

Matrix Methods in the Design Analysis of Mechanisms and Multibody Systems

■ JOHN J. UICKER

■ BAHRAM RAVANI

■ PRADIP N. SHETH



CAMBRIDGE

MATRIX METHODS IN THE DESIGN ANALYSIS OF MECHANISMS AND MULTIBODY SYSTEMS

This book is an integrated approach to kinematic and dynamic analysis. The matrix techniques presented are general and fully applicable to two- or three-dimensional systems. They lend themselves to programming and digital computation and can be the basis of a usable tool for designers. The techniques have broad applicability to the design analysis of all multibody mechanical systems. The more powerful and more flexible the approach, and the less specialization and reprogramming required for each application, the better. The matrix methods presented have been developed using these as primary goals. Although the matrix methods can be applied by hand to such problems as the slider-crank mechanism, this is not the intent of this text, and often the rigor required for such an attempt becomes quite burdensome in comparison with other techniques. The matrix methods have been extensively tested, both in the classroom and in the world of the engineering industry.

John J. Uicker is Professor Emeritus of mechanical engineering at the University of Wisconsin–Madison. Throughout his career, his teaching and research have focused on solid geometric modeling and the modeling of mechanical motion, and their application to computer-aided design and manufacture, including the kinematics, dynamics, and simulation of articulated rigid-body mechanical systems. He founded the UW Computer-Aided Engineering Center and served as its director for its initial ten years of operation. He has served on several national committees of the American Society of Mechanical Engineers (ASME) and the Society of Automotive Engineers (SAE), and he received the ASME Mechanisms Committee Award in 2004 and the ASME Fellow Award in 2007. He is a founding member of the U.S. Council for the Theory of Mechanism and Machine Science (USCToMM), and of the International Federation of Mechanism and Machine Science (IFTToMM). He is a registered mechanical engineer in Wisconsin and has served for many years as an active consultant to industry.

Bahram Ravani is Professor of mechanical and aerospace engineering at the University of California, Davis. He has served as the chair of the department as well as the interim chair of electrical and computer engineering. He is also a member of the graduate programs in biomedical engineering and in forensic science and engineering. Among his honors are the Young Manufacturing Engineer Award from the Society of Manufacturing Engineers, the Gustus L. Larson Memorial Award from ASME for outstanding achievements in mechanical engineering, the Design Automation Award from the Design Engineering Division of ASME for his lifetime of sustained contributions, and the Machine Design Award for eminent achievements. He is currently the technical editor of the *ASME Journal of Computers and Information Science in Engineering* and he was a technical editor of the *Journal of Mechanical Design*. He is a Fellow of ASME and is the former chair of its Design Engineering Division. He is also a member of the Society of Automotive Engineers, the International Society of Biomechanics, and the Association for Advancement of Automotive Medicine.

Pradip N. Sheth (1944–2009) was born in Vadodara, India. He earned his BE (1965) and MS (1968) in mechanical engineering at Maharaja Sayajirao University, Baroda, India, and his PhD in 1972 at the University of Wisconsin–Madison, where John Uicker served as his advisor. In his research, he developed the Integrated Mechanisms Program for the computer-aided design and analysis of multibody mechanical systems. More than 200 industrial and educational organizations worldwide have used the system, which was the first of its kind.

Matrix Methods in the Design Analysis of Mechanisms and Multibody Systems

John J. Uicker

University of Wisconsin–Madison

Bahram Ravani

University of California, Davis

Pradip N. Sheth



CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Mexico City

Cambridge University Press

32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org

Information on this title: www.cambridge.org/9780521761093

© John J. Uicker, Bahram Ravani, and Pradip N. Sheth 2013

This publication is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without the written
permission of Cambridge University Press.

First published 2013

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data

Uicker, John Joseph, author.

Matrix Methods in the Design Analysis of Mechanisms and Multibody Systems /
John Uicker, University of Wisconsin, Madison, Pradip N. Sheth, Bahram Ravani,
University of California, Davis.

pages cm

Includes bibliographical references and index.

ISBN 978-0-521-76109-3 (hardback)

1. Machinery, Dynamics of. 2. Multibody systems – Mathematical models.
3. Dynamics, Rigid – Mathematics. I. Sheth, Pradip N., author. II. Ravani,
Bahram, 1953– author. III. Title.

TJ173.U53 2013

621.8'11–dc23 2012037617

ISBN 978-0-521-76109-3 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy of
URLs for external or third-party Internet Web sites referred to in this publication
and does not guarantee that any content on such Web sites is, or will remain,
accurate or appropriate.

This textbook is dedicated to the memory of the third author, the late Associate Professor **Pradip N. Sheth**, Department of Mechanical Engineering, University of Virginia, Charlottesville, who passed away during his writing of this book after several years of testing it in his classes. His doctoral dissertation included the original development of the Integrated Mechanisms Program (IMP), the first general software system for the simulation of articulated multibody mechanical systems. Much of this text can be traced to that seminal work. His intention was to dedicate his writings to his loving wife, Diane C. Sheth, who provided her encouragement and support throughout his foreshortened career.

This work is also dedicated to the memory of my father, John J. Uicker, Sr., emeritus dean of engineering, University of Detroit; to my mother, Elizabeth F. Uicker; and to my six children, Theresa A. Zenchenko, John J. Uicker III, Joseph M. Uicker, Dorothy J. Winger, Barbara A. Peterson, and Joan E. Horne, and their families.

– John J. Uicker

This work is also dedicated first and foremost to my father, Abraham Ravani, who inspired me from early childhood to pursue science and provided the opportunity for my U.S. education; to my children, Sarah and Samuel Ravani, and Paris Kent; and finally to my wife, Sara Kent, who endured while I spent time working on this book.

– Bahram Ravani

Song of the Screw*

A moving form or rigid mass,
Under whate'er conditions
Along successive screws must pass
Between each two positions.
It turns around and slides along –
This is the burden of my song.

The pitch of screw, if multiplied
By angle of rotation,
Will give the distance it must glide
In motion of translation.
Infinite pitch means pure translation,
And zero pitch means pure rotation.

Two motions on two given screws,
With amplitudes at pleasure,
Into a third screw-motion fuse,
Whose amplitude we measure
By parallelogram construction
(A very obvious deduction).

Its axis cuts the nodal line,
Which to both screws is normal,
And generates a form divine
Whose name, in language formal,
Is "surface-ruled of third degree."
Cylindroid is the name for me.

Rotation round a given line
Is like a force along,
If to say couple you decline,
you're clearly in the wrong –
'Tis obvious upon reflection,
A line is not a mere direction.

So couples with translations too
In all respects agree;
And thus there centers in the screw
A wondrous harmony
Of Kinematics and of Statics –
Sweetest thing in mathematics.

The forces in one given screw,
With motion on a second,
In general some work will do,

Whose magnitude is reckoned
By angle, force, and what we call
The coefficient virtual.

Rotation now to force convert,
And force into rotation;
Unchanged the work, we can assert,
In spite of transformation.
And if two screws no work can claim,
Reciprocal will be their name.

Five numbers will a screw define,
A screwing motion, six;
For four will give the axial line,
One more the pitch will fix;
And hence we always can contrive
One screw reciprocal to five.

Screws – two, three, four, or five,
combined
(No question here of six),
Yield other screws which are confined
Within one screw complex.
Thus we obtain the clearest notion
Of freedom and constraint of motion.

In complex III, three several screws
At every point you find,
Or, if you one direction choose,
One screw is to your mind;
And complexes of order III
Their own reciprocals may be.

In IV, wherever you arrive,
You find of screws a cone,
On every line of complex V
There is precisely one;
At each point of this complex rich,
A plane of screws has given pitch.

But time would fail me to discourse
Of Order and Degree;
Of Impulse, Energy, and Force,
And Reciprocity.
All these and more, for motions small,
Have been discussed by Dr. Ball.

Anonymous

* Published anonymously in *Nature*, 14, 30–30 (11 May 1876). This poem accurately captures in verse the main points of the mathematical theory of screws which forms a common thread of the theory behind this book.

Contents

Preface	<i>page</i> xiii
About the Authors	xvii
1 Concepts and Definitions	1
1.1 Mechanical Design: Synthesis versus Analysis	1
1.2 Multibody Systems and Mechanisms	3
1.3 Planar, Spherical, and Spatial Mechanisms	7
1.4 Mechanical Body	10
1.5 Mechanical Chain and Kinematic Inversion	11
1.6 Joints and Joint Elements	12
1.7 The Six Lower-Pairs	14
1.8 Higher-Pairs and Kinematic Equivalence	19
1.9 Restraints versus Constraints	20
REFERENCES	20
2 Topology and Kinematic Architecture	22
2.1 Introduction	22
2.2 The Incidence Matrix	24
2.3 Connectedness and Assemblies	27
2.4 Kinematic Loops	27
2.5 Kinematic Paths	32
REFERENCES	37
PROBLEMS	37
3 Transformation Matrices in Kinematics	42
3.1 Introduction	42
3.2 Homogeneous Coordinates of a Point	42
3.3 Line Coordinates and Plücker Vectors	45
3.4 Three-dimensional Orientation	47
3.5 Transformation of Coordinates	51
3.6 Positions, Postures, and Displacements	55
3.7 Euler's and Chasles' Theorems	60

3.8 Euler-Rodrigues Parameters	69
3.9 Displacement of Lines	74
3.10 Quaternions	74
REFERENCES	75
PROBLEMS	77
4 Modeling Mechanisms and Multibody Systems with Transformation Matrices	80
4.1 Introduction	80
4.2 Body Coordinate Systems	80
4.3 Joint and Auxiliary Coordinate Systems	81
4.4 Specifying Data for a Coordinate System	82
4.5 Modeling Dimensional Characteristics of a Body	85
4.6 Modeling Joint Characteristics	87
4.6.1 Helical Joint	88
4.6.2 Revolute Joint	90
4.6.3 Prismatic Joint	91
4.6.4 Cylindric Joint	92
4.6.5 Spheric Joint	93
4.6.6 Flat Joint	94
4.6.7 Rigid Joint	96
4.6.8 Open Joint	96
4.6.9 Parallel-Axis Gear Joint	98
4.6.10 Involute Rack-and-Pinion Joint	100
4.6.11 Straight-Tooth Bevel-Gear Joint	102
4.6.12 Point on a Planar-Curve Joint	104
4.6.13 Line Tangent to a Planar-Curve Joint	106
PROBLEMS	108
5 Posture Analysis by Kinematic Equations	111
5.1 Introduction	111
5.2 Consecutive Transformations	112
5.3 Denavit-Hartenberg Transformations	116
5.4 Absolute Position	118
5.5 The Loop-closure Equation (Kinematic Equation for Position Analysis)	119
5.6 Closed-form Solution of Kinematic Equations for Joint-variable Positions	121
5.7 General Styles for Closed-Form Solutions of Kinematic Equations	140
REFERENCES	145
PROBLEMS	145
6 Differential Kinematics and Numeric Solution of Posture Equations	148
6.1 Introduction	148
6.2 Differential Kinematics of a Helical Joint	149

6.3	Derivative Operator Matrices	153
6.3.1	Helical Joint	155
6.3.2	Revolute Joint	155
6.3.3	Prismatic Joint	155
6.3.4	Cylindric Joint	155
6.3.5	Spheric Joint	156
6.3.6	Flat Joint	156
6.3.7	Rigid Joint	156
6.3.8	Open Joint	156
6.3.9	Parallel-axis Gear Joint	157
6.3.10	Involute Rack-and-Pinion Joint	157
6.3.11	Straight-tooth Bevel-gear Joint	158
6.3.12	Point on a Planar-Curve Joint	158
6.3.13	Line Tangent to a Planar-Curve Joint	158
6.4	Screw Axes and Ball Vectors for Differential Displacements	159
6.5	Numeric Solution of Kinematic Posture Equations	163
6.5.1	Solution for a Nearby Posture	164
6.5.2	Avoiding Convergence to a False Solution	168
6.5.3	Numeric Solution of the Loop-closure Equation	169
6.6	Identification of Generalized Coordinates	173
6.7	Scaling Internal Length Units	175
6.8	Quality Index	176
6.9	Convergence and Robustness	177
	REFERENCES	181
	PROBLEMS	182
7	Velocity Analysis	183
7.1	Introduction	183
7.2	Definition of Velocity	184
7.3	First Geometric Derivatives of Joint Variables	186
7.4	Velocities of Joint Variables	189
7.5	First Geometric Derivatives of Body Postures	191
7.6	Velocities of Bodies	194
7.7	First Geometric Derivatives of Point Positions	195
7.8	Velocities of Points	196
	REFERENCE	196
	PROBLEMS	196
8	Acceleration Analysis	197
8.1	Definition of Acceleration	197
8.2	Derivatives of the Q_h Operator Matrices	198
8.2.1	Helical (Screw) Joint	199
8.2.2	Revolute Joint	199
8.2.3	Prismatic Joint	199
8.2.4	Cylindric Joint	200
8.2.5	Spheric Joint	200
8.2.6	Flat Joint	201

8.2.7	Rigid Joint	201
8.2.8	Open Joint	201
8.2.9	Parallel-Axis Gear Joint	202
8.2.10	Involute Rack-and-Pinion Joint	204
8.2.11	Straight-Tooth Bevel-Gear Joint	204
8.2.12	Point on a Planar-Curve Joint	205
8.2.13	Line Tangent to a Planar-Curve Joint	205
8.3	Derivatives of the D_h Operator Matrices	205
8.4	Second Geometric Derivatives of Joint Variables	207
8.5	Accelerations of Joint Variables	214
8.6	Second Geometric Derivatives of Body Postures	216
8.7	Second Geometric Derivatives of Point Positions	220
8.8	Accelerations of Bodies	220
8.9	Accelerations of Points	223
	REFERENCE	223
	PROBLEMS	223
9	Modeling Dynamic Aspects of Mechanisms and Multibody Systems	225
9.1	Introduction	225
9.2	Modeling Kinetic Energy	226
9.3	The Inertia Matrix	227
9.4	Systems of Units	230
9.5	Modeling Gravitational Effects	230
9.6	Modeling Joint Stiffness	232
9.7	Modeling Joint Damping	232
9.8	Modeling Point-to-Point Springs	233
9.9	Modeling Point-to-Point Dampers	234
9.10	Modeling External Forces and Torques Applied with Joint Variables	235
9.11	Modeling External Forces and Torques Applied to Bodies	236
	REFERENCES	241
	PROBLEMS	242
10	Dynamic Equations of Motion	244
10.1	Introduction	244
10.2	Lagrange's Equation	244
10.3	Generalized Momentum	245
10.4	D'Alembert Inertia Forces	246
10.5	Generalized Restoring Forces	249
10.6	Generalized Applied Forces	250
10.7	Complete Equations of Motion	250
	REFERENCES	253
	PROBLEMS	253

11 Linearized Equations of Motion	254
11.1 Introduction	254
11.2 Linearization Assumptions	254
11.3 Linearization	255
11.4 Linearized Equations of Motion	258
11.5 Dynamic Equations with Specified Input Motions	260
PROBLEMS	261
12 Equilibrium Posture Analysis	262
12.1 Introduction	262
12.2 Seeking a Nearby Posture of Equilibrium	263
12.3 Seeking Equilibrium with Some Generalized Coordinates Specified	266
12.4 Large Increments of the Generalized Coordinates	266
12.5 Stable versus Unstable Equilibrium	267
12.6 Postures of Neutral Equilibrium	269
REFERENCE	270
PROBLEM	270
13 Frequency Response of Mechanisms and Multibody Systems	271
13.1 Introduction	271
13.2 Homogeneous First-order Equations of Motion	271
13.3 Modal Coordinates	274
13.4 Laplace Transformed Equations of Motion	276
13.5 Frequency Response	277
REFERENCES	278
PROBLEMS	279
14 Time Response of Mechanisms and Multibody Systems	280
14.1 Inverse Laplace Transform	280
14.2 Cauchy's Residue Theorem	282
14.3 Systems with Repeated Eigenvalues	284
14.4 Time Integration Algorithm	288
14.5 Adaptive Time-step Control	291
REFERENCES	292
PROBLEM	293
15 Collision Detection	294
15.1 Introduction	294
15.2 Vertex-Face Contact	295
15.3 Edge-Edge Contact	296
15.4 Finding the Time Increment until Contact	297
REFERENCES	299

16 Impact Analysis	300
16.1 Applied Impulsive Loads	300
16.2 Location and Type of Contact	303
16.3 Simple Impact Model	303
16.4 Impact Model with Tangential Impulse	305
16.5 Impact Model with Normal Torsional Impulse	306
16.6 Impact Model with Moment Impulse	307
16.7 Integrated Model of Impact	307
16.8 Impact Analysis with SGCs	308
REFERENCES	309
PROBLEM	309
17 Constraint Force Analysis	310
17.1 Introduction	310
17.2 Fictitious Displacements	311
17.3 Fictitious Derivatives	313
17.4 Lagrange Equation for Constraint Force	316
REFERENCES	320
PROBLEMS	320
Index	321

Preface

This text presents a uniform and comprehensive treatment of the theory and use of homogeneous coordinates and transformation matrices in the kinematic and dynamic design analysis and the numeric simulation of mechanisms and multibody systems.

The following observations, originally set down by Reuleaux in 1875,¹ are every bit as true today, and it would be difficult to state them better.

The whole study of the constitution of machines – the Kinematics of Machinery – naturally divides itself into two parts, the one comprehending the theoretical and the other the applied or practical side of the subject; of these the former alone forms the subject of this work. It deals chiefly with the establishment of those ideas which form the foundation of the applied part of the science, and in its treatment of these its method differs in great part essentially from those heretofore employed.

As I have here to do chiefly with theoretical questions, it might seem that I could hardly expect to interest other than those concerned only with the theoretical side of this special study. But Theory and Practice are not antagonists, as is so often tacitly assumed. Theory is not necessarily unpractical, nor Practice unscientific, although both of these things may occur. Indeed in any department thoroughly elucidated by Science the truly practical coincides with the theoretical, if the theory be right. The popular antithesis should rather be between Theory and Empiricism. This will always remain, and the more Theory is extended the greater will be the drawback of the empirical, as compared with the theoretical methods. The latter can never be indifferent, therefore, to any who are able to use them, even if their work be entirely “practical,” and although they may be able for a while longer to get on without them. The theoretical questions, however, which are here to be treated, are of so deep-reaching a nature that I entertain the hope that those who are practically, as well as those who are theoretically concerned with the subject, may obtain help from the new method of treating them.

Certainly, the science of kinematics has grown a great deal and today rests on a much firmer foundation than it did in Reuleaux’s time. However, to a great extent, the gulf between theory and empiricism still exists. On the one hand, we find that

¹ Franz Reuleaux (1829–1905), *Theoretische Kinematik, Grundzüge einer Theorie des Maschinenwesens (Theoretical kinematics: Foundations of the theory of mechanisms)*, Friedrich Vieweg & Sohn, Braunschweig, 1875; English translation by A. B. W. Kennedy, *Reuleaux’ Kinematics of Machinery*, Macmillan and Co., London, 1876; reprinted by Dover Publications, Inc., New York, 1963.

academics have developed a vast body of science, steeped in the elegance and sophistication traditional to their views. However, their efforts, almost to an individual, are still directed toward further understanding of the four-bar linkage, the slider-crank mechanism, and, more recently, the robotic manipulator, and rather simple multibody systems. On the other hand, even today, we find that the inventor – the completely practical person who develops a working machine or performs an analysis of a complex multibody system, despite the richness of modern theoretical developments – finds very little of modern theory truly usable as a practical design technique or as a broadly applicable computational analysis tool.

Among the several reasons for this paradox is the fact that modern theoretic approaches are difficult for the novice to comprehend and, by the very nature of the problem, are quite tedious to apply. A thorough understanding of these methods takes years of specialized study and, very likely, we find in the end that they do not really solve the complex problems encountered in the design or analysis of present-day equipment. Thus, to be of value, the methods presented in the following chapters must accomplish two apparently conflicting goals. First, they must be applicable to an extremely broad category of problems, including the large multifaceted problems represented in the design of modern machinery and analysis of complex multibody systems. Secondly, they must be put into a form that is useful to the practicing engineer without years of advanced study.

It is our firm belief that the sole hope for accomplishing both of these purposes lies with the development of a unified and powerful analytic method that can be programmed for solution by computer. Only in this way can the more sophisticated methods be made usable without requiring significant specialized training of every user. Also, this is the only apparent method of dealing with some of the more complex mechanisms and multibody systems, if only because the number of calculations involved would be prohibitive by any other means. If sufficiently general software can be written, however, the application of even the most sophisticated theoretical approach to very complex multibody systems becomes a feasible goal. In presenting such a general approach, however, we will be careful, from time to time, to also present alternative – less general, but perhaps more intuitive – approaches. This is intended to provide a balanced and better understanding of the methods presented, and to illustrate the power of the more general techniques.

Furthermore, much of the more recent trends toward miniaturization and high performance for mechanisms necessitate the inclusion of dynamic analysis along with kinematics. In the broader category of multibody systems, dynamic analysis has always played a key role. However, this book deals with an integrated approach to both kinematic and dynamic analyses. The transformation matrix techniques presented are general and fully applicable to systems in either two or three dimensions. In addition, they lend themselves to programming and digital computation and can, therefore, be the basis of a usable tool for the designer. This book may appear to place more emphasis on mechanisms because much of the techniques have their roots in the kinematics and design literature. However, the techniques have broad applicability to the design analysis of all multibody mechanical systems.

Another pitfall one must avoid when taking a general approach is that of replacing the effort a designer or an engineer must spend in learning and applying the analysis procedures with an equal or worse task of writing and testing complex

computer programs. Whatever methods proposed for real design and analysis use in the future – it seems to the authors – must include the generality and flexibility to handle a very broad class of problems and give a thorough analysis, without requiring separate programming for each new problem. Only in this way can real usability be achieved. The more powerful and more flexible the approach, and the less specialization and reprogramming required for each application, the better.

The transformation matrix methods presented in the following chapters have been developed using these as primary goals. The reader must keep these firmly in mind throughout the book; they are essential to the appreciation and perhaps even to the comprehension of the methods. Although the transformation matrix methods can be applied by hand to such problems as the slider-crank mechanism, this is not the intent of this text, and often the rigor required for such an attempt becomes quite burdensome in comparison with other techniques.

The transformation matrix methods have been extensively tested, both in the classroom and in engineering industry. In the classroom, the authors have tested the drafts of this text in senior/graduate-level courses at the University of Wisconsin–Madison and the University of Virginia, and more recently at the University of California, Davis, and we are indebted to all of those students for their trials and suggestions for improvements. As for use in engineering industry, the methods presented herein have been the basis for the software system known as the Integrated Mechanisms Program (IMP).² First released in 1972, IMP has been extensively used in many companies and academic institutions to analyze many different kinds of mechanical systems. Although it is still not a perfect tool, IMP continues to be used, and its many users also deserve much credit for the authors' insights and the experience reflected in the methods described herein.³

Developing methods for computer solution requires several radical alterations in the approach taken from those of more traditional methods. It requires simplicity and precision, almost to a fault. Because the computer has no reasoning capability, any possible conflict in interpretation of the user's intent will result in disaster. Definitions of terms must be extremely precise; identification of parts must be unique; sign conventions must be established, once and for all, in a clear understandable manner; and the sequencing of the solution process must take every possible eventuality into account, even those cases that seem trivial in the rational world of humans.

Again, Reuleaux expresses our thoughts very well:⁴

The remodeling which has become necessary requires undisturbed adherence to clear, simple, logical principles. What, however, is to be drawn from our criticism of the system heretofore used – what I have endeavored to illustrate and develop by single instances – what the philosophical sentences I have quoted bring before us in a condensed form – we may contract into one word. So far as our special problem is concerned, the question is to make the science of machinery deductive. The study must be so formed that it rests upon a few fundamental truths peculiar to itself. The whole fabric must be reducible to their

² P. N. Sheth and J. J. Uicker, "IMP (Integrated Mechanisms Program), a Computer-Aided Design Analysis System for Mechanisms and Linkages," *Journal of Engineering for Industry, ASME Transactions*, vol. 94, May 1972, pp. 454–64.

³ For an up-to-date version of IMP in open-source form (GNUPL, version 3), the reader should see <http://code.google.com/p/impsim/>.

⁴ Reuleaux, *op. cit.*

strictness and simplicity, and from them again we must be able, conversely, to develop it. Here again is a point from which the weakness of the method heretofore employed can be surveyed at a glance. Its difference from the ideal method is not that it employs the inductive instead of the deductive method; that indeed would be no advantage but it might still be defensible. No, it has been entirely unmethodical. It has chosen no fixed method of investigation, or rather, it has not found any in spite of zealous search; indeed it has so often cried "*Eureka*" that it now rests quietly in the impression that such fixed standpoint has really been found.

About the Authors



John J. Uicker is Professor Emeritus of mechanical engineering at the University of Wisconsin–Madison. He received his BME degree from the University of Detroit, and his MS and PhD degrees in mechanical engineering from Northwestern University with Professor J. Denavit as his advisor. He joined the University of Wisconsin faculty in 1967, where he served until his retirement in 2007. Throughout his career, his teaching and research have been in solid geometric modeling and the modeling of mechanical motion, and their application to computer-aided design and manufacture; these include the kinematics, dynamics, and

simulation of articulated rigid-body mechanical systems. He was the founder of the UW Computer-Aided Engineering Center and served as its director for its initial ten years of operation. He is a member of the American Society of Mechanical Engineers (ASME); the American Society for Engineering Education (ASEE); Society of Automotive Engineers (SAE); and the Tau Beta Pi, Sigma Xi, and Pi Tau Sigma professional engineering honor societies. He has served on several national committees of ASME and SAE, and he received the ASME Mechanisms Committee Award in 2004 and ASME Fellow Award in 2007. He is one of the founding members of the U.S. Council for the Theory of Mechanism and Machine Science (USCToMM) and of IFToMM, the International Federation of Mechanism and Machine Science. He served for several years as editor-in-chief of the *Mechanism and Machine Theory* journal of the federation. He is a registered mechanical engineer in Wisconsin and has served for many years as an active consultant to industry.

As an ASEE Resident Fellow, he spent 1972–73 at Ford Motor Company. He was also awarded a Fulbright-Hayes Senior Lectureship and became a visiting professor to Cranfield Institute of Technology in Cranfield, England, in 1978–79. After graduate study under the originators, Professors Denavit and Hartenberg, he became the pioneering researcher on transformation matrix methods of linkage analysis, and he was the first to advance their use into the dynamics of mechanical systems. He has been awarded twice for outstanding teaching, three times for outstanding research publications, and twice for historically significant publications.



Bahram Ravani is a professor of mechanical engineering at University of California, Davis. He received his BS degree Magna Cum Laude from Louisiana State University in Baton Rouge; his MS degree, with distinction, from the College of Engineering, Columbia University in New York; and his PhD degree from Stanford University, in Stanford, California, all in mechanical engineering. From 1982–87, he was on the faculty of mechanical engineering at the University of Wisconsin–Madison, first as an assistant professor and later as a tenured associate professor. He then joined the University of California, Davis, where he has been performing teaching and research in the areas of kinematics and dynamics, mechanical design, robotics and mechatronics, collision mechanics, and biomechanics. He served as the chair of the department of Mechanical and Aeronautical Engineering, as well as the interim chair of Electrical and Computer Engineering. At Davis, he is also a member of the graduate program in Biomedical Engineering and the graduate program in Forensic Science and Engineering. In 1985, he was on leave from the University of Wisconsin and worked for the Manufacturing Systems Product Division of IBM Corporation in Boca Raton, Florida. He was also a visiting professor in the Department of Mechanical and Production Engineering at the Katholieke Universiteit of Leuven in Belgium during the summer of 1987. In 1987, he was the recipient of the Young Manufacturing Engineer award from the Society of Manufacturing Engineers, and in 1993 he was the recipient of the Gustus L. Larson Memorial award from ASME for outstanding achievements in mechanical engineering within 10 to 20 years following graduation. In 1997, he received the Design Automation award from the Design Engineering Division of ASME for his lifetime of sustained contributions to the field of design automation. In 2005, he was the recipient of the Machine Design Award of ASME for eminent achievements in mechanical design. He was the technical editor of the ASME *Journal of Mechanical Design* from 1993–97 and is presently the technical editor of ASME's *Journal of Computers and Information Science in Engineering*. He is a Fellow of ASME and is the former chair of the Design Engineering Division of ASME. He is also a member of the Society of Automotive Engineers, the International Society of Biomechanics, and the Association for Advancement of Automotive Medicine.



Pradip N. Sheth (deceased, January 2009) was born in Baroda/Vadodara, India, in 1944. He earned his BE degree in mechanical engineering at Maharaja Sayajirao University, Baroda, India, in 1965, and earned his MS in 1968 and PhD in 1972, in mechanical engineering at the University of Wisconsin–Madison where Professor John Uicker served as his advisor. In his research, he developed the Integrated Mechanisms Program (IMP) for the computer-aided design and analysis of multibody mechanical systems. More than 200 industrial and educational organizations worldwide have used this system, which was the first of its kind.

Pradip was a Postdoctoral Fellow at the University of Michigan from 1972–74 and developed a vehicle analysis program as a research project for Ford Motor Company. From 1974–85, he was manager of Engineering Development at Allis-Chalmers Corporation. From 1985, Pradip served on the faculty of the Department of Mechanical and Aerospace Engineering at the University of Virginia (UVA) where he came to help establish a Master's degree program in Manufacturing Systems Engineering. This program was offered locally as well as through the educational television system, and it has produced hundreds of Master of Engineering graduates.

