. The dynamic finger repositioning problem is to steer the system from an initial configuration (θ_0, x_0) to a desired final configuration (θ_f, x_f). The explicit location of the fingertip on the object at the initial and final configurations can be found by solving the forward kinematics of the system.

The general case of finding $u_1(t)$ and $u_2(t)$ such that the object and the fingers move from an initial to final position (while maintaining contact) can be very difficult. We point out two interesting special cases:

1. If the hand has no redundant degrees of freedom (i.e., K is not present) then it might be possible to move to an arbitrary location/grasp using only u_1. Moving just the contact location requires a carefully chosen closed loop path in x.

2. If we have redundant degrees of freedom, then we can move the fingers along the object while keeping the object position fixed ($\dot{x} = u_1 = 0$). In this case, we use only the vector fields in K to move the fingers.

In the second case, it is sufficient to study the control of a single finger, since the fingers are decoupled if the object is held fixed. We concentrate here on the second case, which is considerably simpler.

4.2 Steering using sinusoids

Consider the case of a single spherical finger rolling on a plane. The kinematics were derived in Chapter 7 and the associated control system is repeated here:

$$\dot{\eta} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \sec q_1 \\ -\sin q_5 \\ -\cos q_5 \\ -\tan q_1 \end{bmatrix} u_1 + \begin{bmatrix} -1 \\ 0 \\ -\cos q_5 \\ \sin q_5 \\ 0 \end{bmatrix} u_2. \tag{8.32}$$

As in the preceding discussion, we will change variables to new ones and keep track of the Taylor series expansions of the nonlinear terms to get a two-chained system. More specifically, with change of state

$$z_1 = q_1 \quad z_2 = q_2 \quad z_3 = -q_5 \quad z_4 = q_3 - q_1 \quad z_5 = q_4 + q_2$$

and change of input

$$v_1 = -u_2 \qquad v_2 = \sec q_1 u_1$$

we get

$$\begin{aligned}
\dot{z}_1 &= v_1 \\
\dot{z}_2 &= v_2 \\
\dot{z}_3 &= z_1 v_2 + \psi_1(z_1)v_2 \\
\dot{z}_4 &= z_3 v_2 + \psi_2(z_3)v_1 + \psi(z_1, z_3)v_2 \\
\dot{z}_5 &= z_3 v_1 + \psi_4(z_3)v_1 + \psi_5(z_1, z_3)v_2,
\end{aligned} \tag{8.33}$$

where the ψ_i are quadratic or higher order in their respective arguments. If these functions ψ_i are neglected, then we have a two-chained system which can be steered using sinusoids. Unlike the case of the hopping robot or car parking, the calculation of the Fourier series coefficients for z_4 and z_5 is not easy. However, a numerical procedure based on using a sines and cosines at integrally related frequencies may be used to steer the finger on the surface of the object. An example of such a path moving a spherical fingertip down the side of a planar object is shown in Figure 8.3. In this figure, we consider the motion of a finger with a spherical tip on a rectangular object (left). The plots to the right of the figure show trajectories which move a finger down the side of the object. The location of the contact on the finger is unchanged as shown in the upper graph which plots the finger contact configurations $(q_1, q_2) = (u_f, v_f)$, while the location of the contact on the face of the object $(q_3, q_4) = (u_o, v_o)$ undergoes a displacement in the v_o direction.

4.3 Geometric phase algorithm

In this subsection, we will describe the use of some techniques from classical differential geometry which can be brought to bear on the specific problem of rolling a spherical finger on a planar surface.

As in Chapter 5, let Q be the configuration space of contact, S_f and S_0 the surface of the fingertip and the object. Then, in local coordinates, Q is parameterized by $\eta = (\alpha_f, \alpha_o, \psi)$, where $\alpha_f = (u_f, v_f), \alpha_o = (u_o, v_o) \in \mathbb{R}^2$ are local coordinates on S_f and S_o respectively, and ψ is the angle of contact. The kinematic equations of rolling contact from Chapter 5 are

$$\begin{aligned}
M_f \dot{\alpha}_f - R_\psi M_o \dot{\alpha}_o &= 0 \\
T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o - \dot{\psi} &= 0.
\end{aligned} \tag{8.34}$$

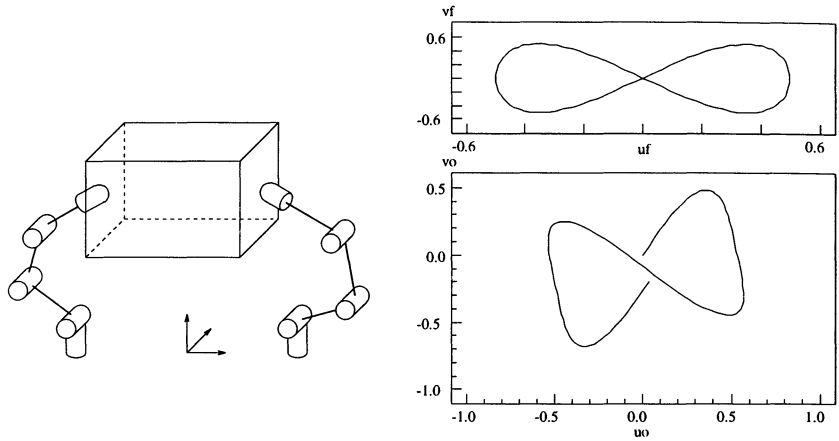


Figure 8.3: Steering applied to a multifingered hand.

In the instance of a spherical finger rolling on a plane, we have that $M_o = I$, $T_o = 0$ and that

$$T_f = \begin{bmatrix} 0 & -\frac{1}{\rho}\tan u_f \end{bmatrix} \qquad M_f = \begin{bmatrix} \rho & 0 \\ 0 & \rho \cos u_f \end{bmatrix},$$

where ρ is the radius of the finger. Since M_f and $R_\psi M_o$ are nonsingular, given either $\alpha_f(t) \in \mathbb{R}^2$ or $\alpha_o(t) \in \mathbb{R}^2$ there exists a unique path $\eta(t) \in Q$ which satisfies the rolling without slipping constraint. More specifically, let $\alpha_f(t), t \in [0,1]$, be a path in S_f and denote $u = \dot{\alpha}_f$. Then the path $\eta(t) \in Q$ is given by integrating the following differential equations

$$\begin{aligned} \dot{\alpha}_f &= u \\ \dot{\alpha}_o &= M_o^{-1} R_\psi M_f u \\ \dot{\psi} &= (T_f + T_o R_\psi) M_f u. \end{aligned} \tag{8.35}$$

This is to say that there exists a well defined *lifting* map $\rho^{-1} : S_f \to Q$ which lifts every path in S_f to a path in Q. The following classical theorem describes how a closed path in S_f generates a change in the angle of contact ψ. Using the formulas for T_f, M_f and T_o above yields

$$\dot{\psi} = -\sin u_f u_2. \tag{8.36}$$

Note that this equation is independent of the radius ρ of the sphere. Thus, the following theorem, though stated for spheres of radius one, actually holds for spheres of arbitrary radius.

Theorem 8.6. Gauss-Bonnet theorem
Let $\alpha_f : [0,1] \to S_f$ be a closed path on the sphere of radius one which

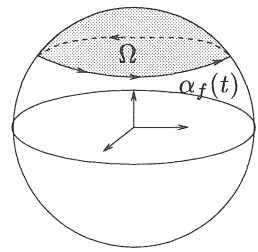


Figure 8.4: Geometric phase of a path in the sphere.

encloses a cap shaped region Ω as shown in Figure 8.4. Let $\Delta\psi$ be the change of the angle of contact as a result of rolling the sphere on the plane along the path given by $\alpha_f(\cdot)$. Then

$$\Delta\psi = -\text{ Area of } \Omega, \tag{8.37}$$

where the area of Ω is measured on the (curved) surface of the sphere of radius one.

Proof. From equation (8.36), we have that

$$\dot{\psi} = T_f M_f \dot{\alpha}_f = -\sin u_f \dot{v}_f.$$

Integrating $\dot{\psi}$ along the curve $\alpha_f(\cdot)$ and applying Green's theorem to the line integral yields

$$\begin{aligned}\Delta\psi = \int_0^1 \dot{\psi}\, dt &= \oint -\sin u_f\, dv_f \\ &= -\iint_\Omega \cos u_f\, dv_f\, du_f = -\text{ Area of } \Omega,\end{aligned}$$

since $\cos u_f\, du_f\, dv_f$ is the infinitesimal area (area form) on a sphere of radius one in local coordinates. □

In the preceding theorem, we saw that the net change in the contact angle $\Delta\psi$ depends on the area enclosed by the curve $\alpha_f(\cdot)$ on a sphere of radius one *regardless of the actual radius of the sphere*! $\Delta\psi$ is referred to as the *geometric phase* or *nonholonomy* of the path $\alpha_f(\cdot)$. It tells us the motion (in fact the area to be covered by a closed path traced out by the finger) to generate a certain change in the contact angle $\Delta\psi$.

In the next proposition, we show how to generate motion on the spherical finger by using a closed path α_o on the planar surface of the object. This specifies the motion of the object surface required to reposition the finger. In turn, we may use this method to produce closed loops in the motion of the finger, which produce changes in phase $\Delta\psi$. The development is completely geometric and does not use the rolling equation (8.35).

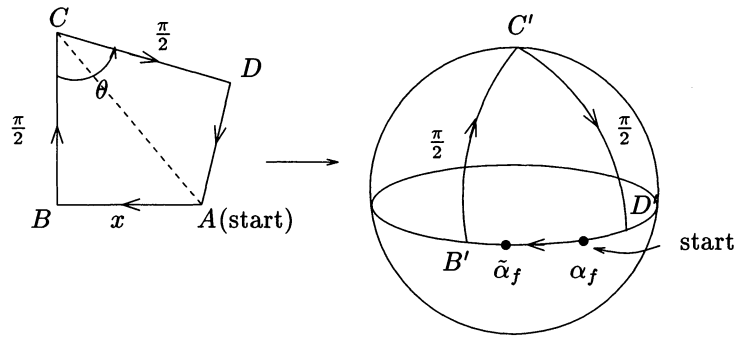


Figure 8.5: Closed path in the plane generates motion on the sphere.

Proposition 8.7. Sphere rolling on a plane
Let α_f and $\tilde{\alpha}_f$ be two points on the sphere which are a distance l apart on the sphere. Assume that $l < \frac{\pi}{2}$. Then, rolling the sphere on the plane along the closed curve ABCDA shown in Figure 8.5 takes α_f to $\tilde{\alpha}_f$ provided the segments AB, DA are of length x and BC, CD are of length $\frac{\pi}{2}$, and x solves the equation

$$2\left(x - \tan^{-1}\frac{x}{\pi/2}\right) = l.$$

Proof. Connect α_f and $\tilde{\alpha}_f$ by an arc of a great circle, and denote by A the starting point on the plane. Tracing the straight line from A to B and then to C on the plane induces a curve in the sphere which starts at α_f, passes through B', and then comes to the north pole, C', as shown in the Figure 8.5. Since the distance traversed by the fingertip on the surface of the object is the same as the distance on the sphere, we have that the distance between α_f and B' is x and $\angle(\alpha_f B'C') = \frac{\pi}{2}$. Going from C to D at an angle θ and a distance $\frac{\pi}{2}$ is equivalent to going down from C' to some point D' on the equator and $d(B', D') = \theta$. By tracing a straight line from D to A on the plane, we follow the equator from D' to some point $\tilde{\alpha}_f$, where $d(\tilde{\alpha}_f, D') = x$. It is clear from the figure that

$$d(\alpha_f, \tilde{\alpha}_f) = 2x - \theta = 2x - 2\tan^{-1}\frac{x}{\pi/2} := f(x).$$

Furthermore, for each $l < \frac{\pi}{2}$ the following equation

$$f(x) = l$$

has a unique solution $x \in [0, \frac{\pi}{2}]$ because $f(0) = 0, f(\pi/2) = \pi/2 > l$ and $f'(x) > 0$. □

To summarize, using techniques from geometry and Green's theorem one can generate strategies for rolling the planar surface of an object

face to cause certain desired motion in the spherical fingertip and the angle of contact. The method appears to be ad hoc and specific to the specific geometry of the finger and the object, however, it may in fact be generalized to other geometries. In fact, there is a generalization of the method to situations other than fingers rolling on objects called the *method of geometric phase.* For example, this method can be used to solve problems of reorienting satellites in space or to give an explanation for how cyclic motions of cilia on the surface of a paramecium cause forward motion in the paramecium.

5 Summary

The following are the key concepts covered in this chapter:

1. Optimal controls (minimizing integral least squares cost) for steering a system with growth vector $(2,3)$ of the form

$$\begin{aligned}\dot{q}_1 &= u_1\\ \dot{q}_2 &= u_2\\ \dot{q}_3 &= q_1u_2 - q_2u_1\end{aligned}$$

   are sinusoidal. Further more when $q_1(0) = q_2(0) = q_1(1) = q_2(1) = 0$, the optimal inputs are sinusoids at frequency 2π.

2. For a system of the form

$$\begin{aligned}\dot{q} &= u\\ \dot{Y} &= qu^T - uq^T\end{aligned}$$

   the optimal steering inputs, minimizing the integral least squares cost, are sinusoidal. Further, when $q(0) = q(1) = 0$ the optimal inputs are sinusoids at integrally related frequencies.

3. Using integrally related sinusoids as (sub-optimal) inputs, one can steer chained form systems. A one-chain system is one of the form

$$\begin{aligned}\dot{q}_1 &= u_1\\ \dot{q}_2 &= u_2\\ \dot{q}_3 &= q_2u_1\\ \dot{q}_4 &= q_3u_1\\ &\vdots\\ \dot{q}_n &= q_{n-1}u_1\end{aligned}$$

   Generalizations to multi-chain systems also exist. Involutivity conditions for converting given control systems into the chained form may be given.

4. While it is difficult to give closed form expressions for the optimal controls associated with solving the least squares steering problem for a nonholonomic control system, one can derive formulas for the time derivatives of the optimal inputs. Further, numerical techniques, such as the Ritz approximation algorithm, may be used to derive approximate algorithms for generating the optimal controls.

5. Piecewise constant inputs can be used to steer a nonholonomic control system in the Philip Hall basis coordinates when the controllability Lie algebra is nilpotent.

6. Dynamic finger repositioning on the surface of an object can be carried out using sinusoids. In the special case of a spherical finger rolling on the surface of a flat object, the geometry of the Gauss-Bonnet theorem may be used to position the finger on the object surface and adjust the angle of contact.

6 Bibliography

While research in nonholonomic behavior of mechanical systems is quite classical, interest in steering and trajectory generation is quite recent. To our knowledge, the connection between nonholonomy and constructive controllability was first pointed by Laumond [56] in the context of mobile robots and Li [59, 61] in the context of fingers rolling on the surface of a grasped object. The literature in controllability for nonlinear systems is quite extensive. A good review of it is to be found in the textbooks of Nijmeijer and van der Schaft [84] and Isidori [44]. However, some of the most important first results on constructive controllability appeared in [11] and [4], where the least squares steering problem was solved for a class of model systems on $\mathbb{R}^n$ and $SO(3)$ respectively.

In this chapter, we have used as source material some of our own recent publications in this area, such as [61] and [78] for steering fingers rolling on the surface of an object, [79] which discusses the use of sinusoids in steering nonholonomic systems, [104] which discusses the structure of optimal controls for steering problems, [32, 33] on the Ritz approximation procedure for solving optimal control problems, and [113] which solves the problem of parking a car with N trailers. A recent collection of papers [62] contains a good cross-section of papers on nonholonomic motion planning for further reading. A discussion of nonholonomic mechanics and geometric phase is in [68, 69].

7 Exercises

1. Show that the following system

$$\begin{aligned}
\dot{q}_1 &= u_1 \\
\dot{q}_2 &= u_2 \\
\dot{q}_3 &= q_1 u_2 \\
\dot{q}_4 &= \frac{1}{2} q_1^2 u_2 \\
\dot{q}_5 &= q_1 q_2 u_2 \\
\dot{q}_6 &= \frac{1}{6} q_1^3 u_1 \\
\dot{q}_7 &= \frac{1}{2} q_1^2 q_2 u_2 \\
\dot{q}_8 &= \frac{1}{2} q_1 q_2^2 u_2
\end{aligned}$$

   is controllable and nilpotent of degree four. Can you find a nonlinear change of coordinates to transform this system into a one-chained form?

2. Show that the following system is controllable and nilpotent:

$$\begin{aligned}
\dot{q}_1 &= u_1 \\
\dot{q}_2 &= u_2 \\
\dot{q}_3 &= q_1 u_2 - q_2 u_1 \\
\dot{q}_4 &= q_1^2 u_2 \\
\dot{q}_5 &= q_2^2 u_1 .
\end{aligned}$$

3. Consider the following control system

$$\begin{aligned}
\dot{q} &= u \\
\dot{Y} &= q u^T - u q^T
\end{aligned}$$

   where $u \in \mathbb{R}^m$ and $Y \in so(m)$.

   (a) Derive the Euler-Lagrange equations for the system, by minimizing the following integral

$$\frac{1}{2} \int_0^1 u^T u \, dt.$$

   (b) For the boundary conditions $q(0) = q(1) = 0$, $Y(0) = 0$ and $Y(1) = \widehat{y}$ for some $y \in \mathbb{R}^3$, solve the Euler Lagrange equations to obtain the optimal inputs u.

(c) Find the input u to steer the system from $(0,0)$ to $(0,\tilde{Y}) \in \mathbb{R}^m \times so(m)$.

4. Consider the following system

$$\begin{aligned}\dot{q}_1 &= u_1\\ \dot{q}_2 &= u_2\\ \dot{q}_{12} &= q_1 u_2\\ \dot{q}_{121} &= q_{12} u_1\\ \dot{q}_{122} &= q_{12} u_2\end{aligned}$$

Apply the inputs

$$\begin{aligned}u_1 &= a_1 \sin 2\pi t + a_2 \cos 2\pi t + a_3 \sin 4\pi t + a_4 \cos 4\pi t\\ u_2 &= b_1 \sin 2\pi t + b_2 \cos 2\pi t + b_3 \sin 4\pi t + b_4 \cos 4\pi t\end{aligned}$$

to this system and integrate $\dot{q}_{121}$ and $\dot{q}_{122}$ from $t = 0$ to $t = 1$ to obtain a system of polynominal equations in (a_i, b_i). Propose a method for solving for the coefficients (a_i, b_i) given the initial and final states.

5. *Two-input two-chained system*
The following system is referred to as a two-input, two-chained system

$$\begin{aligned}\dot{x}_0 &= u_1 & \dot{y}_0 &= u_2\\ \dot{x}_1 &= y_0 u_1 & (\dot{y}_1 &= x_0 u_2)\\ &\vdots & &\vdots\\ x_{n_x} &= x_{n_x-1} u_1 & y_{n_y} &= y_{n_y-1} u_2\end{aligned}$$

where $y_1 := x_0 y_0 - x_1$ to account for skew-symmetry of the Lie bracket.

(a) Prove that the system is controllable.

(b) Show that for each $q_k, k \geq 1$, the inputs $u_1 = a \sin 2\pi t$ and $u_2 = b \cos 2\pi k t$ steer the system to the final value of q_k. Give an explicit formula for the final value of q_k in terms of (a, b).

(c) Show that for each $y_k, k \geq 2$, the inputs $u_1 = b \cos 2\pi k t$ and $u_2 = a \sin 2\pi t$ steers the system to the final value of y_k.

(d) Give an algorithm for steering the system from an initial state to a final state.

6. In the proof of Proposition 8.4, it was asserted that if u satisfies the differential equation

$$\dot{u} = \Omega(t) u$$

for some $\Omega \in so(m)$, then the solution of u is of the form

$$u(t) = U(t)u(0)$$

for some $U(t) \in SO(m)$. Prove this assertion.

7. Proposition 8.4 gave a formula for the first derivative of the optimal inputs. Use the method of the proof of that proposition to obtain a formula for $\ddot{u}$. Express the answer in terms of Lie brackets of the input vector fields. In fact, if you are adventurous, try to find the formula for $u^{(n)}$.