Pradip worked with industry throughout his academic career. He served as regional coordinator for the Manufacturing Action Program of the Virginia Center for Innovative Technology, where he worked with 75 Virginia manufacturing companies. He also helped to establish a Manufacturing Extension Program in Martinsville, Virginia. The Philpott Manufacturing Center is now a statewide resource, partially funded by the National Institute of Standards and Technology. He was also an active participant in UVA's Rotating Machinery and Controls Consortium and the Kluge Rehabilitation Center. He designed consumer products and specialized systems that vary from lawn mowers and tractors to pole-climbing robots, and an array of assistive devices for physically impaired individuals. He held several patents.

Pradip was a member of the American Society of Mechanical Engineers, the American Society for Engineering Education, the Society of Manufacturing Engineers (SME), and the Sigma Xi and Pi Tau Sigma engineering honor societies.

1 Concepts and Definitions

1.1 Mechanical Design: Synthesis versus Analysis

There are two completely different aspects of the study of mechanical systems: *design* and *analysis*. The concept embodied in the word *design* might be more properly termed *synthesis*, the process of contriving a scheme or a device for accomplishing a given purpose. Design is the process of developing the sizes, shapes, material compositions, types and arrangements of parts, and manufacturing processes so that the final system will perform a prescribed task. Although there are many phases of the design process that can be approached in a well-ordered scientific manner, the process is, by its very nature, as much an art as a science. It calls for imagination, intuition, creativity, judgment, and experience. The role of science in the design process can be viewed as providing tools to be used as the designer practices this art. Computer programs and computations that allow a designer to simulate a system and evaluate its potential performance play an important role in helping the designer practice the art. This is why scientific techniques such as the matrix methods discussed in this text play such an important role in dealing with the design of three-dimensional mechanisms and multibody systems.

In the *synthesis* of a mechanical system, from a functional point of view, there are three basic stages that correspond approximately to three basic steps in the design process. The first stage is designated *type synthesis*; it deals with the fundamental decisions a designer makes regarding the style of machine, device, or system to be used. Initially, for example, such decisions include whether a mechanical device should be used at all, or whether an electronic circuit or hydraulic appliance should be chosen instead. After deciding on the use of a mechanism or multibody system, for example, we must then ponder the relative merits of linkages as compared with gear trains or perhaps belts and pulleys.

Once the type synthesis has been accomplished, we have established some general boundaries for the overall system; further study must then go into specifying its basic internal characteristics. The numbers of parts and the types and numbers of joints connecting them must be decided. This process is called *number synthesis*. At this stage, we do not concern ourselves with the detailed shapes of the parts or their strength or wear characteristics, but we are concerned with their

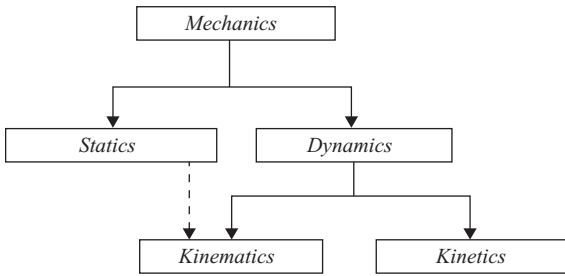


Figure 1.1. The science of mechanics.

overall arrangement. Typical questions considered at this stage include “Will this configuration have the desired degrees of freedom, and can it provide the functionality that is intended?”

Given at least tentative answers to these questions, we are in a position to attempt the third step, *dimensional synthesis*. It is here that we assign dimensions, materials, weights, strengths, and other properties to each of the members or parts of the design. Either by calculation, by experiment, or by intuition and experience, we make all of the detailed decisions that are necessary before the product or system can be manufactured. It is during the process of evaluating the various interacting alternatives and choosing among them that we find the need for a collection of mathematical and scientific methods in the hope of finding at least a valid – and perhaps even an optimal – selection for the given task. These scientific tools do not make decisions for us; we have every right to exert our imagination and creative abilities, even to the extent of overriding mathematical recommendations. Science-based techniques are useful, however, in generating, comparing, and judging various alternatives.

Probably the largest collection of scientific methods at our disposal falls into a category called *analysis*. These are the techniques that allow us to critically examine an already existing or proposed design in order to judge its suitability for a given task. Thus, in itself, analysis is not a creative science, but rather is used for evaluating and rating things already conceived. In fact, it can be used to help the creative process by allowing a formal evaluation of a design and allowing the designer to accept or dismiss a concept or to find ways to improve it. Therefore, analysis is a useful tool in redesign or design improvement, and can be integrated with the creative process. We should always bear in mind, however, that although the majority of our efforts may be spent on analysis, the real goal is synthesis – the design of a product or system. Analysis is simply a tool. It is, however, a vital tool and will invariably be used during the design process. This is particularly true when the analysis techniques lend themselves to computer software and programmed computations because this allows a designer to simulate different concepts and compare the performance of competing design alternatives.

The branch of scientific analysis that deals with motions and forces in a mechanical system is called *mechanics*. As shown in [Figure 1.1](#), it is made up of two parts, called *statics* and *dynamics*. Statics deals with the analysis of stationary systems, that is, those in which time is not a factor. Dynamics, on the other hand, deals with systems that change with time.

Dynamics is also made up of two major disciplines. The great Swiss mathematician, Leonhard Euler (1707–83), was the first to distinguish these [2]:

The investigation of the motion of a rigid body may be conveniently separated into two parts, the one geometrical, the other mechanical. In the first part, the transference of the body from a given position to any other position must be investigated without respect to the causes of the motion, and must be represented by analytical formulae which will define the position of each point of the body after the transference with respect to its initial placement. This investigation will therefore be referable solely to geometry, or rather to stereotomy [the art of stone-cutting].

It is clear that by the separation of this part of the question from the other, which belongs properly to Mechanics, the determination of the motion from dynamic principles will be made much easier than if the two parts were undertaken conjointly.

These two aspects of dynamics were later recognized as the distinct sciences of *kinematics* and *kinetics*, which treat the motion and the forces producing it, respectively. Kinematics was first defined as a separate study by the French mathematician and physicist, André Marie Ampère (1775–1836). He chose the French name *cinématique* from the Greek word *κίνημα* (*kinema*), meaning motion [1]. An interesting narrative on the history of kinematics is found in [3, pp. 1–27].

The field of kinematics, however, has grown to include not only the geometric part of dynamics but also those aspects of statics that deal with the geometry, but not the magnitudes, of the system of forces acting on the bodies. For this reason, Figure 1.1 shows a dashed line indicating the interaction of kinematics with statics. This should not be surprising because there is a well-established duality between the geometry of a system of forces and a set of velocities in kinematics.

The predominant problem in multibody system analysis, as will become evident, is often one of kinematics – a topic of major emphasis in this book. Statics and kinetics, however, are also important parts of any complete design analysis, and these topics are also covered in detail.

1.2 Multibody Systems and Mechanisms

A multibody system can be defined as a collection of bodies (mechanical parts) in which some or all of the bodies may be interconnected by joints that constrain the relative motions between the joined bodies. However, the presence of joints or connections is not an absolute requirement for a multibody system; the bodies may be “restrained,” rather than constrained, by interconnections with other bodies by elements such as springs or dampers. There are a number of abstract concepts that must be further considered for a rigorous understanding and for purposes of modeling a multibody system; these include (1) body, (2) joint, (3) constraint, (4) restraint, (5) spring, and (6) damper.

The general definition of a multibody system covers a very large variety and many different kinds of mechanical systems. The radio-controlled model car shown in Figure 1.2 is one example of a multibody system.

The NASA Mars Exploration Rover, Figure 1.3, is another example of a multibody system. Biomechanical models of the human body, as shown in Figure 1.4, and

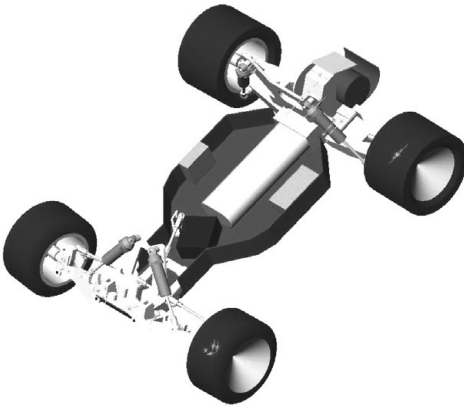
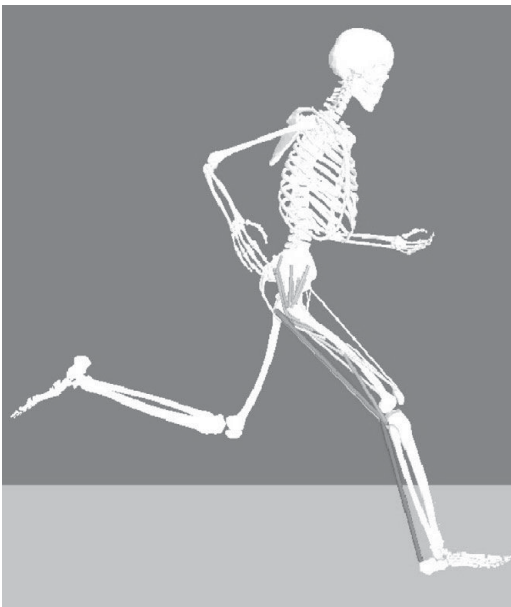


Figure 1.2. Multibody model of a radio-controlled car showing the front and rear suspension systems.



Figure 1.3. NASA Mars Exploration Rover.



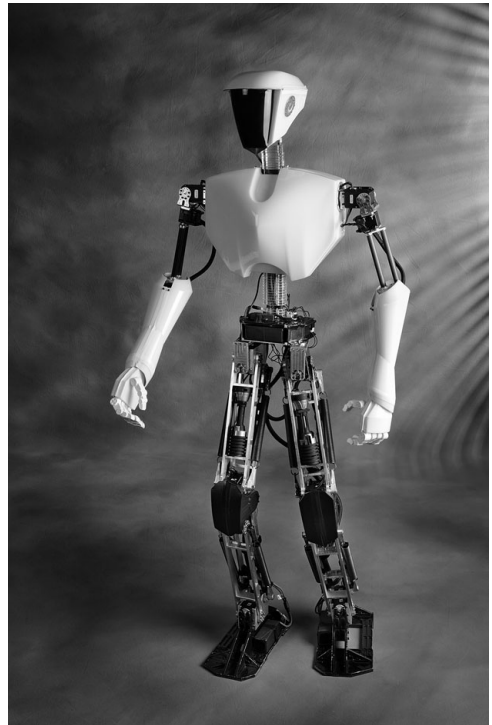
(a)



(b)

Figure 1.4. (a) A biomechanical model for studying human gait, (b) Detailed model of a human knee. (Courtesy Prof. Darryl Thelen, University of Wisconsin, Madison, WI).

Figure 1.5. Humanoid robot CHARLI-2, winner of the RoboCup 2011 World Soccer Competition (Courtesy John McCormick and Prof. Dennis Hong, Robotics and Mechanisms Lab, Virginia Polytechnic Institute, Blacksburg, VA).



also bipedal walking robots, such as that in Figure 1.5, represent additional examples of multibody systems.

The Gough/Stewart platform, shown in Figure 1.6, has been a popular system for a number of applications since the 1960s, including many recent adaptations in parallel robotic systems.

Parallel (Figure 1.7) and serial (Figure 1.8) manipulators are also examples of multibody systems.

Mechanisms constitute an important category of multibody systems. Of course, the variety of possible systems is unlimited. One example of a mechanism is the automotive suspension system shown in Figure 1.9.

Speaking rigorously, a *mechanism* is defined as an assemblage of mechanical bodies, movably connected by joints to form a mechanical system with one body fixed and having the purpose of transforming motion. Whereas a mechanism is

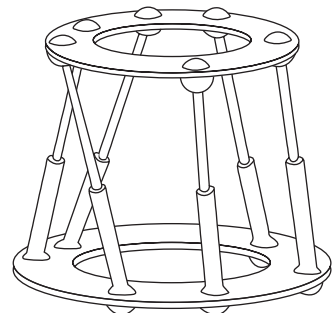


Figure 1.6. The Gough/Stewart platform. Parallel (Figure 1.7) and serial (Figure 1.8) manipulators are also examples of multibody systems.

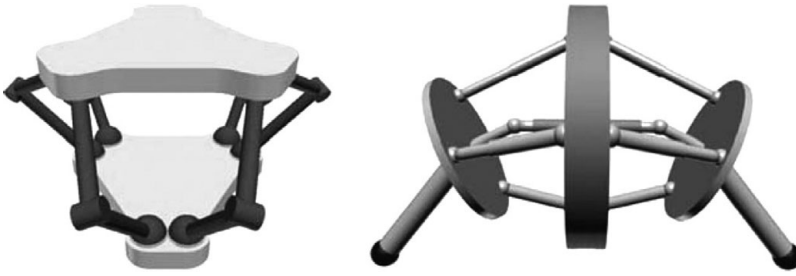


Figure 1.7. Parallel manipulators.

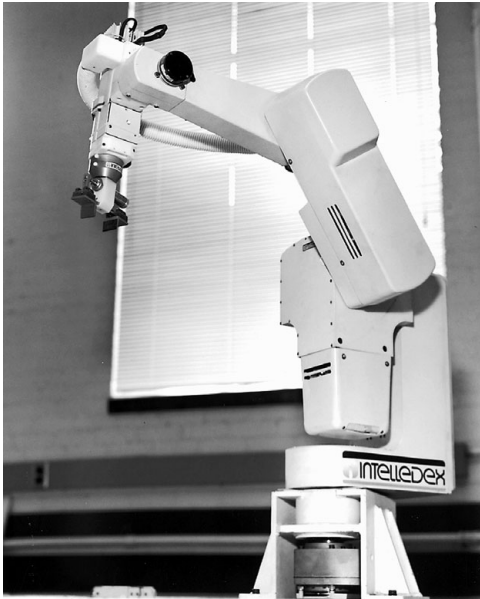


Figure 1.8. Serial manipulator.

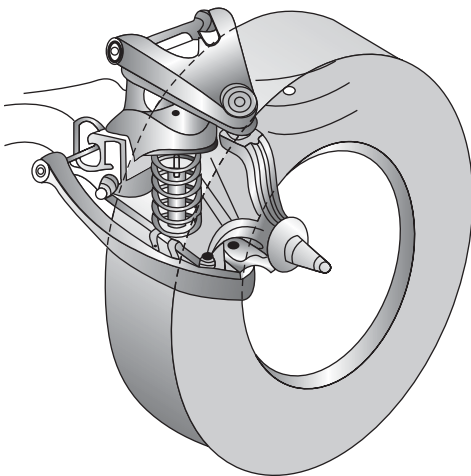


Figure 1.9. Automotive independent front suspension mechanism.

considered to have one of its bodies fixed, a general multibody system, in contrast, may be an unrooted, free-floating system. This definition of a mechanism includes several terms that must themselves be more precisely defined, which is the inherent pitfall of any first definition. However, a start must be made somewhere, and as such, this definition is perhaps as good as any.

Much of the material in this chapter is based on definitions originally established by Professor Franz Reuleaux (1829–1905), a German kinematician whose work [6] marked the beginning of a systematic treatment of kinematics. For an English translation, including additional reading, see British engineer and academic, Alexander Blackie William Kennedy (1847–1928) [5]. Reuleaux's second book [7] also made a lasting impression but, unfortunately, has not been translated into English.

Some light is shed on the meaning of the word “mechanism” by discussing what is *not* meant. Let us distinguish first between the words “mechanism” and “structure.” A *structure* is also an assemblage of mechanical bodies connected by joints, but its purpose is definitely not to transform motion. A structure, such as a truss, is intended to be rigid. It can, perhaps, be mobile in the sense of being movable from place to place. However, it has no internal mobility; no relative motion takes place between its parts or members. A mechanism, on the other hand, does have this freedom among its various members to move relative to one another. Indeed, the whole purpose of a mechanism is to utilize these relative motions for transforming or modifying a given input motion to produce a different output motion. For example, a shaft set in a pair of bearings is not a mechanism, because the intent is to transmit the input motion to the output, rather than to transform it, but it can be viewed as a multibody system. A speed-reducing set of gears between input and output shafts, on the other hand, does form a mechanism.

This brings us to distinguishing between the words “machine” and “mechanism.” A *machine* is an assemblage of fixed and moving bodies for doing work, a device for applying power or changing its direction. It differs from a mechanism in its purpose. In a machine, force, torque, work, and power are the predominant concepts. In a mechanism, even though it may transmit power or force, the predominant concept is one of altering motion. Both machines and mechanisms are multibody systems with multiple masses and may contain elements such as springs and frictional damping elements.

1.3 Planar, Spherical, and Spatial Mechanisms

Mechanisms, like many other things, may be categorized in several different ways in order to emphasize their similarities and differences. One such grouping divides mechanisms into planar, spherical, and spatial categories. Of course, all three groups have many things in common, but there must also be some criterion to distinguish them. In this instance, the criterion is found in the characteristics of the motions of the individual bodies.

A *planar* mechanism is one in which all moving points describe planar curves and in which all of these curves lie in parallel planes. That is, the loci of all points are planar curves, all parallel to a common plane. Owing to this characteristic, it is possible to represent the locus of any chosen point in its true size and shape in a single drawing or figure. The motion transformation of any such mechanism is

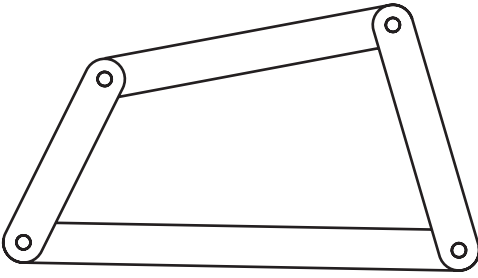


Figure 1.10. Planar four-bar linkage.

called *coplanar*. The planar four-bar linkage (Figure 1.10), the disk-cam and follower (Figure 1.11), and the slider-crank mechanism (Figure 1.12) are familiar examples of planar mechanisms.

A *spherical* mechanism is one in which each moving body (or its extension) has one point that remains stationary as the system moves, and in which the stationary points of all bodies lie at a common location. That is, the locus of any point is a curve contained in a spherical surface and the spherical surfaces defined by arbitrarily chosen points are all concentric. The motions of all particles, therefore, can be completely described by their radial projections on the surface of a sphere with a properly chosen center. The Cardan/Hooke universal joint (Figure 1.13) is perhaps a familiar example of a spherical mechanism.

Spatial mechanisms, on the other hand, include no restrictions on the relative motions of their bodies. The motion transformation is not necessarily coplanar, nor must it be concentric. A spatial mechanism may have particles with loci of double curvature. Any linkage that contains a screw joint, for example, is a spatial mechanism, because the relative motion within a screw joint is helical. Examples

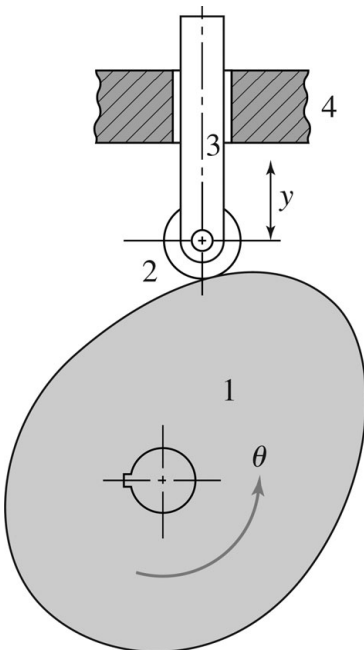
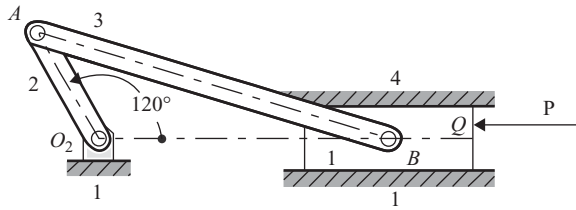


Figure 1.11. Disk-cam and follower.

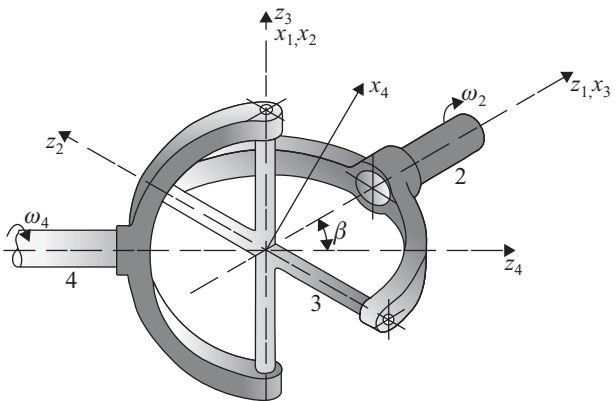
Figure 1.12. Slider-crank mechanism.

of spatial mechanisms are industrial robots (Figure 1.14) and the human skeletal system (Figure 1.4).

It should be pointed out that the overwhelmingly large category of planar mechanisms as well as the category of spherical mechanisms are only special cases or subsets of the all-inclusive category – *spatial* mechanisms. They occur as a consequence of special geometry in the locations and orientations of their joint axes. Unique geometric situations yield their own particular mechanisms.

If planar and spherical mechanisms are only special cases of spatial mechanisms, why is it desirable to identify them separately? Because of the particular geometric conditions that identify these types, simplifications are possible in their design and analysis. As previously mentioned, it is possible to observe the motions of all points of a planar linkage in true size and shape from a single direction. In other words, all motions can be represented graphically in a single view. Thus, graphic techniques are well suited to their analysis, as demonstrated by the abundance of texts such as [9] on the kinematics of mechanisms. Because spatial mechanisms do not enjoy this special geometry, visualization can become difficult, and more powerful techniques are needed for their analysis.

Because the vast majority of mechanisms in use today are planar, we may question the need for the more complicated techniques developed in later chapters. There are several reasons why more powerful methods are of value for such systems, even though the “simpler” graphic techniques may have been mastered. First, they provide new, alternative methods that solve problems in a different way. Thus, they provide a means for checking results. Certain problems by their nature may be more amenable to one method than to another. Second, methods that are analytic in nature are better suited to solution by digital computation than are graphic techniques and, therefore, can be analyzed with higher accuracy. Third, even though

Figure 1.13. Cardan/Hooke universal joint.

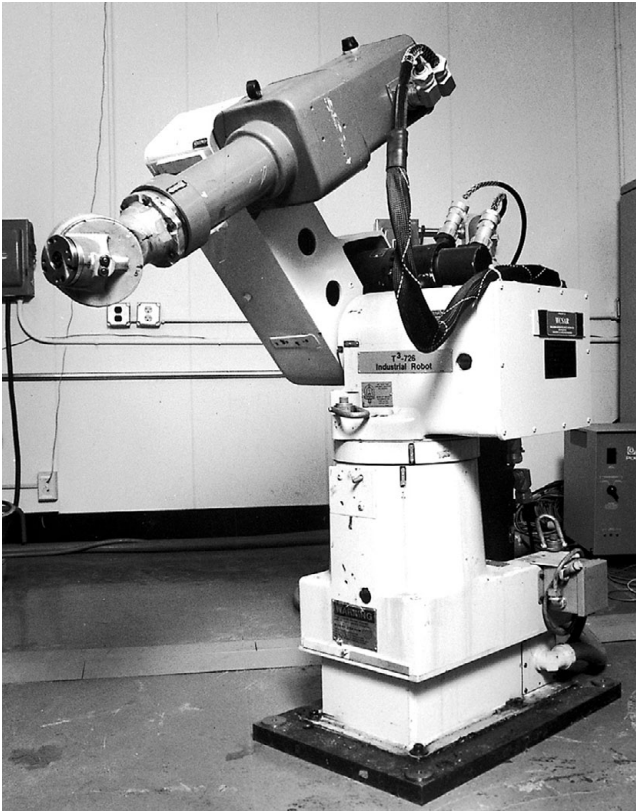


Figure 1.14. An industrial robot.

the majority of useful linkages are planar and well suited to graphic solution, the few remaining must also be analyzed, and techniques should be known for analyzing them. Fourth, a possible reason that planar linkages are so common is that good methods for analysis of the more general spatial systems have not been readily available until recent years. Therefore, their design and use have not been common, even though they may be inherently better suited in certain applications.

Finally, spatial mechanisms are much more common in practice than their formal description indicates. Consider a “planar” four-bar linkage (Figure 1.10). It has four bodies connected by four pin joints whose axes are “parallel.” This parallelism is a mathematical hypothesis; it is not a reality. The joint axes, as produced in a shop – in any shop, no matter how good – are only approximately parallel. If the axes are nearly parallel, the system operates because of looseness in the bearings or flexibility of the bodies. If the joint axes are far out of parallel, there is binding in no uncertain terms, and the system only moves because the bodies flex and twist, producing loads in the bearings. A common way of compensating for non-parallelism is to connect the bodies with self-aligning bearings, actually spheric joints allowing three-dimensional rotation. Such a “planar” linkage is, thus, really a low-grade spatial mechanism.

1.4 Mechanical Body

Let us now look more closely at a term that has been used frequently in previous sections. The term is “body,” or more precisely, “mechanical body.” In this text, a

mechanical body is defined as a physical component of a machine, mechanism, or multibody system that is considered completely rigid, and that may contain joint elements for connecting it to other bodies.

The key concept in this definition is that of *rigidity*. Because the purpose of a mechanism is to transform motion, its analysis usually begins with a study of its kinematics. The assumption that bodies are rigid is a key in isolating kinematic effects from those of kinetics; it allows major simplifications in the analysis process. Stated explicitly, the assumption is that there is no change in distance between arbitrarily chosen points of the same mechanical body no matter what load is applied. Detailed consideration of deformations or flexibilities in mechanical bodies requires a separate and comprehensive treatment and there is much past and recent literature, for example [4], on the subject. For this reason, this topic is not covered in this text.

It is true that no real machine member is completely rigid; each has elastic (and also thermal) properties characteristic of its shape and material. As such, a mechanical body is an idealization of a real machine component. However, it is this idealization that allows the kinematics of a mechanical system to be studied separately from kinetic (and thermal) effects. Machines that depend on flexibility of their members for their motion, such as the four-bar linkage with nonparallel axes discussed earlier, cannot be idealized as consisting of mechanical bodies. Analysis techniques for such systems either must accept this approximation, or they will necessarily be complicated by the need for simultaneous kinematic and kinetic (and perhaps even thermal) analyses [8].

Whereas a real machine member is made up of particles of mass and has material properties, a mechanical body has only geometric properties – that is, points or locations, lines, and planes. This brings us to the concept of the *extended* mechanical body. The entire three-dimensional space that contains a mechanical body and that moves with the body can be thought of as an extension of that body. Because of this concept, it can be quite proper to speak of points on a body that lie outside of the boundaries of its physical shape. In addition, it is permissible to speak of coincident points or locations on two or more bodies, even though two different physical particles cannot occupy the same space at the same time.

As mentioned in its definition, a mechanical body may carry the elements (mating surfaces) of joints that connect it to other bodies. Thus, bodies can be subdivided into categories wherein *nullary* bodies describe those carrying no joint elements, *unary* bodies carry a single-joint element, *binary* bodies carry two, *ternary* bodies carry three, and so on. It should be noticed that, in kinematics, the primary function a body serves is to ensure that the relative locations and orientations of its joint elements do not change – that is, the purpose of a body is to hold its joint elements and other shape features in constant geometric relationships.

1.5 Mechanical Chain and Kinematic Inversion

When several mechanical bodies are movably connected by joints, they are said to form a *mechanical chain*. If every body in the chain is connected to at least two others as in Figure 1.15b,c the chain comprises one or more closed loops and is called a *closed chain*; if not, the chain is referred to as *open*, as in Figure 1.15a. If the chain consists entirely of binary bodies, as in Figure 1.15b, it is a *simple chain*. *Compound*

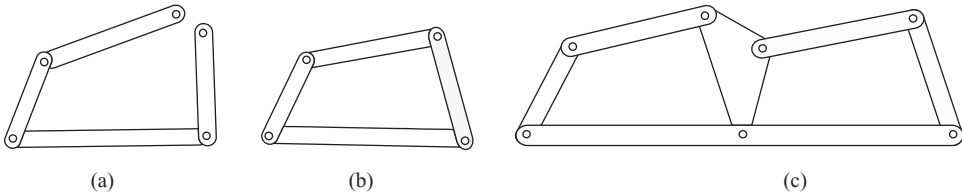


Figure 1.15. (a) Open mechanical chain, (b) simple closed chain, (c) compound chain.

chains, however, contain other than binary bodies and may form more than a single closed loop. An example is shown in [Figure 1.15c](#).

Referring to the previous definition of a mechanism, we see that it is necessary to have one body fixed. When we say that a body is fixed, we mean that it is chosen as the frame of reference for the movement of other bodies; that is, that the motions of other points of the mechanical system are measured with respect to a coordinate system attached to the fixed body. The fixed body in a practical machine usually takes the form of a stationary platform or base or housing rigidly attached to such a base, and is called the *frame* or *ground* or *base*. The question of whether this reference frame is truly stationary (in the sense of being an inertial frame of reference) is immaterial in the study of kinematics because masses are neglected, but does become important in the investigation of kinetics when inertial forces become important. In any case, once a frame member is designated (and other conditions are met), as the inputs are moved through continually changing positions, all other bodies have well-defined motions with respect to the chosen frame.

If, for the same mechanical chain, a different body is chosen as the frame, the *relative* motions between the various bodies are not altered, but their *absolute* motions with respect to the new base may be dramatically different. The process of changing the frame of reference or the base link of a mechanical system – that is, designating a different body as the fixed frame – is known as *kinematic inversion*. An example is shown in [Figure 1.16](#).