8. Use the results of Proposition 8.16 to show that the optimal input of that proposition normalized by $\|u(0)\|$, that is,

$$\frac{u(t)}{\|u(0)\|}$$

solves the minimum time steering problem to steer the system from $q(0) = q_0$ to $q(T) = q_f$ subject to the constraint that $\|u(t)\|^2 \leq 1$ for all t.

9. Apply the methods of Proposition 8.16 to solve the optimal control inputs for the model control system we studied in Section 2, namely

$$\begin{aligned} \dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= q_1 u_2 - q_2 u_1. \end{aligned}$$

10. Extend the method used to find the optimal inputs for Engel's system to find optimal inputs for the system of Exercise 2.

11. Consider the least squares optimal input steering problem for a system with drift:

$$\dot{q} = f(q) + \sum_{i=1}^{m} g_i(q)u_i.$$

(a) Find the expression for the optimal Hamiltonian and prove that the optimal inputs satisfy the differential equation

$$\dot{u} = \Omega(q,p)u = \begin{bmatrix} p^T[f,g_1] \\ \vdots \\ p^T[f,g_m] \end{bmatrix} \tag{8.38}$$

with $\Omega(q,p)$ defined as in (8.16).

(b) Find the second derivatives of the optimal inputs given in equation (8.38).

12. Consider the following system

$$\begin{aligned}\dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= q_2 u_1\end{aligned}$$

with initial condition $q(0) = 0$. Let the inputs be of the form

$$\begin{aligned}u_1 &= \alpha_{1,0}\alpha_{1,1}\cos 2\pi t + \alpha_{1,2}\sin 2\pi t \\ u_2 &= \alpha_{2,0} + \alpha_{2,1}\cos 2\pi t + \alpha_{2,2}\sin 2\pi t.\end{aligned}$$

(a) Integrate the control system symbolically from $t = 0$ to $t = 1$ to obtain $q(1)$ as a function of α, the coefficient vector of the inputs, and compute the Jacobian $A = \frac{\partial q(1)}{\partial \alpha} \in \mathbb{R}^{3\times 6}$.

(b) Prove that if A is full rank then the system is controllable with inputs of the above form.

13. Consider the one-chain system in equation (8.7). For the Philip Hall basis g_1, g_2, $\mathrm{ad}_{g_1}^k g_2$, $k = 1, \ldots, n-1$, derive the Chen-Fliess-Sussmann equation.

14. Consider the following system

$$\begin{aligned}\dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= q_1 u_2 - q_2 u_1 \\ \dot{q}_4 &= q_1^2 u_2.\end{aligned}$$

(a) Derive the Chen-Fliess-Sussmann equation.

(b) Assuming that the inputs are of the form

$$\begin{aligned}u_1 &= a_0 + a_1\cos 2\pi t + a_2\sin 2\pi t \\ u_2 &= b_0 + b_1\cos 2\pi t + b_2\sin 2\pi t,\end{aligned}$$

compute the polynomial equation for the amplitude parameters in terms of the initial and final states.

Chapter 9

Future Prospects

DOI: 10.1201/9781315136370-9
This chapter has been made available under a CC-BY 4.0 license.

In this book, we have tried to give the reader a feel for the sorts of analytical tools that one needs in the study of robotic manipulation. We have adopted a mathematical point of view because of its compactness. One thing that this mathematical point of view masks is the excitement that we feel for robotics technology and its future because of the considerable innovation and the development being made in its use. The robotics industry has reached one plateau with the successful introduction of robots into automotive manufacturing—spot welding and painting are two arenas where robotic usage is almost universal, assembly of engines is an area where the amount of utilization is more varied—and into electronic assembly. There are several other areas where the usage of robotics is in its infancy and this chapter is dedicated to brief descriptions of some of these fields along with a quick assessment of their current status.

One question that comes up often in such a retrospective is the difference between "teleoperation" and "robotics." While precise definitions and distinctions between these two topics are elusive, the rough distinction appears to be in the amount of human interaction: "human intelligence" rather than "machine intelligence" required for the operation of the same set of basic devices. Consequently, it hardly seems surprising to us that in the natural course of evolution of the technology, teleoperation will precede true robotic or autonomous operation. Indeed, the pragmatic point of view would be to favor the introduction of new robotic devices first in teleoperated form. This has also historically been the path of evolution of the field. Nonetheless, there are good reasons for not having a person in the feedback loop in many applications:

1. Communication delays in transmission of sensor information and receipt of command information

2. Slow speed of response of humans to numerical and quantitative data

In our opinion, the teleoperators of today are the autonomous robots of tomorrow and as such we will not make a distinction between them in this chapter.

One other topic that comes up a great deal in the popular science press is the concurrent usage of the terms *virtual reality*, and *telepresence*. The term virtual reality refers to the remote creation of a synthetic environment containing sights, sounds, touch, and forces. While remote sight and sound are easily implemented, remote touch and force are not as easily achievable. In order to manipulate objects remotely (telepresence), it is important to be able to have each of these senses. Thus, there is an intimate relationship between remote manipulation or telepresence and virtual reality. Indeed, we see that the bulk of the current literature on virtual reality is really about telepresence, since the purpose of simulating a remote environment is to allow a person to interact with it.

In summary, we feel that the technological progress in the years to come will be on a broad front spanning teleoperation, virtual reality, and dextrous manipulation. In the rest of the chapter, we will say a little bit about the opportunities. The sections are organized according to the scale of the robots: Section 1 deals with conventionally sized robots, Section 2 with robots at the millimeter scale, and Section 3 with robots at the micrometer and nanometer scale.

1 Robots in Hazardous Environments

One of the chief areas for the future (and current) use of robots of the conventional size is in hazardous environments. In this section, we give a brief description of the sorts of environments in which robots will be (and are) found.

Space

The best known example of a robot in space is the 20 meter long remote manipulator system on board the space shuttle. It has six degrees of freedom and is usually manually teleoperated by an astronaut under direct visual feedback. There is, however, the ability to have the robot be moved under computer control in Cartesian coordinates. There are Japanese plans to build a flight telerobotic servicer which has two cooperating robot arms for repairing satellites and other coordinated activities on board a self-propelled platform. Other examples of robots in space include the Mars rover and other planetary exploring robots which feature tracked or wheeled mobile bases with arms on board for scooping soil samples. It is not anticipated that these devices will be under human control remotely since the transit time delays for commands are too high to allow for meaningful remote feedback actions.

Underwater

In the last decade, several remotely-operated vehicles have been built for inspection of underwater oil derricks and exploration. Their development has been motivated by the high cost of human divers and the risk to life of working under water. For the most part they consist of a mobile platform, either on an umbilical line from the mother ship or completely autonomous, fitted with one or more robotic arms. Most of these robots are remotely piloted and most current undersea manipulators are hydraulic to withstand the high forces and corrosive elements that they need to withstand. In the future, there will be a surge in the number of completely autonomous robots for exploration of the ocean floor and other unstructured environments for which the human reaction time is too large.

Nuclear, toxic waste disposal and mining

Some of the earliest work in robotics came from teleoperators for handling radioactive material. In recent years mobile robots with robotic arms onboard for inspection, maintenance, and even for handling of spent fuel rods, have become more prevalent. With the growth in the extent and nature of hazardous materials that need to be disposed worldwide, robots for handling and disposing toxic materials will need to be developed. Mining environments are similarly hazardous and there are already quite a few different kinds of mining vehicles and arms that can be remotely operated.

Firefighting, construction and agriculture

One can visualize a scenario in which maps of buildings would be downloaded onto robots at the scene of a fire. These robots can then be used for firefighting using onboard heat and smoke detectors and trailing an umbilical cord carrying water as well as fire-retardant chemicals and relaying video data to remote locations. In this application, it seems important to have robots that can negotiate stairways as well as corridors (i.e., legged as well as wheeled robots). Automated construction is a field in its infancy, but robotic tools for accurate and risk-free construction, sometimes in underground or underwater environments, are developing rapidly. In agriculture, multifingered robot hands mounted on an arm and equipped with vision systems have been used for picking oranges and for harvesting crops. It is thought that robots could also be used for tilling and planting.

Robotic systems have also been used for deboning meat in meat packing plants in Australia. Finally, an amazing robotic system that has been used for sheep shearing has been developed at the University of West Aus-

tralia and features the development of a wrist with no singularities in the workspace of the manipulator [115].

2 Medical Applications for Multifingered Hands

In recent years, there has been a great deal of excitement about minimally invasive surgery, including a number of techniques for accessing internal organs through small incisions or orifices in the body (varying in size from about 3 millimeters to 11 millimeters). As a result, trauma to muscles and other tissue which need to be cut in traditional surgery is minimized, resulting in a considerable savings in recovery time, risk to life during the operation and hospital stay. Typically, in these procedures, slender probes are introduced via a puncture and tools such as probes with laser light sources, cameras, and instruments are fed into the body cavity. In some cases, the body cavity is distended with gas (usually the abdominal cavity) to create viewing room. The advances in active optics (CCD imagers and high resolution displays) and fiber optics have made it possible for the surgeon to have very high quality images of the inside of the body through a small aperture. Several instruments can then be used to take advantage of this vision of the inside of the body cavity: the endoscope is used for the inside of the gastro-intestinal tract, the laparoscope for the abdominal cavity, the thoracoscope for the thoracic cavity and the arthroscope for the inside of the joints. Several procedures, such as the removal of the gall bladder (cholecsystectomy) using the laparoscope, removal for biopsy of polyps in the gastro-intestinal tract using the endoscope, repairing hernias in the lung cavity using the thoracoscope, and scraping away scar tissue in the knee joint using the arthroscope, are now commonplace. Of the 600,000 cholecsystectomies performed annually in the U. S. it is thought that up to 500,000 are performed laparoscopically and, according to some practitioners, minimally invasive techniques will dominate "open" surgery in the future.

However, what *limits minimally invasive surgery is manipulator technology*, for the following reasons:

1. *Inadequate degrees of freedom.* Current needle holders, cutters and other tools transmit a surgeon's hand motion through passive mechanisms. Further, tactile feedback is disrupted. Foremost among the limitations imposed by today's tools is their limited number of degrees of freedom. For example, a needle driver that can slide, twist and pivot (up and down as well as left and right) inside the body cavity gives a surgeon only four degrees of freedom without full control of orientation. Thus, suture lines must radiate from the insertion point, since the needle can only be driven by twisting the

driver about its long axis.

2. *The need for fine motion control in surgery.* In open surgery the surgeon braces herself so as to reduce the amount of tremor transmitted to the end of the surgical device. In the instance of minimally-invasive surgery, the fulcrum at the point of entry of the instrument reduces the tremor for pivoting motions of the tool, but does not help positioning accuracy in the other directions caused by shoulder and elbow tremor.

Improved manipulators with many degrees of freedom would increase efficiency, safety, and the range of cases that could utilize these methods by addressing the drawbacks mentioned above. However, the kinematics of useful devices is complex for many reasons. To realize the gains of minimally invasive surgery, we feel that the technology of multifingered robot hands could be brought to bear. The design, construction, and control of a miniature hand-like manipulator requires significant departures from more traditional robot manipulators. Because of the small sizes of the fingers (on the order of millimeters), direct actuation of each rotary or prismatic joint is not practical. In some of our own preliminary work, we have constructed small fingers which are either controlled by cables or by small hydraulic actuators. The fabrication techniques for the manipulator are borrowed from integrated circuit technology. Although common metal and plastic materials are capable of developing biologically significant forces at this small scale, it is a challenge to develop actuators which exhibit large enough ranges of motion.

At the outset, teleoperator technology which is used to reflect the actions of the surgeon into the body cavity will be used to control the surgical manipulators. User interfaces such as sensor gloves worn by the surgeon would provide the surgeon with tactile and force feedback, while the positioning of the fingers would be measured by sensors and transmitted to small multifingered hands. One such system has been proposed by us in [19].

The growth potential of this application is enormous. Remote surgery is being explored for use on the battlefield and in space, and with greater intelligence, control, and sensing built into surgical manipulators, one can conceive of surgical workstations in the not too distant future.

3 Robots on a Small Scale: Microrobotics

In many new applications, it is necessary to handle or manipulate very small objects, for example living cells or parts of semiconductor electronics. The scale of operations that we visualize in these applications are several orders of magnitude smaller than those involved in the surgical applications of Section 2, which we have termed milli-robots. Thus, in

this section we will concentrate on micro- and nano-scale robots. There are two different approaches to dealing with these small objects: the first to use a conventional (large) manipulator with a very precise control system and the second is to miniaturize the manipulator. There are many advantages to shrinking the robots to the same scale as the parts being manipulated:

1. Delicate forces can be applied.
2. Robots can be made more accurate.
3. Robots can be fabricated using silicon processing and photolithographic techniques.

The notion of a micro-robot on a chip has been popularized by Brooks and Flynn [16] and Pister, Fearing, and co-workers [92, 93, 94]. We foresee a scenario in which these robots see wide application in micro-teleoperation in cramped areas, and in massively parallel handling of small biological and electromechanical systems. In this section, we abstract from Fearing [31] some of the technological challenges and opportunities in this rapidly growing area.

There are many engineering issues to be addressed in building micro-robotic systems: the power source, the propulsion method (if they are mobile), control integrated with sensing, and communications with the macro-world. One key new technology that provides new capabilities for sensing and actuation at the micro scale is micromachining. This is the ability to machine at very small scales, including the micron scale, using techniques from integrated circuit fabrication. This can be used, for instance, to produce actuators which have hundreds of miniature cilia (like a paramecium); or to make mechanisms like grippers that can handle parts of the size of 10μ diameter or planar rotary motors of a few microns size and sensors like miniature gyroscopes. At these sizes, forces scale differently so that electrostatic forces are stronger than electromagnetic forces. This necessitates a rethinking of actuation methods for these mechanisms.

Intelligent sensors, actuators and control systems can be integrated on a single chip. A novel integrated system for manipulating dry parts in the plane was proposed by Pister et al. [93]. This system consists of a 1 cm^2 silicon substrate with an air bearing to support individual 1 mm^2 platforms. The individual platforms are driven in the plane by electrostatic forces and can carry grippers, probes for sensing, or tools for processing. Capacitive position sensing of the platforms is added to complete the system. This system has been partially fabricated and a conceptualization of it is presented in Figure 9.1. It was designed in analogy to macro-robot manipulator called Robotworld (made by Automatix Corporation) and was made to automatically align and splice together fiber optic cables.

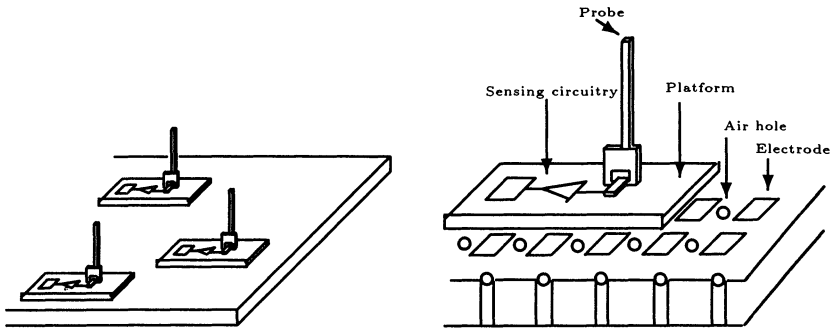


Figure 9.1: (a) View of several probing platforms floating on the same bearing surface. (b) Detailed view of a single platform. (Figures courtesy of Kristofer Pister and Ronald Fearing)

Current micro-mechanical systems are for the most part planar. However, it is clear that for manipulators to extend far beyond the surface that they are mounted on, it is important to have 3-D structures. A promising new approach in this regard is the micro-hinge method of Pister et al. [94]. This approach consists of actually fabricating components in 2-D, but providing them with the ability to rotate or slide into place resulting in the assembly of a 3-D structure. By using techniques drawn from oragami, the Japanese art of paper folding, the structures can be made to self assemble under agitation in a water bath after emergence from the silicon foundry.

In the years to come we feel that there will be an explosive growth of micro-machined robots with propulsion capabilities in fluids and with onboard robots and multifingered robots for manipulation so as to do a variety of tasks both on the biological and integrated circuit fronts.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Appendix A

Lie Groups and Robot Kinematics

Readers familiar with differential geometry will have observed that most of the analytic tools we use in this text are derived from Lie group theory and Riemannian geometry. In this appendix, we give a brief introduction to the basics of Lie group theory and its connections with rigid body kinematics.

1 Differentiable Manifolds

We begin with a brief review of differential geometry, based on the treatment given by Boothby [9]. The material here is intended primarily to fix the notation used in the subsequent sections.

1.1 Manifolds and maps

Let $U \subset \mathbb{R}^n$ and $V \subset \mathbb{R}^m$ be open sets. A mapping $f : U \to V$ is a *smooth map* if all partial derivatives of f, of any order, exist and are continuous. If $m = n$ and f is bijective and both f and f^{-1} are smooth, then f is called a *diffeomorphism* and U and V are said to be *diffeomorphic.*

A *manifold of dimension n* is a set M which is locally homeomorphic to $\mathbb{R}^n$. We parameterize the manifold by using a set of *local coordinate charts.* A local coordinate chart is a pair (ϕ, U), where ϕ is a function which maps points in the set $U \subset M$ to an open subset of $\mathbb{R}^n$. Two overlapping charts (ϕ, U) and (ψ, V) are C^∞ *related* if $\psi^{-1} \circ \phi$ is a diffeomorphism where it is defined. A collection of such charts with the additional property that the U's cover M is called a *smooth atlas.* A manifold M is a *smooth manifold* if it admits a smooth atlas.

The properties of mappings between manifolds are defined in terms of local coordinate charts. Let $F : M \to N$ be a mapping between two smooth manifolds and let (U, ϕ) and (V, ψ) be coordinate charts for M and N, respectively. The mapping $F : M \to N$ is smooth if $\tilde{F} = \psi \circ F \circ \phi^{-1} : \phi(U) \to \psi(V)$ is smooth for all choices of coordinate charts on M and N. Similarly, F is a diffeomorphism if $\tilde{F}$ is a diffeomorphism for all coordinate charts.

1.2 Tangent spaces and tangent maps

Let M be a smooth manifold of dimension n and let p be a point in M. We write $C^\infty(p)$ for the set of smooth, real-valued functions on M whose domain of definition includes some open neighborhood of p. A map $X_p : C^\infty(p) \to \mathbb{R}$ is called a *derivation* if, for all $\alpha, \beta \in \mathbb{R}$ and $f, g \in C^\infty(p)$, it satisfies

$$\begin{aligned} &\text{(i)} \quad X_p(\alpha f + \beta g) = \alpha(X_p f) + \beta(X_p g) \quad \text{(linearity)} \\ &\text{(ii)} \quad X_p(fg) = (X_p f)g(p) + f(p)(X_p g) \quad \text{(Leibniz rule)} \end{aligned}$$

The set of all derivations $X_p : C^\infty(p) \to \mathbb{R}$ defines a vector space over the reals with the operations

$$\begin{aligned} (X_p + Y_p)f &= X_p f + Y_p f \\ (\alpha X_p)f &= \alpha(X_p f). \end{aligned}$$

The *tangent space* of M at a point p, denoted T_pM, is the set of all derivations $X_p : C^\infty(p) \to \mathbb{R}$. Elements of the tangent space are called *tangent vectors*. Let (U, ϕ) be a coordinate chart on M with local coordinates $(x_1, \ldots, x_n)$. Then, the set of derivations $\{\frac{\partial}{\partial x_i}\}$ forms a basis for T_pM and hence we can write

$$X_p = X_1 \frac{\partial}{\partial x_1} + \cdots + X_n \frac{\partial}{\partial x_n}.$$

The vector $(X_1, \ldots, X_n) \in \mathbb{R}^n$ is a local coordinate representation of $X_p \in T_pM$.

Let $F : M \to N$ be a smooth map. We define the *tangent map* of F at p as the linear map $F_{*p} : T_pM \to T_{F(p)}N$ defined by

$$F_{*p}X_p(f) = X_p(f \circ F),$$

where $X_p \in T_pM$ and $f \in C^\infty(F(p))$. We also make use of the notation T_pF to denote the tangent map of F at p. The tangent map satisfies the following properties:

1. If $H = F \circ G$ is the composition of two smooth maps F and G, then $H_{*p} = F_{*G(p)} \circ G_{*p}$ or $T_pH = T_{G(p)}F \circ T_pG$.

2. If $F : M \to N$ is a diffeomorphism, then $F_{*p} : T_pM \to T_{F(p)}N$ is an isomorphism of tangent spaces with inverse $(F_{*p})^{-1} = (F^{-1})_{*F(p)}$.

If $c : \mathbb{R} \to M$ is a curve in M such that $c(0) = p$ and $\dot{c}(0) := c_{*t}(\frac{\partial}{\partial t})|_{t=0} = X_p$, then $T_pF(X_p) = \frac{d}{dt}F(c(t))|_{t=0}$.

Let M be a manifold of dimension n. The *tangent bundle* of M, denoted TM, is a manifold of dimension $2n$ defined by

$$TM = \bigcup_{p \in M} T_pM.$$

An element of TM will be written as (p, X_p) or simply X_p, where $p \in M$ and $X_p \in T_pM$. There is a natural projection $\pi : TM \to M$ given by $\pi(X_p) = p$.

1.3 Cotangent spaces and cotangent maps

Given the tangent space T_pM to a manifold M at a point p, we define the *cotangent space* of M at p, denoted T_p^*M, as the set of all linear functions $\omega_p : T_pM \to \mathbb{R}$. T_p^*M is a vector space having the same dimension as T_pM and elements of T_p^*M are called *cotangent vectors*. We write $\langle \omega_p, X_p \rangle$ for the action of a cotangent vector $\omega_p \in T_p^*M$ on a tangent vector $X_p \in T_pM$. If $\{\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\}$ is a basis for T_pM corresponding to local coordinates $(x_1, \ldots, x_n)$, the dual basis for T_p^*M is given by $\{dx_1, \ldots, dx_n\}$, where

$$\langle dx_i, \frac{\partial}{\partial x_j} \rangle = \delta_{ij}, \qquad i, j = 1, \ldots, n.$$

Given a function $f : M \to \mathbb{R}$, we define a cotangent vector $df(p) \in T_p^*M$ by

$$\langle df(p), X_p \rangle = X_p(f), \qquad X_p \in T_pM.$$

$df(p)$ is called the *differential* of f. Relative to a chart (ϕ, U) with local coordinates $x = (x_1, \ldots, x_n)$, $df(p)$ is written as

$$df(x) = \frac{\partial f}{\partial x_1}(x)dx_1 + \cdots + \frac{\partial f}{\partial x_n}(x)dx_n,$$

where $x = \phi(p)$.

Let $F : M \to N$ be a smooth map. The *cotangent map* of F is the linear map $F_p^* : T_{F(p)}^*N \to T_p^*M$ defined by

$$\langle F_p^*\alpha_{F(p)}, X_p \rangle = \langle \alpha_{F(p)}, F_{*p}X_p \rangle, \qquad \alpha_{F(p)} \in T_{F(p)}^*N, X_p \in T_pM.$$

We also make use of the notation T_p^*F to denote the cotangent map of F at p.

Let M be a manifold of dimension n. The *cotangent bundle* of a manifold M, denoted T^*M is a manifold of dimension $2n$ defined as

$$T^*M = \bigcup_{p \in M} T_p^* M.$$

An element of TM will be written as (p, α_p) or simply $\alpha_p \in T_p^* M$.

1.4 Vector fields

A smooth *vector field* X on a manifold M is defined as a smooth map $X : M \to TM$ satisfying $\pi \circ X = \mathrm{id}$, where $\pi : TM \to M$ is the canonical projection and $\mathrm{id} : M \to M$ is the identity map on M. We let $\mathfrak{X}(M)$ denote the set of all smooth vector fields on M. Relative to a coordinate chart (ϕ, U), a vector field is written as

$$X(x) = X_1(x)\frac{\partial}{\partial x_1} + \cdots + X_n(x)\frac{\partial}{\partial x_n},$$

where each X_i is a smooth function defined on an open neighborhood of $x = \phi(p)$. It is customary to write a vector field as a column vector

$$X(x) = \begin{bmatrix} X_1(x) \\ \vdots \\ X_n(x) \end{bmatrix}.$$

Vector fields represent differential equations on manifolds. Let $c : (a, b) \to M$ be a curve on the manifold. The curve c is said to be an *integral curve* of the vector field X if

$$\dot{c}(t) = X(c(t)).$$

By the existence and uniqueness theorem for ordinary differential equations, the existence of integral curves for a given nonzero vector field is guaranteed locally. The vector field is said to be *complete* if the domain of definition of the integral curves can be chosen to be $(-\infty, \infty)$. In this case, the integral curves of a vector field define a one-parameter family of diffeomorphisms $\Phi_t(q) : M \to M$ with the understanding that $\Phi_t(q)$ is the point on the integral curve starting from initial condition q at $t = 0$. This one parameter family of diffeomorphisms is referred to as the *flow* of the vector field X.

Let X be a smooth vector field and $f \in C^\infty(M)$ a smooth function on M. The *Lie derivative* of f with respect to X is a new function $Xf : M \to \mathbb{R}$ defined by

$$Xf(p) = X_p f.$$

In coordinate chart (ϕ, U), if we write $X = \sum_{i=1}^n X_i(x)\frac{\partial}{\partial x_i}$, then

$$Xf(x) = \sum_{i=1}^n \frac{\partial f}{\partial x_i} X_i(x),$$

where all partial derivatives are evaluated at $x = \phi(p)$.

Let X and Y be two smooth vector fields. The *Lie bracket* of X and Y, denoted $[X, Y]$, is a new vector field defined by

$$[X, Y]f = X(Yf) - Y(Xf).$$

It is not difficult to show that $[X, Y]$ satisfies all the properties of a derivation. In local coordinates, if we write $X = \sum_{i=1}^n X_i(q)\frac{\partial}{\partial x_i}$ and $Y = \sum_{i=1}^n Y_i(q)\frac{\partial}{\partial x_i}$, then the Lie bracket vector field $[X, Y]$ is given by

$$[X, Y] = \sum_{j=1}^n \left(\sum_{i=1}^n \frac{\partial Y_j}{\partial x_i} X_i - \frac{\partial X_j}{\partial x_i} Y_i \right) \frac{\partial}{\partial x_j}.$$

Let $F : M \to N$ be a smooth mapping between manifolds and $X \in \mathfrak{X}(M)$, $Y \in \mathfrak{X}(N)$ smooth vector fields. We say that X and Y are *F-related* if they satisfy

$$Y_{F(p)} = F_{*p} X_p.$$

If F is a diffeomorphism, then given a vector field $X \in \mathfrak{X}(M)$ we can define a new vector field $Y \in \mathfrak{X}(N)$ via the *push forward map* $F_* : TM \to TN$ defined by

$$(F_* X)_q = F_{*F^{-1}(q)} X_{F^{-1}(q)}.$$

Similarly, if $F : M \to N$ is a diffeomorphism and Y is a vector field on N, we can define a new vector field on M as

$$(F^* Y)_p = (F_{*p})^{-1} Y_{F(p)}.$$

The mapping $F^* : TN \to TM$ is called the *pull back map* for F. The pull back is related to the push forward by the formula $F^* = (F^{-1})_*$.

Proposition A.1. *Let X and Y be smooth vector fields on M and $F : M \to N$ be a smooth map. Then*

$$F_*[X, Y] = [F_* X, F_* Y].$$

A vector space V (over $\mathbb{R}$) is a *Lie algebra* if there exists a bilinear operator $V \times V \to V$, denoted $[\cdot, \cdot]$, satisfying

1. Skew-symmetry: $[v, w] = -[w, v]$ for all $v, w \in V$

2. Jacobi identity:

$$[[v,w],z] + [[z,v],w] + [[w,z],v] = 0$$

for all $v, w, z \in V$

A subspace $W \subset V$ is called a *Lie subalgebra* if $[v,w] \in W$ for all $v, w \in W$. The vector space of all smooth vector fields on a manifold M is an infinite-dimensional Lie algebra under the Lie bracket operation on vector fields.

1.5 Differential forms

A smooth *differential one-form*, on a manifold M is a smooth map $\alpha : M \to T^*M$ satisfying $\pi \circ \alpha = \mathrm{id}$ where $\pi : T^*M \to M$ is the canonical projection and $\mathrm{id} : M \to M$ is the identity map. In local coordinates, a differential form is written as

$$\alpha(x) = \alpha_1(x)dx_1 + \cdots + \alpha_n(x)dx_n$$

where each α_i is a smooth function on M.

Let $F : M \to N$ be a smooth mapping between manifolds and $\beta : N \to T^*N$ a smooth differential one-form on N. We can define a new one-form $\alpha : M \to T^*M$ by

$$\langle \alpha_p, X_p \rangle = \langle \beta_{F(p)}, F_{*p} X_p \rangle.$$

We call α the *pull back* of β by F and write $\alpha = F^*\beta$. Note that the pull back of a one-form is defined for any smooth mapping $F : M \to N$, not just diffeomorphisms.

2 Lie Groups

This section collects some basic concepts of Lie groups which prove to be useful in robot kinematics and control. A more detailed treatment of these subjects can be found in Spivak [109]. Explicit formulas for $SO(3)$ and $SE(3)$ are given by Park and Murray [89].

2.1 Definition and examples

A *Lie group* is a group G which is also a smooth manifold and for which the group operations $(g,h) \mapsto gh$ and $g \mapsto g^{-1}$ are smooth. A Lie group is *abelian* if $gh = hg$ for all $g, h \in G$. We will use the symbol e to denote the identity element of the group.

For every $g \in G$, we define *left translation* by g as the map $L_g : G \to G$ given by $L_g(h) = gh$ for $h \in G$. Similarly, *right translation* by g is defined

as the map $R_g : G \to G$ satisfying $R_g(h) = hg$. Since $L_g \circ L_h = L_{gh}$ and $R_g \circ R_h = R_{gh}$, we have that $(L_g)^{-1} = L_{g^{-1}}$ and $(R_g)^{-1} = R_{g^{-1}}$. Thus, both L_g and R_g are diffeomorphisms of G for each g. Moreover, left and right translation commute: $L_g \circ R_h = R_h \circ L_g$. If the group is abelian then $L_g = R_g$.

Example A.1. The Euclidean space under addition
The Euclidean space $\mathbb{R}^n$ with group operation $(x, y) \mapsto x + y$ is an abelian Lie group. The inverse of $x \in \mathbb{R}^n$ is denoted $-x$, and the identity element is the zero vector. Since $x + y = y + x$, such a Lie group is abelian.

Example A.2. The general linear group, $GL(n, \mathbb{R})$
The group of all $n \times n$ nonsingular real matrices is called the *general linear group* and denoted $GL(n, \mathbb{R})$. As a manifold, $GL(n, \mathbb{R})$ can be regarded as an open subset of $\mathbb{R}^{n^2}$. For $A, B \in GL(n, \mathbb{R})$, the group operation is matrix multiplication

$$(A, B) \mapsto A \cdot B$$

and inversion is given by the matrix inverse. Both operations are smooth since the formulas for the product and inverse of matrices are smooth in the matrix components. The identity element is the $n \times n$ identity matrix. Left and right translation are defined as left and right multiplication, respectively.

Example A.3. The special orthogonal group, $SO(n)$
The special orthogonal group is a subgroup of the general linear group, defined as

$$SO(n) = \{R \in GL(n, \mathbb{R}) : RR^T = I, \det R = +1\}.$$

The dimension of $SO(n)$ as a manifold is $n(n-1)/2$. For $n = 3$, the group $SO(3)$ is also referred to as the *rotation group* on $\mathbb{R}^3$.

Example A.4. The special Euclidean group, $SE(3)$
The group of rigid transformations on $\mathbb{R}^3$ is defined as the set of mappings $g : \mathbb{R}^3 \to \mathbb{R}^3$ of the form $g(x) = Rx + p$, where $R \in SO(3)$ and $p \in \mathbb{R}^3$. An element of $SE(3)$ is written as $(p, R) \in SE(3)$. $SE(3)$ can be identified with the space of 4×4 matrices of the form

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix},$$

where $R \in SO(3)$ and $p \in \mathbb{R}^3$. $SE(3)$ is a Lie group of dimension 6.

2.2 The Lie algebra associated with a Lie group

Let X be a vector field on G. X is *left invariant* if $(L_g)_* X = X$, that is

$$T_h L_g X(h) = X(gh) \quad \text{for all } h \in G.$$

Let $\mathfrak{X}_L(G)$ be the set of left invariant vector fields on G. Then, for $X, Y \in \mathfrak{X}_L(G)$ we have

$$L_{g*}[X,Y] = [L_{g*}X, L_{g*}Y] = [X,Y].$$

Thus, $\mathfrak{X}_L(G)$ is a Lie subalgebra of the Lie algebra $\mathfrak{X}(G)$, the set of all vector fields on G.

For each $\xi \in T_e G$, we define a vector field X_ξ on G by

$$X_\xi(g) = T_e L_g \xi.$$

Since

$$\begin{aligned} X_\xi(gh) &= T_e L_{gh} \cdot \xi = T_e(L_g \circ L_h) \cdot \xi \\ &= T_h L_g (T_e L_h \cdot \xi) = T_h L_g (X_\xi(h)), \end{aligned}$$

X_ξ is left invariant. The linear maps $\rho_1 : \mathfrak{X}_L(G) \to T_e G$ given by

$$\rho_1(X) = X(e)$$

and $\rho_2 : T_e G \to \mathfrak{X}_L(G)$ given by

$$\rho_2(\xi) = X_\xi$$

satisfy $\rho_1 \circ \rho_2 = \mathrm{id}_{T_e G}$ and $\rho_2 \circ \rho_1 = \mathrm{id}_{\mathfrak{X}_L(G)}$. Hence $\mathfrak{X}_L(G)$ and $T_e G$ are isomorphic as vector spaces. Defining a Lie bracket in $T_e G$ by

$$[\xi_1, \xi_2] = [X_{\xi_1}, X_{\xi_2}](e), \qquad \xi_1, \xi_2 \in T_e G \tag{A.1}$$

makes $T_e G$ into a Lie algebra. The vector space $T_e G$ with this Lie algebraic structure is called the *Lie algebra* of G and is denoted $\mathfrak{g}$.

A *Lie subalgebra* of $\mathfrak{g}$ is a subspace $\mathfrak{h} \subset \mathfrak{g}$ such that $\xi, \eta \in \mathfrak{h}$ implies that $[\xi, \eta] \in \mathfrak{h}$. It can be shown that if H is a Lie subgroup of G with Lie algebra $\mathfrak{h}$, then $\mathfrak{h}$ is a Lie subalgebra of $\mathfrak{g}$.

Example A.5. The Lie algebra of $(\mathbb{R}^n, +)$
For the group $\mathbb{R}^n$ we have $e = 0$, $T_0 \mathbb{R}^n \cong \mathbb{R}^n$, and it is easy to see that the left invariant vector field defined by $v \in T_0 \mathbb{R}^n$ is the constant vector field: $X_v(x) = v$ for all $x \in \mathbb{R}^n$. Therefore, the Lie algebra of $\mathbb{R}^n$ is $\mathbb{R}^n$ itself, with the trivial Lie bracket $[v_1, v_2] = 0$ for all $v_1, v_2 \in \mathbb{R}^n$.

Example A.6. The Lie algebra of $GL(n, \mathbb{R})$
The Lie algebra of $GL(n, \mathbb{R})$ is the set of all $n \times n$ real matrices, denoted $\mathfrak{gl}(n, \mathbb{R})$, with the bracket structure

$$[A, B] = AB - BA \qquad A, B \in \mathfrak{gl}(n, \mathbb{R}).$$

To derive this, note that $GL(n,\mathbb{R})$ is an open subset of $\mathbb{R}^{n\times n}$ and hence $T_eGL(n,\mathbb{R}) \cong \mathbb{R}^{n\times n}$. A vector field on $GL(n,\mathbb{R})$ can be written as

$$X(x) = \sum_{i,j} X_{ij}(x)\frac{\partial}{\partial x_{ij}}$$

where each x_{ij} represents a coordinate in $\mathbb{R}^{n\times n}$. One can now proceed to define left-invariant vector fields on $\mathbb{R}^{n\times n}$ (relative to left matrix multiplication) and compute the Lie bracket from equation (A.1). See [109, pp. 509–511] for the details of this calculation.

Example A.7. The Lie algebra of $SO(3)$
The Lie algebra of $SO(3)$, denoted $so(3)$, may be identified with the 3×3 skew-symmetric matrices of the form

$$\widehat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{A.2}$$

with the bracket structure

$$[\widehat{\omega}_1, \widehat{\omega}_2] = \widehat{\omega}_1\widehat{\omega}_2 - \widehat{\omega}_2\widehat{\omega}_1, \qquad \widehat{\omega}_1, \widehat{\omega}_2 \in so(3).$$

We can identify $so(3)$ with $\mathbb{R}^3$ using the mapping in equation (A.2), which maps a vector $\omega \in \mathbb{R}^3$ to a matrix $\widehat{\omega} \in so(3)$. It is straight forward to show that

$$[\widehat{\omega}_1, \widehat{\omega}_2] = (\omega_1 \times \omega_2)^\wedge, \qquad \omega_1, \omega_2 \in \mathbb{R}^3.$$

Thus $\omega \mapsto \widehat{\omega}$ is a Lie algebra isomorphism between the Lie algebra $(\mathbb{R}^3, \times)$ and the Lie algebra $(so(3), [\cdot,\cdot])$.

Example A.8. The Lie algebra of $SE(3)$
The Lie algebra of $SE(3)$, denoted $se(3)$, can be identified with 4×4 matrices of the form

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \qquad \omega, v \in \mathbb{R}^3$$

with the bracket structure $[\widehat{\xi}_1, \widehat{\xi}_2] = \widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1$. Let

$$\widehat{\xi}_1 = \begin{bmatrix} \widehat{\omega}_1 & v_1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \widehat{\xi}_2 = \begin{bmatrix} \widehat{\omega}_2 & v_2 \\ 0 & 0 \end{bmatrix}.$$

Then

$$[\widehat{\xi}_1, \widehat{\xi}_2] = \widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1 = \begin{bmatrix} (\omega_1 \times \omega_2)^\wedge & \omega_1 \times v_2 - \omega_2 \times v_1 \\ 0 & 0 \end{bmatrix}.$$

The vector space $se(3)$ is isomorphic to $\mathbb{R}^6$ via the mapping $\widehat{\xi} \mapsto \xi = (v, \omega) \in \mathbb{R}^6$.

2.3 The exponential map

For every $\xi \in T_eG$, let $\phi_\xi : \mathbb{R} \to G$ denote the integral curve of the left invariant vector field X_ξ passing through e at $t = 0$. That is, $\phi_\xi(0) = e$ and $\frac{d}{dt}\phi_\xi(t) = X_\xi(\phi_\xi(t))$. It follows from its definition that

$$\phi_\xi(s+t) = \phi_\xi(s)\phi_\xi(t),$$

which means that $\phi_\xi(t)$ is a one-parameter subgroup of G. Indeed, as functions of t, both sides are equal at $t = 0$ and both satisfy the differential equation $\dot{\sigma}(t) = X_\xi(\sigma(t))$ by left invariance of X_ξ, so they are equal by the uniqueness of integral curves. Using either left invariance or the fact that $\phi_\xi(t+s) = \phi_\xi(t)\phi_\xi(s)$ shows that $\phi_\xi(t)$ is defined for all $t \in \mathbb{R}$.