1.6 Joints and Joint Elements

One contributing factor in determining the relative motions of two points in a mechanical system is the assumption that all bodies are rigid and that, therefore, two points of the same body can only move on spherical loci with respect to each other. However, this fact alone is not enough to completely specify the kinematics of a mechanism or multibody system because it tells nothing about the relative motions of points on different bodies. These relative motions between bodies cannot be arbitrary. These too must be constrained or, at least, restrained to have the

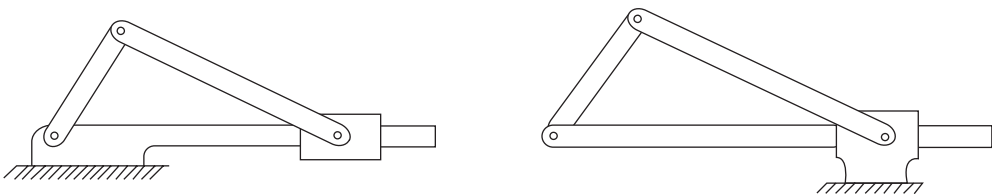


Figure 1.16. Example of kinematic inversion.

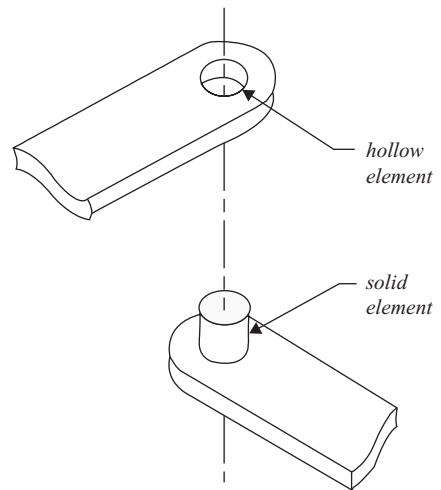


Figure 1.17. Hollow and solid elements of a revolute joint.

proper relative motions – those chosen by the designer for the particular task to be performed.

In kinematics, a *joint* is defined as a mechanical connection between two (and only two) bodies. The designer restricts, but does not necessarily eliminate, the relative motions allowed between bodies by selecting the types and locations of joints used in connecting the bodies. Thus, we are led to the conclusion that, in addition to the constant (rigid) geometric relationship between joint elements on the same body, the nature of the joints and the relative motions that they permit between the attached bodies are essential in determining the kinematics of a mechanism or multibody system. For example, in the human musculoskeletal system, it is the joints that allow movement of the limbs with respect to one another, providing both mobility and dexterity. For other mechanical systems such as robotic manipulators, joints are chosen such that they can be easily driven by actuators, such that their positions and/or velocities can be determined by sensors, and such that they can be controlled by automatic control systems. It is therefore important to look closely at the nature of joints in both general terms and through common types.

Another important reason to classify some of the common types of joints has to do with developing a general method of analysis that will lend itself to digital computation. In general, the bodies of different types of multibody systems come in an unlimited variety of shapes, sizes, mass properties, and so on. The types and variety of joints, however, is more limited and therefore, we can separately study and catalogue the more common types. These can then be chosen as needed to provide the governing equations for different multibody systems.

Because a joint connects two mechanical bodies, the joint is not a physical entity in itself; it is composed of two constituents, the mating surfaces on the two connected bodies. The two mating surfaces, when considered separately, are each referred to as a *joint element* and, when they are joined together, they form a *kinematic pair* (of elements) or a *joint*. If it is desirable to distinguish between the two elements making up a joint, their shapes usually make it natural to refer to one as the *hollow element* and the other as the *solid* or *full element* as shown in [Figure 1.17](#). Some joint elements do not have this obvious hollow and solid geometry, as for example in the

case of a flat joint (see Figure 1.20f). The two may then be arbitrarily distinguished by assigning one term to each.

The controlling factor that determines the relative motion(s) allowed by a given joint are the shapes of the mating surfaces or elements. Each type of joint has its own characteristic shapes for its elements, and each allows a given type of motion that is determined by the possible ways in which these two elemental surfaces can move with respect to each other. For example, a pin joint usually has cylindrical elements with provision on the ends to prevent axial motion. Assuming a good fit, without backlash, these surfaces only permit relative rotational motion. Thus, a pin joint allows the two connected parts to experience relative rotation with respect to each other about their common axes. So, too, other joint types each have their own characteristic element shapes and relative motions. These element shapes restrict the otherwise arbitrary motions of two unconnected bodies to some prescribed type of relative motion, thus producing *constraints* on the motion of the total mechanical system.

It should be pointed out that the element shapes may sometimes be subtly disguised and difficult to recognize. For example, a pin joint might include a roller bearing, and the two mating surfaces, as such, may not be recognized as those of a pin joint. Nevertheless, if the motions of the individual rollers are not of interest, the overall relative motion allowed by such a joint is not different and the joint is still of the same generic type; it allows relative rotation about a single axis. So, too, the diameter of the pin used (and, in most cases, other dimensional data) are of no more importance in kinematics than the exact sizes and shapes of the connected bodies. Thus, the criteria that distinguish different joint types are the relative motions that they permit rather than the detailed physical shapes of the elements, even though these may provide vital clues.







As previously stated, the primary function of a mechanical body is to hold fixed geometric relationships between its joint elements. In a similar way, the primary kinematic function of a joint is to constrain the relative motions allowed between the connected bodies. Other features are determined for other reasons and are unimportant in the study of kinematics.

When a kinematic problem is formulated, it is necessary to recognize the types of relative motion permitted by each of the joints and to assign to each some variable parameters for describing these relative motions. Because many of these parameters are required as degrees of freedom in relative motion allowed by the joint in question, they are referred to as *joint variables*. Thus, the only joint variable of a pinned joint is an angle measured between reference lines fixed in the adjacent joint elements, whereas a ball-and-socket joint has three joint variables (perhaps three angles) to describe the relative three-dimensional rotation allowed between its elements.

1.7 The Six Lower-Pairs

Because a joint implies joining the elements of a pair of bodies, a *kinematic pair* was the name given to a joint by Reuleaux [6]. He also divided pairs into two categories that he called *higher-* and *lower-pairs*, the latter category consisting of precisely the six types listed in Table 1.1, to be described in this section. He distinguished between

Table 1.1. Characteristics of the lower-pairs

Joint type	Symbol	Diagram	Joint variable(s)	Degrees of freedom	Relative motion	Motion type
Helical	$H(\sigma)$		$\Delta\theta$ or Δs	1	helical	spatial
Revolute	R		$\Delta\theta$	1	rotation	planar
Prismatic	P		Δs	1	rectilinear	planar
Cylindric	C		$\Delta\theta$ and Δs	2	cylindric	spatial
Spheric	S		$\Delta\theta, \Delta\theta', \Delta\theta''$	3	spheric	spheric
Flat	F		$\Delta s, \Delta s', \Delta\theta$	3	planar	spatial

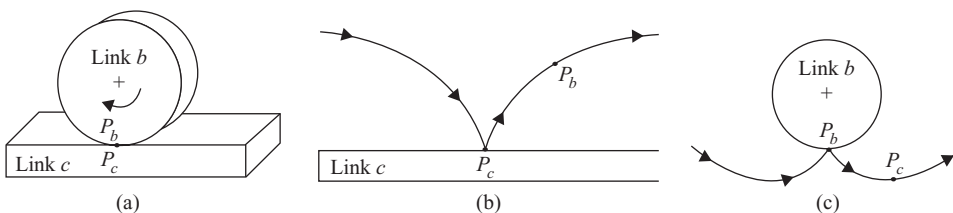
the categories by noting that lower-pairs, such as the pin joint, have *surface contact* between their joint elements, whereas higher-pairs, such as the connection between a disk-cam and its roller follower, have *line or point contact* between their elemental surfaces. However, as previously noted, this criterion may be misleading. Instead, we should look for distinguishing features in the relative motion(s) that the joint allows.

Consider two machine components – that is, two bodies of a multibody system – labeled body b and body c (one circular and the other straight) rolling on each other without slip as shown in [Figure 1.18a](#). Choosing two coincident points – P_b and P_c of bodies b and c , respectively – at the point of contact ([Figure 1.18a](#)), notice the locus that each traces in a coordinate system fixed to the other body as the joint moves. In this case, the point P_b traces a cycloid on body c , ([Figure 1.18b](#)), whereas the point P_c of the straight body c traces an involute on the circular body b , ([Figure 1.18c](#)). The characteristic to be noticed is the dissimilarity between the two curves; this is a mark of a higher-pair.

In a lower-pair, such as a pin joint, on the other hand, the loci traced by a pair of coincident points of the connected bodies are *similar* curves (circles for a pin joint), differing only in the directions in which they are traced. A corollary to this is the observation that interchanging the hollow and solid elements of a lower-pair between the two bodies does not affect their relative motion. This is illustrated for a pin joint in [Figure 1.19](#).

The six joint types defined by Reuleaux [6] to form the category referred to as lower-pairs are listed in [Table 1.1](#) and are illustrated in [Figure 1.20](#).

The simple kinematic notation for each joint, shown in column two of [Table 1.1](#), can be used to provide the following information about a mechanism or a robotic manipulator as part of a multibody system: namely, the types of joints between

**Figure 1.18.** Contrast of relative motions between elements of a higher-pair.

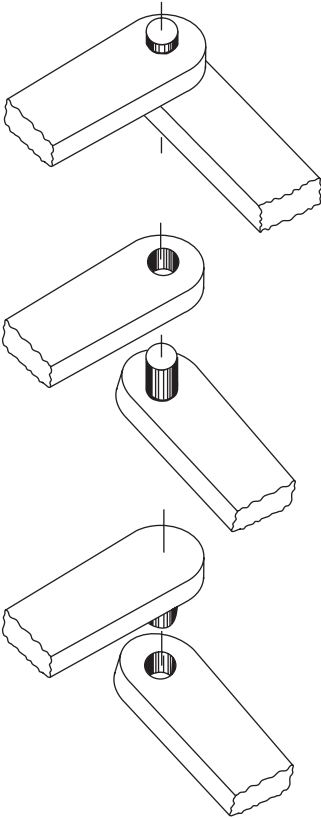


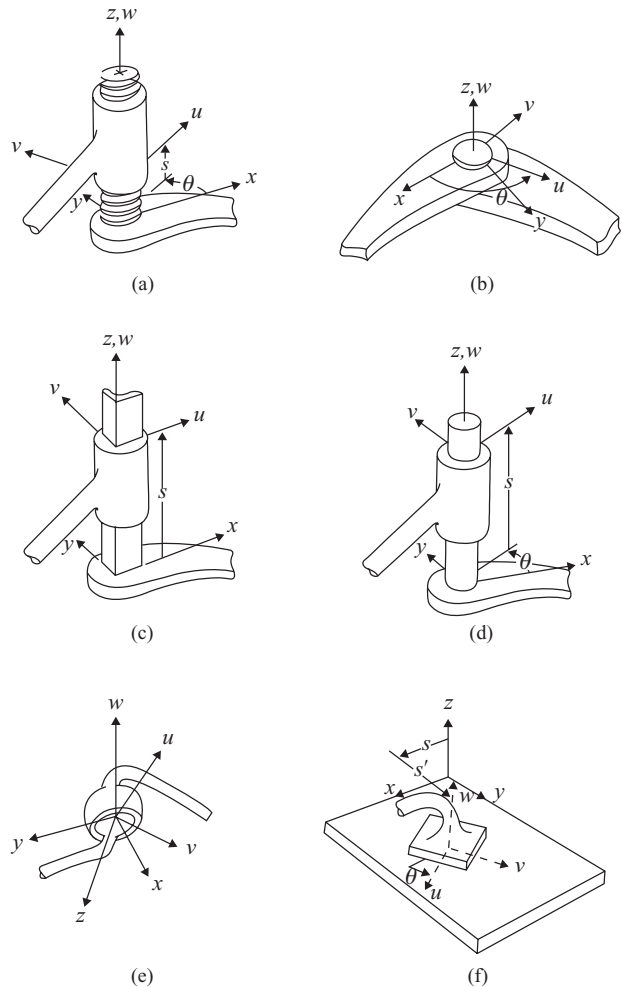
Figure 1.19. Inversion of the hollow and solid elements of a revolute joint.

mechanical bodies and the order in which the joints appear in a chain. For example, one can refer to a multibody system as an *RSSR* mechanism. This means that the first joint in the chain is a revolute (*R*) joint connecting the fixed (or the first) body to the second body of the system. Next, the second body is connected to the third body by a spheric (*S*) joint and this third body is also connected to the fourth body by another spheric (*S*) joint. Finally, the fourth body is connected back to the first or fixed body by a revolute (*R*) joint.

The six lower-pair Reuleaux joints are sometimes described symbolically using skeletal diagrams that simplify the sketching of different mechanisms, robotic manipulators, or some multibody systems. These are shown for each of the six lower-pairs in column three of [Table 1.1](#).

The most general of the six lower-pairs defined by Reuleaux, the one from which all others may be derived, is the *helical* or *screw* joint. It carries the symbol $H(\sigma)$ where σ represents the pitch of the helix. The helical joint consists of a solid element borne by the “bolt” and a hollow element carried by the “nut” and, as with other lower-pairs, the loci traced by coincident points are identical or, in this case, helical. If one of the joint elements is turned through an angle $\Delta\theta$ relative to the other, it also advances by an axial distance Δs . The ratio of Δs to $\Delta\theta$ is an invariant property of the helix defined as its *pitch*, $\sigma = \Delta s / \Delta\theta$, the axial advance per unit rotation. When $\Delta\theta$ is specified then Δs is determined and conversely. Therefore, either $\Delta\theta$ or Δs may be used as the joint variable defining the relative displacement of the elements.

Figure 1.20. The six lower-pairs: (a) helical joint, (b) revolute joint, (c) prismatic joint, (d) cylindric joint, (e) spheric joint, (f) flat joint.



However, because Δs and $\Delta\theta$ are interrelated through the pitch, the joint has only one degree of freedom in relative motion, and only one of these variables may be used as the joint variable.

Two other lower-pairs are formed by allowing the pitch of a helical joint to take one of its two extreme values. On setting the pitch to zero, we see that Δs becomes zero and the “threads” become circumferential bands; in this case, only relative rotation is possible. This type of joint, which could be written $H(0)$, is so common, appearing as a pinned joint, that it is given its own symbol R and is called a *revolute* joint. The revolute joint has only one degree of freedom and, because there is no axial advance, the joint variable is the relative rotation $\Delta\theta$ between the elements. The revolute joint is especially important in the design of many mechanisms or multibody systems because it can be simply actuated with an electric or hydraulic motor; it is in common use, for example, in robotic manipulators.

When the pitch of a helical joint is made infinite, the “threads” align themselves axially, allowing only relative axial translation. This type of joint also has one degree of freedom, and the joint variable is the relative axial translation Δs between the joint elements. Again, this pair could be denoted symbolically as $H(\infty)$ but it is so

important, appearing when there is rectilinear translation, that it is given the name *prismatic* pair and is denoted by the symbol P . The prismatic joint is also very simply actuated, as by a hydraulic or pneumatic cylinder.

Of these three types of joints – the helical joint and its two descendants, the revolute and prismatic joints – each has a single degree of freedom; each has its own unique type of relative motion – helical, circular, and rectilinear, respectively; each is described by a single joint variable; and each is located by its distinctive geometric axis. It is now clear that the motion of a helical joint can be obtained by a coaxial combination of a revolute and a prismatic joint with a constant ratio of the translational to the rotational motion. The remaining three lower-pairs each have more than one degree of freedom and can be thought of as equivalent to series combinations of revolute and prismatic joints.

The *cylindric* joint is equivalent to a revolute joint mounted coaxially in series with a prismatic joint. It has two degrees of freedom because the axial advance and the rotation are independent. There is no constraint between these two motions, as there was in a helical joint. Thus, $\Delta\theta$ and Δs are both joint variables. The cylindric joint is denoted by the symbol C .

The ball-and-socket joint is an example of a *spheric* or *globular* joint. It may be visualized as a combination of three consecutive revolute joints with three non-coplanar axes all intersecting at a central point. Although any three intersecting non-coplanar axes are sufficient, mutually perpendicular axes are often used for convenience in visualization and calculation. A spheric joint has three degrees of freedom and the three joint variables may be chosen as $\Delta\theta$, $\Delta\theta'$, and $\Delta\theta''$, the relative motions in each of the three independent revolute joints. The symbol S is used to denote a spheric joint. In robotic applications a spheric wrist is often used and consists of three revolute joints all intersecting at one point that is referred to as the wrist center point.

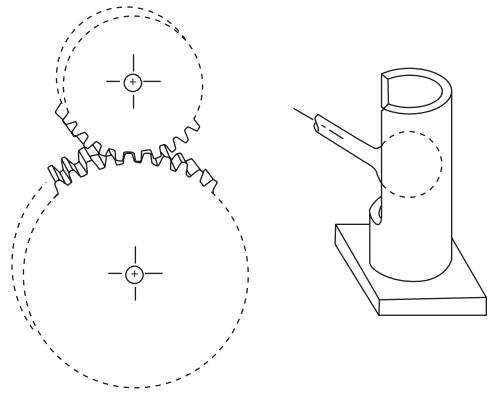
The sixth type of lower-pair is the *flat* or *planar* joint, denoted by the symbol F . The joint elements of a flat joint are planar surfaces, constrained to remain in contact but free to move by sliding on one another. The relative motion is general planar motion, having three degrees of freedom. The motion may be visualized as two non-parallel translations in the plane and one rotation about an axis normal to the plane. Thus, it is equivalent to two non-parallel prismatic joints and a properly oriented revolute joint (perpendicular to the axis directions of the two prismatic joints) connected in series. Its three joint variables can be chosen as Δs , $\Delta s'$, and $\Delta\theta$.

As pointed out earlier, the lower-pairs are very common in the construction of practical mechanisms and multibody systems and, in some cases, all of a mechanical system's joints are lower-pairs. A mechanism that consists entirely of lower-pairs is referred to as a *linkage*.

Planar linkages utilize only revolute and prismatic joints. Although a flat joint might theoretically be included, this would impose no constraints on the relative motion in the plane and thus would be equivalent to an opening in the chain. Planar motion also requires that the axes of all prismatic joints be parallel to the plane of motion and that the axes of all revolute joints be normal to the plane of motion.

Spheric linkages must be constituted entirely of revolute joints because a spheric joint would be equivalent to an opening in the chain and all other lower-pairs have

Figure 1.21. Examples of higher-pairs.



non-spheric motion. In spheric linkages, the axes of all revolute joints must intersect at the center of the spheric motion.

Robotic manipulators usually use single-degree-of-freedom joints because they can easily be controlled by independent actuators. The most common joints used in these are revolute and prismatic joints.

One could argue that Reuleaux should have included two more joint types in his compilation of lower-pairs. One of these might be called a *rigid* joint; it is defined here as a joint that has zero degrees of freedom and allows *no* motion between the connected bodies. The other might be called an *open* joint; it is defined here as a *logical* connection with six degrees of freedom and allows complete freedom between the “connected” bodies. On the other hand, one could argue that these two joint types do not truly fit the definition of a joint in the first place, and should not be included. Nevertheless, these two additional joint types, although not lower-pairs, do allow significant advantages in the simulation of multibody systems, and are included in later chapters of this text.

1.8 Higher-Pairs and Kinematic Equivalence

In spite of the common occurrence of lower-pairs in practical systems, the methods developed in later chapters would be quite inadequate if no consideration were given to higher-pairs. However, by definition, the higher-pairs include all joint types not mentioned in the list of lower-pairs, and thus are practically unlimited in variety. A systematic accounting of all higher-pairs and their motion characteristics is, therefore, not a realistic objective. Examples of higher-pairs are shown in [Figure 1.21](#).

Let us now return to the observation that two unconnected bodies have six degrees of freedom in relative spatial motion – three in relative translation and three in relative rotation – and that the objective of including a joint is to impose conditions or *constraints* on these six independent relative motions, that is to make them dependent on one another in a chosen relationship. For example, in a helical joint, two translations and two rotations are completely constrained (eliminated) and the remaining translation and rotation are made dependent on each other through the ratio defined as the pitch.

One way of analyzing mechanical systems that include higher-pairs is to find a proper combination of lower-pairs that enforces the same relative motion constraints

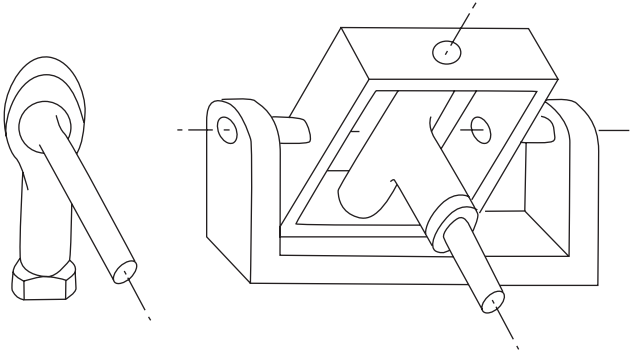


Figure 1.22. Example of kinematic equivalence.

and thus can be used as a model for the higher-pair. Such a combination is referred to as a *kinematic equivalent*. An example is shown in [Figure 1.22](#). There are no hard-and-fast rules for discovering these proper combinations and the process may require a certain amount of ingenuity on the part of the analyst, especially as the constraint conditions become more complex.

It should be pointed out that a kinematically equivalent combination of lower-pairs may be found that properly restricts the relative motions as desired, but has force transmission characteristics entirely different from the higher-pair being modeled. Thus, attention must also be given to static and dynamic equivalence if these types of analyses are to be performed on the resulting model.

1.9 Restraints versus Constraints

Joints provide connections between bodies and, in our modeling paradigm, they restrict the relative motions between the connected bodies to specific types determined by the choice of the joint. For example, a revolute joint only allows relative rotation about the joint axis between the two connected bodies. For modeling, joints are thus considered to strictly enforce specific constraints on the relative motions of the connected bodies.

On the other hand, bodies may be restrained by interconnections such as springs or dampers, which do not strictly enforce relative motion constraints, but nevertheless restrain the relative motions of the interconnected bodies through energy storage and/or release. Of course, a joint, however stiff in the directions of its constraints, may be modeled as providing restraint in the directions of its relative motion. It should be clear that the use of a joint (a constraint) or the use of a softer restraint is a modeling decision, determined for the specific system and the complexity and precision desired in modeling the system.

REFERENCES

1. A. M. Ampere, *Essai sur le Philosophie des Sciences, ou exposition analytique d'une classification naturelle de toutes les connaissances humaines* [*Essay on the philosophy of science, an analytical exhibition or natural classification of all human knowledge*], Paris, 1834.

2. L. Euler, *Theoria motus corporum solidorum seu rigidorum* [*Theory of the motion of solid or rigid bodies*], Rostock, 1765; also in *Novi commentarii Academiae Petropolitanae* [*New memoirs of the imperial academy of sciences in St. Petersburg*], vol. 20, 1776.
3. R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*, McGraw-Hill, Inc., New York, 1964.
4. L. L. Howell, *Compliant Mechanisms*, John Wiley & Sons, Inc., New York, 2001.
5. A. B. W. Kennedy, *Kinematics of Machinery: Outlines of a Theory of Machines*, McMillan & Co., Ltd., London, 1876. This is an English translation of [6], and has been reprinted by Dover Publications, Inc., New York, 1963.
6. F. Reuleaux, *Theoretische Kinematik: Grundzüge einer Theorie des Maschinenwesens*, [*Theoretical kinematics: Foundations of the theory of mechanisms*], Friedrich Vieweg & Sohn, Braunschweig, 1875.
7. F. Reuleaux, *Lehrbuch der Kinematik, Zweiter Band: Die praktischen Beziehungen der Kinematik zur Geometrie und Mechanik* [*Textbook for kinematics, volume 2: The practical relations of kinematics to geometry and mechanics*], Friedrich Vieweg & Sohn, Braunschweig, 1900.
8. A. A. Shabana, *Computational Continuum Mechanics*, Cambridge University Press, New York, 2008.
9. J. J. Uicker, Jr., G. R. Pennock, and J. E. Shigley, *Theory of Machines and Mechanisms*, 4th ed., Oxford University Press, New York, 2011.

2.1 Introduction

In order to make a systematic study of mechanisms and multibody systems and to develop general methods for their analysis by digital computer, we must be able to recognize and precisely describe certain basic information that governs their operation. For example, it is clear that, at some point, we must explicitly identify certain dimensional information, such as part shapes and dimensions, in order to perform the analysis. However, before we reach this stage, another even more fundamental problem confronts us. We must first study each system enough to determine how its various parts are interrelated – that is, which part is connected to which, and what is the nature of each connection. In other words, we need to understand the *kinematic architecture* of the multibody mechanical device. In the kinematics literature, the term “structural analysis” has sometimes been used for this type of analysis. Here, however, we use the term “kinematic architecture” to avoid confusion with the statics use of structural analysis.

In the classic methods of analysis, both graphic and analytic, the task of recognizing the architecture of a mechanism or multibody system did not require reduction to a step-by-step procedure. No real difficulties arose because the analyst, through experience, developed an intuitive feeling for analyzing problems of a given type. As the analysis progressed, he or she could continually make decisions based on experience as to what steps should be taken in what order and what techniques might be applied to accomplish each step.

However, if a general method suitable for automated computation is to be developed here, the problem of recognizing a system’s topology – the associations between its parts – must be dealt with in a more systematic manner. It is not possible to depend on a computer to “know” how to proceed through the solution process. It is necessary to discover and adopt a unified procedure, common to all situations that might be encountered.

The success of a generalized approach, therefore, depends on the development of a general algorithm whereby a computer can recognize and manipulate systems of widely differing kinematic architecture without placing awkward requirements or special conventions on the task of data preparation. Insofar as possible, all problems should be handled by one consistent procedure. The challenge, therefore, is to devise

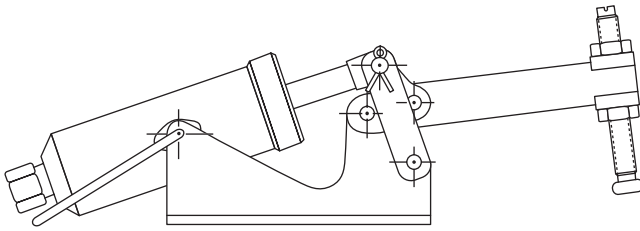


Figure 2.1. Hydraulic clamp (Lapeer Mfg. Co., Lapeer, MI).

a procedure powerful enough that it is applicable to as wide a variety of mechanisms and multibody systems as possible.

In keeping with this philosophy, the purpose of this chapter is to develop a general algorithm for studying the *topology* of a mechanism or multibody system – that is, the number of bodies, the number and types of joints, the pattern in which the bodies and joints are arranged, the number and pattern of closed loops, if any, and other such characteristics that are determined solely by the connectivity of the system. This phase of the analysis of a system is referred to as *topological analysis* or analysis of the kinematic architecture of the system.

In order to describe the kinematic architecture of a mechanism or multibody system explicitly and precisely, it is necessary to choose a format that can be understood by others for purposes of communication. It is also advantageous if this format can be manipulated directly to determine such features as closed loops. Several formats are possible and each has its own advantages depending on what features are sought through manipulation.

One way of describing topological characteristics, for example, is through a photograph or drawing such as that shown in [Figure 2.1](#). Clearly, this can provide a complete description; however, it also includes a plethora of irrelevant information that tends to disguise the true nature of the topology.

Another approach, common throughout the history of traditional kinematics, is the use of schematic drawings such as that shown in [Figure 2.2](#). Such schematics suppress many of the features that are unimportant in topological analysis. However, they still retain an aura of “shape” and can become misleading for this reason. Also, although such diagrams are well suited to human communication, they are not well suited to manipulation by computer; a numeric format would be preferable.

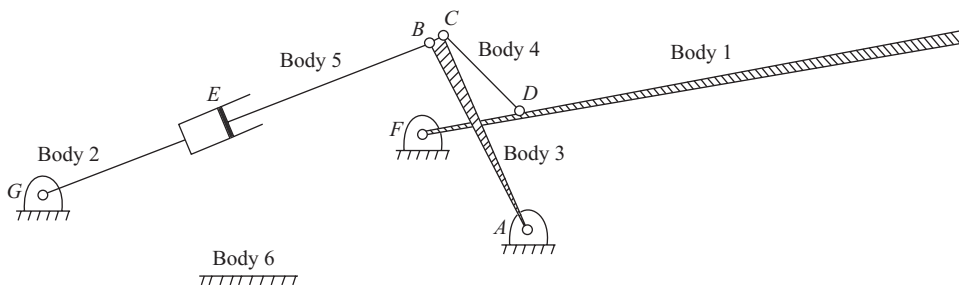


Figure 2.2. Schematic drawing for the hydraulic clamp of [Figure 2.1](#).

2.2 The Incidence Matrix

We recall from Chapter 1 that all mechanisms and multibody systems are made up of only two basic types of components: mechanical parts and joints. Their definitions are repeated here:

Mechanical Part (Body): a mechanical part is a rigid body that may contain joint elements for connecting it to other bodies.

Joint: a joint is a connection between two (and only two) joint elements of separate bodies that provides constraints on the relative motions of the two bodies joined.

Notice that a joint does not exist merely by the fact that both joint elements exist; it also requires that they be joined. Therefore, a joint is considered a separate entity in itself, distinct from the two elements.

Once all of the bodies and joints have been identified for a given system, the problem of topological analysis requires describing the relationships that exist between them. Much of the remainder of this chapter relies on the branch of mathematics titled *graph theory*; see [2] for example. Precise discussion of these relationships requires two further definitions:

Incidence: a joint is said to be *incident* with each of the two bodies containing its joint elements.