The function $\exp : T_eG \to G$ defined by $\exp(\xi) = \phi_\xi(1)$ is called the *exponential map* of the Lie algebra $\mathfrak{g}$ into G. The exponential map takes the line $\xi s \in \mathfrak{g}$, $s \in \mathbb{R}$, into a one-parameter subgroup of G, i.e.,

$$\exp(\xi s) = \phi_\xi(s).$$

To verify this formula, note that for fixed s, the curve $t \mapsto \phi_\xi(ts)$ which at $t = 0$ passes through e, satisfies the differential equation

$$\frac{d}{dt}\phi_\xi(ts) = sX_\xi(\phi_\xi(ts)) = X_{s\xi}(\phi_{s\xi}(t)).$$

Since $\phi_{s\xi}(t)$ and $\phi_\xi(ts)$ satisfy the same differential equation and both pass through e at $t = 0$, it follows that $\phi_{s\xi}(t) = \phi_\xi(ts)$. Putting $t = 1$ yields $\exp(\xi s) = \phi_\xi(s)$. In fact, all connected one-parameter subgroups of G are obtained in this way.

Differentiating the map $\exp(\xi s) = \phi_\xi(s)$ with respect to s at $s = 0$ shows that $T_e \exp = \mathrm{id}_{T_eG}$. Therefore, by the inverse function theorem, $\exp : \mathfrak{g} \to G$ is a local diffeomorphism from a neighborhood of zero in $\mathfrak{g}$ onto a neighborhood of e in G. If G is compact it can be shown that the exponential map is surjective.

Example A.9. The exponential map on $(\mathbb{R}^n, +)$
Consider $\mathbb{R}^n$ with addition as the group operation. The Lie algebra of $\mathbb{R}^n$ is $\mathbb{R}^n$ with the trivial bracket and the integral curve of a left invariant vector field $X_v(x) = v$ is given by $\phi_v(t) = vt$. Thus, $\phi_v(t) \circ x = x + vt$ and $\exp : \mathbb{R}^n \to \mathbb{R}^n$ is the identity.

Example A.10. The exponential map on $GL(n, \mathbb{R})$
Let $G = GL(n, \mathbb{R}) \subset \mathbb{R}^{n \times n}$ so that $\mathfrak{g} = \mathfrak{gl}(n, \mathbb{R}) = \mathbb{R}^{n \times n}$ with $[A, B] = AB - BA$. For every $A \in \mathfrak{gl}(n, \mathbb{R})$, the mapping $\Phi_A : \mathbb{R} \to GL(n, \mathbb{R})$ given by

$$\Phi_A(t) = \sum_{n=0}^{\infty} \frac{t^n A^n}{n!}$$

is a one-parameter subgroup because $\Phi_A(0) = I$ and

$$\frac{d}{dt}\Phi_A(t) = \sum_{n=1}^{\infty} \frac{t^{n-1}A^n}{(n-1)!} = \Phi_A(t)A,$$

which shows that Φ_A is an integral curve of the left invariant vector field X_A. Therefore, the exponential map $\exp : \mathfrak{gl}(n,\mathbb{R}) \to GL(n,\mathbb{R})$ is given by

$$\exp(A) = \Phi_A(1) = \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

Example A.11. The exponential map on $SO(3)$
Let $G = SO(3)$. It was shown in Chapter 2 that $\exp\widehat{\omega}$ corresponds to a rotation about the vector $\omega \in \mathbb{R}^3$ by an angle $\|\omega\|$. An explicit formula is given by Rodrigues's formula:

$$e^{\widehat{\omega}} = I + \frac{\widehat{\omega}}{\|\omega\|}\sin\|\omega\| + \frac{\widehat{\omega}^2}{\|\omega\|^2}(1 - \cos\|\omega\|). \tag{A.3}$$

Example A.12. The exponential map on $SE(3)$
For $G = SE(3)$, the Lie algebra can be identified with 4×4 matrices of the form

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}, \qquad \omega, v \in \mathbb{R}^3,$$

with $[\widehat{\xi}_1, \widehat{\xi}_2] = \widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1$. The exponential map is given by

$$\exp\widehat{\xi} = \begin{bmatrix} I & v \\ 0 & 1 \end{bmatrix}, \quad \omega = 0 \qquad \text{and} \qquad \exp\widehat{\xi} = \begin{bmatrix} e^{\widehat{\omega}} & Av \\ 0 & 1 \end{bmatrix}, \quad \omega \neq 0,$$

where

$$A = I + \frac{\widehat{\omega}}{\|\omega\|^2}(1 - \cos\|\omega\|) + \frac{\widehat{\omega}^2}{\|\omega\|^3}(\|\omega\| - \sin\|\omega\|).$$

The exponential map $\exp : \mathfrak{g} \to G$ defined by $\exp(\xi) = \phi_\xi(1)$, is a local diffeomorphism from a neighborhood of zero in $\mathfrak{g}$ onto a neighborhood of e in G. Thus, restricted to a small neighborhood U of e there is a function $\log : U \to \mathfrak{g}$ such that $\exp \circ \log(g) = g$ for all $g \in U$. The function $\log : U \subset G \to \mathfrak{g}$ is the inverse of $\exp : \mathfrak{g} \to G$. For the general linear group, it can be computed explicitly.

Example A.13. Log function on $GL(n,\mathbb{R})$
Let $G = GL(n,\mathbb{R})$ and $A \in G$. Then, the log function is defined by the following matrix polynominal

$$\log A = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(A-I)^n}{n},$$

which converges for all $\|A - I\| < 1$.

Example A.14. Log function on $SO(3)$
Let $G = SO(3)$. Then the log function is given by $\log R = \widehat{a} = \theta\widehat{\omega}$, where $\theta \in \mathbb{R}$ and $\widehat{\omega} \in se(3)$ are given by

$$2\cos\theta + 1 = \text{trace}(R) \quad \text{and} \quad \widehat{\omega} = \frac{1}{2\sin\theta}(R - R^T) \qquad R \neq I.$$

When $R = I$, $\theta = 2\pi k$ for any integer k and ω can be chosen arbitrarily. Note that the log function on $SO(3)$ is multi-valued since θ is not unique.

Example A.15. Log function on $SE(3)$
The log function on $SE(3)$ is given by

$$\widehat{\xi} = \log \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & A^{-1}p \\ 0 & 0 \end{bmatrix},$$

where $\widehat{\omega} = \log R$ and

$$A^{-1} = I - \frac{1}{2}\widehat{\omega} + \frac{2\sin\|\omega\| - \|\omega\|(1 + \cos\|\omega\|)}{2\|\omega\|^2 \sin\|\omega\|}\widehat{\omega}^2 \qquad \omega \neq 0.$$

If $\omega = 0$ then $A = I$. The log function on $SE(3)$ is multi-valued since ω is not unique.

2.4 Canonical coordinates on a Lie group

Since $\exp : \mathfrak{g} \to G$ is a local diffeomorphism, we can use the exponential map to define local coordinates for G. Let $\{X_1, \ldots, X_n\}$ be a basis of $\mathfrak{g}$. The mapping

$$g = \exp(X_1\sigma_1 + \cdots + X_n\sigma_n) \tag{A.4}$$

defines a local diffeomorphism between the real numbers $\sigma \in \mathbb{R}^n$ and $g \in G$ for g sufficiently near the identity. Hence we can consider σ as a coordinate mapping, $\sigma : U \to \mathbb{R}^n$, where $U \subset G$ is an arbitrarily small neighborhood of the origin. Using this coordinate chart and left translation, we can construct an entire atlas for the Lie group G. Define a chart (U_g, ψ_g) about $g \in G$ by letting

$$U_g = L_g(U) = \{L_g h | h \in U\}$$

and defining $\psi_g = \sigma \circ L_{g^{-1}} : U_g \to \mathbb{R}^n$, so that

$$\psi_g(h) = \sigma(g^{-1}h).$$

It is not difficult to show that the set of charts $\{(U_g, \psi_g)\}$ indeed forms an atlas for G. The functions $(\sigma_1, \ldots, \sigma_n)$ defined in equation (A.4) are called *canonical coordinates of the first kind* around the identity and relative to the basis $\{X_1, \ldots, X_n\}$.

If we write

$$g = \exp X_1\theta_1 \exp X_2\theta_2 \cdots \exp X_n\theta_n \tag{A.5}$$

for g near the identity, then the functions $(\theta_1, \theta_2, \ldots, \theta_n)$ are called the *canonical coordinates of the second kind.* Examples of such coordinates are the product of exponentials formula studied in Chapter 3, and the Euler angle parameterizations of the rotation group.

2.5 Actions of Lie groups

In Chapter 2 we transformed points, vectors, twists, and wrenches using matrix multiplication with either g or some form of Ad_g. All of these transformations can be described as the left action of $SE(3)$ on an appropriate space. In this section, we give the definition of a left action of a Lie group on a manifold and give several examples related to robot kinematics. More specific examples which make connections with the material in Chapter 2 are presented in the next section.

Let M be a smooth manifold and G a Lie group. A *left action* of G on M is a smooth map $\Phi : G \times M \to M$ such that

(i) $\Phi(e, x) = x$ for all $x \in M$

(ii) For every $g, h \in G$ and $x \in M$, $\Phi(g, \Phi(h, x)) = \Phi(gh, x)$

We will often write $\Phi_g(x)$ for $\Phi(g, x)$. If Φ is an action of G on M and $x \in M$, the *orbit* of x is defined by

$$\mathrm{Orb}_x = \{\Phi_g(x) : g \in G\}.$$

Example A.16. Action of $GL(n, \mathbb{R})$ on $\mathbb{R}^n$
$GL(n, \mathbb{R})$ acts on $\mathbb{R}^n$ by $(A, x) \mapsto Ax$. The orbit of $x \neq 0$ is the open set $\mathbb{R}^n / \{0\}$.

Example A.17. Action of G on itself via conjugation
The map $I_g : G \to G$ given by $I_g(h) = R_{g^{-1}} L_g(h) = ghg^{-1}$ is called the *conjugation map* or the *inner automorphism* associated with g. The map I_g defines a left action on G since $I_e = \mathrm{id}$ and

$$I_g \circ I_h(x) = ghxh^{-1}g^{-1} = I_{gh}(x).$$

Orbits of this action are called *similarity classes.*

Example A.18. Adjoint action of G on its Lie algebra
Differentiating the conjugation map I_g at e, we get the *adjoint action* of G on $\mathfrak{g}$, $\mathrm{Ad} : G \times \mathfrak{g} \to \mathfrak{g}$ defined as

$$\mathrm{Ad}_g(\xi) = (T_e I_g)\xi = T_e(R_g^{-1} \circ L_g)\xi.$$

If G is a subgroup of $GL(n, \mathbb{R})$ then $\mathfrak{g} \subset \mathbb{R}^{n \times n}$ and $\mathrm{Ad}_g \xi = g\xi g^{-1}$ for $\xi \in \mathfrak{g}$.

Example A.19. Coadjoint action of G on the dual of its Lie algebra
The *coadjoint action* of G on $\mathfrak{g}^*$, the dual of the Lie algebra $\mathfrak{g}$ of G, is defined as follows. Let $\mathrm{Ad}_g^* : \mathfrak{g}^* \to \mathfrak{g}^*$ be the dual of Ad_g defined by

$$\langle \mathrm{Ad}_g^* \alpha, \xi \rangle = \langle \alpha, \mathrm{Ad}_g \xi \rangle$$

for $\alpha \in \mathfrak{g}^*$ and $\xi \in \mathfrak{g}$. Then the map $\Phi^* : G \times \mathfrak{g}^* \to \mathfrak{g}^*$ given by

$$\Phi^*(g, \alpha) = \mathrm{Ad}_{g^{-1}}^* \alpha$$

is the coadjoint action of G on $\mathfrak{g}^*$.

Example A.20. Lifted action of G from M to TM
Let $\Phi : G \times M \to M$ be an action of G on M, where $\Phi_g : M \to M$ is defined by $\Phi_g(x) = \Phi(g, x)$. One can lift this action to an action on TM, $\Phi_* : G \times TM \to TM$, defined by

$$\Phi_*(g, (x, v_x)) = (\Phi_g(x), T_x \Phi_g \cdot v_x).$$

Φ_* is called the *lifted action* of G on TM.

3 The Geometry of the Euclidean Group

In this section we study the geometric properties of the Euclidean group and discuss their implications on robot kinematics and control. The material in this section is based in part on the dissertation of Loncaric [64] and also on the work of the authors [60, 78, 87].

3.1 Basic properties

In Chapter 2 we presented the theory of rigid body motion and showed its connections with homogeneous matrices and the theory of screws. We now show that the various tools available in the study of rigid motion are special cases of the more general tools defined for general Lie groups.

Rigid body kinematics

The configuration of a rigid body with respect to some reference configuration is described by an element $g = (p, R) \in SE(3)$. If A is a fixed coordinate frame and B a frame attached to the rigid body, then we write $g_{ab} = (p_{ab}, R_{ab}) \in SE(3)$ to denote the configuration of B with respect to A. p_{ab} represents the location of the origin of the B frame and $R_{ab} \in SO(3)$ its orientation. The group operation on $SE(3)$ allows us to determine the configuration of a frame C relative to A via an intermediate frame B:

$$g_{ac} = g_{ab} \cdot g_{bc} = (p_{ab} + R_{ab} p_{bc}, R_{ab} R_{bc}).$$

If we represent $g \in SE(3)$ as a 4×4 homogeneous matrix,

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix},$$

then the group operation is given by matrix multiplication and we may regard $SE(3)$ as a subgroup of the general linear group, $GL(4, \mathbb{R})$.

The configuration $g_{ab} \in SE(3)$ can also be interpreted as a mapping from the coordinates of a point written relative to the B frame into the coordinates of the same point written relative to the A frame. Formally, this defines an action of $SE(3)$ on $\mathbb{R}^3$ given by $\Phi_g(q) = p + Rq$. In homogeneous coordinates this action can be written as

$$\begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix}.$$

It follows from associativity of matrix multiplication that this actually defines an action of $SE(3)$ on $\mathbb{R}^3$. The use of a 1 in the last row of the homogeneous representation for a point allows the action of $SE(3)$ on points to be represented as multiplication between a matrix and a vector.

The action of $SE(3)$ on vectors describes how the velocity of a point is mapped from one coordinate frame to another. Formally, we represent the velocity of a point as an element of $T_x\mathbb{R}^3$ and the action of $SE(3)$ on tangent vectors (velocities) is the lifted action of $SE(3)$ on $M = \mathbb{R}^3$. The lifted action of $SE(3)$ on $T\mathbb{R}^3$ is given by $\Phi_{g*}(v_q) = (g(q), Rv_q)$ where $g(q)$ denotes the action of g on the point q. In homogeneous coordinates, the tangent space (velocity) portion of the action can be written as

$$\begin{bmatrix} v_a \\ 0 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ 0 \end{bmatrix}.$$

By defining the homogeneous representation of a vector to have a zero in the bottom row, we are able to once again use multiplication of a matrix and a vector to represent the action.

Since $SE(3)$ is a Lie group, the exponential map can be used to map elements of the Lie algebra into the group. In homogeneous coordinates, the Lie algebra of $SE(3)$ is a Lie subalgebra of $\mathfrak{gl}(4, \mathbb{R})$ consisting of matrices of the form

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \qquad \widehat{\omega} \in so(3),\ v \in \mathbb{R}^3,$$

with the Lie bracket given by the matrix commutator. We call an element of the Lie algebra $se(3)$ a *twist*. The vector space $se(3)$ has dimension 6 and is isomorphic to $\mathbb{R}^6$ via the mapping $\widehat{\xi} \mapsto \xi = (\omega, v)$.

A twist can be interpreted geometrically using the theory of screws. Consider the motion generated by simultaneously rotating and translating about an axis in the direction $\omega \in \mathbb{R}^3$ going through a point $q \in \mathbb{R}^3$.

Let h represent the ratio of translational motion to rotational motion. If h is finite, then the resulting rigid motion is the exponential of the twist $\widehat{\xi} \in se(3)$ given by

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & -q \times \omega + h\omega \\ 0 & 0 \end{bmatrix}.$$

The one-parameter subgroup $\phi_{\widehat{\xi}}(\theta) = \exp(\widehat{\xi}\theta)$ generated by this twist corresponds to a rotation about an axis followed by translation along that same axis. Thus the exponential of a twist generates a screw motion.

It follows from the general properties of the exponential map that, near the identity, any element of $SE(3)$ can be written as the exponential of some twist. For $SE(3)$ it can be shown that the exponential map is actually surjective and hence any rigid transformation can be written as the exponential of some twist. This statement may be regarded as a restatement of Chasles' theorem, which states that every rigid motion can be realized as a screw motion.

Although every rigid motion can be written as the exponential of a twist, the set of twists do not define a parameterization for $SE(3)$. The exponential map is not injective, and hence there may be many twists which give the same rigid motion. One example of this is a pure rotation, which can be written as either a rotation about an axis ω by an angle θ or a rotation about an axis $-\omega$ by an angle $2\pi - \theta$. These two different twists give the same motion.

Velocities and forces

If $c(t) \in M$ is a curve on a manifold, then the velocity of that curve is an element of the tangent space to M at $c(t)$, i.e., $\dot{c}(t) \in T_{c(t)}M$. If M happens to be a Lie group, then the tangent space T_gG is isomorphic to T_eG. Hence, by left translation, we can identify the instantaneous velocity of a trajectory on a Lie group with an unique element of the corresponding Lie algebra.

Returning to $SE(3)$, there are two ways to map $T_gSE(3)$ to $T_eSE(3)$—left and right translation. Consider left translation first. We make use of the fact that since $SE(3)$ can be viewed as a matrix Lie group (using homogeneous coordinates), so the tangent map to $L_h : SE(3) \to SE(3)$ is given by matrix multiplication:

$$T_gL_hX_g = hX_g,$$

where $X_g \in T_gSE(3)$. To show this relationship, let $g(t) \in SE(3)$ be a curve which is tangent to X_g at time $t = 0$. Then from the definition of the tangent map,

$$T_gL_h\dot{g}(0) = \frac{d}{dt}\left(L_h \circ g(t)\right)|_{t=0} = h\dot{g}(0).$$

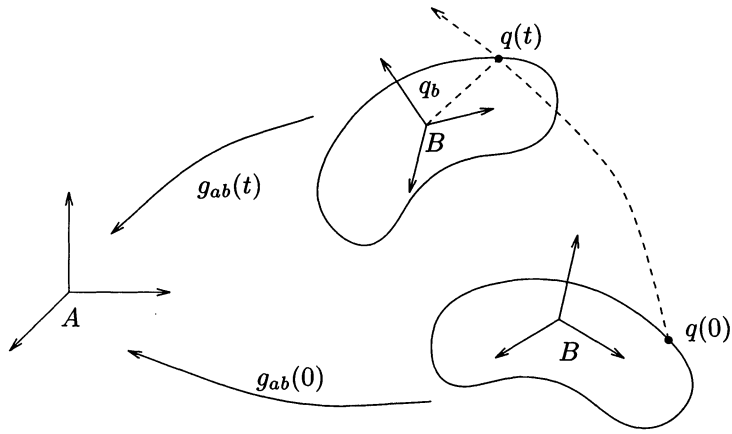


Figure A.1: Trajectory of a rigid body relative to a fixed frame.

Letting $h = g^{-1}$ we see that, using left translation, the velocity of the rigid body can be shifted to the identity and written as a twist

$$\widehat{V}^b = g^{-1}\dot{g}.$$

This is precisely the *body velocity* that was defined in Chapter 2.

If we use right translation to map a velocity $\dot{g} \in T_g SE(3)$ to the tangent space at the identity, the resulting quantity is the *spatial velocity:*

$$\widehat{V}^s = \dot{g}g^{-1}.$$

The derivation of this formula follows exactly as in the body velocity case, replacing L_h with R_h.

One reason for the terminology "body" velocity and "spatial" velocity becomes clear if we consider the action of twists on points. Let $q \in \mathbb{R}^3$ be a point attached to a rigid body and let $g_{ab}(t) \in SE(3)$ describe the trajectory of a frame B attached to the rigid body relative to a fixed reference frame A, as shown in Figure A.1. In homogeneous coordinates, the trajectory of the point q as a function of time can be written as

$$q_a(t) = g_{ab}(t)q_b,$$

where q_a and q_b are the coordinates of the point relative to the A and B frames. The velocity of the point, relative to the A frame, is given by

$$v_a(t) = \dot{q}_a(t) = \dot{g}_{ab}(t)q_b, \tag{A.6}$$

where we have used the fact that q_b is constant since the point is fixed in the body frame. Thus, $\dot{g}_{ab}(t) \in T_g SE(3)$ can be viewed as a mapping

between the body coordinates of a point and the spatial velocity of that same point.