Mechanical Graph (Assembly): a mechanical graph or assembly consists of a set of bodies, a set of joints, and a prescribed incidence relationship between these two sets.

Thus, a mechanical graph uniquely defines the characteristics of a mechanism or multibody system that are essential to its topological analysis, and at the same time strips away all other features such as the geometry and material properties of the bodies. These features will, of course, be reintroduced at a later stage in the analysis, but are not pertinent to the analysis of a system's kinematic architecture. The incidence relationships for the clamp example of [Figure 2.2](#) are given in [Table 2.1](#). In this example, letters rather than numbers are used for joint labels to reduce confusion between joint labels and body labels, which are numeric. Notice also that the stationary body is intentionally chosen to have the highest body label (6); the reason for this becomes clear in section 2.5.

Table 2.1. Incidence table for [figure 2.2](#)

Joint	Connects <i>from</i> body	<i>to</i> body
<i>A</i>	6	3
<i>B</i>	3	5
<i>C</i>	3	4
<i>D</i>	1	4
<i>E</i>	2	5
<i>F</i>	6	1
<i>G</i>	2	6

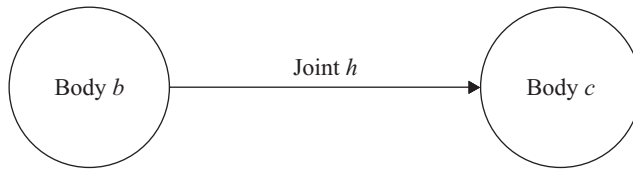


Figure 2.3. Graphic representation of kinematic architecture where each body is represented as a node and each joint is represented as a directed arc connecting two nodes.

One further property is defined in order to achieve maximum capability in the upcoming manipulation. Each joint is assigned an arbitrary *orientation*. When the orientation of joint h is chosen as oriented *from* body b *to* body c , then joint h is said to be *negatively incident* with body b and *positively incident* with body c . A mechanical graph in which every joint is assigned an orientation is called an *oriented mechanical graph* or an *oriented assembly*.

Another way to represent the topology of a complex multibody system, which perhaps aids in visualization of its kinematic architecture, is through a graphic representation in which each body is represented as a node (vertex) and each joint is represented as a directed arc (edge) of a graph, and the arc (edge) connections of the nodes (vertices) correspond to the joint connections of the bodies. When the orientation of joint h is directed *from* body b *to* body c , then the arc for joint h is directed from the node representing body b to the node representing body c , as shown in Figure 2.3.

With this convention, the oriented graph for the clamp of Figure 2.2, using the orientations given in Table 2.1, is depicted in Figure 2.4.

With the aforementioned definitions, we are now prepared to define a new and very useful format for expressing the incidence relations of a mechanism or multibody system. The information contained in the incidence table or in the oriented graph can be written in the form of an *incidence matrix*, in which each row represents a body, each column represents a joint, and the matrix entries define the incidence between corresponding rows and columns. For a system having ℓ bodies and n joints,

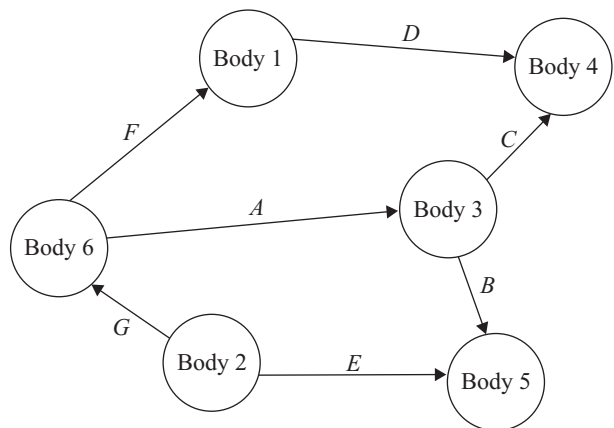


Figure 2.4. The oriented graph of the hydraulic clamp of Figure 2.2, using the joint orientations given in Table 2.1.

the incidence matrix Γ is $(\ell \times n)$ matrix. The entry in row b and column h is defined as follows:

$$\Gamma(b, h) = \begin{cases} +1 & \text{if joint } h \text{ is positively} \\ & \text{incident with body } b; \\ & b = 1, 2, \dots, \ell, \\ -1 & \text{if joint } h \text{ is negatively} \\ & \text{incident with body } b; \\ & h = 1, 2, \dots, n, \\ 0 & \text{if joint } h \text{ is not incident} \\ & \text{with body } b. \end{cases} \quad (2.1)$$

Therefore, for the clamp example of Figure 2.2, with the orientations of the joints chosen as shown in Table 2.1, the incidence matrix is

$$\Gamma = \begin{array}{cccccccc} \left[\begin{array}{cccccccc} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 5 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 6 \end{array} \right] & \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \\ \begin{array}{cccccccc} A & B & C & D & E & F & G & \end{array} \end{array}$$

The incidence matrix completely describes the topology of an oriented mechanical graph and, therefore, completely describes the kinematic architecture of a mechanism or multibody system. It also has the advantage that it can easily be communicated to and manipulated by a computer, and thus provides a working tool as well as a convenient symbolism. A few further definitions bring out some simple tests that can be performed on the incidence matrix.

A system is said to contain a *self-loop* whenever there is a joint for which both joint elements are contained in the same body; that is, when a joint connects a body to itself. Self-loops, such as that shown in Figure 2.5, are of no interest in kinematics because they permit no relative motion. They can be easily detected, however, because each valid joint is represented by a column of the incidence matrix having exactly two nonzero entries, one positive and one negative. Any self-loops should be eliminated from the incidence matrix (and from the count of joints) before further manipulations are performed.

The *degree* of a body is defined as the number of joints that are incident with that body. Thus, a nullary body has degree zero, a unary body has degree one, a binary body has degree two; a ternary body has degree three, and so on. The degree of a body can be found by summing the absolute values of the elements of the corresponding row of the incidence matrix. Notice that the degree of a body should

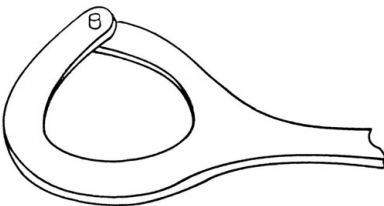


Figure 2.5. Self-loop.

count joint elements only when they are contained in joints. Thus, a “ternary” body, for example, in which one joint element is not joined to another body, is only of degree two. Equivalently, the degree of all bodies can be found as the diagonal elements of the product of the incidence matrix with its transpose:

$$d_b = (\Gamma \Gamma^t)(b, b), \quad b = 1, 2, \dots, \ell. \quad (2.2)$$

Another interesting property can be found by summing this equation over all bodies:

$$\sum_{b=1}^{\ell} d_b = \text{trace}(\Gamma \Gamma^t) = 2n. \quad (2.3)$$

This shows that because the number of joints in a mechanism or multibody system is an integer, the number of bodies of odd degree must be even.

2.3 Connectedness and Assemblies

An assembly (mechanical graph) of a multibody system is said to be *connected* if and only if its bodies *cannot* be grouped into subsets such that no joint is incident with bodies from more than one subset; that is, if and only if it is not possible to group the bodies and joints such that the incidence matrix becomes block diagonal:

$$\Gamma = \begin{bmatrix} \Gamma_{11} & 0 \\ 0 & \Gamma_{22} \end{bmatrix}. \quad (2.4)$$

In a connected mechanical graph or assembly, the degree of any node must not be less than unity. Mechanisms and multibody systems usually have connected mechanical graphs. If a mechanical graph is not connected, it does not represent a single assembly, and its unconnected assemblies may (although they need not) be analyzed as separate systems. An example of a situation where simultaneous analysis may be desired, or even required, is when the analysis is considering possible collision between bodies of unconnected assemblies. Physically unconnected assemblies may be logically connected by means of an open joint; see section 4.6.8.

Because there are $(\ell!n!)$ permutations of body and joint labels, Eq. (2.4) does not provide a convenient test for connectivity. Such a test is provided, however, by the following theorem, taken from graph theory [2]:

THEOREM 2.1. *The incidence matrix of a mechanical graph containing ℓ bodies and including a connected subassemblies has rank $(\ell - a)$.*

2.4 Kinematic Loops

A task of utmost importance in the topological analysis of a mechanism or multibody system is the determination of any closed kinematic loops, if they exist. All mechanism and multibody analysis techniques in one way or another depend on the recognition of such loops for formulation of the essential constraint equations. In any computer-aided analysis scheme involving mechanisms or multibody systems, either the program must be restricted to a specified kinematic architecture, or the user must

identify any kinematic loops, or an algorithm must be incorporated into the software that can identify closed loops from known information. In this section, an algorithm for discovering kinematic loops based on the incidence matrix is presented.

First, the terminology must be precisely defined:

Kinematic Path: a kinematic path is a sequence of joints (h_1, h_2, \dots, h_k) for which there exists a corresponding sequence of bodies (b_0, b_1, \dots, b_k) such that joint h_j is incident with bodies b_{j-1} and b_j for each joint of the sequence.

Kinematic Loop: a closed kinematic path (where $b_k = b_0$) is called a kinematic loop.

Also, a kinematic path or kinematic loop is said to be *oriented* if the joints in the sequence have designated orientations and a “direction” is chosen for the kinematic path or loop.

Because we have chosen a matrix representation to denote the incidence relationships of a given system, it is natural to consider a similar notation for expressing its kinematic loops. Therefore, we define the *oriented-loop matrix* L such that each row represents an oriented loop and columns again correspond to the joints of the system, taken in the same order as for the incidence matrix. We define the entry in row i , column h , of the oriented-loop matrix as follows:

$$L(i, h) = \begin{cases} \begin{array}{l} \text{if joint } h \text{ is contained in loop } i, \text{ and} \\ +1 \text{ if the orientation of loop } i \text{ and the} \\ \text{orientation of joint } h \text{ are the same;} \end{array} \\ \begin{array}{l} \text{if joint } h \text{ is contained in loop } i, \text{ and} \\ -1 \text{ if the orientation of loop } i \text{ and the} \\ \text{orientation of joint } h \text{ are opposite;} \end{array} \\ 0 \text{ if joint } h \text{ is not contained in loop } i. \end{cases} \quad h = 1, 2, \dots, n, \quad (2.5)$$

For the clamp example of the preceding section, therefore, an oriented-loop matrix containing all possible loops can be found by careful inspection of [Figure 2.2](#):

$$L = \begin{bmatrix} 1 & 0 & 1 & -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 1 & -1 & 1 & -1 & -1 \\ -1 & 0 & -1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 & 1 & 1 \end{bmatrix}. \quad (2.6)$$

However, the desired algorithm need not generate all possible loops, as shown here. It is clear, for instance, that the final three rows of matrix (2.6) represent loops that are equivalent to those of the first three rows, but which are oppositely oriented. Also, inspection verifies that the third row is equal to the sum of the first two rows. It is only necessary to generate enough loops to completely characterize the system. That is, we seek a complete set of *independent* kinematic loops.

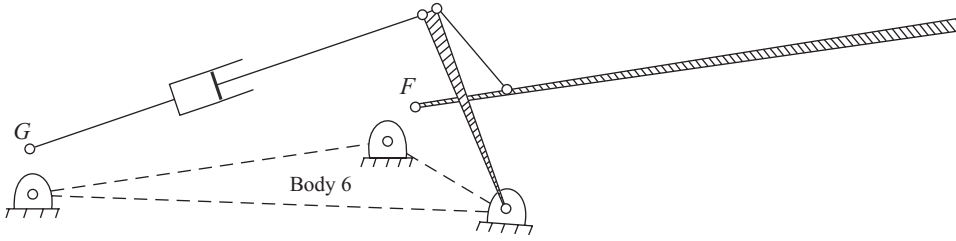


Figure 2.6. Kinematic tree for Figure 2.2.

Returning our attention to the incidence matrix, consideration of theorem 2.1 ensures that it can be partitioned as follows:

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix}, \tag{2.7}$$

where Γ_{11} is the largest possible nonsingular square submatrix. Theorem 2.1 tells us that Γ_{21} and Γ_{22} , together, form a rows where a is the number of unconnected subassemblies and that, therefore, Γ_{11} is of order $(\ell - a)$. Some reordering of the rows and columns may be required to achieve the largest possible nonsingular submatrix for Γ_{11} . However, this is assuredly possible because, according to theorem 2.1, the rank of Γ is $(\ell - a)$.

For the clamp example that we have been following, reordering is not required. The submatrices for this example are:

$$\begin{bmatrix} \Gamma_{11} \\ \Gamma_{21} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 & 0 \\ A & B & C & D & E \end{bmatrix} \quad \begin{bmatrix} \Gamma_{12} \\ \Gamma_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline -1 & 1 \\ F & G \end{bmatrix}.$$

It is interesting to note the significance of this partitioning. The submatrices Γ_{11} and Γ_{21} , taken together, also represent a mechanical graph, but not the same assembly as the original. This new system has the same bodies as the original, but has certain of its joints (F and G) disconnected as shown in Figure 2.6. It is clear in this example that a judicious selection of joints are disconnected so that the assembly remains connected but no longer contains any kinematic loops.

Is this a coincidence? No. Theorem 2.1 ensures that a mechanism or multibody system retains a connected assemblies because Γ_{11} is nonsingular and contains $(\ell - a)$ rows. Therefore, it is always possible to disconnect at least $[n - (\ell - a)]$ joints, although the choice of *which* joints may not be arbitrary, without dividing any of the assemblies. However, no more than this number of joints can be disconnected because if one more joint were disconnected, Γ_{11} would contain only $(\ell - a - 1)$ columns and, according to theorem 2.1, the system would then consist of $(a + 1)$ assemblies.

These partially disconnected assemblies have such interesting properties with regard to topological studies that they are given a special name. A mechanical graph that is connected but contains no loops is called a *tree* or *kinematic tree*. A kinematic tree of a mechanical graph, therefore, is a subgraph that contains all the nodes of the original mechanical graph but has no loops. One fundamental property of a kinematic tree is that it possesses a unique kinematic path between any two of its bodies. Therefore, as can be seen from Figure 2.6, when any of the disconnected joints is reconnected, a unique loop is formed, consisting of the reconnected joint and the joints of the unique path through the tree. It should be pointed out that most mechanisms contain at least one closed loop, but some multibody systems, such as serially connected robots, are characterized by tree-type kinematic architectures.

The set of $[n - (\ell - a)]$ joints that are disconnected to form the tree are called the *cut-set*. Each joint of the cut-set can be reconnected, one at a time, and a unique loop is formed for each. Also, because each of these contains at least one joint (the reconnected joint) that appears in no other loop, the loops found in this manner must be independent. Because joints are represented by arcs in a mechanical graph, the arcs corresponding to these reconnections (that form independent loops) are called chords of the mechanical graph. This means that the number of independent loops in a mechanical graph is equal to the number of chords, which is the same as the difference between the number of arcs of a mechanical graph and the number of arcs of a tree of the same graph. However, because $[n - (\ell - a)]$ joints are disconnected in forming the tree, this procedure shows that there are at least $NL = (n - \ell + a)$ independent kinematic loops in a mechanism or multibody system. The digraph NL is used to avoid duplicate symbolism in later chapters. This number, NL , has also found much significance in other applications of graph theory and is alternatively referred to as the *nullity*, the *connectivity*, the *cyclomatic number*, or the *first Betti number* of the graph [named after Italian mathematician, Enrico Betti (1823–92)]. Later in this section we will show that the rank of the oriented-loop matrix L cannot be greater than NL . Therefore, the set of kinematic loops generated by this process also provides a complete set of independent kinematic loops for the system.

Suppose that we are successful in finding the NL kinematic loops by the previously described procedure. Suppose also that we form the oriented-loop matrix L , including only these NL independent loops. If they are taken in the proper order and if the proper orientation is chosen for each, then the oriented-loop matrix can be partitioned as follows:

$$L = [L_1 \quad -I], \quad (2.8)$$

where $-I$ is the $(NL \times NL)$ negative-identity matrix corresponding to the reconnected joints of the cut-set and the rows of L_1 correspond to the unique, oriented paths through the tree. The arbitrary negative sign is chosen to produce a positive result in Eq. (2.10).

An extremely simple algorithm for generating the paths in L_1 is based on another important theorem from graph theory. Proof is omitted here; see, for example, [3, p. 92]:

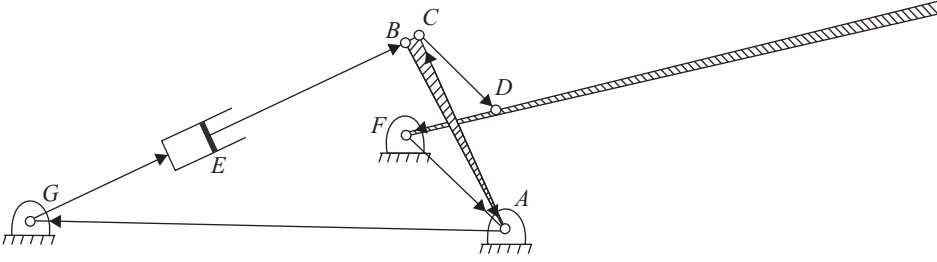


Figure 2.7. Oriented kinematic loops of Eq. (2.12) for Figure 2.2.

THEOREM 2.2. *With the columns of the incidence matrix Γ and the oriented-loop matrix L ordered consistently, these two matrices are orthogonal. That is*

$$\Gamma L^t = 0, \quad (2.9)$$

where L^t indicates the transpose of the matrix L .

Partitioning the matrices as indicated and applying this theorem, therefore,

$$\begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \begin{bmatrix} L_1^t \\ -I \end{bmatrix} = 0,$$

and on expanding the top row of submatrices,

$$\Gamma_{11} L_1^t - \Gamma_{12} = 0.$$

Because Γ_{11} was chosen to be nonsingular, this equation can be readily solved for the unknown portion of the oriented-loop matrix:

$$L_1 = (\Gamma_{11}^{-1} \Gamma_{12})^t. \quad (2.10)$$

Using this procedure on our clamp example, by Eq. (2.10) we find that

$$\Gamma_{11}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad \Gamma_{11}^{-1} \Gamma_{12} = L_1^t = \begin{bmatrix} 1 & -1 \\ 0 & -1 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (2.11)$$

and

$$L = \left[\begin{array}{ccccc|cc} 1 & 0 & 1 & -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & -1 \end{array} \right]. \quad (2.12)$$

$A \quad B \quad C \quad D \quad E \quad F \quad G$

Successfully finding the rank of Γ for this example to be five while the number of bodies is six verifies that the system consists of $a = 1$ assemblies. Checking the entries of Eq. (2.12) against the first two rows of Eq. (2.6), we verify that we have indeed found a valid, complete, and independent set of oriented kinematic loops for the system. These are shown in Figure 2.7.

Previously we made use of the fact that the rank of Γ cannot be greater than that which gives $NL = (n - \ell + a)$ in order to show that NL loops form a complete and independent set. This can now be shown directly from theorem 2.2. Because Γ and

L are orthogonal, we know that the rank of Γ plus the rank of L cannot exceed the dimension of their common vector space, the vector space of all joints in the system. This is Sylvester's law of nullity [3, p. 67], named after British mathematician, James Joseph Sylvester (1814–97):

$$\text{rank}(\Gamma) + \text{rank}(L) \leq n.$$

Theorem 2.1 tells us that the rank of Γ is $(\ell - a)$. Therefore,

$$\text{rank}(L) = n - \ell + a = NL. \quad (2.13)$$

In this section, we have seen that the incidence matrix Γ can be manipulated to produce a complete set of independent kinematic loops that can be used for analysis of the system. It should be noted from Eq. (2.10) that the matrix L_1 can be found directly by performing the row operations that are required to reduce the original incidence matrix Γ to row-echelon form:

$$\begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} I & L_1^t \\ 0 & 0 \end{bmatrix}.$$

The valid row operations allowed for reducing to this form are:

- (a) a row may be replaced by its negative,
- (b) a row may be added to or subtracted from another,
- (c) a row or column may be switched with another.

Switching rows is equivalent to relabeling bodies whereas switching columns is equivalent to relabeling joints. If these operations are used, however, the rearranged order of the original row and column labels must be recorded for later use when interpreting the body and joint labels in the result.

2.5 Kinematic Paths

Another topological matrix for which we will find many uses in the coming chapters is called the *kinematic path matrix*, and is labeled P . It is an $(\ell \times n)$ matrix in which each row shows the joints that appear in the *kinematic path* from the base body to the corresponding body. These paths become useful, for example, in calculating the motion of a chosen body by summing the relative motions of the joints along the path from the base body to the chosen body.

The entry in row b , column h of the oriented path matrix, with columns taken in the same order as in the incidence matrix, is defined as follows:

$$P(b, h) = \begin{cases} +1 & \text{if joint } h \text{ is contained in the path from the base} \\ & \text{body to body } b, \text{ and if the orientation of joint} \\ & h \text{ and the orientation of that path are the same;} \\ -1 & \text{if joint } h \text{ is contained in the path from the base} \\ & \text{body to body } b, \text{ and if the orientation of joint} \\ & h \text{ and the orientation of that path are opposite;} \\ 0 & \text{if joint } h \text{ is not contained in the path from the} \\ & \text{base body to body } b. \end{cases} \quad (2.14)$$

The oriented-path matrix can be found by a procedure very similar to that used in finding the oriented-loop matrix. The process is nicely shown by continuing the clamp example previously shown. If, for that example, we augment the original incidence matrix with an $(\ell \times \ell)$ identity matrix to its right, it appears as follows:

$$[\Gamma \mid I] = \left[\begin{array}{cccccc|cccc} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

Using the row operations previously enumerated, this augmented incidence matrix can be reduced to row-echelon form, exactly as was done to find the loop matrix, but working also on the additional augmented columns of the identity matrix. The final reduced form is:

$$\left[\begin{array}{c|c|c|c} I & L_1^t & \Gamma_{11}^{-1} & 0 \\ \hline 0 & 0 & 1 & I \end{array} \right] = \left[\begin{array}{cccc|cc|cccc|c} 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right].$$

The sixth and seventh columns of this result confirm the L_1^t portion of the loop matrix. The operations that correctly reduce the top-left portion of the matrix to the identity are the same operations that modify a portion of the augmented identity matrix to give Γ_{11}^{-1} ; this shows how the inverse was found for Eq. (2.11). The final row of unit entries in the later ℓ columns will now be explained.

If the upper-right $(\Gamma_{11}^{-1})^t$ portion of this result is transposed, it forms the upper-left portion of the path matrix:

$$P = \left[\begin{array}{c|c} (\Gamma_{11}^{-1})^t & 0 \\ \hline 0 & 0 \end{array} \right]. \quad (2.15)$$

$$P = \left[\begin{array}{cccccc|cc} 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix},$$

$$\begin{matrix} A & B & C & D & E & F & G \end{matrix}$$

where the last two columns of zeroes have been added for the cut-set joints F and G . We can see from row 2 of this matrix, for example, that the path from the base (body 6) to body 2 passes through joints A and B with positive orientation and through joint E with reversed orientation. This and the paths to other bodies can be confirmed by comparison with [Figure 2.2](#).

From this example it is clear that the path matrix can be found directly from the transpose of Γ_{11}^{-1} . This is true for this example and is also true in general. However, for mechanisms and multibody systems containing more than one assembly, another phenomenon arises. We will see this by following through another example. Without taking any notice of its unconnected, multi-assembly characteristics (that a computer would not know), we form the incidence matrix for the system shown in [Figure 2.8](#). This is

$$\Gamma = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix},$$

$A \quad B \quad C \quad D \quad E \quad F \quad G$

where, again, row and column labels have been added to indicate the body labels (numeric) and joint labels (literals).

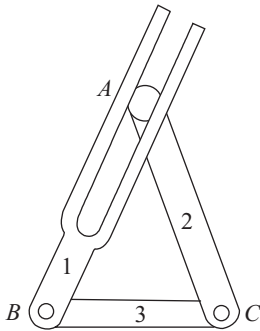
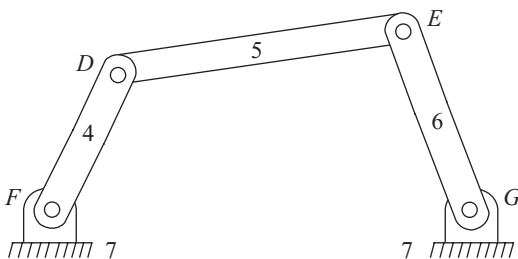


Figure 2.8.



and

$$P = \begin{array}{cccccc|c} A & B & D & E & F & G & C & \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 2/3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4/7 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 5/7 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 6/7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7/7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/3 \end{array} .$$

We note here that each path formed in this multi-assembly problem falls totally within an individual subassembly. The fact that the rows that originally correspond to bodies 3 and 7 are those that reduce to zeroes shows that bodies 3 and 7 may be chosen as the two reference bodies of the two separate subassemblies. The entries in the bottom-right quadrant of the row-reduced matrix show that the paths to bodies 1, 2, and 3 all have body 3 as the reference body for their subassembly, whereas the paths for bodies 4, 5, 6, and 7 all have body 7 as their reference body.

If desired, we can now rearrange the columns of the oriented-loop and path matrices to match the original order of the joint labels. This gives:

$$L = \begin{array}{cccccc|c} A & B & C & D & E & F & G & \\ \hline 0 & 0 & 0 & 1 & -1 & 1 & -1 & \\ 1 & 1 & -1 & 0 & 0 & 0 & 0 & \end{array}$$

and

$$P = \begin{array}{cccccc|c} A & B & C & D & E & F & G & \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 2/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4/7 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 5/7 \\ 0 & 0 & 0 & 1 & -1 & 1 & 0 & 6/7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7/7 \end{array} .$$

In order to find paths with respect to the body that is actually stationary, even for a single-assembly mechanism or multibody system, it is necessary to choose the original body labels such that the fixed body is represented by the final row of the incidence matrix. We note that this is done in the previous examples. If it is not done, then paths are found with respect to the highest body label of each subassembly.

We can see from the examples that the row-reduction operations are extremely simply performed, even by hand computation. All entries remain either 1, 0, or -1 throughout the procedure. This property, called the *unimodular property*, is more fully discussed in [1]. Also, on a computer, integer arithmetic operations can be used, avoiding any risk of rounding or truncation errors.

REFERENCES

1. I. Cederbaum, “Matrices all of whose elements and subdeterminants are 1, -1 , or 0,” *Journal of Mathematics and Physics*, 1958, vol. **36**, pp. 351–61.
2. F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.
3. S. Seshu and M. B. Reed, *Linear Graphs and Electrical Networks*, Addison-Wesley, Reading, MA, 1961.

PROBLEMS

2.1 Find the oriented-loop and -path matrices for the mechanism shown by **Figure P2.1** and the following incidence table P2.1:

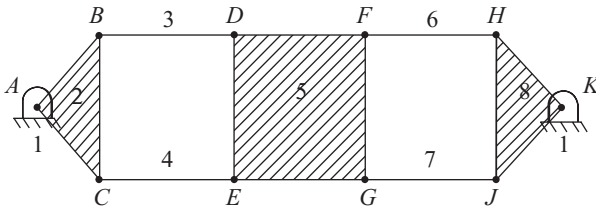


Figure P2.1

Table P2.1. Incidence table P2.1

Joint	from body	to body
A	1	2
B	2	3
C	2	4
D	3	5
E	4	5
F	5	6
G	5	7
H	6	8
J	7	8
K	8	1

2.2 **Figure P2.2b** shows a model of a Gough/Stewart platform, one type of parallel robotic system, pictured in **Figure P2.2a**.¹ In this type of system, there are two main bodies: the moving platform and the fixed base. The motion of the moving platform relative to the fixed base is controlled by six struts. Each of these struts is attached to the fixed base by a universal joint and to the moving platform by a ball joint. These struts are linear actuators and we assume that hydraulic cylinders are used. The schematic on the right in **Figure P2.2c** shows a typical strut. For modeling, the

¹ An animation can be viewed at http://en.wikipedia.org/wiki/File:Hexapod_general_Anim.gif.

actuator is shown as composed of two bodies, $PISTON_h$ and CYL_h , and these two bodies are connected by a prismatic joint P_h .

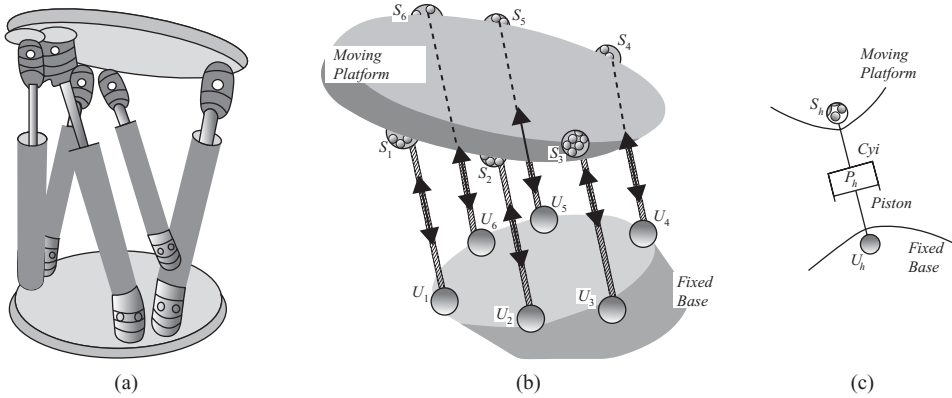


Figure P2.2a-c.

A directed graph of the Gough/Stewart platform, with the orientations of the joints selected arbitrarily, is shown in Figure P2.2d.