A more appealing representation of velocity is one which does not require switching between coordinate frames. That is, suppose we wish to find the relationship between the coordinates of a point and its velocity, when both quantities are specified with respect to a single frame. We can accomplish this by transforming either the coordinates of the point or the coordinates of velocity to the appropriate frame. For example, if we are given the coordinates of the point q with respect to the spatial frame A, then the velocity of q with respect to A is given by

$$v_a = \dot{g} q_b = (\dot{g} g^{-1}) q_a.$$

This is precisely the spatial velocity that is defined by using right translation to pull back the velocity $\dot{g} \in T_g SE(3)$ to $T_e SE(3)$. A similar argument shows that the body velocity, $g^{-1}\dot{g}$ can be viewed as a map from the body coordinates of a point to the body velocity of that point.

The body and spatial velocities are physically interpreted as the instantaneous translational and rotational velocity written relative to the body or spatial frame, respectively. They are related to one another by the adjoint action of $SE(3)$ on $se(3)$. Namely, if $\widehat{V}^b$ is the body velocity for a rigid motion $g(t)$, then the spatial velocity is given by

$$\widehat{V}^s = \mathrm{Ad}_g \widehat{V}^b.$$

This relationship can be derived by direct calculation, as in Chapter 2.

A generalized force on $SE(3)$ can be regarded as a covector on $SE(3)$; i.e., an element of $T_g^* SE(3)$. As with velocities, we can map the cotangent space $T_g^* SE(3)$ onto the dual of the Lie algebra by either left or right translation. We call the resulting object a *wrench*. Left translation corresponds to representing the force in the body coordinate frame while right translation corresponds to representing it in the spatial coordinate frame. The natural action of a wrench on a twist gives the instantaneous work due to applying the given wrench along the infinitesimal motion generated by the twist.

If we identify $se(3)^*$ with $\mathbb{R}^6$, then a wrench can be written as

$$F = \begin{bmatrix} f \\ \tau \end{bmatrix},$$

where f is the translation component of the force and τ is the angular component. The natural action of a wrench on a twist becomes the inner product between $F \in \mathbb{R}^6$ and $V \in \mathbb{R}^6$. It is important to keep in mind, however, that this action is the natural action of a covector on a vector, and is defined independently of any inner product structure on $\mathbb{R}^6 \cong se(3)$.

Although wrenches are dual to twists, we choose to represent them slightly differently. We always represent a wrench on $SE(3)$ as a vector in $\mathbb{R}^6$ since the matrix representation of a wrench does not prove to be particularly useful. Furthermore, we write a wrench using a single subscript to denote the frame with respect to which it is written. Thus, if A is an inertial reference frame and B is a frame attached to a rigid body, we write F_a for the spatial representation of a wrench applied to the rigid body and F_b for the body representation of the wrench. In taking the action of a wrench on a twist, one must always insure that the twist and the wrench are represented relative to the same frame. Thus the instantaneous work generated by a twist V and a wrench F can be written either as $V^s_{ab} \cdot F_a$ or $V^b_{ab} \cdot F_b$.

Transformation laws

Let $V^s_{bc} \in \mathbb{R}^6$ be the spatial velocity of a frame C relative to an inertial frame B and suppose that we wish to know the spatial velocity with respect to a different inertial frame, A. Using the definition of the spatial velocity, we have

$$\begin{aligned}\widehat{V}^s_{ac} &= \dot{g}_{ac}\, g^{-1}_{ac} = g_{ab}\, \dot{g}_{bc}\, g^{-1}_{bc}\, g^{-1}_{ab} \\ &= g_{ab}\, \widehat{V}^s_{bc}\, g^{-1}_{ab} = \mathrm{Ad}_{g_{ab}}\, \widehat{V}^s_{bc},\end{aligned}$$

and hence the velocity of the rigid body is transformed according to the adjoint action of $SE(3)$ on $se(3)$. If we represent the velocities as vectors in $\mathbb{R}^6$ then we write $V^s_{ac} = \mathrm{Ad}_{g_{ab}}\, V^s_{bc}$ where $\mathrm{Ad}_g : \mathbb{R}^6 \to \mathbb{R}^6$ is the matrix

$$\mathrm{Ad}_g = \begin{bmatrix} R & \widehat{p}R \\ 0 & R \end{bmatrix}.$$

Note that here we use the symbol Ad_g to represent the adjoint mapping both on $se(3) \subset \mathbb{R}^{4\times 4}$ and on $\mathbb{R}^6$, which is isomorphic to $se(3)$.

Suppose instead that we change the body coordinate frame. Let g_{ab} represent the configuration of the frame B relative to A and let g_{bc} represent a fixed transformation corresponding to a new choice of body frame. The spatial velocity of C with respect to A is given by

$$\widehat{V}^s_{ac} = \dot{g}_{ac} g^{-1}_{ac} = \dot{g}_{ab} g_{bc} g^{-1}_{bc} g^{-1}_{ab} = \widehat{V}^s_{ab}.$$

Thus, changing the body coordinate frame does not affect the spatial velocity of the object.

Similar relationships can be derived for body velocities. Changing the spatial coordinate frame does not affect the body velocity of a rigid object. However, if g_{ab} represents the motion of the rigid body with respect to the frame A and g_{bc} represents a new choice of body frame, then

$$\widehat{V}^b_{ac} = \mathrm{Ad}_{g^{-1}_{bc}}\, \widehat{V}^b_{ab}.$$

Thus the adjoint action of $SE(3)$ on $se(3)$ represents the effect of a change of body frame on the body velocity.

The transformations described above assume that new and old body or spatial coordinate frames have a fixed relative configuration. If we are given two rigid motions $g_{ab}(t)$ and $g_{bc}(t)$, then in order to compute the velocity of frame C relative to frame A, we must add the velocities between the frames. This addition must occur in a single coordinate frame, and hence we use the adjoint mapping to transform the velocities appropriately. For example, the spatial velocity between frames A and C is given by

$$V^s_{ac} = V^s_{ab} + \mathrm{Ad}_{g_{ab}} V^s_{bc}.$$

The adjoint mapping in the second term converts the instantaneous velocity V^s_{bc}, which is written in the coordinates of frame B, into an instantaneous velocity written relative to frame A.

The coadjoint action is used to model the transformation of wrenches. If $F_a \in \mathbb{R}^6 \cong se(3)^*$ is a wrench written relative to a fixed frame A, then the coordinates of the wrench relative to a new fixed frame, B, are given by

$$F_b = \mathrm{Ad}^T_{g^{-1}_{ab}} F_a$$

where g_{ab} is the configuration of frame B relative to A. This expression follows directly from the definition of the coadjoint action given in the previous section.

3.2 Metric properties of $SE(3)$

Since the Lie algebra of $SE(3)$ can be identified with $\mathbb{R}^6$, there is an inner product structure on $se(3)$ induced by the usual inner product on $\mathbb{R}^6$. However, it turns out that this inner product is not invariant under change of coordinate frame and hence can be misleading. Suppose, for example, that we are given two twists $\xi_1 \in \mathbb{R}^6$ and $\xi_2 \in \mathbb{R}^6$ that satisfy $\xi_1 \cdot \xi_2 = 0$. If we transform the coordinate frame with respect to which the twists are written, then the twists transform as $\xi'_i = \mathrm{Ad}_g \xi_i$, where $g \in SE(3)$ represents the change of frame. The inner product between the twists in the new coordinate frame is given by

$$\xi'_1 \cdot \xi'_2 = \xi_1^T \begin{bmatrix} R^T & 0 \\ -R^T\hat{p} & R^T \end{bmatrix} \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix} \xi_2 = \xi_1^T \begin{bmatrix} I & R^T\hat{p}R \\ -R^T\hat{p}R & I - R^T\hat{p}^2R \end{bmatrix} \xi_2.$$

If $p \neq 0$, then the dot product is not preserved and hence two twists which are orthogonal relative to one choice of coordinate frame may not be orthogonal relative to a different choice of frame.

This lack of frame independence has caused some confusion in the robotics literature, due to incorrect use of the inner product on $\mathbb{R}^6$ as an inner product in $se(3)$. In this section we show that there is no inner

product structure on $se(3)$ which is invariant under change of coordinate frame. The implications of this fact are also discussed.

Invariant metrics on $\mathbb{R}^n$

Consider first the Euclidean space $\mathbb{R}^n$. A *metric* Ψ on $\mathbb{R}^n$ is a bilinear function $\Psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that

1. Ψ is symmetric: $\Psi(v, w) = \Psi(w, v)$, for all $v, w \in \mathbb{R}^n$.
2. Ψ is positive-definite: $\Psi(v, v) \geq 0$ and $\Psi(v, v) = 0$ if and only if $v = 0$.

Let $\{e_1, \ldots, e_n\}$ be an orthonormal basis of $\mathbb{R}^n$ (note that such a basis defines a Cartesian coordinate frame in $\mathbb{R}^n$). With respect to this basis Ψ has the matrix representation

$$\Psi_{ij} = \Psi(e_i, e_j), \quad i, j = 1, \ldots, n.$$

From now on, we will use Ψ to represent both the mapping and its matrix representation. Thus $\Psi(v, w) = v^T \Psi w$.

Because a metric is positive definite, the matrix representation of a metric is always invertible. We will occasionally be interested in symmetric bilinear mappings which have invertible matrix representations, but are not positive definite. A bilinear function $\Psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is *nondegenerate* if $\Psi(v, w) = 0$ for all $w \in \mathbb{R}^n$ implies that $v = 0$. We call a bilinear mapping which is symmetric and nondegenerate, but not necessarily positive-definite, a *pseudo-metric.*

Let v_a be the representation of a vector in frame A and v_b be the representation of the same vector in frame B. These vectors are related by the lifted action of $SE(n)$ on $\mathbb{R}^n$ and hence

$$v_a = R_{ab} v_b, \tag{A.7}$$

where $g_{ab} = (p_{ab}, R_{ab}) \in SE(n)$ is the position and orientation of B relative to A. A metric is said to be *invariant* under change of coordinate frames if

$$\Psi(Rv_1, Rv_2) = \Psi(v_1, v_2) \qquad \forall v_1, v_2 \in \mathbb{R}^n,\ R \in SO(n). \tag{A.8}$$

Equation (A.8) yields the following constraint on the matrix representation of Ψ:

$$\Psi R = R \Psi \qquad \forall R \in SO(n). \tag{A.9}$$

Lemma A.2. *Let $\Psi \in \mathbb{R}^{n \times n}$ be a symmetric matrix which satisfies equation* (A.9) *for all $R \in SO(n)$ (i.e., Ψ commutes with all $n \times n$ rotation matrices). Then*

$$\Psi = \alpha I$$

for some $\alpha \in \mathbb{R}$.

Proof. Let v be a unit eigenvector for Ψ with eigenvalue λ. Multiplying both sides of equation (A.9) by v, we have

$$\Psi Rv = R\Psi v = \lambda Rv.$$

It follows that Rv is an eigenvector of Ψ for *any* $R \in SO(n)$ and hence $\Psi w = \lambda w$ for any unit vector $w \in \mathbb{R}^n$. This is only possible if Ψ is a scalar times the identity matrix. □

The scalar α corresponds to a choice of *length scale* on $\mathbb{R}^n$. In the case of $\mathbb{R}^3$, a length scale assigns a physical unit such as "meters" or "feet" to a unit vector in $\mathbb{R}^3$. Once a fixed length scale is chosen, there is a unique metric on $\mathbb{R}^3$ which is invariant under change of coordinate frames.

Proposition A.3. Invariant metrics on $\mathbb{R}^n$
Up to a choice of length scale, the Euclidean space $\mathbb{R}^n$ has a unique positive definite metric which is invariant under change of coordinate frames.

Invariant metrics on $SE(3)$

A natural question that follows from the preceding discussion is whether $SE(3)$ also supports a metric that is invariant under change of coordinate frames. If so, we can associate a length measure to twists, generalizing the notion of *speed* in $\mathbb{R}^3$ to $se(3)$. Since $SE(3)$ is not a Euclidean space, but a manifold, we need to define what we mean by a metric on a manifold.

Let M be a manifold. A *quadratic form* Ψ on M is a bilinear mapping $\Psi_x : T_xM \times T_xM \to \mathbb{R}$ which is symmetric for all x and varies smoothly with x. If Ψ is positive definite then we call Ψ a *metric*. If Ψ is non-degenerate but not positive-definite, we say that Ψ is a *pseudo-metric*. Let $F : M \to M$ be a smooth map. The pull back of a quadratic form Ψ by F is also a quadratic form, denoted $F^*\Psi$ defined by

$$F^*\Psi_x(v_x, w_x) = \Psi_{F(x)}(T_xF\, v_x, T_xF\, w_x), \quad v_x, w_x \in T_xM.$$

Ψ is said to be *F-invariant* if $F^*\Psi = \Psi$.

To define a metric on a Lie group, we may first define it at the identity and then translate it to the whole group by either left or right translation. The resulting metric is automatically left or right invariant, respectively. However, a left invariant metric may not be right invariant and vice versa. A metric that is both left and right invariant is called *bi-invariant*. Let $I_h : SE(3) \to SE(3)$ be the conjugation map

$$I_h(g) = hgh^{-1}.$$

Since $I_h = R_{h^{-1}} \circ L_h$, a metric Ψ is bi-invariant if and only if it is I_h-invariant for all $h \in SE(3)$. That is, we require that a bi-invariant metric satisfy

$$I_h^*\Psi = \Psi, \qquad \forall h \in SE(3).$$

These same definitions hold for any quadratic form on $SE(3)$, even if it is not positive definite.

Lemma A.4. *Let Ψ be a left (or right) invariant quadratic form on $SE(3)$. Then Ψ is I_h-invariant if and only if it is I_h-invariant at the identity, i.e.,*

$$\Psi_e(\widehat{\xi},\widehat{\eta}) = \Psi_e(\mathrm{Ad}_h\,\widehat{\xi}, \mathrm{Ad}_h\,\widehat{\eta})$$

for all $h \in SE(3)$ and $\widehat{\xi},\widehat{\eta} \in se(3)$.

Proof. The proof of necessity is immediate. To prove sufficiency, we only need to show that Ψ is right invariant since $I_h = R_{h^{-1}} \circ L_h$. For this, observe the following equations, which hold for all $\widehat{\xi},\widehat{\eta} \in se(3)$ and $h \in SE(3)$:

$$\begin{aligned}
\Psi_e(\widehat{\xi},\widehat{\eta}) &= \Psi_e(T_eI_h\widehat{\xi},\, T_eI_h\widehat{\eta}) \\
&= \Psi_e(T_e(L_h \circ R_{h^{-1}})\widehat{\xi},\, T_e(L_h \circ R_{h^{-1}})\widehat{\eta}) \\
&= \Psi_e(T_{h^{-1}}L_h \circ T_eR_{h^{-1}}\widehat{\xi},\, T_{h^{-1}}L_h \circ T_eR_{h^{-1}}\widehat{\eta}) \\
&= \Psi_{h^{-1}}(T_eR_{h^{-1}}\widehat{\xi},\, T_eR_{h^{-1}}\widehat{\eta}).
\end{aligned}$$

Here, the first equation follows from Ψ being I_h-invariant at the identity, and the last equation follows from left invariance of Ψ. This shows that Ψ is right invariant and hence bi-invariant. □

Proposition A.5. Bi-invariant quadratic forms on $SE(3)$
Let Ψ be a left invariant quadratic form on $SE(3)$, where we identify $se(3)$ with $\mathbb{R}^6$. Then Ψ is bi-invariant if and only if the matrix representation of Ψ at e has the form

$$\Psi_e = \begin{bmatrix} 0 & \beta I \\ \beta I & \gamma I \end{bmatrix}, \qquad \beta,\gamma \in \mathbb{R}.$$

Proof. By Lemma A.4, Ψ is bi-invariant if and only if for all $\widehat{\xi},\widehat{\eta} \in se(3)$ and $h \in SE(3)$,

$$\Psi_e(T_eI_h\widehat{\xi},\, T_eI_h\widehat{\eta}) = \Psi_e(\widehat{\xi},\, \widehat{\eta}). \tag{A.10}$$

Identifying $se(3)$ with $\mathbb{R}^6$ and representing Ψ_e by the matrix

$$\Psi_e = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

where $A,B,C \in \mathbb{R}^{3\times 3}$ and A,C are symmetric, we have from equation (A.10) that

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \mathrm{Ad}_h^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \mathrm{Ad}_h, \qquad \forall h \in SE(3). \tag{A.11}$$

Using the definition of the adjoint mapping on $\mathbb{R}^6$,

$$\mathrm{Ad}_h = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix},$$

equation (A.11) becomes

$$\begin{aligned} A &= R^T A R \\ B &= R^T A \hat{p} R + R^T B R \\ C &= -R^T \hat{p} A \hat{p} R - R^T \hat{p} B R + R^T B^T \hat{p} R + R^T C R. \end{aligned} \tag{A.12}$$

By Lemma A.2, A in equation (A.12) must be of the form

$$A = \alpha I, \qquad \alpha \in \mathbb{R}.$$

Since (A.12) holds for all $h \in SE(3)$, by letting $p = 0$ in the second and third equations we obtain

$$B = \beta I \quad C = \gamma I \qquad \beta, \gamma \in \mathbb{R}.$$

Finally, using $B = \beta I$ and $C = \gamma I$ in the third equation of (A.12), we conclude that $\alpha = 0$ and the theorem is proven. □

Among the set of bi-invariant quadratic forms, we have the following two special cases.

Example A.21. Hyperbolic metric on $SE(3)$: ($\beta = 1$, $\gamma = 0$) The quadratic form whose matrix representation given by

$$\langle \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix}, \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} \rangle_H = (v_1^T \omega_2 + v_2^T \omega_1) \tag{A.13}$$

is called the *hyperbolic metric* on $SE(3)$. The eigenvalues of the matrix representation of $\langle \cdot, \cdot \rangle_H$ are all ± 1 and occur pairwise. Hence it is possible for a vector to have negative "length" with respect to the hyperbolic metric.

The choice of a hyperbolic metric on $SE(3)$ depends on a choice of length scale on $\mathbb{R}^6$. This is clear from the formula in equation (A.13), where a change of length scale results in a linear factor in the hyperbolic metric. Thus, the choice of a particular hyperbolic metric fixes the ratio between a unit translation vector and a unit rotation vector.

Although it is not positive definite, the hyperbolic metric is non-degenerate and hence it defines a pseudo-metric on $SE(3)$. In fact, $\langle \cdot, \cdot \rangle_H$ coincides with the reciprocal product defined in Chapter 2. The hyperbolic metric is a special case of the *Klein form*, which can be defined on any Lie group [13].

Example A.22. Killing form on $SE(3)$: ($\beta = 0$, $\gamma = -4$)
The quadratic form given by

$$\langle \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix}, \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} \rangle_K = -4\omega_1^T \omega_2 \tag{A.14}$$

is called the *Killing form* on $SE(3)$. It is degenerate, and hence defines neither a metric nor a pseudo-metric. The Killing form on $SE(3)$ does not depend on the choice of length scale.

The Killing form on $SE(3)$ is a special case of the Killing form for a general Lie group, defined as

$$\langle \xi_1, \xi_2 \rangle_K = \text{trace}(\text{ad}_{\widehat{\xi}_1} \cdot \text{ad}_{\widehat{\xi}_2}),$$

where $\text{ad}_{\widehat{\xi}} : \mathfrak{g} \to \mathfrak{g} : \widehat{\eta} \mapsto [\widehat{\xi}, \widehat{\eta}]$. The factor of -4 in equation (A.14) was chosen to make it agree with the general definition.

Observe that in neither of the above cases is the quadratic form positive definite. In fact, it is easy to see that for any bi-invariant metric $\Psi((v,0),(v,0)) = 0$. Hence we have the following:

Corollary A.5.1. Lack of a bi-invariant metric on $SE(3)$
There does not exist a bi-invariant (positive-definite) metric on $SE(3)$.