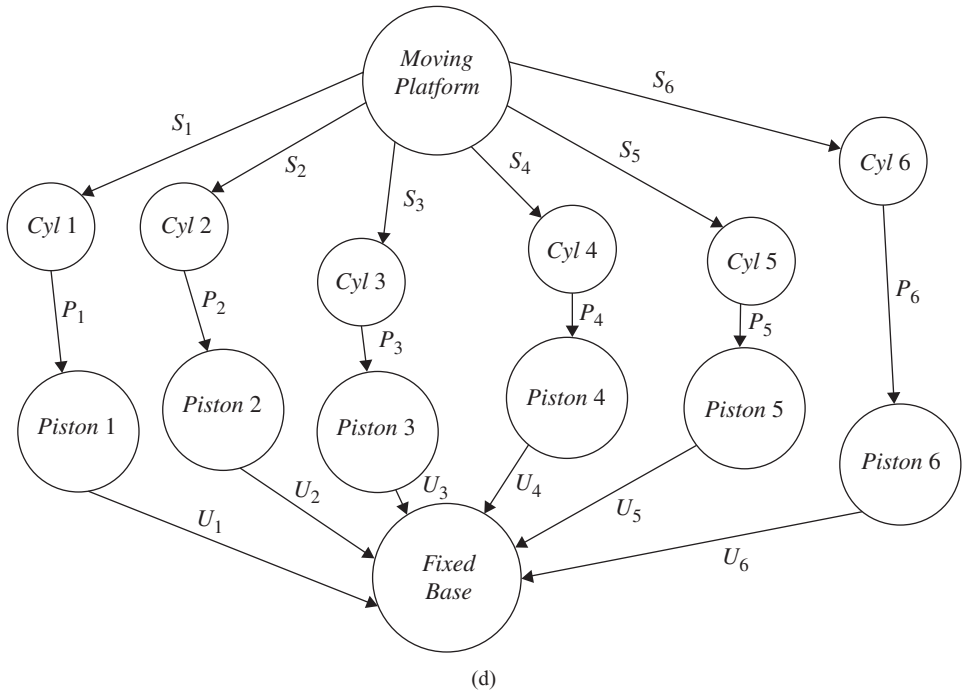


Figure P2.2d. Directed graph of the Gough/Stewart platform of Figure P2.2

The incidence matrix corresponding to the directed graph of Figure P2.2d is shown in the following table:

Table/Matrix

	S1	P1	U1	S2	P2	U2	S3	P3	U3	S4	P4	U4	S5	P5	U5	S6	P6	U6
Moving Platform	-1	0	0	-1	0	0	-1	0	0	-1	0	0	-1	0	0	-1	0	0
CYL1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PISTON1	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CYL2	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
PISTON2	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0
CYL3	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
PISTON3	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
CYL4	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0
PISTON4	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0
CYL5	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
PISTON5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0
CYL6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
PISTON6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
Fixed Base	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

Utilize the incidence matrix and appropriate software to compute the oriented loops for this system, and answer the following questions.

1. Which joints are cut to create trees?
2. How many independent loops are identified?

Write the oriented-loop matrix computed by your code arranged in the column sequence of the incidence matrix corresponding to the columns of Table P2.2.

2.3 Figure P2.3a shows a model of a multibody system that represents the stance posture of a walking bug robot. All joints are revolute and are designated J1, J2, . . . , J10. The various bodies are identified as B1 (ground), B2, and so forth and there are a total of nine (9) bodies.

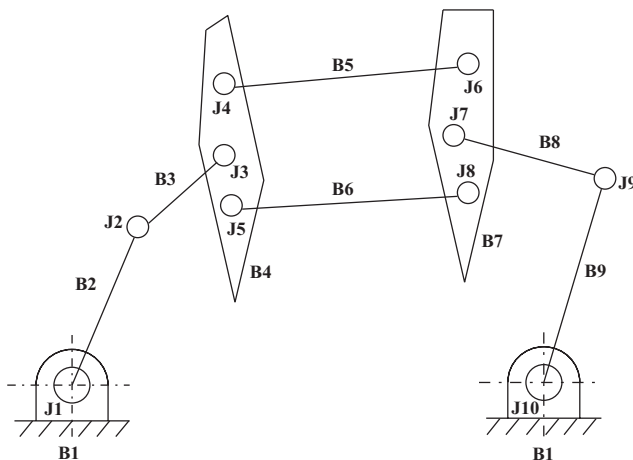


Figure P2.3a. Schematic diagram of a bug robot.

Select orientations for the various joints and, based on your choices, construct the incidence matrix for this model.

Utilize appropriate software to compute the independent loops and show the oriented-loop matrix, with columns in the sequence J1, J2, J3, . . . , J10.

Suppose that this bug robot raises one of its feet as shown in Figure P2.3b and that a new body (B10) is extended from B7 and a new revolute joint J11 is formed between B7 and B10. The original joint J1 that formerly connected B1 and B2 in the stance posture, now connects B2 to the new B10.

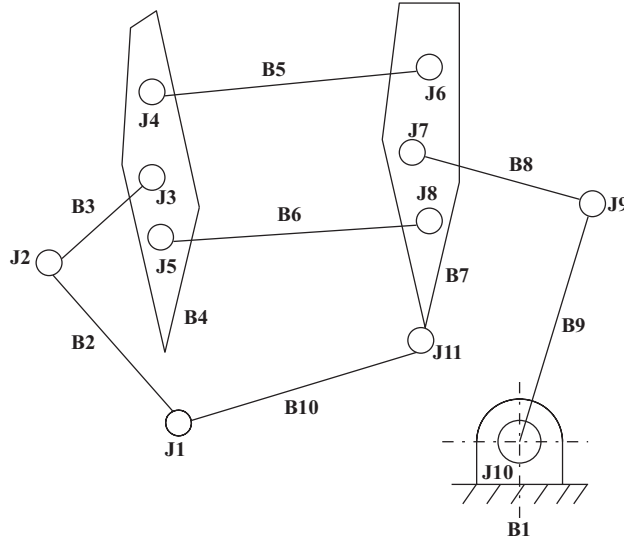


Figure P2.3b. Second posture of the bug robot.

Modify your incidence matrix and utilize appropriate software to compute the new oriented-loop matrix.

2.4 Figure P2.4a shows a model of a folding/unfolding deployable system.

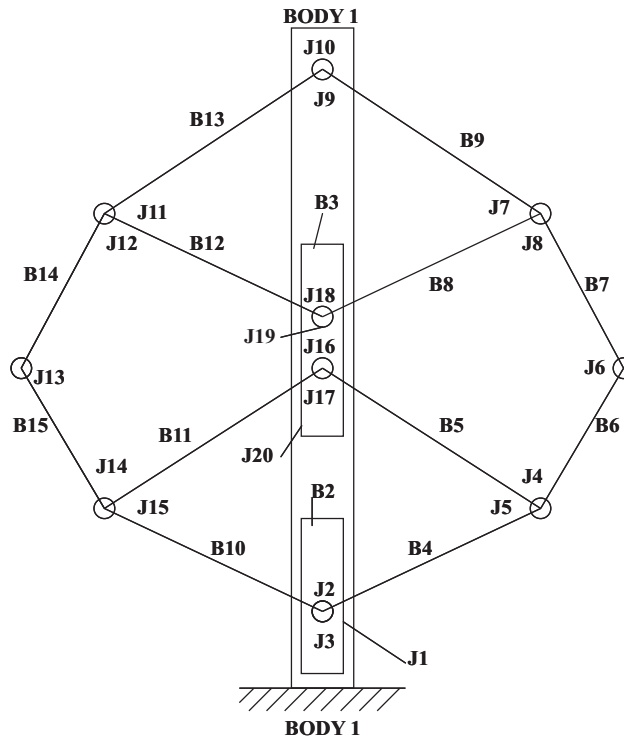


Figure P2.4a. Model of a deployable system.

In this system, there are fifteen (15) bodies and twenty (20) joints. Joints J1 and J20 are prismatic whereas all others are revolute joints. A directed graph of this system, with the orientations of the joints arbitrarily selected, is shown in Figure P2.4b.

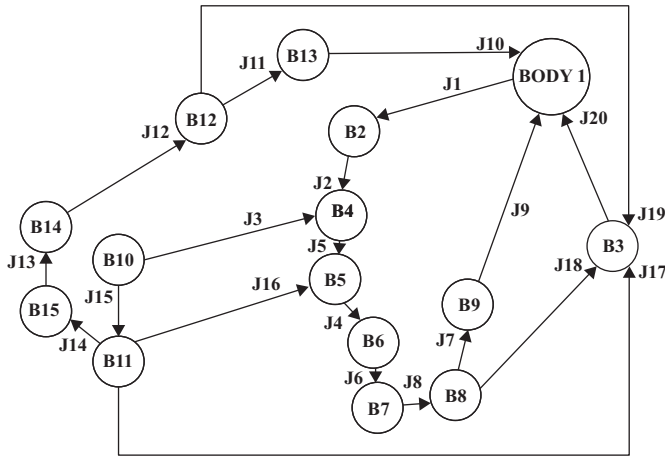


Figure P2.4b. Directed graph of the deployable system of Figure P2.4a.

The incidence matrix corresponding to the directed graph of Figure P2.4b is shown in the table of Figure P2.4c.

	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	
B1	-1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1
B2	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	-1
B4	0	1	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B5	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
B6	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B7	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
B8	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	-1	0	0
B9	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
B10	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
B11	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	-1	-1	0	0	0	0
B12	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	-1	0
B13	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0
B14	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0
B15	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0

Figure P2.4c. Incidence matrix for the deployable system of Figure P2.4b.

Utilize the incidence matrix and appropriate software to compute the oriented loops for this system, and answer the following questions.

1. Which joints are cut to create a tree?
2. How many independent loops are identified?
3. Write the oriented-loop matrix computed by your code arranged in the column sequence corresponding to the columns in the table of Figure P2.4c.

3.1 Introduction

Before formulating a numeric method for design analysis of mechanisms and multibody systems, let us first consider the essential characteristics of the problem being addressed. What are the chief difficulties encountered in the design analysis of a mechanism or multibody system? It is not the laws of mechanics as such that cause difficulty. It is the fact that, once a problem has been formulated, it is often too formidable algebraically to be easily solved. This complexity does not arise from static and dynamic force relationships, but from the kinematics – the changing geometry. The basic constraint equations that govern the motions within a machine or multibody system come from the fact that rigid bodies hold their respective joint elements in constant spatial relationships to one another. This type of constraint invariably leads to a set of highly nonlinear simultaneous algebraic equations.

Because the difficulties in an analytic approach to mechanism and multibody system analysis stem from the geometry, it is wise to choose a mathematical formulation suited to this type of problem. One such formulation is based on the use of matrices to represent the transformation equations between strategically located coordinate systems fixed in successive bodies. This approach has been developed into an extremely general and powerful technique for mechanism and multibody system analysis, and the next several chapters are devoted to its presentation. Before this can be presented effectively, however, we must become familiar with a number of basic operations that render matrix algebra so useful in performing coordinate transformations. The purpose of this chapter, therefore, is to develop this foundation.

3.2 Homogeneous Coordinates of a Point

The position of a point in three-dimensional Euclidean space is determined once three independent coordinates are given. We are all familiar, for example, with the Cartesian coordinates of a point as the projections of the point onto the three axes of a Cartesian reference frame, first introduced in [5] by the French philosopher

and mathematician, René Descartes (1596–1650). Once the point's position is known, unique x, y, z coordinates are determined. Also, conversely, if values corresponding to the x, y, z coordinates are known, the position of the point is uniquely determined.

However, it is also possible to define the position of a point in many other ways. We will recall, for example, using cylindric or spheric coordinates in problems for which these were more suitable. As a less common example, suppose we follow the German geometer, Karl Wilhelm Feuerbach (1800–34), and instead of a set of Cartesian reference axes, we define a reference tetrahedron whose fixed vertices are A, B, C , and D . Suppose also that we wish to determine the location of a point P that, for brevity, we assume to be inside of our tetrahedron of reference. Let us define symbols for the volumes of the four smaller tetrahedra determined by the point P and each of the four faces of the tetrahedron of reference. We identify these volumes by the symbols $V_A = \text{Vol}(PBCD)$, $V_B = \text{Vol}(PCDA)$, $V_C = \text{Vol}(PDAB)$, and $V_D = \text{Vol}(PABC)$. We could think of these four volumes as being four coordinates of the point P because once the tetrahedron of reference and the point are chosen, these coordinates are uniquely determined, and conversely. The fact that this method uses four coordinates to locate a point in three dimensions is explained by the fact that the four coordinates are not independent. They sum to the volume of the tetrahedron of reference.

These Feuerbach coordinates [7] are not the type we shall use in the coming chapters. They are presented here solely to illustrate the fact that coordinates can be defined in a variety of different ways. There is nothing sacrosanct about Cartesian, or cylindric, or spheric coordinates; in fact, in certain applications, a completely strange point-coordinate definition may add appreciably to the power, flexibility, and ease of use of the formulae developed. The Feuerbach example also illustrates that, although *at least* three coordinates are necessary to define the location of a point in three dimensions, a redundant set with more than three coordinates may be used as long as three and only three are independent.

Throughout the remainder of this text we will work with a definition of point coordinates that forms a special case of *homogeneous coordinates*, sometimes referred to as *barycentric* coordinates, introduced in [13] by the German mathematician and astronomer, August Ferdinand Möbius (1790–1868). With homogeneous coordinates, four coordinates are used to define the location of a point in three dimensions [12]. When these are written in column vector form, the position of a point using homogeneous coordinates is given by

$$r = \begin{bmatrix} r^1 \\ r^2 \\ r^3 \\ r^4 \end{bmatrix}. \quad (3.1)$$

The conditions that relate the four homogeneous coordinates to 3-D Cartesian coordinates of the same point are as follows:

$$r^x = \frac{r^1}{r^4}, \quad r^y = \frac{r^2}{r^4}, \quad r^z = \frac{r^3}{r^4}. \quad (3.2)$$

Thus, if we are given the homogeneous coordinates of a point, we can quite easily find the Cartesian coordinates of the same point by dividing all coordinates of the four-dimensional homogeneous position vector r by its last coordinate r^4

$$r = \begin{bmatrix} r^x \\ r^y \\ r^z \\ 1 \end{bmatrix}. \quad (3.3)$$

One reason that these coordinates are referred to as homogeneous is that any polynomial relating Cartesian position coordinates can be written in homogeneous form by recasting it in terms of homogeneous coordinates. For example, if the Cartesian coordinates are related by the quadratic equation

$$(r^x)^2 + 2r^x + 12(r^y)^2 - 2r^y r^z + 3r^z - 18 = 0,$$

this can be rewritten in homogeneous coordinates as

$$(r^1)^2 + 2r^1 r^4 + 12(r^2)^2 - 2r^2 r^3 + 3r^3 r^4 - 18(r^4)^2 = 0.$$

We note that because each term is now of the same degree, all of the homogeneous position coordinates r^i can be multiplied by an arbitrary nonzero scalar without affecting the validity of the equation. Thus, using homogeneous coordinates, we are free to rescale the position vector r by any nonzero multiplier at will. When the problem solution is completed, the Cartesian coordinates can be found by dividing r by its fourth element r^4 as shown in Eq. (3.2).

Another interesting characteristic of homogeneous point coordinates is that, with them, we can describe the location of a point infinitely distant from the origin of coordinates. For example, the homogeneous position vector

$$r = \begin{bmatrix} r \cdot i \\ r \cdot j \\ r \cdot k \\ 0 \end{bmatrix} = \begin{bmatrix} r \cos(\widehat{r, i}) \\ r \cos(\widehat{r, j}) \\ r \cos(\widehat{r, k}) \\ 0 \end{bmatrix} \quad (3.4)$$

describes a point whose Cartesian coordinates are all infinite. This point is located infinitely far from the origin along a line whose direction is determined by the vector r , made up of the first three components. Note that the homogeneous coordinates of such a point at infinity are all finite. Thus, a point at infinity is uniquely described in a manner that can be manipulated on a computer without numeric difficulty. This illustrates that homogeneous coordinates representing points at infinity can also be used to represent free vectors such as unit vectors; this is described more fully later in this chapter.

The study of homogeneous coordinates is rich in history and, since its beginnings, has been closely connected with the study of kinematics [4]. Although we shall not have need for all of the elegance in this book, you will find that this and the closely related fields of *affine* and *projective* geometry provide fascinating reading and add further insight into the work of the coming chapters.

Next, let us consider another characteristic of homogeneous coordinates and the underlying projective geometry, and that is the concept of *geometric duality*. When

using homogenous coordinates, the equation of a plane in three-dimensional space can be written as a linear equation of the form

$$Ax + By + Cz + Dw = 0$$

or, in matrix form, as

$$P^t R = R^t P = 0, \quad (3.5)$$

where the superscript t indicates the transpose of the superscripted matrix, and

$$P = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}, \quad R = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}. \quad (3.6)$$

Just as R represents the homogeneous coordinates of a point, P also has four coordinates and represents the homogeneous coordinates of a plane. The quantities A, B, C, D are called plane coordinates of P and they are homogenous in the same way that x, y, z, w are homogenous – namely, multiplying them by any nonzero scalar does not change the plane that they represent.

Equation (3.5) illustrates the duality between points and planes in three-dimensional projective space. Put into words, this equation represents a plane P passing through point R when one highlights the homogeneous plane coordinates or, alternatively, it represents a point R lying on plane P when one emphasizes the homogeneous point coordinates. In three articles in his own journal [9], the French mathematician, Joseph Diaz Gergonne (1771–1859), states the general principle of *geometric duality* that, in the projective space of three dimensions, every theorem connecting points and planes corresponds to another theorem in which the terms points and planes are interchanged, provided that no metric relations are involved. For example: three points determine a unique plane, and three planes determine a unique point. Similarly: two points determine a unique line, and two planes determine a unique line. In three dimensions, a point is the geometric dual of a plane. A line is a self-dual; that is, a line is the dual of a line.

3.3 Line Coordinates and Plücker Vectors

Another situation where more than the minimal number of coordinates is useful to describe the location of a geometric entity in space is the use of Plücker line coordinates. Line geometry is a branch of algebraic geometry that is closely related to homogeneous coordinates and to screw coordinates (to be discussed later) and was first elucidated in [15] by the German mathematician and physicist, Julius Plücker (1801–68).

A homogeneous representation of a straight line in three dimensions is provided by a 6-D vector of *line coordinates* or by a pair of 3-D *Plücker vectors*. The Plücker vectors of a line are defined by two vectors with Cartesian coordinates

$$\boldsymbol{\Omega} = \mathbf{r} \times \hat{\boldsymbol{\Omega}} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \hat{\boldsymbol{\Omega}} = \begin{bmatrix} d \\ e \\ f \end{bmatrix}, \quad (3.7)$$

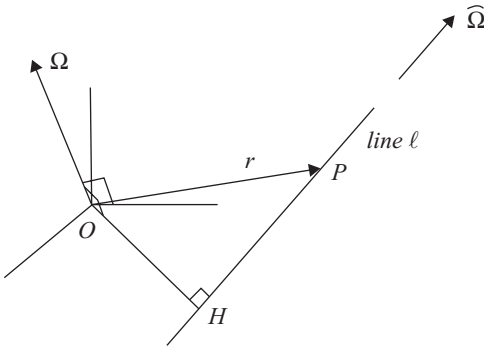


Figure 3.1. Geometry of Plücker vectors.

where $\hat{\Omega}$ is a vector directed along the line and r is the Cartesian position vector of a point on the line as shown in [Figure 3.1](#).

The vector $\hat{\Omega}$ is often normalized to form a unit vector but need not be so. The vector Ω is the moment that the $\hat{\Omega}$ vector makes about the origin of coordinates. Thus, one Plücker vector, $\hat{\Omega}$, defines the orientation of the line whereas the other, Ω , uniquely defines its location in space and distinguishes this particular line from all other lines parallel to it. The vector Ω is perpendicular to the plane OHP ([Figure 3.1](#)) and, when the vector $\hat{\Omega}$ is a unit vector, the magnitude of the moment Ω is the distance from the origin O to the line ℓ , namely, the distance OH . It is clear from vector algebra that Plücker vectors satisfy the Plücker identity:

$$\hat{\Omega} \cdot \Omega = \hat{\Omega}^t \Omega = \Omega^t \hat{\Omega} = 0. \quad (3.8)$$

It should be pointed out that Plücker vectors represent an oriented line in space because the vector $\hat{\Omega}$ describes the direction of the line, which includes both its orientation and its sense.

The elements of Plücker vectors of a line are called Plücker coordinates and can be found from the homogeneous coordinates of two points located on the line, or from the homogeneous coordinates of two planes that intersect in the line, as follows:

$$\begin{aligned} a &= y_1 z_2 - z_1 y_2, & a &= B_1 C_2 - C_1 B_2, \\ b &= z_1 x_2 - x_1 z_2, & b &= C_1 A_2 - A_1 C_2, \\ c &= x_1 y_2 - y_1 x_2, & c &= A_1 B_2 - B_1 A_2, \\ d &= x_1 w_2 - w_1 x_2, & d &= A_1 D_2 - D_1 A_2, \\ e &= y_1 w_2 - w_1 y_2, & e &= B_1 D_2 - D_1 B_2, \\ f &= z_1 w_2 - w_1 z_2, & f &= C_1 D_2 - D_1 C_2. \end{aligned} \quad (3.9)$$

Here, x_i, y_i, z_i, w_i ($i = 1, 2$) are homogeneous coordinates of two points on the line and A_i, B_i, C_i, D_i ($i = 1, 2$) are their geometric duals, the plane coordinates of two planes intersecting in the line.

Conversely, the homogenous coordinates of points on a line can be found from the Plücker coordinates of the line in terms of a parameter λ as follows:

$$\begin{aligned} x &= ec - fb + \lambda d(d^2 + e^2 + f^2), \\ y &= fa - dc + \lambda e(d^2 + e^2 + f^2), \\ z &= db - ea + \lambda f(d^2 + e^2 + f^2), \\ w &= d^2 + e^2 + f^2. \end{aligned} \quad (3.10)$$

Different values of the parameter λ correspond to different points along the line. Equations (3.10) are scalar forms of the vector equation

$$\mathbf{r} = \widehat{\Omega}^{-2}(\widehat{\Omega} \times \boldsymbol{\Omega}) + \lambda \widehat{\Omega}. \quad (3.11)$$

This is derived from the geometry of Plücker vectors (see Figure 3.1) considering the fact that $\mathbf{r} \times \widehat{\Omega} = \boldsymbol{\Omega}$, and $\widehat{\Omega} \cdot \boldsymbol{\Omega} = 0$, and using linear algebra.

This text presents matrix methods based on the use of (4×4) transformation matrices. Therefore, it is desirable to represent Plücker vectors in a (4×4) matrix format. Noting that the second of the two Plücker vectors is a free vector because it represents the direction of a line, we can form a (3×3) skew-symmetric matrix by arranging the components of this vector as follows:

$$\widetilde{\widehat{\Omega}} = \begin{bmatrix} 0 & -\widehat{\Omega}^z & \widehat{\Omega}^y \\ \widehat{\Omega}^z & 0 & -\widehat{\Omega}^x \\ -\widehat{\Omega}^y & \widehat{\Omega}^x & 0 \end{bmatrix}. \quad (3.12)$$

The first Plücker vector $\boldsymbol{\Omega}$ can then be written in vector form as $\boldsymbol{\Omega} = \mathbf{r} \times \widehat{\Omega}$, or in (3×1) matrix form as

$$\boldsymbol{\Omega} = -\widetilde{\widehat{\Omega}}\mathbf{r}. \quad (3.13)$$

The matrix form of the two Plücker vectors of a line can now be assembled into a single (4×4) matrix as follows:

$$\begin{bmatrix} 0 & -\widehat{\Omega}^z & \widehat{\Omega}^y & \Omega^x \\ \widehat{\Omega}^z & 0 & -\widehat{\Omega}^x & \Omega^y \\ -\widehat{\Omega}^y & \widehat{\Omega}^x & 0 & \Omega^z \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \widehat{\Omega} & \boldsymbol{\Omega} \\ 0 & 0 \end{bmatrix}. \quad (3.14)$$

3.4 Three-dimensional Orientation

One of the most fundamental problems of kinematics is that of describing three-dimensional angular direction or, more precisely, of specifying the *orientation* of a rigid body with respect to a known frame of reference. The word *attitude* can also be used for orientation. Clearly, the orientation or attitude of a rigid body is completely determined once a set of coordinate axes fixed to the body has been located relative to the known reference frame. In this section we assume that the origins of the two

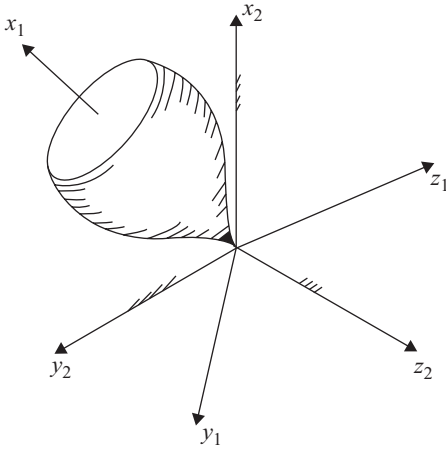


Figure 3.2. Orientation of a spinning top.

coordinate systems are coincident so that we may concentrate solely on their relative orientation.

A typical problem of this type is that of describing the orientation of a spinning top such as that shown in [Figure 3.2](#). The top carries a right-hand Cartesian coordinate system $x_1y_1z_1$ that must be located relative to a fixed coordinate system $x_2y_2z_2$. Notice that, as decided in Chapter 2, the higher label is used for the stationary body.

One fruitful way of describing the relative angular relationship between two coordinate systems is through what are called *direction cosines*. If \mathbf{r} is the position vector of some arbitrary point on the top, and if $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ represent unit vectors along the $x_1y_1z_1$ and $x_2y_2z_2$ axes, respectively, then the component of \mathbf{r} along the x_2 axis can be expressed as follows:

$$\begin{aligned} r^{x_2} &= \mathbf{i} \cdot \mathbf{r} = (\mathbf{i} \cdot \mathbf{i}')r^{x_1} + (\mathbf{i} \cdot \mathbf{j}')r^{y_1} + (\mathbf{i} \cdot \mathbf{k}')r^{z_1} \\ &= \cos(\widehat{\mathbf{i}, \mathbf{i}'})r^{x_1} + \cos(\widehat{\mathbf{i}, \mathbf{j}'})r^{y_1} + \cos(\widehat{\mathbf{i}, \mathbf{k}'})r^{z_1}, \end{aligned}$$

where $\cos(\widehat{\mathbf{i}, \mathbf{j}'})$, for example, denotes the cosine of the angle between the \mathbf{i} and \mathbf{j}' unit vectors.

Taking dot products with \mathbf{j} and \mathbf{k} to express equations for the r^{y_2} and r^{z_2} components of \mathbf{r} in the same format, and grouping the results into a single matrix equation, we obtain

$$\begin{bmatrix} r^{x_2} \\ r^{y_2} \\ r^{z_2} \end{bmatrix} = \begin{bmatrix} \cos(\widehat{\mathbf{i}, \mathbf{i}'}) & \cos(\widehat{\mathbf{i}, \mathbf{j}'}) & \cos(\widehat{\mathbf{i}, \mathbf{k}'}) \\ \cos(\widehat{\mathbf{j}, \mathbf{i}'}) & \cos(\widehat{\mathbf{j}, \mathbf{j}'}) & \cos(\widehat{\mathbf{j}, \mathbf{k}'}) \\ \cos(\widehat{\mathbf{k}, \mathbf{i}'}) & \cos(\widehat{\mathbf{k}, \mathbf{j}'}) & \cos(\widehat{\mathbf{k}, \mathbf{k}'}) \end{bmatrix} \begin{bmatrix} r^{x_1} \\ r^{y_1} \\ r^{z_1} \end{bmatrix},$$

or

$$\mathbf{r}_2 = \Theta_{21}\mathbf{r}_1. \quad (3.15)$$

Such a (3×3) matrix is called a *rotation matrix* and is useful for transforming any free vector from one coordinate system to another. The nine *direction cosines* in the matrix Θ_{21} completely describe the instantaneous orientation of the x_1, y_1, z_1 coordinate system with respect to the x_2, y_2, z_2 coordinate system.

We know, however, that the nine direction cosines cannot be independent because a rigid body has only three degrees of freedom in spatial rotation. There must be six additional equations relating these nine direction cosines. These may be found by considering the fact that the magnitude of the vector \mathbf{r} must be identical in both coordinate systems,

$$(r^{x_2})^2 + (r^{y_2})^2 + (r^{z_2})^2 = (r^{x_1})^2 + (r^{y_1})^2 + (r^{z_1})^2.$$

Writing this in matrix form, we see that

$$r_2^t r_2 = r_1^t r_1,$$

where the superscript t indicates the transpose of the superscripted matrix. Now, substituting Eq. (3.15) into this expression yields

$$r_1^t \Theta_{21}^t \Theta_{21} r_1 = r_1^t r_1,$$

which can only be satisfied for an arbitrary choice of the point r_1 if

$$\Theta_{21}^t \Theta_{21} = I. \quad (3.16)$$

This, in turn, shows that

$$\Theta_{21}^t = \Theta_{21}^{-1}. \quad (3.17)$$

A matrix having this interesting property is said to be an *orthogonal* matrix, and Eq. (3.16) is one expression for what are called the *orthogonality conditions*. However, because we know that the orientation of a rigid body or a coordinate system in three-dimensional space depends on only three parameters, this matrix equation includes six conditions relating the nine direction cosines of Eq. (3.15). In other words, the rotation matrix is not a minimal representation of orientation.

In two dimensions the rotation matrix reduces to a (2×2) matrix and the positions of points in a rotated frame with respect to a fixed frame with coincident origin are related by

$$\begin{bmatrix} r^{x_2} \\ r^{y_2} \end{bmatrix} = \begin{bmatrix} \cos(\widehat{\mathbf{i}, \mathbf{i}'} & \cos(\widehat{\mathbf{i}, \mathbf{j}'} \\ \cos(\widehat{\mathbf{j}, \mathbf{i}'} & \cos(\widehat{\mathbf{j}, \mathbf{j}'} \end{bmatrix} \begin{bmatrix} r^{x_1} \\ r^{y_1} \end{bmatrix}. \quad (3.18)$$

The direction cosines are again not independent and the orientation of a rigid body or a coordinate system in two dimensions can be described by a minimal representation of just a single angle, say ϑ , of the x_1 axis with respect to the x_2 axis as shown in [Figure 3.3](#).

The description of relative orientation in two dimensions in terms of a single angle ϑ is a minimal representation and is very convenient for use in two-dimensional applications. However, the question now becomes: what minimal representation can be used for orientation in three dimensions? For example, what minimal representation may be used as coordinate(s) to describe the attitude of the spinning top? Initially, it may seem desirable to find a set of three independent parameters, hopefully angles like the two-dimensional case, for describing such a three-dimensional orientation.

The great Swiss mathematician and physicist, Leonhard Euler (1707–83) is usually credited with being the first to show that a set of three angles is sufficient to

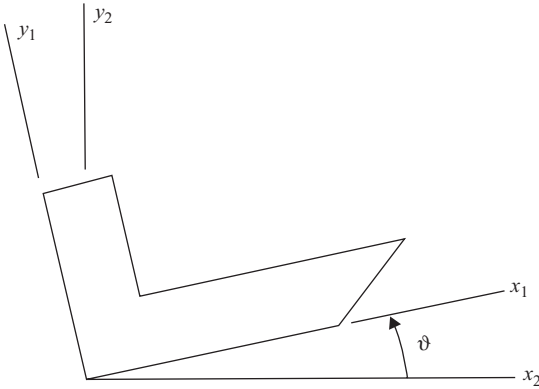


Figure 3.3. Minimal representation of orientation of a rigid body in two dimensions.

specify the relative orientation of two Cartesian coordinate systems in three dimensions. Although he used various choices in his writings, any such set of three angles now bears his name; they are called *Euler angles*. In order to illustrate them, the two coordinate systems used to describe the orientation of the spinning top are shown again in [Figure 3.4](#), but now without the top. One form of a set of three Euler angles: ξ , ϑ , and φ (see Euler, [6]) are shown in the figure. Various other definitions of Euler angles are shown in Goldstein, [10]. Note in [Figure 3.4](#) that the planes x_1y_1 and x_2y_2 intersect in a line ℓ that is referred to as the *line of nodes*.

The three Euler angles shown here are defined as follows: the angle ξ is measured from the positive x_2 axis to the line of nodes ℓ about the positive z_2 axis in the plane x_2y_2 . The angle ϑ is measured from the positive z_2 axis to the positive z_1 axis about the line of nodes ℓ in the plane z_2z_1 . The angle φ is measured from the line of nodes ℓ to the positive x_1 axis about the positive z_1 axis in the plane x_1y_1 .

The right-hand rule is used to define the positive sense of each angle; thus, counterclockwise angles are positive. The convention, used throughout this text, that counterclockwise angles are positive is not arbitrary. It is required by such conventions as $\mathbf{k} = \mathbf{i} \times \mathbf{j}$ and the assumption that we wish to use right-hand Cartesian coordinate systems, with their y axes taken counterclockwise from their x axes as seen from their positive z axes.

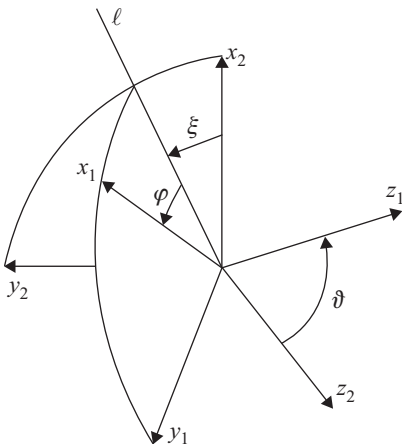


Figure 3.4. Euler angles ξ , ϑ , and φ of $x_1y_1z_1$ with respect to $x_2y_2z_2$.

Using elementary geometry to relate the orientations of the two coordinate systems, we can show that the expanded form of Eq. (3.15) in terms of these three Euler angles becomes:

$$\begin{bmatrix} r^{x_2} \\ r^{y_2} \\ r^{z_2} \end{bmatrix} = \begin{bmatrix} -\sin\xi \cos\vartheta \sin\varphi + \cos\xi \cos\varphi & -\sin\xi \cos\vartheta \cos\varphi - \cos\xi \sin\varphi & \sin\xi \sin\vartheta \\ \cos\xi \cos\vartheta \sin\varphi + \sin\xi \cos\varphi & \cos\xi \cos\vartheta \cos\varphi - \sin\xi \sin\varphi & -\cos\xi \sin\vartheta \\ \sin\vartheta \sin\varphi & \sin\vartheta \cos\varphi & \cos\vartheta \end{bmatrix} \begin{bmatrix} r^{x_1} \\ r^{y_1} \\ r^{z_1} \end{bmatrix}. \quad (3.19)$$

We will further discuss the representation of orientation of a rigid body in subsequent sections of this chapter. However, let us first consider the coordinates of points in coordinate systems that do not share coincident origins.

3.5 Transformation of Coordinates

In section 3.4, we demonstrated that the position vector of a point in a Cartesian coordinate system can be transformed into an equivalent position vector in another Cartesian coordinate system by multiplying by the rotation matrix as was shown in Eq. (3.15). That transformation, however, was restricted to the case where the two coordinate systems share coincident origins. In this section we will remove this restriction and develop a general transformation of coordinates in three dimensions.

Consider the arbitrarily chosen point r of Figure 3.5. The two Cartesian position vectors \mathbf{r}_c in coordinate system c and \mathbf{r}_b in coordinate system b are related by the vector equation

$$\mathbf{r}_c = \mathbf{r}_b + \mathbf{r}_{O_b}. \quad (3.20)$$

If we write each of these vectors in terms of its components along its own coordinate axes, then, referring to Figure 3.6, we have

$$\begin{aligned} \mathbf{r}_c &= r^x_c \mathbf{i} + r^y_c \mathbf{j} + r^z_c \mathbf{k}, \\ \mathbf{r}_b &= r^x_b \mathbf{i}' + r^y_b \mathbf{j}' + r^z_b \mathbf{k}', \\ \mathbf{r}_{O_b} &= r^x_{O_b} \mathbf{i} + r^y_{O_b} \mathbf{j} + r^z_{O_b} \mathbf{k}. \end{aligned} \quad (3.21)$$

Adding these according to Eq. (3.20) we obtain

$$r^x_c \mathbf{i} + r^y_c \mathbf{j} + r^z_c \mathbf{k} = r^x_b \mathbf{i}' + r^y_b \mathbf{j}' + r^z_b \mathbf{k}' + r^x_{O_b} \mathbf{i} + r^y_{O_b} \mathbf{j} + r^z_{O_b} \mathbf{k}. \quad (3.22)$$

Now taking the dot product of this equation with each of the unit vectors \mathbf{i} , \mathbf{j} , and \mathbf{k} in turn, we obtain three scalar equations:

$$\begin{aligned} r^x_c &= r^x_b \cos(\widehat{\mathbf{i}, \mathbf{i}'}) + r^y_b \cos(\widehat{\mathbf{i}, \mathbf{j}'}) + r^z_b \cos(\widehat{\mathbf{i}, \mathbf{k}'}) + r^x_{O_b}, \\ r^y_c &= r^x_b \cos(\widehat{\mathbf{j}, \mathbf{i}'}) + r^y_b \cos(\widehat{\mathbf{j}, \mathbf{j}'}) + r^z_b \cos(\widehat{\mathbf{j}, \mathbf{k}'}) + r^y_{O_b}, \\ r^z_c &= r^x_b \cos(\widehat{\mathbf{k}, \mathbf{i}'}) + r^y_b \cos(\widehat{\mathbf{k}, \mathbf{j}'}) + r^z_b \cos(\widehat{\mathbf{k}, \mathbf{k}'}) + r^z_{O_b}. \end{aligned}$$

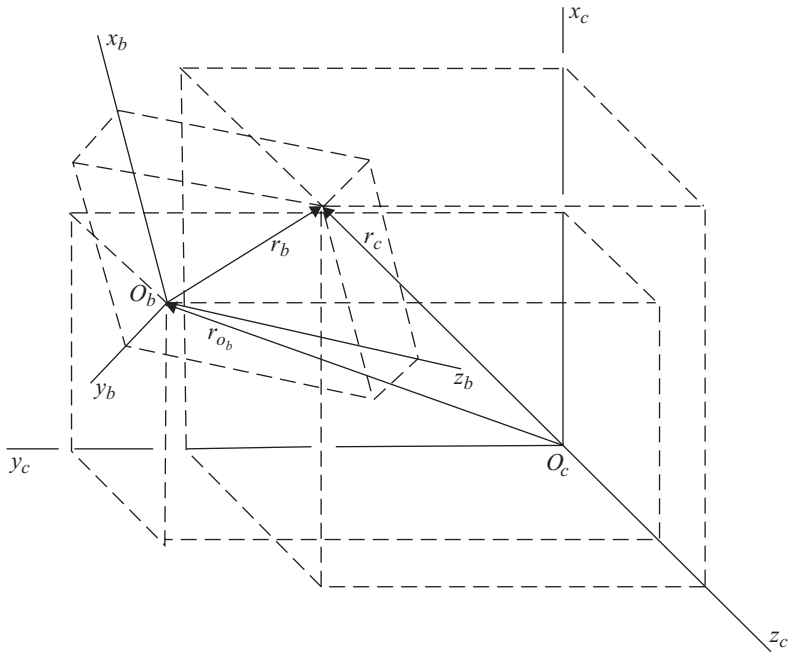


Figure 3.5. General 3-D coordinate transformation.

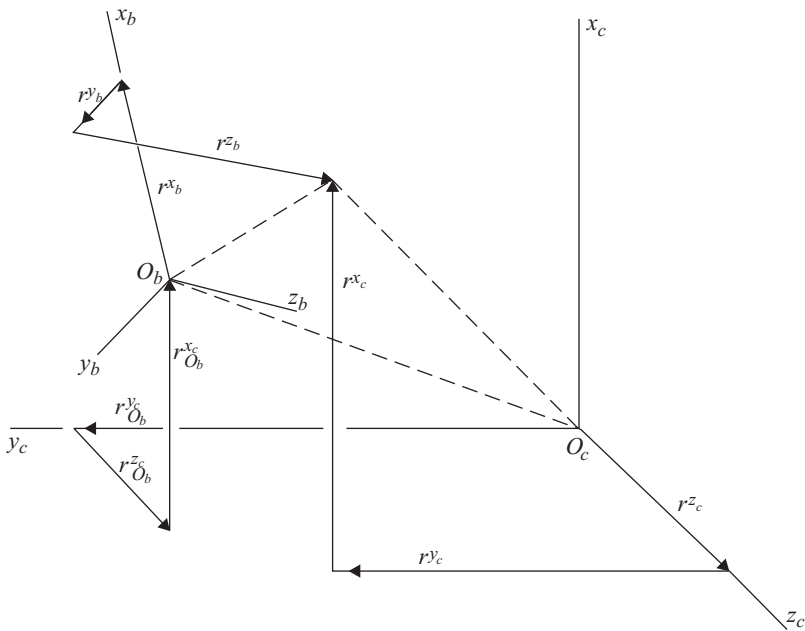


Figure 3.6. General 3-D coordinate transformation.

In terms of homogeneous position coordinates, these can be written

$$\begin{bmatrix} r^x \\ r^y \\ r^z \\ 1 \end{bmatrix}_c = \begin{bmatrix} \cos(\widehat{i,i}') & \cos(\widehat{i,j}') & \cos(\widehat{i,k}') & r_{O_b}^{x_c} \\ \cos(\widehat{j,i}') & \cos(\widehat{j,j}') & \cos(\widehat{j,k}') & r_{O_b}^{y_c} \\ \cos(\widehat{k,i}') & \cos(\widehat{k,j}') & \cos(\widehat{k,k}') & r_{O_b}^{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r^x \\ r^y \\ r^z \\ 1 \end{bmatrix}_b, \quad (3.23)$$

where the final row of the new (4×4) transformation matrix simply expresses the identity, $1 = 1$. The simplicity of the bottom row of this matrix results from our assumption of rigid body transformations; this is not the case in the more general fields of affine or projective transformations. It can also be viewed as a constraint that a unit distance in the c coordinate system is equal to a unit distance in the b coordinate system.

We can extend the utility of the aforementioned equation by noting that a point at infinity [in the sense of Eq. (3.4)] is also properly transformed by the same (4×4) matrix, no matter what its orientation. The reader may wish to verify that, after transformation, a point at infinity remains at infinity. Note how this results from the special rigid body form of the last row of the matrix. Therefore, we can say in complete generality that the transformation between *any* two three-dimensional Cartesian coordinate systems labeled c and b is expressed by the equation

$$r_c = T_{cb} r_b, \quad (3.24)$$

where r_c and r_b represent the homogeneous position vectors of a point expressed in the c and b coordinate systems, respectively, and T_{cb} is a (4×4) transformation matrix of the form

$$T_{cb} = \begin{bmatrix} \cos(\widehat{i,i}') & \cos(\widehat{i,j}') & \cos(\widehat{i,k}') & r_{O_b}^{x_c} \\ \cos(\widehat{j,i}') & \cos(\widehat{j,j}') & \cos(\widehat{j,k}') & r_{O_b}^{y_c} \\ \cos(\widehat{k,i}') & \cos(\widehat{k,j}') & \cos(\widehat{k,k}') & r_{O_b}^{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.25)$$

Note that the homogeneous transformation matrix T_{cb} completely describes the location of the origin and the orientation of a coordinate system b with respect to another coordinate system c and depends on six independent parameters. In other words, the location and orientation of a coordinate system (or a rigid body) in three dimensions can be characterized by six degrees of freedom. Note that the orientation portion has three degrees of freedom and the portion dealing with the location of the origin of the moving coordinate system contains the other three degrees of freedom.

We should also notice that Eq. (3.24) not only describes the relationship between the coordinates of a point measured with respect to two different coordinate systems but also the coordinate transformation of a unit vector as measured in the two coordinate systems. A unit vector describes a direction (an orientation and sense) in space and, therefore, can be viewed as a vector toward a particular point at infinity. Its three direction cosines can be used as the first three homogeneous coordinates of a point at infinity with a fourth coordinate of zero. Consider, for example, two coordinate systems as shown in [Figure 3.7](#).

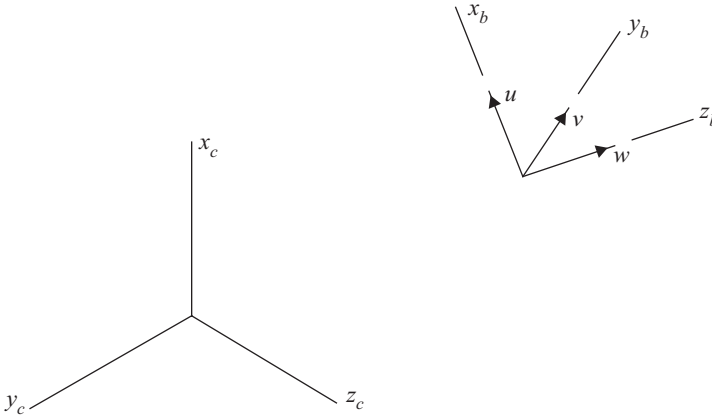


Figure 3.7. Coordinate transformation for unit vectors.

Consider the three unit vectors \mathbf{u} , \mathbf{v} , and \mathbf{w} as shown; these three unit vectors in coordinate systems $x_b y_b z_b$ and $x_c y_c z_c$ can be expressed, respectively, as

$$\mathbf{u}_b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}_b = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{w}_b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \text{ and } \mathbf{u}_c = \begin{bmatrix} u^{x_c} \\ u^{y_c} \\ u^{z_c} \end{bmatrix}, \mathbf{v}_c = \begin{bmatrix} v^{x_c} \\ v^{y_c} \\ v^{z_c} \end{bmatrix}, \mathbf{w}_c = \begin{bmatrix} w^{x_c} \\ w^{y_c} \\ w^{z_c} \end{bmatrix}.$$

The homogeneous coordinates of a point at infinity corresponding to the unit vector \mathbf{u} can be written in the $x_b y_b z_b$ and $x_c y_c z_c$ coordinate systems, respectively, as $r_b = [1, 0, 0, 0]^t$, and $r_c = [u^{x_c}, u^{y_c}, u^{z_c}, 0]^t$.

Now, writing Eq. (3.24) with the objective of trying to determine the elements of the homogeneous transformation matrix T_{cb} , we get

$$\begin{bmatrix} u^{x_c} \\ u^{y_c} \\ u^{z_c} \\ 0 \end{bmatrix} = \begin{bmatrix} \Theta(1,1) & \Theta(1,2) & \Theta(1,3) & r_{O_b}^{x_c} \\ \Theta(2,1) & \Theta(2,2) & \Theta(2,3) & r_{O_b}^{y_c} \\ \Theta(3,1) & \Theta(3,2) & \Theta(3,3) & r_{O_b}^{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \Theta(1,1) \\ \Theta(2,1) \\ \Theta(3,1) \\ 0 \end{bmatrix}.$$

Equating elements of both sides we obtain: $\Theta(1,1) = u^{x_c}$, $\Theta(2,1) = u^{y_c}$, and $\Theta(3,1) = u^{z_c}$. In a similar fashion, if we apply Eq. (3.24) to unit vectors \mathbf{v} and \mathbf{w} , we obtain $\Theta(1,2) = v^{x_c}$, $\Theta(2,2) = v^{y_c}$, $\Theta(3,2) = v^{z_c}$, and $\Theta(1,3) = w^{x_c}$, $\Theta(2,3) = w^{y_c}$, $\Theta(3,3) = w^{z_c}$. The corresponding homogeneous transformation matrix becomes:

$$T_{cb} = \begin{bmatrix} u^{x_c} & v^{x_c} & w^{x_c} & r_{O_b}^{x_c} \\ u^{y_c} & v^{y_c} & w^{y_c} & r_{O_b}^{y_c} \\ u^{z_c} & v^{z_c} & w^{z_c} & r_{O_b}^{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.26)$$

It should be pointed out that the resulting elements of the matrix T_{cb} are obvious from the definitions of the unit vectors and the location of the origin of the moving coordinate system. They were found formally here only to show an example of the utility of Eq. (3.24).

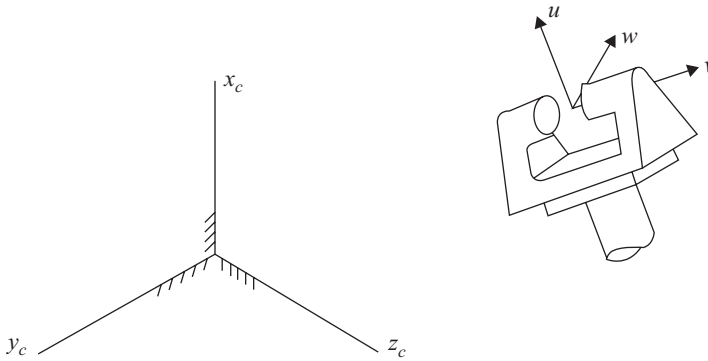


Figure 3.8. Posture of a robotic end-effector with respect to a base or world coordinate system.

3.6 Positions, Postures, and Displacements

We are all familiar with the meaning of the word “position.” It is a term that tells where an item is located. But let us be more precise. If the item in question is a particle or point, its position can be specified by its distance from and direction with respect to some set of reference axes whose location is assumed already known. If we choose to work with a Cartesian coordinate system, we can specify the position of a point by giving its x , y , and z coordinates. If we choose to work in cylindric or spheric coordinates, on the other hand, another set of parameters would be given. In section 3.2, we discussed a system of homogeneous coordinates for the position of a point. In any case, the position of a point in three-dimensional space is a vector quantity, having at least three scalar components called *coordinates*.

Let us next consider the term *position* when applied to something other than a point. In order to specify the location of a body or system of points, for example, it is necessary to specify more than just three coordinates. It is necessary to specify enough coordinates that the location of every point of the item being located is uniquely determined. If all of these coordinates are grouped into a single quantity according to some agreed upon set of conventions, then the result describes the location of the system of points.

In the case of a single rigid body, the location and orientation of a coordinate system fixed to the body with respect to a reference or world coordinate system describes the posture of that body. In section 3.5, we saw that this position can be described in terms of the location of the origin of the coordinate system fixed to the body as well as a (3×3) matrix describing the orientation of this coordinate system, both specified with respect to the same world coordinate system. In fact, we assembled these into a (4×4) matrix shown in Eq. (3.26).

In robotics, for example, this matrix is used to describe the location and orientation (the posture) of a coordinate system attached to the end-effector with respect to a base or world coordinate system. In this case, as shown in Figure 3.8, the unit vector \mathbf{u} is called the approach vector showing the direction in which the end-effector would approach an object to grasp it. The unit vector \mathbf{v} is called the orientation vector showing the orientation of the palm of the hand of the end-effector. Finally the unit vector \mathbf{w} is called the normal vector indicating the direction of the normal to

the surface defined by the two fingers of the end-effector. It is clear by its definition that $\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}$.

In some robotics literature, the word position is used loosely to describe the location of only a single point (such as the origin) of a coordinate system attached to the end-effector. In such literature, the end-effector is said to have a certain position; the orientation may then be added and the term “pose” is often used for the combination of the two. It should be pointed out, however, that the term *posture* is more appropriate¹ and is utilized throughout this text.

The term posture becomes even more suitable when dealing with a mechanism or multibody system because we are not concerned with the position of only a single point or of only a single rigid body, but of an assembly of rigid bodies, and we wish to describe the position of the entire system. We use the term posture to describe the configuration of a system, including both the locations and the orientations of every body of the system, all at a given moment in time.

If a point or a body or a coordinate system has a known position or posture and is subsequently moved to a new position or posture, it is said to have undergone a *displacement*, defined as the difference between its later and initial positions or postures. For a point, a displacement can be written in terms of a vector whose components consist of changes in the position coordinates; that is, the differences between the later position coordinates and those of the earlier reference position. For a rigid body, a displacement results from a change in posture of the body; that is, a change in location of a reference point and/or a change in orientation of the body. Using the notation of the previous section and representing the posture of a rigid body by a (4×4) transformation matrix, the displacement of a rigid body can be described by the difference between two such matrices representing the later and earlier postures (locations and orientations). For a mechanism or multibody system consisting of ℓ rigid bodies, the displacement of body b between some initial time (t_0) and some later time (t_1) is given by

$$\Delta T_{0b} = T_{0b}(t_1) - T_{0b}(t_0), \quad b = 1, 2, \dots, \ell. \quad (3.27)$$

Because a point remains fixed with respect to the coordinate system of its own rigid body, the displacement of a point of rigid body b is

$$\Delta R_b = R_b(t_1) - R_b(t_0) = T_{0b}(t_1)r_b - T_{0b}(t_0)r_b = \Delta T_{0b}r_b, \quad b = 1, 2, \dots, \ell. \quad (3.28)$$

We notice that displacement depends only on the two bounds of the interval. No information regarding *how* the later position or posture is achieved is necessary for its determination. If we fly from Chicago to London and return, we travel a great distance but our displacement is zero.

Suppose we wish to describe the displacement of a rigid body as it moves from some initial posture to a later posture. Suppose also that we wish to describe this change in posture of the body starting with data for the displacements of only a few points attached to the body. In three dimensions, a minimum of three non-collinear points must be measured to define a body's posture because their nine coordinates,

¹ The *Webster Comprehensive Dictionary: International Edition* states under its definition of attitude, “Synonyms: pose, position, posture. A posture is assumed without any special reference to expression of feeling; . . . A pose is a position studied for artistic effect or considered with reference to such an effect.”

subject to the three constant-distance constraints between them, give sufficient data for determining the six degrees of freedom of the body.

Suppose that we choose to measure the initial positions of three chosen points of the body with respect to some stationary coordinate system, and denote these data by Cartesian coordinate position vectors, $\mathbf{R}_1(t_0)$, $\mathbf{R}_2(t_0)$, and $\mathbf{R}_3(t_0)$. Suppose that we also measure the position coordinates of the same three points after the body has moved, $\mathbf{R}_1(t_1)$, $\mathbf{R}_2(t_1)$, and $\mathbf{R}_3(t_1)$. From these data, we wish to describe the displacement of the body in such a way that we are able to find the new positions of other points of the same body.

The moving body will carry a *body* coordinate system, x, y, z . Suppose we choose this moving coordinate system coincident with the global system at the initial time t_0 . This choice is convenient because it gives $T(t_0) = I$ and, by Eq. (3.24), $R(t_0) = r$ identifies the homogeneous body coordinates of the points.

Notice that the subscript for body label is not used in this section. Because there is only one moving body, moving with respect to the absolute frame, this should not cause confusion. Also, the subscripts showing point number or time state are used in this section only.

Now, to simplify the following calculations, we generate data for an additional fourth point of the moving body. Suppose we choose this additional point by the equation

$$\mathbf{r}_4 = \mathbf{r}_1 + [\mathbf{r}_2 - \mathbf{r}_1] \times [\mathbf{r}_3 - \mathbf{r}_1]. \quad (3.29)$$

Because the original three points are assumed distinct and non-collinear, this fourth point is linearly independent of the first three.

Because all four points belong to the same rigid body, the later position of this fourth point can be found by a similar calculation

$$\mathbf{R}_4(t_1) = \mathbf{R}_1(t_1) + [\mathbf{R}_2(t_1) - \mathbf{R}_1(t_1)] \times [\mathbf{R}_3(t_1) - \mathbf{R}_1(t_1)].$$

According to Eq. (3.27), we can write in homogeneous coordinates that

$$R(t_1) = (I + \Delta T)r,$$

and, from this, we define a new matrix T that transforms the points from the initial position r to the altered position $R(t_1)$.

$$T = I + \Delta T, \quad (3.30)$$

$$R(t_1) = Tr. \quad (3.31)$$

Writing this equation four times, once for each of our points, we can group these into a single matrix equation as follows:

$$[R_1(t_1) \mid R_2(t_1) \mid R_3(t_1) \mid R_4(t_1)] = T[r_1 \mid r_2 \mid r_3 \mid r_4],$$

and because the matrix of initial point position data is square (4×4) and non-singular (because the four points are not coplanar), this equation can be solved to find the T matrix as follows:

$$T = [R_1(t_1) \mid R_2(t_1) \mid R_3(t_1) \mid R_4(t_1)][r_1 \mid r_2 \mid r_3 \mid r_4]^{-1}. \quad (3.32)$$

Once this T matrix is known, any point of that same moving body can be specified by its initial position $r = R(t_0)$, and its displaced position $R(t_1)$ can be found by Eq.(3.31). A numeric example will illustrate.

EXAMPLE 3.1 Suppose we have data for the position coordinates of three points of a moving body such that we know:

$$\mathbf{r}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_2 = \begin{bmatrix} 2 \text{ in} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_3 = \begin{bmatrix} 0 \\ 5 \text{ in} \\ 0 \\ 1 \end{bmatrix},$$

$$\mathbf{R}_1(t_1) = \begin{bmatrix} 2.872 \text{ in} \\ -4.757 \text{ in} \\ 6.469 \text{ in} \\ 1 \end{bmatrix}, \quad \mathbf{R}_2(t_1) = \begin{bmatrix} 1.493 \text{ in} \\ -3.927 \text{ in} \\ 5.281 \text{ in} \\ 1 \end{bmatrix}, \quad \mathbf{R}_3(t_1) = \begin{bmatrix} 6.098 \text{ in} \\ -1.137 \text{ in} \\ 5.251 \text{ in} \\ 1 \end{bmatrix}.$$

Using Eqs. (3.29) and its successor, we calculate data for two positions of an independent fourth point

$$\mathbf{r}_4 = \begin{bmatrix} 0 \\ 0 \\ 10 \text{ in} \\ 1 \end{bmatrix}, \quad \mathbf{R}_4(t_1) = \begin{bmatrix} 6.162 \text{ in} \\ -10.270 \text{ in} \\ -1.197 \text{ in} \\ 1 \end{bmatrix}.$$

We now use Eq. (3.32) to find the T matrix

$$T = \begin{bmatrix} 2.872 \text{ in} & 1.493 \text{ in} & 6.098 \text{ in} & 6.162 \text{ in} \\ -4.757 \text{ in} & -3.927 \text{ in} & -1.137 \text{ in} & -10.270 \text{ in} \\ 6.469 \text{ in} & 5.281 \text{ in} & 5.251 \text{ in} & -1.197 \text{ in} \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 \text{ in} & 0 & 0 \\ 0 & 0 & 5 \text{ in} & 0 \\ 0 & 0 & 0 & 10 \text{ in} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1},$$

$$T = \begin{bmatrix} -0.68936 & 0.64538 & 0.32904 & 2.87154 \text{ in} \\ 0.41461 & 0.72396 & -0.55134 & -4.75653 \text{ in} \\ -0.59404 & -0.24366 & -0.76668 & 6.46932 \text{ in} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the displacement matrix for the body over this time interval is found from Eq. (3.30),

$$\Delta T = T - I = \begin{bmatrix} -1.68936 & 0.64538 & 0.32904 & 2.87154 \text{ in} \\ 0.41461 & -0.27604 & -0.55134 & -4.75653 \text{ in} \\ -0.59404 & -0.24366 & -1.76668 & 6.46932 \text{ in} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

From the T matrix we can find the displaced position of any additional point(s) of the body as follows

$$\mathbf{r}_5 = \begin{bmatrix} 2 \text{ in} \\ 5 \text{ in} \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{R}_5(t_1) = T\mathbf{r}_5 = \begin{bmatrix} 4.720 \text{ in} \\ -0.308 \text{ in} \\ 4.063 \text{ in} \\ 1 \end{bmatrix},$$

or, from the displacement matrix ΔT , we can find the displacement of any additional point(s) of the body:

$$\Delta R_5 = \Delta T \begin{bmatrix} 2 \text{ in} \\ 5 \text{ in} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.720 \text{ in} \\ -5.308 \text{ in} \\ 4.063 \text{ in} \\ 0 \end{bmatrix}.$$

From this example, we see that the T matrix appears similar in many ways to our transformation matrix of previous sections. We see the characteristic zero and unit entries in the bottom row and a quick check verifies that the upper-left (3×3) submatrix is orthogonal with determinant of positive one, as a rotation submatrix must be. Was this accidental or peculiar to this example? No; these properties are true in the general case and proof is shown in the next section.