The fact that a bi-invariant metric does not exist on $SE(3)$ does not mean that one cannot define a notion of length on $SE(3)$. Rather, it implies that the definition of a metric is not intrinsic. It involves a *choice*. If we restrict ourselves to metrics which are defined at the identity and extended to the entire group via left (or right) translation, we have the following characterization:

Proposition A.6. Left invariant metrics on $SE(3)$
The set of all left- (or right-) invariant metrics on $SE(3)$ are parameterized by the choice of reference frame origin and the choice of length scale.

A proof of this proposition can be found in [64]. The need for a choice of reference frame origin follows from the lack of a bi-invariant metric on $SE(3)$. The choice of length scale affects how translational motion and rotational motion are weighted in the metric.

We discuss some implications of the preceding results on robot kinematics and control in the following examples.

Example A.23. Geometric attributes of twists
Let $\xi = (v, \omega)$ be a twist with $\omega \neq 0$ (the exponential of such a twist is referred to as a *proper* screw motion). The magnitude M_ξ and pitch h_ξ of

the twist, as defined in Chapter 2, can be re-interpreted using the Killing form and the hyperbolic metric as follows:

$$M_\xi = -\frac{1}{4}\langle \xi, \xi \rangle_K \qquad \text{and} \qquad h_\xi = -2\frac{\langle \xi, \xi \rangle_H}{\langle \xi, \xi \rangle_K}.$$

Thus, both quantities are coordinate-frame independent attributes of a twist with nonzero angular component.

The pitch of a screw obviously depends on the choice of length scale since it is defined as the ratio of translational and rotational motion. This extra freedom is reflected in the choice of length scale which is required for the hyperbolic metric.

Example A.24. Constraints on $SE(3)$ and hybrid force control Consider the hybrid control problem, in which a manipulator pushes against a surface in $\mathbb{R}^3$. This surface can be described as a set of holonomic constraints of the form

$$f_i(g) = 0 \qquad i = 1, \ldots, k,$$

where each $f_i : SE(3) \to \mathbb{R}$ is a smooth function. Differentiating this position constraint with respect to time, we get a constraint on $\dot{g} \in T_g SE(3)$:

$$df_i(g)\dot{g} = \langle df_i(g), \dot{g} \rangle = 0, \tag{A.15}$$

where $df_i(g) \in T_g^* SE(3)$ and $\dot{g} \in T_g SE(3)$. This constraint can be translated back to the identity using left (or right) translation,

$$\begin{aligned} \langle df_i(g), \dot{g} \rangle &= \langle (df_i(g), (L_g)_* (L_{g^{-1}})_* \dot{g} \rangle \\ &= \langle L_g^* df(g), (L_{g^{-1}})_* \dot{g} \rangle \\ &= \langle L_g^* df_i(g), \widehat{\xi} \rangle = 0, \end{aligned}$$

where $\widehat{\xi} = g^{-1}\dot{g} \in se(3)$ and $L_g^* df_i(g) \in se(3)^*$.

The quantity $L_g^* df_i(g)$ is an element of $se(3)^*$, the dual of the Lie algebra, and hence we can interpret it as a force. The subspace of $se(3)^*$ spanned by $L_g^* df_1, \ldots, L_g^* df_k$ is called the *space of constraint forces*, and the subspace of $se(3)$ annihilated by $L_g^* df_i, i = 1, \ldots, m$, is called the *space of free motions*. These subspaces annihilate each other since any constraint force acting on a free motion gives zero instantaneous work. Although the subspaces themselves are not invariant under change of coordinates, it is always true that these subspaces will annihilate each other, independent of the choice of frames used to represent the problem.

A common mistake in the robotics literature is to treat the constraint directions as twists. In this case, one tries to define the action of a constraint force on a velocity as the usual dot product on $\mathbb{R}^6$. Constraint

forces are then identified with those twists that are orthogonal to the space of free motions. However, *this description is not frame invariant.* If we change the frames which are used to describe the problem, the space of constraint twists may no longer be orthogonal to the free motions. This is because there is no invariant metric on $SE(3)$ and hence there is no invariant notion of orthogonality.

This apparent problem is completely due to associating the constraint directions with vectors instead of covectors. The proper way to describe a constraint is as an element of the dual of the Lie algebra. In this case, if we shift coordinate frames, then a twist $V \in \mathbb{R}^6 \cong se(3)$ and a wrench $F \in \mathbb{R}^6 \cong se(3)^*$ will transform according to

$$V' = \mathrm{Ad}_g V \qquad \text{and} \qquad F' = \mathrm{Ad}_{g^{-1}}^T F.$$

The action of the new twist on the new wrench is given by

$$F' \cdot V' = (F')^T \mathrm{Ad}_{g^{-1}} \mathrm{Ad}_g V = F \cdot V.$$

Note that the action of a wrench on a twist also scales correctly with respect to a change of length scale. So a wrench which annihilates a twist does so independently of the units in which length is described.

Example A.25. Left invariant constraints
The constraints described in the previous example generate a subspace of twists and wrenches which annihilate each other. These subspaces are defined on the Lie algebra and its dual, but they may vary as the manipulator moves along the constraint surface. That is, the constraint $df(g) \in T_g^* SE(3)$ pulled back to the identity, $L_g^* df(g) \in T_e^* SE(3)$, may depend on $g \in SE(3)$.

A constraint is said to be *left invariant* if $L_g^* df(g) = df(e)$. In other words, a left invariant constraint gives directions of free motion which are constant relative to a body fixed frame. Left invariant constraints arise frequently in applications, for example when moving across a flat surface. The following result, due to Loncaric [64] asserts that all left (or right) invariant constraints can be characterized in terms of subgroups of $SE(3)$.

Proposition A.7 (Loncaric). *A constraint $f(g) = 0$ is left (or right) invariant if and only if $f^{-1}(0)$ is a subgroup of $SE(3)$.*

Example A.26. Manipulability measures
In Chapter 3, we defined several manipulability measures based on properties of the manipulator Jacobian. Some of these manipulability measures rely on the Euclidean metric on $\mathbb{R}^6 \cong se(3)$, which we now know is not invariant under change of coordinate frames. Thus, care should be exercised when one applies these manipulability measures to manipulator design and control.

To illustrate our point, let us place two tool frames T_1 and T_2 at the end-effector of a robot and then write down the (body) manipulator Jacobian at a joint position θ. Let $J_1(\theta)$ and $J_2(\theta)$ be the representation of the body manipulator Jacobian with respect to frame T_1 and T_2, respectively. Then, we have

$$J_1(\theta) = \mathrm{Ad}_h J_2(\theta),$$

where $h \in SE(3)$ is the position and orientation of T_2 relative to T_1. Since Ad_h does not preserve the Euclidean metric, we have in general

$$\sigma_{\min}(J_1(\theta)) \neq \sigma_{\min}(J_2(\theta)).$$

Furthermore, the point in the joint space at which the minimum singular value of J_1 is achieved could be different from that of J_2. This shows that manipulator design based on maximizing the minimum singular value of J is ill-posed, as the result depends on placement of the body coordinate frames.

A similar result holds for the condition number.

3.3 Volume forms on $SE(3)$

In the last section we saw that there does not exist a bi-invariant metric on $SE(3)$ and hence the definition of the length of a curve on $SE(3)$ depends on the choice of a reference frame and a length scale. One would like to know to what extent the choice of a reference frame and length scale affects the definition of *volume* on $SE(3)$. This is important, for example, when talking about the size of the workspace of a robot manipulator, which is a compact subset of $SE(3)$. We begin with a very brief review of the definition of volume on a manifold. See Boothby [9] for a more complete treatment.

Let M be a manifold of dimension n. A *volume form* Ω on M is a skew-symmetric multilinear map

$$\Omega_p : \underbrace{T_pM \times \cdots T_pM}_{n} \to \mathbb{R}$$

such that $\Omega_p \neq 0$ for all $p \in M$. The volume form defines the volume of a parallelepiped formed by n tangent vectors. By integrating the volume form over a manifold (using local coordinates), the volume of the manifold (or a subset of the manifold) can be defined. If $M = \mathbb{R}^n$ then the standard volume form which corresponds to ordinary integration is given by the determinant function, as follows:

$$\Omega(V_1, \ldots, V_n) = \det \begin{bmatrix} V_1 & \cdots & V_n \end{bmatrix} \qquad V_i \in \mathbb{R}^n.$$

Let $F : M \to M$ be a smooth map from M to itself. The *pull back* of Ω by F, is a volume form $F^*\Omega$ defined by

$$F^*\Omega_x(v_1, \ldots, v_n) = \Omega_{F(x)}(T_xFv_1, \ldots, T_xFv_n).$$

In local coordinates, if J is the Jacobian of F evaluated at x and $\det J$ the determinant of J, then

$$F^*\Omega = (\det J)\Omega. \tag{A.16}$$

Thus the pull back of a volume form generalizes the change of variables formula from ordinary calculus. A volume form on $SE(3)$ is said to be *left invariant* if $(L_g)^*\Omega = \Omega$ and *right invariant* if $(R_g)^*\Omega = \Omega$. A volume form which is both left and right invariant is called *bi-invariant.*

Theorem A.8. Bi-invariant volume forms on $SE(3)$
Let Ω be a left invariant volume form on $SE(3)$. Then Ω is also right invariant and hence bi-invariant.

Sketch of proof. In order to show that $I_h^*\Omega = \Omega$ it suffices to show that

$$\Omega(e) = \det(T_eI_h)\Omega(e).$$

However, the Jacobian of I_h at the identity is just the adjoint mapping associated with h, and computing the determinant of Ad_h gives

$$\det \mathrm{Ad}_h = \det \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix} = 1.$$

Hence Ω is bi-invariant. Right invariance follows from bi-invariance and left invariance. □

A corollary of this theorem is that $SE(3)$ supports a bi-invariant volume form, since we can define the volume element on $se(3) \cong \mathbb{R}^6$ and then use left (or right) translation to define a volume form on all of $TSE(3)$. One possible choice for such a volume form is to multiply the standard volume form $\Omega_{\mathbb{R}^3}$ on $T_e\mathbb{R}^3 = \mathbb{R}^3$ with a volume form on $so(3)$, the Lie algebra of $SO(3)$. It can be shown that a volume form on $SO(3)$ can be defined by making use of the Lie bracket and Killing form on $SO(3)$:

$$\Omega_{SO(3)}(\hat{\omega}_1, \hat{\omega}_2, \hat{\omega}_3) = -\frac{1}{4}\langle \hat{\omega}_1, [\hat{\omega}_2, \hat{\omega}_3] \rangle_K = \omega_1 \cdot \omega_2 \times \omega_3.$$

As seen from the last equality, this corresponds to the triple product if we identify $so(3)$ with $\mathbb{R}^3$ and, in fact, this is equivalent to the volume form defined by using the standard volume on $\mathbb{R}^3$. With this definition, $SO(3)$ has a volume of $8\pi^3$ rad^3.

A volume form defined by combining the volume on $\mathbb{R}^3$ with that on $SO(3)$, i.e., $\Omega_{SE(3)} = \Omega_{\mathbb{R}^3} \times \Omega_{SO(3)}$, clearly depends on the choice of

length scale. In essence, this describes the tradeoff between translational volume and angular volume. However, once a length scale has been fixed, the volume form will not depend on the choice of coordinate frames. Also, since changing the length scale results in a corresponding scaling of volume, one can still say that one volume is larger than another, independent of this choice.

Example A.27. Well-posed manipulability measures
The use of the determinant of $J(\theta)$ as a measure of manipulability was described in Chapter 3. This manipulability measure makes use of a volume form on $SE(3)$ defined in terms of the standard volume on $\mathbb{R}^6 \cong se(3)$. Using the notation defined in Example A.26, the coordinate independence of this measure can be seen from the relationships

$$\det(J_1(\theta)) = \det(\mathrm{Ad}_h\, J_2(\theta)) = \det \mathrm{Ad}_h \det J_2(\theta) = \det J_2(\theta).$$

Example A.28. Optimal manipulator design [86, 87]
Consider a robot manipulator consisting of six revolute joints (commonly called a *6R manipulator*). Let

$$g(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_6\theta_6} g(0)$$

be the forward kinematics map of the manipulator relative to some reference configuration, and l_{ξ_i} be the twist axis associated with the i^{th} joint. Let C_ξ denote the set of all curves which join the axes of the joints of the manipulator. Thus $c \in C_\xi$ is a curve which links the axes of each ξ_i and, without loss of generality, is parameterized such that $c(i) \cap l_{\xi_i} \neq 0$. In other words, $c : [1, 6] \to \mathbb{R}^3$ and $c(i)$ is a point on the axis of the i^{th} joint twist. The line seqments $c_{[i,i+1]}$, $i = 1, \ldots, 5$, are referred to as *links* of the manipulator, and c_t denotes c evaluated at t. We define the *manipulator length* as

$$L_M := \min_{c \in C_\xi} \sum_{i=2}^{6} \|c_i - c_{i-1}\|.$$

It is left as an exercise to show that L_M is independent of the choice of reference configurations.

The work volume V_M of the manipulator is given by the following volume integral

$$V_M = \int_{g(Q)} \Omega,$$

where $Q = \mathbb{S}^1 \times \cdots \times \mathbb{S}^1 = \mathbb{T}^6$ is the manipulator joint space, g is the forward kinematics map, and Ω is a bi-invariant volume form on $SE(3)$. If we use the volume form $\Omega_{SE(3)} = \Omega_{\mathbb{R}^3} \times \Omega_{SO(3)}$, the manipulator work volume in terms of the manipulator length, L_M, is

$$V_M \leq \frac{4}{3}\pi (L_M)^3 \cdot 8\pi^3.$$

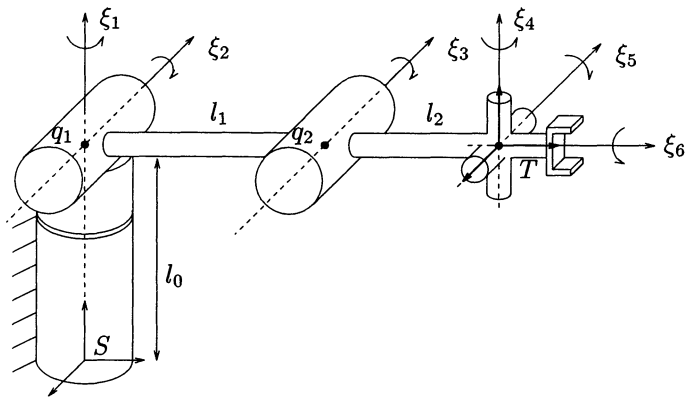


Figure A.2: An elbow manipulator

A manipulator consisting of six revolute joints has maximal work volume if it achieves this bound. This bound is independent of choice of base and tool frames, but it does depend on the choice of length scale. However, as noted above, this choice of scale does not affect comparisons between volumes, and hence we can use this choice of volume to define an optimal manipulator.

For the case of 6R manipulators, it is possible to completely describe the class of manipulators which achieve maximal work volume. Define an *elbow manipulator* to be one which consists of a shoulder, an elbow, and a wrist aligned as shown in Figure A.2. The wrist and shoulder joints of an elbow manipulator consist of mutually orthogonal axes intersecting at a point. The inverse to an elbow manipulator is one in which the wrist and the shoulder joints are switched (so that the "wrist" is located at the base of the manipulator).

Theorem A.9. Optimal manipulator design
A 6R manipulator with given length L_M has maximal work volume if and only if it is an elbow manipulator or the inverse of an elbow manipulator, with the elbow midway between the shoulder and the wrist.

A detailed discussion and proof of this result is given by Paden and Sastry [86, 87]. The wrist at one end of the manipulator insures that all orientations can be reached at any configuration. The location of the elbow, at the midpoint between the shoulder and wrist, insures that there is not a hole in the center of the workspace.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Appendix B

A Mathematica Package for Screw Calculus

This appendix contains a brief description of a Mathematica package, `Screws.m`, which facilitates the use of screws, twists, and wrenches for analyzing robot kinematics. The `Screws` package implements all of the functions described in Chapter 2 and, when combined with the supplementary package `RobotLinks.m`, allows symbolic and numerical computation of the kinematics of open-chain robot manipulators as well as many other functions. The Mathematica program itself is described in [121].

The `Screws` package is available via anonymous ftp from the host `avalon.caltech.edu` and may be used free of charge. Documentation and installation instructions are included with the source code for the package. The `Screws` package was written by R. Murray and S. Sur at the California Institute of Technology. All correspondence concerning the software should be sent to via e-mail to `murray@avalon.caltech.edu`. The authors assume no responsibility for the correctness or maintenance of the `Screws` package. The source code is currently available *only* via anonymous ftp.

The remainder of this appendix contains a brief description of the `Screws` package, describing the functions which are available and their syntax. Although not strictly necessary, some familiarity with Mathematica is assumed. This appendix can also be used as a guide for implementing a screw calculus package in other symbolic and numerical programming languages.

Using the Screws package

The `Screws` package implements screw theory in 3-dimensional Euclidean space, $\mathbb{R}^3$. It uses homogeneous coordinates to represent points, vectors,

and rigid motions, making it easy to integrate into other Mathematica packages.

The `Screws` package consists of two groups of functions. The first group operates on rotation matrices and implements all of the mathematical operations described in Section 2 of Chapter 2. The following functions are defined for computing in $SO(3)$:

- `AxisToSkew[w]`
  Generate a skew-symmetric matrix give a vector `w` $\in \mathbb{R}^3$.
- `RotationAxis[R]`
  Calculate the axis of rotation for a matrix `R` $\in SO(3)$.
- `SkewExp[S, theta]`
  Calculate the exponential of a skew-symmetric matrix. If `theta` is not specified, it defaults to 1. If the first argument to `SkewExp` is a vector, `SkewExp` first converts it to a skew-symmetric matrix and then takes its exponential.
- `SkewToAxis[S]`
  Generates a vector given a skew-symmetric matrix.

Limited error checking is used to insure that the arguments to the functions are in the proper form.

The second group of functions implements calculations on $SE(3)$. Rigid body transformations are represented using 4×4 matrices. Functions are provided for transforming points and vectors to and from homogeneous coordinates, as well as converting a translation and rotation pair into a 4×4 matrix. The following functions are defined for use in $SE(3)$:

- `HomogeneousToTwist[xi]`
  Convert `xi` from a 4×4 matrix to a 6-vector.
- `PointToHomogeneous[q]`
  Generate the homogeneous representation of a point `q` $\in \mathbb{R}^3$.
- `RigidAdjoint[g]`
  Generate the adjoint matrix corresponding to `g`.
- `RigidOrientation[g]`
  Extract the rotation matrix `R` from a homogeneous matrix `g`.
- `RigidPosition[g]`
  Extract the position vector `p` from a homogeneous matrix `g`.
- `RigidTwist[g]`
  Compute the twist `xi` $\in \mathbb{R}^6$ which generates the homogeneous matrix `g`.

- `RPToHomogeneous[R,p]`
  Construct a 4×4 homogeneous matrix from a rotation matrix `R` and a translation `p`.

- `ScrewToTwist[h, q, w]`
  Return the twist coordinates of a screw with pitch `h` through the point `q` and in the direction `w`. If `h == Infinity`, then a pure translational twist is generated. In this case, `q` is ignored and `w` gives the direction of translation.

- `TwistAxis[xi]`
  Compute the axis of the screw corresponding to a twist. The axis is represented as a pair `{q, w}`, where `q` is a point on the axis and `w` is a unit vector describing the direction of the axis. The twist `xi` can be specified either as a 6-vector or a 4×4 matrix.

- `TwistExp[xi, theta]`
  Compute the matrix exponential of a twist `xi`. The default value of `theta` is 1. If the first argument to `TwistExp` is a 6-vector, it is automatically converted to a 4×4 matrix.

- `TwistPitch[xi]`
  Compute the pitch of a twist.

- `TwistMagnitude[xi]`
  Compute the magnitude of a twist.

- `TwistToHomogeneous[xi]`
  Convert `xi` from a 6-vector to a 4×4 matrix.

- `VectorToHomogeneous[q]`
  Generate the homogeneous representation of a vector.

Limited error checking is used to insure that the arguments to the functions are in the proper form.

Manipulator kinematics

The functions defined in the `Screws` package can be used to analyze the kinematics of a robot manipulator. This section describes this process and defines some new functions which streamline the analysis of manipulator kinematics. These functions are contained in the package `RobotLinks.m`, which is included with in `Screws` package distribution.

The forward kinematics for a robot manipulator can be written as a product of exponentials (of twists). The following functions are defined for creating twists specifically for robot manipulators:

- `RevoluteTwist[q, w]`
  Construct the unit twist corresponding to a revolute joint in the direction `w` going through the point `q`.

- `PrismaticTwist[q, w]`
  Construct the unit twist corresponding to a prismatic joint in the direction `w` going through the point `q`.

These functions use the `ScrewToTwist` function defined in `Screws.m`.

Once the twists are defined, the forward kinematic map and the manipulator Jacobian can be calculated using matrix multiplication combined with the `TwistExp` and `RigidAdjoint` functions. These computations are automated by the following functions:

- `ForwardKinematics[{xi1, th1}, {xi2, th2}, ..., gst0]`
  Compute the forward kinematics map using the product of exponentials formula. The pairs `{xi, th}` define the joint twist and joint angle (or displacement) for each joint of the manipulator.

- `SpatialJacobian[{xi1, th1}, {xi2, th2}, ..., gst0]`
  Compute the spatial manipulator Jacobian for the manipulator. The pairs `{xi, th}` are given as in the `ForwardKinematics` function.

An example of the usage of `Screws` and `RobotLinks` packages is shown below for computing the kinematics of a SCARA manipulator. The notation corresponds to the notation used to describe the SCARA manipulator in Chapter 2.

```
<<Screws.m                          (* screws package       *)
<<RobotLinks.m                      (* additional functions *)

(* Twist axes for SCARA robot, starting from the base *)
xi1 = RevoluteTwist[{0,0,0},     {0,0,1}];      (* base  *)
xi2 = RevoluteTwist[{0,l1,0},    {0,0,1}];      (* elbow *)
xi3 = RevoluteTwist[{0,l1+l2,0}, {0,0,1}];      (* wrist *)
xi4 = PrismaticTwist[{0,0,0},    {0,0,1}];

(* Location of the tool frame at reference configuration *)
gst0 = RPToHomogeneous[IdentityMatrix[3], {0,l1+l2,0}];

(* Forward kinematics map *)
gst = Simplify[
  ForwardKinematics[
    {xi1,th1}, {xi2,th2}, {xi3,th3}, {xi4,th4}, gst0
  ]
];
```

```
(* Spatial manipulator Jacobian *)
Js  = Simplify[
  SpatialJacobian[{xi1,th1}, {xi2,th2}, {xi3,th3}, {xi4,th4}, gst0]
];
```

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Bibliography

[1] J. Angeles. *Rational Kinematics.* Springer-Verlag, 1988.

[2] H. Asada and J. J. Slotine. *Robot Analysis and Control.* John Wiley, 1986.

[3] K. J. Astrom and B. Wittenmark. *Computer Controlled Systems: Theory and Design.* Prentice-Hall, 1984.

[4] J. Baillieul. Geometric methods for nonlinear optimal control problems. *Journal of Optimization Theory and Applications*, 25(4):519–548, 1978.

[5] J. Baillieul and D. Martin. Resolution of redundancy. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41*, pages 49–90. American Mathematical Society, 1990.

[6] R. S. Ball. *A Treatise on the Theory of Screws.* Cambridge University Press, 1900.

[7] A. Bicchi, J. K. Salisbury, and D. L. Brock. Contact sensing from force measurements. *International Journal of Robotics Research*, 12(3):249–262, 1993.

[8] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch. Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control*, 37(11):1746–1757, 1992.

[9] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, second edition, 1986.

[10] O. Bottema and B. Roth. *Theoretical Kinematics.* North-Holland, 1979.

[11] R. W. Brockett. Control theory and singular Riemannian geometry. In P. Hinton and G. Young, editors, *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.

[12] R. W. Brockett. Robotic manipulators and the product of exponentials formula. In P. A. Fuhrman, editor, *Mathematical Theory of Networks and Systems*, pages 120–129. Springer-Verlag, 1984.

[13] R. W. Brockett, editor. *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41.* American Mathematical Society, 1990.

[14] R. W. Brockett and L. Dai. Non-holonomic kinematics and the role of elliptic functions in constructive controllability. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 1–22. Kluwer, 1993.

[15] R. W. Brockett, A. Stokes, and F. Park. A geometrical formulation of the dynamical equations describing kinematic chains. In *IEEE International Conference on Robotics and Automation*, pages 637–642, 1993.

[16] R. A. Brooks and A. M. Flynn. Rover on a chip. *Aerospace America*, pages 22–26, October 1989.

[17] P. Chiacchio and B. Siciliano. A closed-loop Jacobian transpose scheme for solving the inverse kinematics of nonredundant and redundant wrists. *Journal of Robotics Systems*, 6(5):601–630, 1989.

[18] D. S. Childress. Artificial hand mechanisms. In *Mechanisms Conference and International Symposium on Gearing and Transmissions*, San Francisco, CA, October 1972.

[19] M. Cohn, D. C. Deno, S. S. Sastry, and J. Wendlandt. Actuating and force-sensing for cable-driven, teleoperated manipulators. In *Medicine Meets Virtual Reality*, San Diego, 1992. Aligned Managment Associates.

[20] P. Coiffet et al. *Robot Technology.* McGraw-Hill, 1983. Translation from the French, *Les Robots* (8 volumes).

[21] J. J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley, second edition, 1989.

[22] M. R. Cutkosky. *Robotic Grasping and Fine Manipulation.* Kluwer, 1985.

[23] A. De Luca and G. Oriolo. The reduced gradient method for solving redundancy in robot arms. *Robotersysteme*, 7(2):117–122, 1991.

[24] J. Demmel, G. Lafferriere, J. Schwartz, and M. Sharir. Theoretical and experimental studies using a multifinger planar manipulator. In *IEEE International Conference on Robotics and Automation*, pages 390–395, 1988.

[25] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, pages 215–221, June 1955.

[26] D. C. Deno, R. M. Murray, K. S. J. Pister, and S. S. Sastry. Finger-like biomechanical robots. In *IEEE International Conference on Robotics and Automation*, 1992.

[27] M. P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.

[28] J. Duffy. *Analysis of Mechanisms and Robot Manipulators.* Edward Arnold Ltd., London, 1980.

[29] J. Duffy and C. Crane. A displacement analysis of the general spatial 7R mechanism. *Mechanisms and Machine Theory*, 15:153–169, 1980.

[30] A. G. Erdman and G. N. Sandor. *Mechanism Design: Analysis and Synthesis.* Prentice-Hall, 1984.

[31] R. S. Fearing. Micro structures and micro actuators for implementing sub-millimeter robots. In H. S. Tzou and T. Fukuda, editors, *Precision Sensor, Actuators and Systems.* Kluwer Academic Publishers, 1992.

[32] C. Fernandes, L. Gurvits, and Z. Li. Attitude control of a space platform manipulator system using internal motion. *International Journal of Robotics Research*, 1994. (to appear).

[33] C. Fernandes, L. Gurvits, and Z. Li. Near optimal nonholonomic motion planning for a system of coupled rigid bodies. *IEEE Transactions on Automatic Control*, March 1994.

[34] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, second edition, 1991.

[35] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw Hill, 1987.

[36] B. Gorla and M. Renaud. *Modeles des Robots Manipulateurs: applications á leur commande*. Cepadues-Éditions, 1984.

[37] M. Grayson and R. Grossman. Models for free nilpotent Lie algebras. *Journal of Algebra*, 135(1):177–191, 1990.

[38] M. Hall. *The Theory of Groups*. Macmillan, 1959.

[39] H. Hanafusa and H. Asada. A robotic hand with elastic fingers and its application to assembly process. In M. Brady et al., editor, *Robot Motion: Planning and Control*, pages 337–360. MIT Press, 1982.

[40] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, AC–22:728–740, 1977.

[41] G. Hinton. Some computational solutions to Bernstein's problems. In H. T. A. Whiting, editor, *Human Motor Actions—Bernstein Reassessed*, chapter 4b. Elsevier Science Publishers B.V., 1984.

[42] K. H. Hunt. *Kinematic Geometry of Mechanisms*. Oxford University Press, 1978.

[43] T. Iberall and T. Venkataraman, editors. *Dextrous Robot Hands*. Springer-Verlag, 1988.

[44] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 2nd edition, 1989.

[45] S. Jacobsen, J. Wood, K. Bigger, and E. Iverson. The Utah/MIT hand: Work in progress. *International Journal of Robotics Research*, 4(3):21–50, 1984.

[46] A. Jain and G. Rodriguez. Recursive flexible multibody system dynamics using spatial operators. *Journal of Guidance, Control, and Dynamics*, 15(6):1453–1466, 1992.

[47] W. Kahan. Lectures on computational aspects of geometry. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Unpublished, July 1983.

[48] I. Kao and M. R. Cutkosky. Quasi-static manipulation with compliance and sliding. *International Journal of Robotics Research*, 11(1):20–40, 1992.

[49] J. Kerr. *An Analysis of Multi-fingered Hands*. PhD thesis, Department of Mechanical Engineering, Stanford University, 1984.

[50] H. K. Khalil. *Nonlinear Systems.* Macmillan, 1992.

[51] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, RA-3(1):43–53, February 1987.

[52] D. Koditschek. Natural motion for robot arms. In *IEEE Control and Decision Conference*, pages 733–735, 1984.

[53] A. J. Koivo. *Fundamentals for Control of Robotic Manipulators.* Wiley, 1989.

[54] K. Kreutz. On manipulator control by exact linearization. *IEEE Transactions on Automatic Control*, 34(7):763–767, 1989.

[55] G. Lafferriere and H. J. Sussmann. A differential geometric approach to motion planning. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 235–270. Kluwer, 1993.

[56] J.-P. Laumond. Finding collision-free smooth trajectories for a nonholonomic mobile robot. In *International Joint Conference on Artificial Intelligence*, pages 1120–1123, 1987.

[57] D. F. Lawden. *Elliptic Functions and Applications.* Springer-Verlag, 1980.

[58] H. Y. Lee and C. G. Liang. A new vector theory for the analysis of spatial mechanisms. *Mechanisms and Machine Theory*, 23(3):209–217, 1988.

[59] Z. Li. *Kinematics, Planning and Control of Dextrous Robot Hands.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1989.

[60] Z. Li. Geometrical considerations of robot kinematics. *International Journal of Robotics and Automation*, 5(3):139–145, 1990.

[61] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–71, 1990.

[62] Z. Li and J. F. Canny, editors. *Nonholonomic Motion Planning.* Kluwer, 1992.

[63] Z. Li, P. Hsu, and S. S. Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *International Journal of Robotics Research*, 8(4):33–50, 1989.

[64] J. Loncarić. *Geometric Analysis of Compliant Mechanisms in Robotics.* PhD thesis, Division of Applied Sciences, Harvard University, 1985.

[65] D. Manocha and J. F. Canny. Real time inverse kinematics for general 6R manipulators. Technical Report ESRC 92-2, University of California, Berkeley, 1992.

[66] R. Manseur and K. L. Doty. A robot manipulator with 16 real inverse kinematic solutions. *International Journal of Robotics Research*, 8(5):75–79, 1989.

[67] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *International Journal of Robotics Research*, 9(1):61–74, 1990.

[68] J. E. Marsden. *Lectures on Mechanics.* London Mathematical Society, 1992.

[69] J. E. Marsden, R. Montgomery, and T. S. Ratiu. *Reduction, Symmetry, and Phases in Mechanics*, volume 436 of *Memoirs.* American Mathematical Society, 1990.

[70] M. T. Mason and J. K. Salisbury. *Robot Hands and the Mechanics of Manipulation.* MIT Press, 1985.

[71] J. M. McCarthy. *An Introduction to Theoretical Kinematics.* MIT Press, 1990.

[72] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifingered positive grips. *Algorithmica*, 2:541–558, 1987.

[73] D. J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, 1988.

[74] D. J. Montana. The kinematics of contact with compliance. In *IEEE International Conference on Robotics and Automation*, pages 770–775, 1989.

[75] R. M. Murray. Nilpotent bases for a class of non-integrable distributions with applications to trajectory generation for nonholonomic systems. Technical Report CIT/CDS 92-002, California Institute of Technology, October 1992.

[76] R. M. Murray, D. C. Deno, K. S. J. Pister, and S. S. Sastry. Control primitives for robot systems. *IEEE Transactions on Systems, Man and Cybernetics*, 22(1):183–193, 1992.

[77] R. M. Murray and S. S. Sastry. Control experiments in planar manipulation and grasping. In *IEEE International Conference on Robotics and Automation*, pages 624–629, 1989.

[78] R. M. Murray and S. S. Sastry. Grasping and manipulation using multifingered robot hands. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41*, pages 91–128. American Mathematical Society, 1990.

[79] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[80] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization.* Addison-Wesley, 1991.

[81] Y. Nakamura, K. Nagai, and T. Yoshikawa. Dynamics and stability in coordination of multiple robotic mechanisms. *International Journal of Robotics Research*, 8(2):44–61, 1989.

[82] Ju. I. Neimark and N. A. Fufaev. *Dynamics of Nonholonomic Systems*, volume 33 of *Translations of Mathematical Monographs.* American Mathematical Society, 1972.

[83] V.-D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, 1988.

[84] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag, 1990.

[85] T. Okada. Computer control of multijointed finger system for precise object-handling. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-12(3):289–299, 1982.

[86] B. Paden. *Kinematics and Control Robot Manipulators.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1986.

[87] B. Paden and S. S. Sastry. Optimal kinematic design of 6R manipulators. *International Journal of Robotics Research*, 7(2):43–61, 1988.

[88] F. C. Park, J. E. Bobrow, and S. R. Ploen. A Lie group formulation of robot dynamics. UCI Mechanical Systems Technical Report, Department of Mechanical Engineering, University of California, Irvine, April 1993.

[89] F. C. Park and A. P. Murray. Computational aspects of the product-of-exponentials formula for robot kinematics. *IEEE Transactions on Automatic Control*, 1994. (to appear).

[90] L. A. Pars. *A Treatise on Analytical Dynamics.* Wiley, 1965.

[91] R. P. Paul. *Robot Manipulators: Mathematics, Programming and Control.* MIT Press, 1981.

[92] K. S. J. Pister. *Hinged Polysilicon Structures with Integrated CMOS Thin Film Transistors.* PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1992.

[93] K. S. J. Pister, R. S. Fearing, and R. T. Howe. A planar air levitated space electrostatic actuator system. In *IEEE Workshop on Micro Electro Mechanical Systems*, pages 67–71, 1990.

[94] K. S. J. Pister, M. W. Judy, S. R. Burgett, and R. S. Fearing. Microfabricated hinges. *Sensors and Actuators A—Physical*, 33(3):249–256, 1992.

[95] E. J. F. Primrose. On the input-output equation of the general 7R mechanism. *Mechanisms and Machine Theory*, 21:509–510, 1986.

[96] M. Raghavan. Manipulator kinematics. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41*, pages 21–48. American Mathematical Society, 1990.

[97] M. Raghavan and B. Roth. Inverse kinematics of the general 6R manipulator and related linkages. *Journal of Mechanical Design*, 115:502–508, 1993.

[98] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[99] G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A spatial operator algebra for manipulator modeling and control. *International Journal of Robotics Research*, 10(4):371–381, 1991.

[100] R. M. Rosenberg. *Analytical Dynamics of Discrete Systems.* Plenum Press, New York, 1977.

[101] B. Roth, J. Rastegar, and V. Scheinman. On the design of computer controlled manipulators. *On the Theory and Practice of Robots and Manipulators, Proceedings of the First CISM-IFToMM Symposium*, pages 93–113, 1973.

[102] J. K. Salisbury. *Kinematic and Force Analysis of Articulated Hands.* PhD thesis, Department of Mechanical Engineering, Stanford University, 1982.

[103] S. S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice-Hall, 1989.

[104] S. S. Sastry and R. Montgomery. The structure of optimal controls for a steering problem. In *IFAC Symposium on Nonlinear Control Systems Design (NOLCOS)*, pages 385–390, 1992.

[105] J-P. Serre. *Lie Algebras and Lie groups.* W. A. Benjamin, New York, 1965.

[106] T. Shamir and Y. Yomdin. Repeatability of redundant manipulators: Mathematical solution of the problem. *IEEE Transactions on Automatic Control*, 33(11):1004–1009, 1988.

[107] F. Skinner. Designing a multiple prehension manipulator. *Journal of Mechanical Engineering*, 97(9):30–37, 1975.

[108] J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, 6:49–59, 1987.

[109] M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume I. Publish or Perish, Inc., Houston, second edition, 1979.

[110] M. W. Spong, F. L. Lewis, and C. T. Abdallah, editors. *Robot Control: Dynamics, Motion Planning and Analysis.* IEEE Press, 1991.

[111] M. W. Spong and M. Vidyasagar. *Dynamics and Control of Robot Manipulators.* John Wiley, 1989.

[112] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institute of Mechanical Engineering*, 180, part I(5):371–186, 1954. 1965–66.

[113] D. Tilbury, R. M. Murray, and S. S. Sastry. Trajectory generation for the N-trailer problem using Goursat normal form. In *IEEE Control and Decision Conference*, pages 971–977, 1993.

[114] R. Tomović and G. Boni. An adaptive artificial hand. *IRE Transactions on Automatic Control*, 7(3):3–10, 1962.

[115] J. P. Trevelyan. *Robots for Shearing Sheep: Shear Magic.* Oxford University Press, York, 1992.

[116] V. S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations.* Springer-Verlag, 1984.

[117] A. M. Vershik and V. Ya. Gershkovich. Nonholonomic problems and the theory of distributions. *Acta Applicandae Mathematicae*, 12:181–209, 1988.

[118] M. Vidyasagar. *Nonlinear Systems Analysis.* Prentice-Hall, second edition, 1993.

[119] G. Walsh and S. S. Sastry. On reorienting rigid linked bodies using internal motions. In *IEEE Control and Decision Conference*, pages 1190–1195, 1991. (to appear in *IEEE Transactions on Robotics and Automation*, 1994).

[120] J. T. Wen and D. S. Bayard. New class of control laws for robot manipulators. Part 1: Non-adaptive case. *International Journal of Control*, 47(5):1361–1385, 1988.

[121] S. Wolfram. *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley, 1992.

[122] T. Yoshikawa. *Foundations of Robotics: Analysis and Control.* MIT Press, 1990.

[123] L. C. Young. *Lectures on the Calculus of Variations and Optimal Control Theory.* Chelsea, New York, second edition, 1980.