We see here that there are two quite different ways in which a transformation can be viewed. It can be seen as a relationship between the postures of one coordinate system and another. The matrix T_{cb} of the previous sections answers the question, “Where is the coordinate system of body b when measured with respect to the coordinate system of body c at a chosen moment in time?” Alternatively, it can be seen as a relationship between two different postures of the same moving coordinate system at two different values of time; the $T(t_1)$ matrix of this section answers the question, “What is the displaced posture of this particular coordinate system at time t_1 with respect to its initial posture?”

The posture of a rigid body such that its body coordinate system coincides with the fixed or global coordinate system, as shown in [Figure 3.9](#), is usually referred to as the *reference posture*. At this posture, vectors from the origins of the two coincident coordinate systems to a point of the rigid body can be represented by r_b in the body coordinate system or by R_b in the fixed or global coordinate system. Thus, at the reference or zero posture, $T(t_0) = I$ and $R_b(t_0) = r_b$.

When the body moves from this reference or zero posture to a new posture at time t_1 , the new position of the point can be written as that given by Eq. (3.31).

$$R_b = T r_b. \quad (3.33)$$

In this equation, $T = I + \Delta T$, but $\Delta T = T(t_1) - T(t_0) = T(t_1) - I$. Therefore, $T = I + T(t_1) - I = T(t_1)$ and $T(t_1)$ is the (4×4) transformation matrix representing the posture of the moving body at time t_1 . This is the same (4×4) transformation

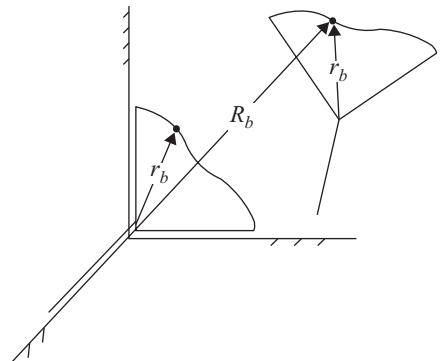


Figure 3.9. Zero or reference posture of a rigid body or coordinate system.

matrix of Eq. (3.24) that was used to represent a coordinate transformation. In other words, Eq. (3.33) can be used to represent the displacement of a point of a rigid body as the body moves from a reference or zero posture to an arbitrary posture or, alternatively, it can be used to represent a coordinate transformation between the body coordinate system of a rigid body and the global coordinate system.

As a rigid body moves from a reference posture through a set of displacements in space, a point attached to the moving body takes positions R_0, R_1, \dots, R_N in the fixed space. Because these points represent different locations of the same point of the rigid body, they are referred to as *homologous* points. If the body goes through N displacements then there are N homologous points associated with a chosen point of the body. As a rigid body moves in space, different points of the body produce different sets of homologous points. In a mechanism or multibody system, for example, a human musculoskeletal system, these sets of homologous points might be sets of points representing the positions of the joints of the system.

3.7 Euler's and Chasles' Theorems

Euler's and Chasles' theorems are two fundamental theorems dealing with rigid body displacements. Euler's theorem deals only with a change in orientation of a rigid body whereas Chasles' theorem deals with a general displacement involving a change in location as well as a change in orientation; that is, a change in posture.

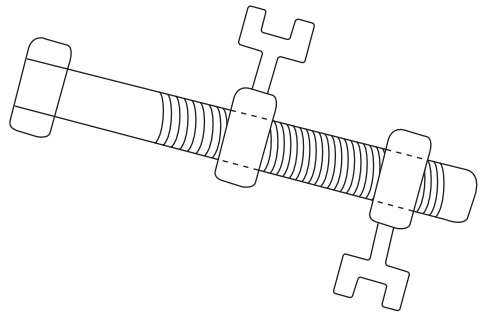
Chasles' theorem [3], named after the French mathematician, Michel Chasles (1793–1880), states that,

Any general three-dimensional displacement of a rigid body can be achieved by a uniform helical motion about a fixed axis.

We note that a helical motion consists of a rotation about and a translation along the axis of a helix that is fixed. For a uniform helical motion, the ratio of the translation along the axis to the rotation about the axis remains constant and is referred to as the pitch of the helical motion. This means that, given any two separated postures of a rigid body, there exists an axis in space with which the body can be brought from its initial posture to its final posture by a rotation about, and a translation along this axis with a fixed pitch. Such a motion can be mechanically realized by a bolt and a nut (see Figure 3.10) where the motion of the nut on the bolt consists of a translation along the axis of the bolt and a rotation about the same axis. This axis is referred to as the *screw axis*. This is why a general rigid body displacement is sometimes referred to as a screw displacement. The screw axis was first identified and defined by the Italian mathematician, Giulio Mozzi (1730–1813), and was published in [14], as is described in detail in [2] by Ceccarelli.

The amount of the relative translation (denoted here as ϕ) together with the amount of the relative rotation (denoted here as θ) about the screw axis define the extent of the displacement and are referred to as screw parameters. The pitch of the displacement (denoted here as σ) is defined in the same way that the pitch of a screw is defined, and indicates the ratio of the magnitude of the relative translation to the magnitude of the relative rotation about the helical axis. In this text, positive rotation is taken to be counterclockwise and positive translation is taken in the direction of the positive sense along the screw axis.

Figure 3.10. A screw displacement performed by a bolt and nut.



Euler's theorem [6], defined six decades earlier, can be considered a special case of Chasles' more general theorem in that only the rotational component of a displacement is considered. It can be stated as,

Any general three-dimensional displacement of a rigid body with one point fixed can be achieved by a single rotation about a fixed axis through the fixed point.

In this case, where there is no translation, the axis is similar to the screw axis, but it now becomes a rotation axis.

In conjunction with these two theorems, it is sometimes desirable to write the equations, Eq. (3.33), governing rigid body displacements in terms of the screw axis and screw parameters identified in Chasles' theorem. Such equations were first derived by the French mathematician, Benjamin Olinde Rodrigues (1794–1851), in his work on the composition of multiple displacements [16]. Here a slightly different derivation is presented.

Consider a rigid body in two different postures as shown in Figure 3.11. This body can be considered to have gone through a spatial displacement from its first posture to its second posture. A chosen point of the body assumes two different

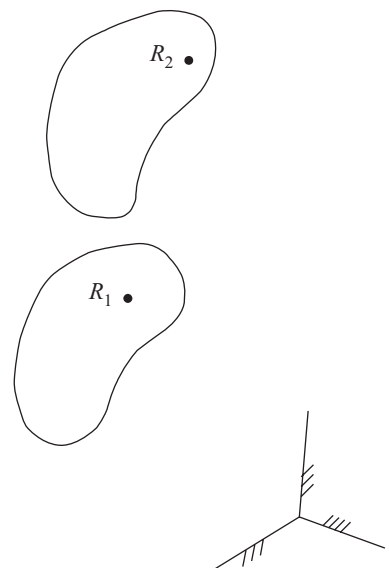


Figure 3.11. A spatial displacement of a rigid body.

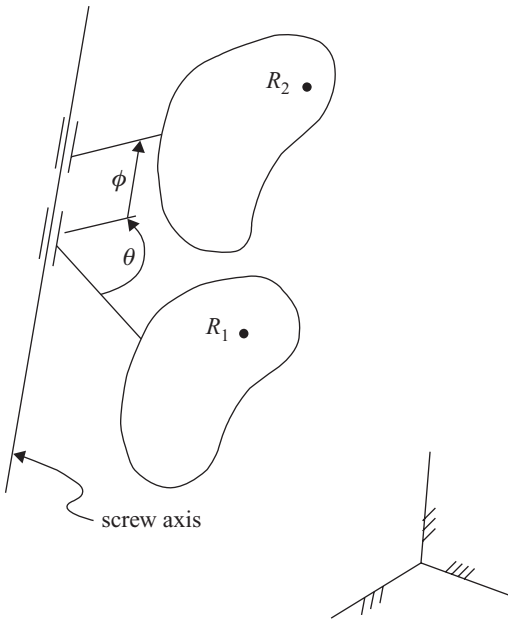


Figure 3.12. The screw axis and screw parameters of a displacement.

positions R_1 and R_2 with respect to the fixed coordinate system. Points R_1 and R_2 are homologous points because they are different positions of the same point as the body goes through a displacement. The screw axis and the screw parameters, θ and ϕ , for the displacement of the rigid body are shown in [Figure 3.12](#).

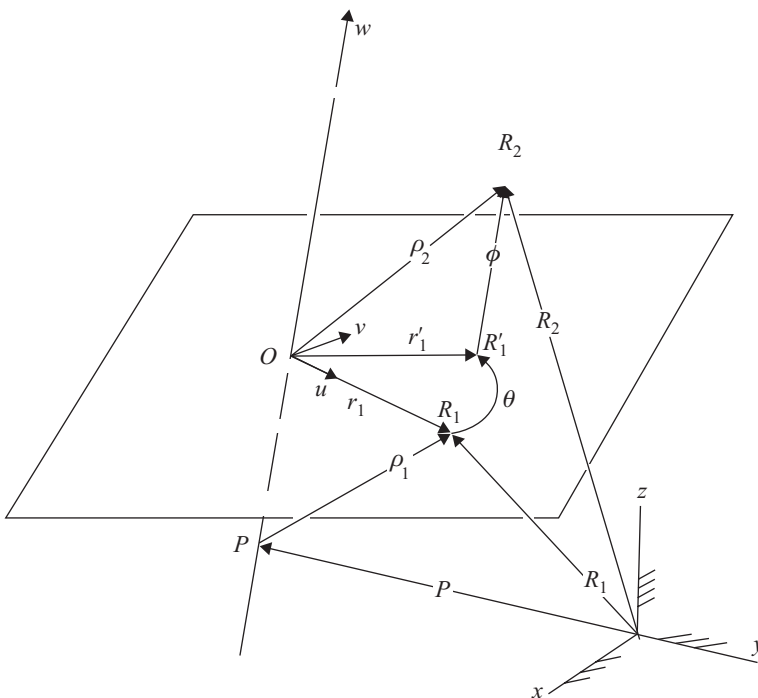


Figure 3.13. Geometry of a screw displacement.

We are now interested in developing equations relating the positions of the two homologous points R_1 and R_2 in terms of the screw axis and the screw parameters. Recall that these are two positions of the same point of the moving body in its two postures in space. In the fixed coordinate system, the screw axis can be specified in terms of a vector from the origin of the coordinate system to one point $\mathbf{P} = [P^x, P^y, P^z]^t$ on the screw axis and a unit vector $\mathbf{w} = [w^x, w^y, w^z]^t$ parallel to the screw axis as shown in Figure 3.13. The screw parameters, namely the counterclockwise rotation about and the translation along the screw axis, are θ and ϕ , respectively. We are interested in finding the functional relationship

$$\mathbf{R}_2 = f(\mathbf{R}_1, \mathbf{P}, \mathbf{w}, \theta, \phi).$$

In this expression \mathbf{R}_1 and \mathbf{R}_2 are two position vectors of point \mathbf{R} of the moving body, both measured in the fixed coordinate system, as the body goes through a helical or screw displacement with the screw axis defined by vectors (\mathbf{P}, \mathbf{w}) and with screw parameters (θ, ϕ) .

From Figure 3.13, it is clear that:

$$\boldsymbol{\rho}_1 = \mathbf{R}_1 - \mathbf{P}, \quad (a)$$

and

$$\mathbf{R}_2 = \mathbf{P} + \overrightarrow{P\mathbf{O}} + \boldsymbol{\rho}_2. \quad (b)$$

Also, considering $|r_1| = |r'_1|$, it can be seen from Figure 3.13 that

$$\boldsymbol{\rho}_2 = r_1 \cos \theta \mathbf{u} + r_1 \sin \theta \mathbf{v} + \phi \mathbf{w}.$$

In addition, we see that

$$\overrightarrow{P\mathbf{O}} = (\boldsymbol{\rho}_1 \cdot \mathbf{w}) \mathbf{w},$$

and

$$\mathbf{v} = \mathbf{w} \times \mathbf{u}.$$

Substituting these into Eq. (b), we get

$$\mathbf{R}_2 = \mathbf{P} + (\boldsymbol{\rho}_1 \cdot \mathbf{w}) \mathbf{w} + r_1 [\cos \theta \mathbf{u} + \sin \theta \mathbf{w} \times \mathbf{u}] + \phi \mathbf{w}.$$

Now, substituting

$$r_1 \mathbf{u} = \boldsymbol{\rho}_1 - \overrightarrow{P\mathbf{O}},$$

and $\boldsymbol{\rho}_1$ from Eq. (a) and simplifying, we obtain our desired equation:

$$\mathbf{R}_2 = (\mathbf{R}_1 - \mathbf{P}) \cos \theta + [(\mathbf{R}_1 - \mathbf{P}) \cdot \mathbf{w}] \mathbf{w} (1 - \cos \theta) + \mathbf{w} \times (\mathbf{R}_1 - \mathbf{P}) \sin \theta + \mathbf{P} + \phi \mathbf{w}. \quad (3.34)$$

Equation (3.34) is the vector form of the general spatial displacement equation relating two positions of a point of a moving body as it goes through a screw displacement expressed in terms of the parameters defined in Chasles' theorem. The equation given here explicitly finds the second position \mathbf{R}_2 of a point of the moving body in terms of its first position \mathbf{R}_1 , the screw axis (\mathbf{P}, \mathbf{w}) , and screw parameters

(θ, ϕ) , and uses the total displacement angle. Olinde Rodrigues was the first to derive such equations, although his equations were in implicit form and were expressed in terms of half-angles.

Before we discuss the use of half-angles in section 3.8, we would like to express Eq. (3.34) in matrix form. We note that, if we have two vectors \mathbf{a} and \mathbf{b} ,

$$\mathbf{a} = \begin{bmatrix} a^x \\ a^y \\ a^z \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b^x \\ b^y \\ b^z \end{bmatrix},$$

then the vector dot product $\mathbf{a} \cdot \mathbf{b}$ can be written in matrix form as $\mathbf{a}^t \mathbf{b}$. Also, if we define the notation

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a^z & a^y \\ a^z & 0 & -a^x \\ -a^y & a^x & 0 \end{bmatrix},$$

then the vector cross product $\mathbf{a} \times \mathbf{b}$ can be written in matrix form as $\tilde{\mathbf{a}} \mathbf{b}$.

It can also be seen that

$$(\mathbf{b} \cdot \mathbf{a}) \mathbf{a} = (\mathbf{a} \cdot \mathbf{b}) \mathbf{a} = (\mathbf{a}^t \mathbf{b}) \mathbf{a} = \mathbf{a} (\mathbf{a}^t \mathbf{b}) = (\mathbf{a} \mathbf{a}^t) \mathbf{b}. \quad (c)$$

If we take the vector \mathbf{a} to be the unit vector \mathbf{w} that defines the orientation of the screw axis, then

$$w w^t = \begin{bmatrix} (w^x)^2 & w^x w^y & w^x w^z \\ w^y w^x & (w^y)^2 & w^y w^z \\ w^z w^x & w^z w^y & (w^z)^2 \end{bmatrix} \quad \text{and} \quad \tilde{w} = \begin{bmatrix} 0 & -w^z & w^y \\ w^z & 0 & -w^x \\ -w^y & w^x & 0 \end{bmatrix}.$$

However, because \mathbf{w} is a unit vector – that is, $w^t w = 1$ – it can be verified that $I - w w^t = \tilde{w} \tilde{w}^t$, where I is the (3×3) identity matrix. Furthermore, because \tilde{w} is a skew-symmetric matrix, $\tilde{w}^t = -\tilde{w}$, and we can write $I - w w^t = \tilde{w} \tilde{w}^t = -\tilde{w}^2$, or $w w^t = I + \tilde{w}^2$. Substituting this into Eq. (c), we can write:

$$(\mathbf{b} \cdot \mathbf{w}) \mathbf{w} = (\mathbf{w} \cdot \mathbf{b}) \mathbf{w} = (\mathbf{w}^t \mathbf{b}) \mathbf{w} = \mathbf{w} (\mathbf{w}^t \mathbf{b}) = (w w^t) \mathbf{b} = (I + \tilde{w}^2) \mathbf{b}.$$

Using these vector-to-matrix conversion identities, with $\mathbf{b} = (\mathbf{R}_1 - \mathbf{P})$, we can rewrite Eq. (3.34) in matrix form as

$$\mathbf{R}_2 = [I + \tilde{w} \sin \theta + \tilde{w}^2 (1 - \cos \theta)] (\mathbf{R}_1 - \mathbf{P}) + \mathbf{P} + \phi \mathbf{w}. \quad (3.35)$$

Equation (3.35) can also be written as

$$\mathbf{R}_2 = \Theta \mathbf{R}_1 + d, \quad (3.36)$$

where

$$\Theta = [I + \tilde{w} \sin \theta + \tilde{w}^2 (1 - \cos \theta)], \quad (3.37)$$

and

$$d = (I - \Theta) \mathbf{P} + \phi \mathbf{w}. \quad (3.38)$$

If we consider the moving body to have a coordinate system that is initially coincident with the coordinate system of the fixed body (see Figure 3.9), then Eq. (3.36) can be written in terms of a (4×4) homogeneous transformation matrix as

$$\begin{bmatrix} R_2^x \\ R_2^y \\ R_2^z \\ 1 \end{bmatrix} = \begin{bmatrix} \Theta(1,1) & \Theta(1,2) & \Theta(1,3) & d^x \\ \Theta(2,1) & \Theta(2,2) & \Theta(2,3) & d^y \\ \Theta(3,1) & \Theta(3,2) & \Theta(3,3) & d^z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_1^x \\ R_1^y \\ R_1^z \\ 1 \end{bmatrix}, \quad (3.39)$$

or $R_2 = T_{21}R_1$, with

$$T = \begin{bmatrix} \Theta(1,1) & \Theta(1,2) & \Theta(1,3) & d^x \\ \Theta(2,1) & \Theta(2,2) & \Theta(2,3) & d^y \\ \Theta(3,1) & \Theta(3,2) & \Theta(3,3) & d^z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \Theta & d \\ 0 & 1 \end{bmatrix},$$

where the elements of the (4×4) matrix T are given by

$$\begin{aligned} \Theta(1,1) &= [(w^x)^2 - 1](1 - \cos \theta) + 1, \\ \Theta(1,2) &= w^y w^x (1 - \cos \theta) - w^z \sin \theta, \\ \Theta(1,3) &= w^z w^x (1 - \cos \theta) + w^y \sin \theta, \\ \Theta(2,1) &= w^x w^y (1 - \cos \theta) + w^z \sin \theta, \\ \Theta(2,2) &= [(w^y)^2 - 1](1 - \cos \theta) + 1, \\ \Theta(2,3) &= w^z w^y (1 - \cos \theta) - w^x \sin \theta, \\ \Theta(3,1) &= w^x w^z (1 - \cos \theta) - w^y \sin \theta, \\ \Theta(3,2) &= w^y w^z (1 - \cos \theta) + w^x \sin \theta, \\ \Theta(3,3) &= [(w^z)^2 - 1](1 - \cos \theta) + 1, \end{aligned} \quad (3.40)$$

and

$$\begin{aligned} d^x &= \phi w^x - [\Theta(1,1) - 1]P^x - \Theta(1,2)P^y - \Theta(1,3)P^z, \\ d^y &= \phi w^y - \Theta(2,1)P^x - [\Theta(2,2) - 1]P^y - \Theta(2,3)P^z, \\ d^z &= \phi w^z - \Theta(3,1)P^x - \Theta(3,2)P^y - [\Theta(3,3) - 1]P^z. \end{aligned} \quad (3.41)$$

We will refer to Eq. (3.39) as the *screw displacement equation* because the elements of its (4×4) homogeneous transformation matrix are given in terms of the screw axis and the screw parameters as shown in Eqs. (3.40) and (3.41).

If we consider only the rotation part, Eq. (3.39) becomes

$$\begin{bmatrix} R_2^x \\ R_2^y \\ R_2^z \end{bmatrix} = \begin{bmatrix} \Theta(1,1) & \Theta(1,2) & \Theta(1,3) \\ \Theta(2,1) & \Theta(2,2) & \Theta(2,3) \\ \Theta(3,1) & \Theta(3,2) & \Theta(3,3) \end{bmatrix} \begin{bmatrix} R_1^x \\ R_1^y \\ R_1^z \end{bmatrix}, \quad (3.42)$$

or symbolically,

$$R_2 = \Theta_{21}R_1, \quad (3.43)$$

where Θ_{21} is the (3×3) rotation matrix whose elements $\Theta_{21}(i, j)$, with $i, j = 1, 2, 3$, are given by Eqs. (3.40).

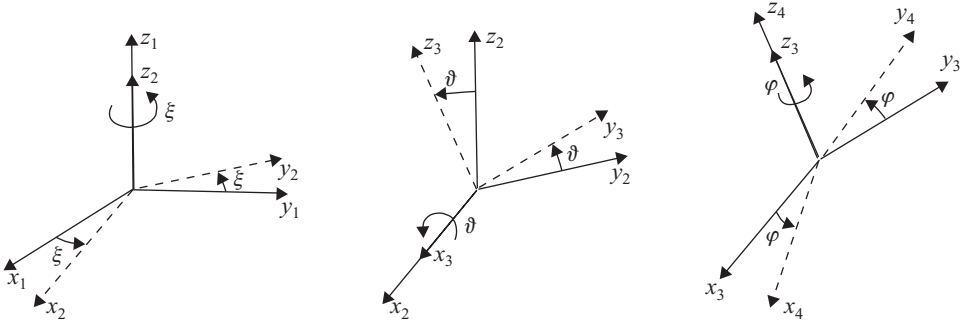


Figure 3.14. Precession, nutation, and spin Euler angles.

Equations (3.42) and (3.43) provide an analytic representation of Euler's theorem. They can also be used to derive an equation for representing orientation using Euler angles by considering the final orientation of a body as a composition of three rotations. For example, as shown in Figure 3.14, the first rotation moves the line of nodes around the fixed z_1 axis, the second rotation is around the line of nodes, and the third rotation is around the z_4 axis fixed in the moving body. These three rotations are called *precession*, *nutation*, and *spin*, respectively, and they are illustrated in Figure 3.14.

The first rotation (called precession) is counterclockwise by the angle ξ about the z_1 axis as shown in Figure 3.14a. The screw or rotation axis for this rotation is z_1 with direction cosines $\mathbf{w}_1 = (0, 0, 1)^t$. Using Eqs. (3.40), the corresponding rotation matrix for precession becomes:

$$\Theta_{12} = \begin{bmatrix} \cos \xi & -\sin \xi & 0 \\ \sin \xi & \cos \xi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Next, we start with the coordinate system $x_2y_2z_2$ as the reference and rotate about the x_2 axis by the counterclockwise angle ϑ (called nutation) to a new orientation indicated as $x_3y_3z_3$ in Figure 3.14b. The screw or rotation axis is x_2 with direction cosines $\mathbf{w}_2 = (1, 0, 0)^t$. Using Eqs. (3.40), the rotation matrix for nutation becomes:

$$\Theta_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \vartheta & -\sin \vartheta \\ 0 & \sin \vartheta & \cos \vartheta \end{bmatrix}.$$

Finally we perform one more rotation to a new orientation $x_4y_4z_4$ as shown in Figure 3.14c. This time the rotation (called spin) is by the counterclockwise angle φ about the rotation or screw axis z_3 that is fixed in the moving body. The direction cosines of this axis are $\mathbf{w}_3 = (0, 0, 1)^t$, and the rotation matrix for spin becomes:

$$\Theta_{34} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

If we now consider the coordinates of a point in the coordinate system $x_4y_4z_4$ and relate it to the coordinates of the same point in the coordinate system $x_1y_1z_1$,

we can write $r_1 = \Theta_{12}r_2$, $r_2 = \Theta_{23}r_3$, and $r_3 = \Theta_{34}r_4$. Substituting these equations into one another, we find $r_1 = \Theta_{12}\Theta_{23}\Theta_{34}r_4$ and the final orientation is described by:

$$\Theta_{14} = \begin{bmatrix} \cos \xi & -\sin \xi & 0 \\ \sin \xi & \cos \xi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \vartheta & -\sin \vartheta \\ 0 & \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

or

$$\Theta_{14} = \left[\begin{array}{c|c|c} -\sin \xi \cos \vartheta \sin \varphi + \cos \xi \cos \varphi & -\sin \xi \cos \vartheta \cos \varphi - \cos \xi \sin \varphi & \sin \xi \sin \vartheta \\ \cos \xi \cos \vartheta \sin \varphi + \sin \xi \cos \varphi & \cos \xi \cos \vartheta \cos \varphi - \sin \xi \sin \varphi & -\cos \xi \sin \vartheta \\ \sin \vartheta \sin \varphi & \sin \vartheta \cos \varphi & \cos \vartheta \end{array} \right]. \quad (3.44)$$

From Eq. (3.44) we can also write

$$\begin{aligned} \Theta_{41} &= \Theta_{14}^{-1} = \Theta_{14}^t = \Theta_{34}^t \Theta_{23}^t \Theta_{12}^t \\ &= \left[\begin{array}{c|c|c} -\sin \xi \cos \vartheta \sin \varphi + \cos \xi \cos \varphi & \cos \xi \cos \vartheta \sin \varphi + \sin \xi \cos \varphi & \sin \vartheta \sin \varphi \\ -\sin \xi \cos \vartheta \cos \varphi - \cos \xi \sin \varphi & \cos \xi \cos \vartheta \cos \varphi - \sin \xi \sin \varphi & \sin \vartheta \cos \varphi \\ \sin \xi \sin \varphi & -\cos \xi \sin \vartheta & \cos \vartheta \end{array} \right]. \end{aligned} \quad (3.45)$$

The reader may wish to verify that $|\Theta_{41}| = |\Theta_{14}| = +1$.

It should be pointed out that there are many different ways that Euler angles can be defined by changing the choice of axes or the order in which the rotations take place. In the previous derivation, for example, we could have taken the second rotation by the amount ϑ in the counterclockwise direction about the y_2 axis instead of about x_2 . In such a case, the rotation matrix for the second rotation would have become:

$$\Theta_{23} = \begin{bmatrix} \cos \vartheta & 0 & \sin \vartheta \\ 0 & 1 & 0 \\ -\sin \vartheta & 0 & \cos \vartheta \end{bmatrix},$$

and the final orientation would be described by

$$\Theta_{14} = \left[\begin{array}{c|c|c} \cos \xi \cos \vartheta \cos \varphi - \sin \xi \sin \varphi & -\cos \xi \cos \vartheta \sin \varphi - \sin \xi \cos \varphi & \cos \xi \sin \vartheta \\ \sin \xi \cos \vartheta \cos \varphi + \cos \xi \sin \varphi & -\sin \xi \cos \vartheta \sin \varphi + \cos \xi \cos \varphi & \sin \xi \sin \vartheta \\ \sin \vartheta \cos \varphi & \sin \vartheta \sin \varphi & \cos \vartheta \end{array} \right].$$

Consider yet another case where we take the first rotation to be a counterclockwise angle α about the x_1 axis, the second rotation to be a counterclockwise angle β about the new y_2 axis, and the final rotation to be a counterclockwise angle γ about the modified z_3 axis. The resulting rotation matrices are

$$\begin{aligned} \Theta_{12} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, & \Theta_{23} &= \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \\ \Theta_{34} &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

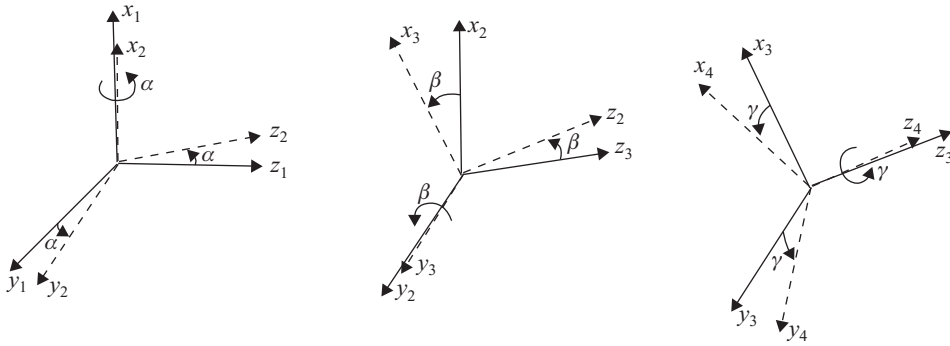


Figure 3.15. Cardan angles.