Index

actions of Lie groups, 415–416
actuator redundancy, 286
actuator singularities, 135, 141
actuators, types of, 155
AdeptOne robot, 5, 83
adjoint action, 415, 420, 421
adjoint transformation, 56
  between body and spatial manipulator Jacobian, 117, 125
  between body and spatial velocities, 56
  for general Lie groups, 415
  for planar motions, 76
  properties of, 77
  of twists, 56, 60, 94
  of velocities, 59, 421
  of wrenches, 63, 64, 422
admissible velocities, for parallel manipulators, 135
angular velocity, *see* rotational velocity
antipodal grasp, 232, 233
asymptotic stability, 179, 180
atan2, 32
automobile, *see* kinematic car
axis of a screw, 45
  choice of point on, 49
axis of a twist, 48
axis of a wrench, 66

ball and socket joint, *see* spherical joint
Ball, R. S., 19
base frame, 84, 91
biological motor control, 303, 307
body angular velocity, 52
body frame, 22, 23, 51
body manipulator Jacobian, *see* manipulator Jacobian
body velocity, 55, 419
  geometric interpretation, 55
  relationship with spatial velocity, 56, 61, 420
  transformation and addition of, 59
body wrench, 63

Campbell-Baker-Hausdorff formula, 381
car with N trailers, 349
Caratheodory's theorem, 230, 299
Cayley parameters, 73
center of mass, 161
chained form, 363, 364, 392
  conversion to, 369
change of coordinates, *see* coordinate transformations
Chasles' theorem, 19, 49, 418
Chen-Fliess series, 376, 378
Chow's theorem, 329, 341
Christoffel symbols, 170, 246
closed-chain manipulators, *see* parallel manipulators
coadjoint action, 416, 422
coefficient of friction, 216, 218
collinear revolute joints, 124
commutator, 324
complete workspace, 95
completely nonholonomic, 320, 339
computed torque, 191–193, 198, 199, 205, 301
condition number of a matrix, 129
configuration of a rigid body, 22
configuration space, 25, 35, 83, 165, 265
conservation of angular momentum, 335
constrained Lagrangian, 275
constrained manipulators
  control of, 202–203, 209, 300, 428
  dynamics of, 200–202, 284
  planar example, 203
constraints, 157, 266, 428
  forces of, 157, 200, 266–269, 428
    see also internal forces
  holonomic, 157, 266, 318
  integrable, 267
  in multifingered grasps, 234–242, 253
  nonholonomic, 268, 274
  Pfaffian, 266–268
contact coordinates, 249, 254
contact forces, 215–218, 224, 238, 260, 277, 280

contact frame, 214, 246
contact kinematics, 248–253
  planar, 262
contact models, 214–218, 259
control
  of constrained manipulators, 202–203, 209, 300
  of multifingered hands, 300–310
  of open-chain manipulators, 190–199
  problem description, 156
  of tendon-driven fingers, 298
  in workspace coordinates, 196–199
controllability, 329–332
controllability Lie algebra, 329
controllability rank condition, 330
convex hull, 225, 229
convex set, 225
coordinate chart, 243, 403
coordinate frame, 20
coordinate transformations
  on inertia matrix, 208
  invariance under, 78, 422–433
  on twists, 60, 77
  use in analyzing singularities, 125
  on velocities, 59, 421
  on wrenches, 63, 422
coordinated lifting, 213, 263, 281
coplanar revolute axes, 125, 151
Coriolis and centrifugal forces, 165, 170
Coriolis matrix, 171, 176, 279
cotangent space, 326, 405
Coulomb friction, 216
coupling matrix, 295, 297
covector, 326
cross product
  2-dimensional, 232
  and Lie bracket, 175, 411
  matrix representation, 26
  preservation by rigid body transformations, 21
  properties of, 26, 73
curvature tensor, 245
cylindrical joint, 81

d'Alembert's principle, 268, 271
degree of nonholonomy, 340
degrees of freedom, 84, 130, 303, 398
  of four-bar mechanism, 136
  loss of, 124, 127
  for parallel mechanisms, 133
  redundant, 285
Denavit-Hartenberg parameters, 93, 110
dextrous manipulation, 9, 213
dextrous workspace, 95, 129
dialytical elimination, 108
diffeomorphism, 403
direct method of Lyapunov, 182
disk rolling on a plane, 272, 314, 336
dispacement, rigid, 20
distribution, 325
drift-free control systems, 329
dynamic finger repositioning, 382–388
dynamics, 155
  constrained manipulators, 200–202, 284
  multifingered hands, 276–285
  nonmanipulable grasps, 290–291
  open-chain manipulators, 168–179
  passivity property, 171, 209
  in presence of constraints, 265–276
  redundant manipulators, 286–290
  structural properties, 171, 198, 279, 314
  using the product of exponentials formula, 175
  in workspace coordinates, 282

eigenvalues of a rotation matrix, 30, 73
elastic tendons, 296–299
elbow manipulator, 148, 433
  forward kinematics, 89
  inverse kinematics, 104
end-effector, 8, 83
end-effector velocity
  using manipulator Jacobian, 115
  for parallel manipulators, 134
end-effector wrench
  using manipulator Jacobian, 121–123, 131
  for parallel manipulators, 134
  for redundant manipulators, 131
  at singular configuration, 124, 152
Engel's system, 373
equilibrium point, 179
equivalent axis representation, 30
equivalent wrenches, 62
Euler angles, 31, 151
Euler's equation, 166, 167, 209
Euler's theorem, 30
Euler-Lagrange equations, 359
exact one-form, 327
exceptional surface, 230
exponential coordinates
  on a Lie group, 414
  for rigid motion, 39–45
    see also twists
  for rotation, 27–31
exponential map

as relative transformation, 42, 45, 49
on general Lie group, 412
for rigid body transformations, 41, 413, 417
for rotations, 28–29, 413
surjectivity onto $SE(3)$, 43
surjectivity onto $SO(3)$, 29
exponential of a matrix, *see* matrix exponential
exponential stability, 181
extension function, 294

falling cat example, 352
feedback linearization, 193
feedforward control, 192, 309
Fick angles, 32
filtration, 340
finger kinematics, 234–237, 253–254
fingertip frame, 234
firetruck example, 350
first fundamental form, 244
first-order controllable systems, 358
fixed contact kinematics, 214
flow of a vector field, 322, 406
foliation, 326
force control, *see* constrained manipulators, control of
force-closure, 213
for antipodal grasps, 232
convexity conditions for, 226
for grasping, 223
number of contacts required, 230
for tendon network, 299
forward kinematics, 83–97
for elbow manipulator, 89
for parallel manipulators, 132
product of exponentials formula, 85–91
for redundant manipulators, 130
for SCARA manipulator, 87, 92
four-bar linkage, 135–139, 314
frame, *see* coordinate frame, tool frame, base frame, etc.
frame invariance, 78, 422
free vector, *see* vector
friction cone, 216, 218, 228, 229
frictionless point contacts, 215, 220, 224
Frobenius' theorem, 326
fundamental grasp constraints, *see* grasp constraints

Gauss frame, 245
Gauss map, 245
Gauss-Bonnet theorem, 385
general linear group, $GL(n,\mathbb{R})$, 409, 410, 412
generalized coordinates, 158, 265, 274
generalized forces, 158
generalized inertia matrix, 162
geometric parameters for a surface, 246
geometric phase, 385
global stability, 180
grasp constraints
fundamental grasp constraint, 237
nonmanipulable case, 291
redundant case, 289
grasp map, 218–223
grasping
basic assumptions, 213, 214
control, 300–310
dynamics, 276–285
effect of fingers, 234–242
fixed contact kinematics, 214–223
force relationships, 238
kinematics and statics, 211–255
versus parallel mechanisms, 281
planar case, 222, 231, 232
planning problem, 213, 229–234
properties, *see* force-closure, manipulability
representation of grasps, 220, 237
rolling contact kinematics, 243–255
similarity to parallel mechanisms, 134
summary of properties, 239
velocity constraints, 237
group
definition, 24
of rigid body transformations, 37
of rotations, 24
growth vector, 341
Gruebler's formula, 133

hand Jacobian, 236, 285
harmonic oscillator, 185
hazardous environments, 396
helical joint, 81
Helmholtz angles, 32
hierarchical control, 302
holonomic constraints, 157, 266, 318
homogeneous coordinates, 19, 36–39, 417–419
for points and vectors, 36, 417
for rigid body transformations, 36, 417
homunculus diagram, 9
hopping robot, 333, 341
hybrid force control, *see* constrained manipulators, control of

hyperbolic metric on $SE(3)$, 426

indirect method of Lyapunov, 184
inelastic tendons, 294–296
inertia matrix
    effect of coordinate transformation, 208
    effective, in grasping, 279
    for open-chain manipulators, 168, 176
    for rigid bodies, 162, 208
inertia tensor, 162, 166
infinite pitch screw, 49
integrable constraints, 267, 318
integrable distribution, 326
integral manifolds, 326
integrating factor, 319
internal forces, 134, 223, 279, 301
    due to motion, 280, 290
    in grasping, 279–281
    regulation of, 301, 302
    in tendon network, 299
internal motions, 130, 238, 285, 287
intersecting joint axes, 126, 152
invariant set, 189
inverse elbow manipulator, 148, 433
inverse kinematics, 97–115
    for elbow manipulator, 104
    general solutions, 108
    number of solutions, 98, 114
    for parallel manipulators, 133, 134, 140
    for redundant manipulators, 130
    for SCARA manipulator, 106
    simple example, 97
    solving using subproblems, 98, 104
    for Stewart platform, 140
involutive closure, 325
involutive distribution, 325
isotropic points, 151

Jacobi identity, 325, 408
Jacobian transpose, 122, 124
Jacobian, manipulator, *see* manipulator Jacobian
joint angle, 84
joint space
    for open-chain manipulators, 83
    for parallel manipulators, 133
joint space control, 156
    versus workspace control, 196, 199
joint torques
    choice of, in grasping, 301
    and end-effector forces, 122, 289
    and tendon forces, 295
joint twists, 87
    given Denavit-Hartenberg parameters, 94
joint types, 81

Killing form, 427
kinematic car, 318, 336, 343
kinematic redundancy, 286
    see also redundant manipulators
kinematic singularities, 123–127, 151–152
    versus actuator singularities, 135, 141
    for four-bar mechanism, 138
    for open-chain manipulators, 124–127
    for parallel manipulators, 134
kinematics, 81
kinetic energy, 161
Klein form, 426

Lagrange multipliers, 157, 269–271
    formula for, 270
    relationship with contact forces, 280
Lagrange's equations, 158
    for constrained systems, 269, 275
    for mechanical systems, 156–167
    for open-chain manipulators, 169
Lagrange-d'Alembert equations, 271, 272, 275
Lagrangian, 158
    for multifingered hand, 277
    for open-chain manipulators, 168
Lasalle's invariance principle, 188, 194
leaf of a foliation, 326
left invariant vector field, 410
length scale, 424
Lie algebra, 326, 407, 410
Lie bracket, 175, 323–325, 407
Lie bracket motion, 323
Lie derivative, 322, 406
Lie group, 408
Lie product, 324, 344
line contact, 260
linearization, 184
link frames, 93
local controllability, 331
local stability, 179, 180, 185
locally positive definite functions, 182
log function on a Lie group, 413
loop equation, *see* structure equations
lower pair joints, 81
Lyapunov functions
    choosing, 183
    skewed energy, 187, 195
Lyapunov stability, 179–189

basic theorem, 183
direct method, 182–184
indirect method, 184–185

magnitude of a twist, 48, 427
magnitude of a wrench, 66
manifold, 318
manifold, definition of, 403
manipulability measures, 128–129, 150, 152, 429
well-posed, 432
manipulable grasp, 213, 238
manipulator inertia matrix, 168
manipulator Jacobian, 115–129
body, 117
geometric interpretation, 117
versus Jacobian of a mapping, 115, 120
and manipulability measures, 128
for mapping forces, 121–123, 131
for parallel manipulators, 134
for redundant manipulators, 130
relationship between body and spatial, 117
for SCARA manipulator, 118, 123
singularities, *see* kinematic singularities
spatial, 116
for Stanford manipulator, 119
manipulator workspace, *see* workspace
mass matrix, *see* manipulator inertia matrix
Mathematica, 435
matrix exponential, 19, 27, 40
properties of, 74
maximally independent contact regions, 233
medical robotics, 398
metric tensor, 244
microrobotics, 399
minimally invasive surgery, 398
Motoman, 282
multifingered grasp, 237
see also grasping
multifingered hand, 8
limitations and advantages, 212

Newton's law, 157, 159, 166, 167
Newton-Euler equations, 165–167, 314
nilpotent Lie algebra, 344, 376
nonholonomic constraints, 268, 274–276, 318
classification, 340
versus holonomic constraints, 274
integrating, 319
see also Pfaffian constraints
nonholonomic motion planning, 319, 331
nonmanipulable grasps, 239, 290
normal vector, 245
normalized Gauss frame, 245
numbering conventions for a robot, 83

ω limit set, 188
one-forms, 326, 408
open loop control, 190
open-chain manipulators, 82
optimal manipulator design, 432
optimal steering, 371
orthogonal coordinate chart, 244
orthogonal matrices, *see* rotation matrices

Paden-Kahan subproblems, 99–103, 148–149
solving inverse kinematics using, 104
palm frame, 215
parallel manipulators, 132–143
inverse kinematics, 134, 140
kinematic singularities, 134
see also four bar linkage, Stewart platform
passivity, 171, 188, 209
PD control, 193–195
perspective transformations, 37
Pfaffian constraints, 266–268
converting to control system, 320, 327
integrability conditions, 328
Philip Hall basis, 344
pitch of a screw, 45
pitch of a twist, 48, 427
pitch of a wrench, 66
planar grasping, 222, 231, 232, 262
planar joint, 82
planar rigid body transformations, 76
planar rotational motion, 75
planar Stewart platform, 142
plane contact, 260
Poinsot's theorem, 19, 65
point contact with friction, 217
points
rigid transformation of, 35, 36, 417
rotational transformation of, 25
versus vectors, 21, 36, 322
position control, 190–199
positive definite functions, 182
positive span, 225, 230
positively dependent, 225

potential energy for an open-chain manipulator, 169
prehensile grasp, 260
prismatic joint, 40, 81, 84
  twist associated with, 49, 87
product of exponentials formula, 82, 85–91
  basic formula, 87
  choice of base frame, 91
  versus Denavit-Hartenberg parameters, 93
  dynamics using, 175, 207
  independence of order of joint motions, 147
  independence on order of joint motions, 86
  manipulator Jacobian using, 116
projection maps, 75
prosthetic hands, 9
pseudo-inverse for resolving redundancy, 131
pull back map, 407, 408
PUMA manipulator, 4, 5
  see also elbow manipulator
pure quaternion, 74
pure rolling, 249, 252, 338
push forward map, 407

quaternions, 33–34, 74

rank of structure equations, 135
rate of convergence, 181, 184
reachable set, 318, 320
reachable workspace, 95
reciprocal product, 67
reciprocal screws, 66–69
  definition, 66
  systems of, 69, 78
  use in analyzing mechanisms, 68, 69, 126
redundant manipulators, 122
  dynamics, 286–290
  in grasping, 238
  kinematic versus actuator redundancy, 286
  kinematics, 130–132
reference configuration, 87
  choice of, 91
regular distribution, 325
regular filtration, 340
relative curvature form, 250
relative growth vector, 341
relative motion, representation using the exponential map, 42
revolute joint, 81, 84
  twist associated with, 49, 87
right-handed coordinate frame, 22
rigid bodies, 20
  dynamics, 165–167
  inertial properties, 160–163
  kinetic energy, 161
rigid body motion, 34–51
  definition of, 20
  representation using $SE(3)$, 34, 416
  representation using body frame, 22
rigid body transformations, 20–22
  actions on points and vectors, 21, 35–37, 417
  composition rule, 37
  formal definition, 21
  group properties, 37
  homogeneous representation, 36
  planar, 76
rigid body velocity, 54–61, 418–420
  see also body velocity, spatial velocity
rigid displacement, 20
rigid transformations, *see* rigid body transformations
robot, origin of word, 1
robustness of control laws, 190
Rodrigues' formula, 28, 76
roll, pitch, yaw angles, 32
rolling contact kinematics, 243–255
rolling penny, *see* disk rolling on a plane
rotation about a line, 38, 87, 99
  as a screw motion, 50
  twist coordinates, 44
rotation about two axes, 100
rotation group, 24
rotation matrices, 23
  actions on points and vectors, 25
  eigenvalues of, 30
  properties of, 23–26, 73
rotation to a given distance, 102
rotational motion, 22–34
  composition rule, 25
  equivalent axis representation, 30
  Euler angle representation, 31
  exponential coordinates, 27–31
  about a fixed axis, 27, 29
  parameterization singularities, 31, 32
  planar, 75
  quaternion representation, 33
  representation using rotation matrices, 23
rotational velocity, 51–54
  body versus spatial, 52

Salisbury Hand, 11, 12
SCARA manipulator, 5, 83
  dynamics, 177
  forward kinematics, 87, 92
  grasp using, 240, 291
  inverse kinematics, 106
  manipulator Jacobian, 118, 120, 123
screw motions, 19, 45, 46
  instantaneous velocity of, 58
screw system, 68
screw theory
  advantages of, 20
  origins of, 19
screws, 45–51
  associated with wrenches, 65
  Chasles' theorem, 49
  geometric attributes of, 45–46
  infinite pitch, 49
  rigid body transformations associated with, 46
  twists associated with, 48
$SE(3)$, 35, 409
  bi-invariant quadratic forms, 425
  bi-invariant volume forms, 431
  hyperbolic metric, 426
  invariant metrics, 423
  lack of bi-invariant metric, 427
  metric properties, 422
$se(3)$, 40, 411
second fundamental form, 245
second-order controllable systems, 362
self-motion manifold, 130
separating hyperplane, 226
setpoint stabilization, 193
singular configurations, 124, 152
  for parallel manipulators, 135
singular values of a matrix, 128, 149
singularities, *see* kinematic singularities
skew-symmetric matrices, 27
  properties of, 26, 28, 73
  for representing cross product, 26
slider-crank mechanism, 152, 203, 314
sliding, 249, 268
small-time locally controllable, *see* locally controllable
$SO(3)$, 24, 409
  see also rotation matrices
$so(3)$, 28, 411
  see also skew-symmetric matrices
soft-finger contact, 217
space robots, 334, 342, 351, 396
spatial angular velocity, 52
spatial frame, 51
spatial manipulator Jacobian, *see* manipulator Jacobian
spatial operator algebra, 207
spatial velocity, 54, 419
  addition of, 59, 422
  geometric interpretation, 54
  relationship with body velocity, 56, 61, 420
  transformation of, 59, 421
spatial wrench, 63
special Euclidean group, *see* $SE(3)$
special orthogonal group, *see* $SO(3)$
sphere rolling on a plane, 252, 338, 343
sphere rolling on a sphere, 349
spherical joint, 81, 140
spherical wrist, 125
  effect on workspace, 96
  versus spherical joint, 140
spring mass system, 185, 187, 189
stability by linearization, 185
stability definitions, 179–182
stable, 179, 180
Stanford manipulator, 2, 4, 148
  manipulator Jacobian, 119
Stanford/JPL hand, *see* Salisbury Hand
Steinitz's theorem, 230, 299
Stewart platform, 139–143, 154
strictly internal forces, 223
structurally dependent forces, 123, 239
structure equations, 132–134
  for four-bar mechanism, 136
  for Stewart platform, 140
supporting hyperplane, 226
surface models, 243

tangent space, 244, 404
teleoperation, 395
tendon kinematics, 293–300
tool frame, 84
torsion form, 246
trajectory generation, using manipulator Jacobian, 117
trajectory tracking, *see* position control
translational motion, 34, 48
transpose of Jacobian, *see* Jacobian transpose
twists, 19, 417
  definition of, 41
  geometric attributes, 45–51, 427
  Lie bracket between, 175
  parameterizing manipulators via, 91–95
  reciprocal to a wrench, 66
  for revolute and prismatic joints, 87

screw coordinates, 47
screw motions corresponding to, 48
transformation of, 60, 77
twist coordinates, 41
two-link planar manipulator
constrained, 315
dynamics, 164
inverse kinematics, 97
moving in a slot, 203

U-joint, 154
uncertainty configuration, 138
underwater robots, 397
uniform stability, 180, 185
unit quaternions, 33, 74
unit twist, 49
Utah/MIT hand, 10, 12, 212

variable geometry truss, 153
vector field, 322, 406
vectors, 21
versus points, 21, 36
rigid transformation of, 21, 37, 417
rotational transformation of, 25
velocity
end-effector, 115
rigid body, *see* rigid body velocity
rotational, *see* rotational velocity
of a screw motion, 58
velocity of a point
attached to end-effector, 117
for rotational motion, 53
virtual displacement, 271
virtual reality, 396
virtual work, 271
viscous friction, 170
volume forms on $SE(3)$, 430

work, between twist and wrench, 62
workspace control, 156, 196–199, 209
versus joint space control, 196, 199
workspace dynamics, 197, 282
workspace of a manipulator, 95–97, 432
dextrous, 95, 129
maximal, 433
wrench basis for a contact, 218, 235
wrenches, 19, 61–66, 420
addition of, 63
body and spatial representations, 63
reciprocal to a twist, 66
screw coordinates of, 65
transformation of, 62, 422

zero pitch screw, 49, 66