The final orientation of the body is now given by

$$\Theta_{14} = \begin{bmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{bmatrix}. \quad (3.46)$$

This particular choice of Euler angles is shown in [Figure 3.15](#) and, when taken in this order and about these axes, is called a set of *Cardan angles*, named after the Italian mathematician, Gerolamo Cardano (1501–76), who was the first to develop such a minimal representation of orientation, two centuries before Euler (1707–83). They are sometimes also called *Tait-Bryan* angles, after the Scottish mathematical physicist, Peter Guthrie Tait (1831–1901), and George Hartley Bryan (1864–1928), professor at University College, Bangor, Wales.

Furthermore, Cardan angles are still used to represent some very practical situations, such as describing the orientation of a spacecraft, a car or truck, or a nautical craft. In these cases, if the x -axis is aligned along the fore-aft axis, with the y axis lateral and the z axis vertical, then the three Cardan angles describe the *roll*, *pitch*, and *yaw* of the vessel (see [Figure 3.16](#)) and are sometimes referred to as nautical angles.

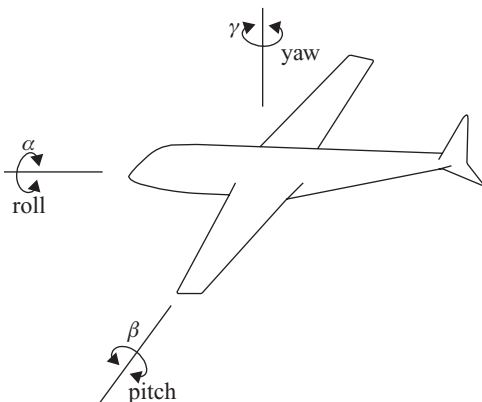


Figure 3.16. Roll, pitch, and yaw axes for an aircraft.

Once the axes, the order, and the sign conventions are specified, three angles are usually unique and sufficient to specify the relative orientation of two arbitrarily oriented Cartesian coordinate systems. However, there are certain orientations (for example, with the conventions used here, when the angle β is equal to $\pm 90^\circ$), for which there are not unique values for α and γ for a given orientation because, at such an attitude, the α and γ angles are measured about collinear axes. The same problem occurs in Eq. (3.44) when angle $\vartheta = 0$ or $\vartheta = \pm 180^\circ$. Such an orientation does not have three independent angles and full three-dimensional rotation about such an orientation cannot be described by this choice of axes. All sets of Euler angles share this difficulty. There is always at least one orientation where two of the rotation axes become collinear, causing a singularity called “gimbal lock” in the description. In fact, it has been shown [17] that it is not possible to have a parameterization of the rotation matrix in terms of three parameters without singularities. However, using four parameters, as with the Euler-Rodrigues parameters discussed in the next section, this problem is eliminated.

3.8 Euler-Rodrigues Parameters

As mentioned in the previous section, there is a major difficulty with the use of any set of Euler angles for the specification of spatial orientation. No matter which set of conventions is chosen, there is always a singularity in the specification for some values of the angles. It is desirable to have a method of specifying three-dimensional rotation that does not display such a singularity for any orientation. The Euler-Rodrigues parameters provide such a description. These parameters, referred to in some literature as just Euler parameters, are based on half-angles. It is pointed out here, however, that Euler only used full angles in all of his work and it was Rodrigues who first introduced the use of half-angles for parameterization of rotation in his derivation of the Rodrigues equations.

Here we start with Eqs. (3.40) and substitute the following two trigonometric identities:

$$\cos \theta = \cos^2(\theta/2) - \sin^2(\theta/2) \quad \text{and} \quad \sin \theta = 2 \sin(\theta/2) \cos(\theta/2).$$

With these, the elements of the rotation matrix given by Eqs. (3.40) become:

$$\begin{aligned} \Theta(1, 1) &= 2[(w^x)^2 - 1] \sin^2(\theta/2) + 1, \\ \Theta(1, 2) &= 2 \sin(\theta/2)[w^y w^x \sin(\theta/2) - w^z \cos(\theta/2)], \\ \Theta(1, 3) &= 2 \sin(\theta/2)[w^z w^x \sin(\theta/2) + w^y \cos(\theta/2)], \\ \Theta(2, 1) &= 2 \sin(\theta/2)[w^x w^y \sin(\theta/2) + w^z \cos(\theta/2)], \\ \Theta(2, 2) &= 2[(w^y)^2 - 1] \sin^2(\theta/2) + 1, \\ \Theta(2, 3) &= 2 \sin(\theta/2)[w^z w^y \sin(\theta/2) - w^x \cos(\theta/2)], \\ \Theta(3, 1) &= 2 \sin(\theta/2)[w^x w^z \sin(\theta/2) - w^y \cos(\theta/2)], \\ \Theta(3, 2) &= 2 \sin(\theta/2)[w^y w^z \sin(\theta/2) + w^x \cos(\theta/2)], \\ \Theta(3, 3) &= 2[(w^z)^2 - 1] \sin^2(\theta/2) + 1, \end{aligned} \tag{3.47}$$

Next we define the vector

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \end{Bmatrix} = \begin{Bmatrix} w^x \\ w^y \\ w^z \end{Bmatrix} \sin(\theta/2).$$

We note that this vector is aligned with a possible axis of rotation and, therefore, the components of this vector are the same in both coordinate frames. However, we also note that, although the w vector is aligned with a *possible* axis of rotation for the total displacement through the finite angle θ , it may not be aligned with the *actual* instantaneous axis of rotation at the beginning or end of the displacement.

We also define a fourth parameter, $e^4 = \cos(\theta/2)$, so that the total set becomes a four-dimensional unit vector

$$e = \begin{Bmatrix} e^1 \\ e^2 \\ e^3 \\ e^4 \end{Bmatrix} = \begin{Bmatrix} w^x \sin(\theta/2) \\ w^y \sin(\theta/2) \\ w^z \sin(\theta/2) \\ \cos(\theta/2) \end{Bmatrix}, \quad (3.48)$$

where

$$(e^1)^2 + (e^2)^2 + (e^3)^2 + (e^4)^2 = e^t e = 1. \quad (3.49)$$

Expressed in terms of these Euler-Rodrigues parameters, the rotation matrix for the orientation of the $x_1 y_1 z_1$ coordinate frame with respect to the $x_2 y_2 z_2$ coordinate frame is

$$\Theta_{21} = \begin{bmatrix} (e^1)^2 - (e^2)^2 - (e^3)^2 + (e^4)^2 & 2e^1 e^2 - 2e^3 e^4 & 2e^1 e^3 + 2e^2 e^4 \\ 2e^1 e^2 + 2e^3 e^4 & -(e^1)^2 + (e^2)^2 - (e^3)^2 + (e^4)^2 & 2e^2 e^3 - 2e^1 e^4 \\ 2e^1 e^3 - 2e^2 e^4 & 2e^2 e^3 + 2e^1 e^4 & -(e^1)^2 - (e^2)^2 + (e^3)^2 + (e^4)^2 \end{bmatrix}. \quad (3.50)$$

For situations in which a numeric form of the rotation matrix is already known and we wish to compute the corresponding Euler-Rodrigues parameters, we may desire a numeric procedure to do this. The following procedure is adapted from Friberg [8].

Let us suppose that the elements of the rotation matrix, with the symbolism $\Theta(i, j)$ referring to the element of matrix Θ in row i and column j , have known numeric values. Because the trace of a square matrix is defined as the sum of the terms on the major diagonal, from Eq. (3.50) we see that

$$\text{trace}(\Theta) = \{\Theta(1, 1) + \Theta(2, 2) + \Theta(3, 3)\} = \{-(e^1)^2 - (e^2)^2 - (e^3)^2 + 3(e^4)^2\}.$$

However, because we know that $(e^1)^2 + (e^2)^2 + (e^3)^2 + (e^4)^2 = 1$, then $(e^4)^2 = \frac{1}{4}\{\text{trace}(\Theta) + 1\}$, or

$$e^4 = \pm y \sqrt{\text{trace}(\Theta) + 1}, \quad (3.51)$$

where the sign may be selected arbitrarily.

Computation of the other three Euler-Rodrigues parameters now involves consideration of numeric issues such as division by a small value or subtraction of

numeric values that are nearly equal but not identical. For these reasons, we establish a precision threshold ε such that e^4 is considered close to zero if $|e^4| \leq \varepsilon$.² There are two possible cases:

Case 1: In this case, $|e^4| > \varepsilon$.

We note, by subtracting off-diagonal elements of Θ , that

$$\begin{aligned} e^1 &= [\Theta(3, 2) - \Theta(2, 3)]/(4e^4), \\ e^2 &= [\Theta(1, 3) - \Theta(3, 1)]/(4e^4), \\ e^3 &= [\Theta(2, 1) - \Theta(1, 2)]/(4e^4), \end{aligned} \tag{3.52}$$

and, from the diagonal terms,

$$\begin{aligned} e^1 &= \pm \frac{1}{2} \sqrt{1 + 2\Theta(1, 1) - \text{trace}(\Theta)}, \\ e^2 &= \pm \frac{1}{2} \sqrt{1 + 2\Theta(2, 2) - \text{trace}(\Theta)}, \\ e^3 &= \pm \frac{1}{2} \sqrt{1 + 2\Theta(3, 3) - \text{trace}(\Theta)}. \end{aligned} \tag{3.53}$$

Equations (3.52) might be used to compute the three parameters e^1, e^2, e^3 once the sign of e^4 has been selected. However, a better numeric approach is to utilize the magnitudes given by Eqs. (3.53) together with sign information from the numerators of Eqs. (3.52). The reason for this choice is that when e^4 approaches zero, the rotation matrix Θ approaches symmetry and, in that situation, numerical cancellation effects can occur in the numerators of Eqs. (3.52). It should also be noted that when $e^4 \rightarrow 0$, then $\text{trace}(\Theta) \rightarrow -1$ might result, and nearly “zero divided by zero” situations may occur in Eqs. (3.52) if the divisions are attempted.

Case 2: In this case, $|e^4| \leq \varepsilon$.

When $|e^4|$ is determined to be effectively zero numerically according to the threshold ε , then we have

$$\begin{aligned} e^2 e^3 &= \frac{1}{4} [\Theta(2, 3) + \Theta(3, 2)], \\ e^3 e^1 &= \frac{1}{4} [\Theta(3, 1) + \Theta(1, 3)], \\ e^1 e^2 &= \frac{1}{4} [\Theta(1, 2) + \Theta(2, 1)]. \end{aligned} \tag{3.54}$$

Also,

$$\begin{aligned} e^1 &= \pm \sqrt{\frac{1}{2} [\Theta(1, 1) + 1]}, \\ e^2 &= \pm \sqrt{\frac{1}{2} [\Theta(2, 2) + 1]}, \\ e^3 &= \pm \sqrt{\frac{1}{2} [\Theta(3, 3) + 1]}. \end{aligned} \tag{3.55}$$

Because $(e^1)^2 + (e^2)^2 + (e^3)^2 + (e^4)^2 = 1$, with $|e^4|$ being numerically small, at least one of the parameters e^1, e^2 , or e^3 must be nonzero. From Eqs. (3.55), we can select the largest numeric value and arbitrarily assign it a plus or minus sign, and then we can utilize Eqs. (3.54) to compute the remaining two Euler-Rodrigues parameters.

² The IMP software system, for example, uses a default value of $\varepsilon = 10^{-6}$ for this threshold.

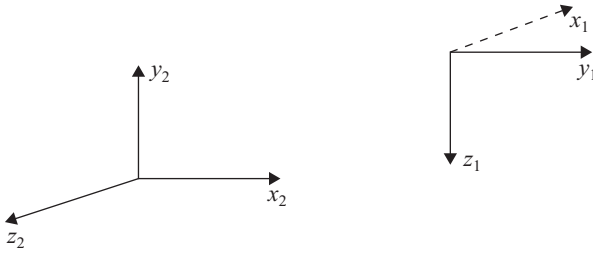


Figure 3.17. Example 3.2.

Thus, in this case, when the rotation matrix Θ is essentially symmetric, it is convenient to use only the row and column of Θ corresponding to the largest diagonal element.

EXAMPLE 3.2 As an illustrative example, consider the coordinate frames shown in [Figure 3.17](#). In this case, the $x_1y_1z_1$ frame is oriented such that the x_1 axis is precisely opposite to positive z_2 , the z_1 axis is opposite to positive y_2 , and the y_1 axis completes the right-handed coordinate system by aligning itself parallel to x_2 . We wish to find the corresponding Euler-Rodrigues parameters for this change of posture.

By inspection, the rotation matrix is:

$$\begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}.$$

From this (3×3) rotation matrix and Eq. (3.51), we find

$$e^4 = \pm \frac{1}{2} \sqrt{\text{trace}(\Theta) + 1} = \pm \frac{1}{2} \sqrt{1} = +\frac{1}{2},$$

where we have arbitrarily selected the positive sign. We note that this is Case 1.

Next, Eqs. (3.53) give:

$$e^1 = \pm \frac{1}{2}, \quad e^2 = \pm \frac{1}{2}, \quad e^3 = \pm \frac{1}{2},$$

and using the signs of the numerators of Eqs. (3.52) gives:

$$e^1 = +\frac{1}{2}, \quad e^2 = +\frac{1}{2}, \quad e^3 = -\frac{1}{2}.$$

Thus, we have both the magnitudes and the signs of the four Euler-Rodrigues parameters:

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \\ e^4 \end{Bmatrix} = \begin{Bmatrix} 0.500 \\ 0.500 \\ -0.500 \\ 0.500 \end{Bmatrix}.$$

Because $e^4 = \cos(\theta/2) = +\frac{1}{2}$, we see that the angle of rotation is $\theta = 120^\circ$ and the unit vector along the axis of rotation to achieve the orientation of the

$x_1y_1z_1$ frame is

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \end{Bmatrix} = \begin{Bmatrix} 0.500 \\ 0.500 \\ -0.500 \end{Bmatrix} = \begin{Bmatrix} w^x \\ w^y \\ w^z \end{Bmatrix} \sin(\theta/2),$$

which shows that

$$\begin{Bmatrix} w^x \\ w^y \\ w^z \end{Bmatrix} = \frac{1}{\sin 60^\circ} \begin{Bmatrix} 0.500 \\ 0.500 \\ -0.500 \end{Bmatrix} = \begin{Bmatrix} 0.577 \\ 0.577 \\ -0.577 \end{Bmatrix}.$$

EXAMPLE 3.3 As another example, we wish to find the Euler-Rodrigues parameters for the relative orientations of the two coordinate frames shown in [Figure 3.18](#).

In this case, the rotation matrix is:

$$\begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix},$$

from which we find

$$e^4 = \pm \frac{1}{2} \sqrt{\text{trace}(\Theta) + 1} = 0.$$

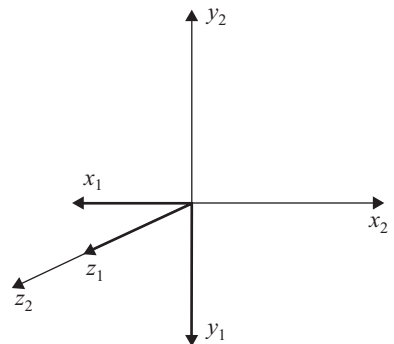
Note that this is Case 2. Therefore, from Eqs. (3.55):

$$e^1 = 0, \quad e^2 = 0, \quad e^3 = \pm 1.$$

We select $e^3 = +1$. Then, Eqs. (3.54) give $e^1 = 0$ and $e^2 = 0$. Thus, we now have all four Euler-Rodrigues parameters,

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \\ e^4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix}.$$

Figure 3.18. Example 3.3.



Because $e^4 = \cos(\theta/2) = 0$, the angle of rotation is $\theta = 180^\circ$. The unit vector along the axis of rotation to achieve this orientation of the $x_1y_1z_1$ frame is

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \end{Bmatrix} = \begin{Bmatrix} w^x \\ w^y \\ w^z \end{Bmatrix} \sin(\theta/2) = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix},$$

which shows that

$$\begin{Bmatrix} w^x \\ w^y \\ w^z \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}.$$

3.9 Displacement of Lines

The general rigid body displacement equations, Eqs. (3.39), describes displacement of points of a moving body from its zero reference posture with respect to the fixed axes. In some applications, however, it may be more appropriate to describe displacement of lines of a moving body with respect to the fixed axes. In section 3.3, we showed that Plücker vectors provide a method for representing lines in space that is based on use of homogeneous quantities. It is interesting to investigate the form of the general rigid body displacement equations if the emphasis were on the displacement of lines represented by Plücker vectors rather than on points represented by homogeneous coordinates.

Let us assume that Plücker vectors of a line ℓ of a moving body are given by $(\sigma, \hat{\sigma})$ in the coordinate system of the moving body and by $(\Omega, \hat{\Omega})$ in the fixed coordinate system. Considering the (4×4) matrix representation for the Plücker vectors, and the definition of these vectors, we can show that the equation for the displacement of such a line is be given by:

$$\begin{bmatrix} \hat{\Omega} & \Omega \\ 0 & 0 \end{bmatrix} = T \begin{bmatrix} \hat{\sigma} & \sigma \\ 0 & 0 \end{bmatrix} T^{-1}, \quad (3.56)$$

where T is the usual homogeneous transformation matrix, namely, $T = \begin{bmatrix} \theta & d \\ 0 & 1 \end{bmatrix}$.

3.10 Quaternions

Quaternions are another mathematical notation, developed by the Irish physicist, astronomer, and mathematician, Sir William Rowan Hamilton (1805–65), to represent three-dimensional orientation and rotation [11]. Like the two-dimensional geometric algebra formed by complex numbers, quaternions form a geometric algebra but in four dimensions.

A quaternion q is represented by an expression of the form $q = q^1i + q^2j + q^3k + q^4$ where the coefficients q^i , $i = 1, \dots, 4$, are real numbers and the basis units i, j, k multiply as follows:

$$i^2 = j^2 = k^2 = -1 \quad \text{and} \quad ij = k; \quad jk = i; \quad ki = j; \quad ji = -k; \quad kj = -i; \quad ik = -j. \quad (3.57)$$

Using these rules for multiplication, it is clear that quaternion multiplication, in general, is not commutative. In other words, $\mathbf{q}\mathbf{q}'$, in general, is not equal to $\mathbf{q}'\mathbf{q}$.

The norm of a quaternion is the non-negative number defined by:

$$|\mathbf{q}| = \sqrt{(q^1)^2 + (q^2)^2 + (q^3)^2 + (q^4)^2}, \quad (3.58)$$

which is zero only when $q^1 = q^2 = q^3 = q^4 = 0$. The conjugate of a quaternion \mathbf{q} is defined as

$$\bar{\mathbf{q}} = -q^1\mathbf{i} - q^2\mathbf{j} - q^3\mathbf{k} + q^4. \quad (3.59)$$

From the last two equations, it follows that

$$\mathbf{q}\bar{\mathbf{q}} = (q^1)^2 + (q^2)^2 + (q^3)^2 + (q^4)^2 = |\mathbf{q}|^2. \quad (3.60)$$

The inverse of a quaternion is derived from Eq. (3.60) as:

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{|\mathbf{q}|^2}. \quad (3.61)$$

A unit quaternion is a quaternion with unit norm. From Eq. (3.61), it is clear that the inverse of a unit quaternion is given by its conjugate.

A quaternion can be viewed as having a vector part ($q^1\mathbf{i} + q^2\mathbf{j} + q^3\mathbf{k}$), and a scalar part (q^4). Because the first three Euler-Rodrigues parameters can be viewed as components of a vector parallel to a screw axis and the fourth is a scalar, we can assign them to the vector and scalar parts of a quaternion. The resulting quaternion is

$$\mathbf{q} = w^x \sin(\theta/2)\mathbf{i} + w^y \sin(\theta/2)\mathbf{j} + w^z \sin(\theta/2)\mathbf{k} + \cos(\theta/2). \quad (3.62)$$

We note that this is a unit quaternion because $[(w^x)^2 + (w^y)^2 + (w^z)^2]\sin^2(\theta/2) + \cos^2(\theta/2) = 1$. The unit quaternion given by (3.62) can be interpreted as representing a rotation of magnitude θ around a unit vector with direction $\mathbf{w} = w^x\mathbf{i} + w^y\mathbf{j} + w^z\mathbf{k}$.

Although quaternions provide a compact representation of rotation, they are not used in the remainder of this text. They are briefly presented here because of their historic significance and their prevalence in some current literature. In this book, however, the emphasis is on matrix methods that facilitate the application of unified computer-aided techniques for design analysis. The Euler-Rodrigues parameters, which are equivalent to a quaternion, are used for representing three-dimensional orientation or rotation in matrix form.

REFERENCES

1. G. Cardano, *Opus nouum de proportionibus numerorum, motuum, ponderum, sonorum aliarumque rerum mensurandarum. Item de aliza regula*. Basel, 1570. A schematic drawing of the Cardan joint is pictured in this manuscript. However, there is no evidence that Cardan ever constructed such a device.
2. M. Ceccarelli, "Screw axis defined by Giulio Mozzi in 1763 and early studies on helicoidal motion," *Mechanism and Machine Theory*, vol. **35**, 2000, pp. 761–70.
3. M. Chasles, "Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre, [Notes on general properties of a system of two identical bodies arbitrarily located in space; and on the finite or infinitesimal motion

- of a free solid body],” *Bulletin des Sciences Mathématiques, Astronomiques, Physiques et Chimiques de Ferrussac* [Bulletin of the Sciences of Mathematics, Astronomy, Physics, and Chemistry by Ferrussac], vol. 14, Paris, 1830, pp. 321–26.
4. J. L. Coolidge, *A History of Geometrical Methods*, Dover Publications, Inc., New York, 1963. Includes an extensive bibliography.
 5. R. Descartes, *Discours de la méthode pour bien conduire sa raison et chercher la vérité dans les sciences*, [Discourse on methods for reasoning and seeking truth in science], Leiden: Jan Maire, 1637. Cartesian coordinates were introduced in the third appendix that is titled *La Géométrie*, which focuses on the connections between geometry and algebra.
 6. L. Euler, “Theoria motus corporum solidorum seu rigidorum, [Treatise on the motion of solids or rigid bodies],” *Opera omnia II*, vol. 9, Rostock, 1765, pp. 84–98; also in “Formulae generales pro translatione quacumque corporum rigidorum, [General formulae for the motion of rigid bodies],” *Novi Comentarum Academiae Scientiarum Petropolitanae*, [New memoirs of the imperial academy of sciences in St. Petersburg], vol. 20, 1776, pp. 189–207.
 7. K. W. Feuerbach, *Grundriss zu analytischen Untersuchungen der dreieckigen Pyramide* [Foundations of the analytic theory of the triangular pyramid], Nürnberg, 1827.
 8. O. Friberg, “Computation of Euler Parameters from Multipoint Data,” *Journal of Mechanisms, Transmissions, and Automation in Design*, ASME Transactions, vol. 110, June 1988, pp. 116–121.
 9. J. D. Gergonne, *Annales de mathématique pures et appliquées* [Annals of pure and applied mathematics] better known as *Annales de Gergonne*, Neimes, 1824–27.
 10. H. Goldstein, *Classical Mechanics*, Addison-Wesley Publishing Co., Inc., Reading, MA, 2nd ed., 1980.
 11. W. R. Hamilton, *Lectures on quaternions containing a systematic statement of a new mathematical method: of which the principles were communicated in 1843 to the Royal Irish Academy and which has since formed the subject of successive courses of lectures delivered in 1848 and subsequent years, in the halls of Trinity College, Dublin: with numerous illustrative diagrams, and with some geometrical and physical applications*, Hodges and Smith, Dublin, 1853.
 12. E. A. Maxwell, *General Homogeneous Coordinates in Space of Three Dimensions*, Cambridge University Press, London, 1951. An excellent general reference, reissued in 2008.
 13. A. F. Möbius, “Der barycentrische Calcul, [Barycentric calculus],” *Crelle’s Journal für die reine und angewandte Mathematik* [Crelle’s Journal for Pure and Applied Mathematics], Leipzig, 1827.
 14. G. Mozzi, “Discorso matematico sopra il rotamento momentaneo dei corpi [Mathematical treatise on temporally revolving bodies],” Stamperia di Donato Campo, Napoli, 1763.
 15. J. Plücker, *Neue Geometrie des Raumes gegründet auf die Betrachtung der geraden Linie als Raumelement* [A new geometry of space based on consideration of the straight line as the spatial element], B. G. Teubner, Leipzig, 1868–69, pp. 1–374.
 16. O. Rodrigues. “Des lois géométriques qui régissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire [Geometric laws that govern the movement of a solid system in space, and the change of coordinates from these displacements considered independently of the causes that produce them],” *Journal de Mathématiques Pures et Appliquées* [Journal of pure and applied mathematics], Bachelier, Paris, 5, 1840, pp. 380–440, better known as *Annales de Gergonne*, described in detail in H. Cheng and K. C. Gupta, “An Historical Note on Finite Rotations,” *Journal of Applied Mechanics*, ASME Transactions, vol. 56, no. 1, 1989, pp. 139–45.
 17. J. Stuelpnagel, “On the Parametrization of the Three-Dimensional Rotation Group,” *SIAM Review*, vol. 6, no. 4. (Oct., 1964), pp. 422–30.

PROBLEMS

3.1 Prove that the four points whose homogeneous coordinates are $A(1,2,3,4)$, $B(4,3,2,1)$, $C(1,1,1,1)$, and $D(3,1,-1,-3)$ are collinear.

3.2 Two circles, each of radius r , have the equations $x^2 + y^2 = r^2$ and $(x - r)^2 + y^2 = r^2$. Find the four points of intersection between these two quadratic equations. (Hint: recast the two quadratic equations into homogeneous coordinates.)

3.3 Two of the axes of a moving body 1 are in directions described by the vectors $\mathbf{i}_1 = [0.0, 1.0, 1.0]^t$ and $\mathbf{j}_1 = [1.0, -1.0, 1.0]^t$ with respect to the fixed axes of body 2. Determine the rotation matrix Θ_{21} .

3.4 Two of the basis vectors of stationary body 2 are in directions described by $\mathbf{j}_2 = [0.0, 1.0, 1.0]^t$ and $\mathbf{k}_2 = [1.0, -1.0, 1.0]^t$ with respect to the moving axes of body 1. Determine the rotation matrix Θ_{21} .

3.5 The end effector of a robot is to be located at $[500, 150, 225]^t$ where all distances have units of millimeters. The approach vector is to be $[1, 0, 0]^t$, and the orientation vector is to be $[0, 0, 1]^t$. Find the transformation matrix for this posture.

3.6 Verify that $|T_{ik}| = +1$ for Eq. (3.26).

3.7 Measured data in millimeters for the position coordinates of three points of a moving body are known such that:

$$\mathbf{r}_1 = \begin{bmatrix} 50 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_3 = \begin{bmatrix} 0 \\ 125 \\ 0 \\ 1 \end{bmatrix},$$

$$\mathbf{R}_1(t_1) = \begin{bmatrix} 37.325 \\ -98.175 \\ 132.045 \\ 1 \end{bmatrix}, \quad \mathbf{R}_2(t_1) = \begin{bmatrix} 71.800 \\ -118.925 \\ 161.725 \\ 1 \end{bmatrix}, \quad \mathbf{R}_3(t_1) = \begin{bmatrix} 152.450 \\ -28.425 \\ 131.225 \\ 1 \end{bmatrix}.$$

Find the (4×4) homogeneous transformation matrix for this displacement.

3.8 Verify by direct computation that $|\Theta_{41}| = +1$ for Eq. (3.44).

3.9 Find the precession, nutation, and spin Euler angles that yield the rotation matrix of problem 3.3.

3.10 Determine the roll, pitch, and yaw Euler angles that yield the rotation matrix of problem 3.4.

3.11 A rigid body rotates by 60° about an axis defined by the vector $[3, 4, 0]^t$. Find the corresponding rotation matrix.

3.12 Determine the Euler-Rodrigues parameters for the displacement matrix of problem 3.3.

3.13 Show that, when components of Plücker vectors of lines are used in the form of the (4×4) matrix as discussed in this chapter, the displacement of lines is given by a similarity transformation. In other words, using the notation of this chapter prove that:

$$\begin{bmatrix} \widehat{\Omega} & \Omega \\ 0 & 0 \end{bmatrix} = T \begin{bmatrix} \widehat{\sigma} & \sigma \\ 0 & 0 \end{bmatrix} T^{-1}.$$

3.14 Consider two lines whose Plücker vectors are $(\widehat{\Omega}_i, \Omega_i); i = 1, 2$. Show that if these two lines are parallel, then $\widehat{\Omega}_1 \widehat{\Omega}_2 = 0$ and if they are perpendicular, then $\widehat{\Omega}_1 \cdot \widehat{\Omega}_2 = 0$.

3.15 Consider a rigid body b with a body coordinate system x_b, y_b, z_b . The orientation of this body with respect to the coordinate system x_c, y_c, z_c of another body c is specified in terms of three direction cosines derivable from the three angles: $\angle(x_b, x_c) = 45^\circ$, $\angle(x_b, y_c) = 60^\circ$, $\angle(x_b, z_c) = 45^\circ$. Show that these three independent direction cosines lead to multiple possible orientations for body b with respect to body c . This problem illustrates an ambiguity inherent in representation of three-dimensional orientation of a rigid body when only three independent direction cosines are specified.

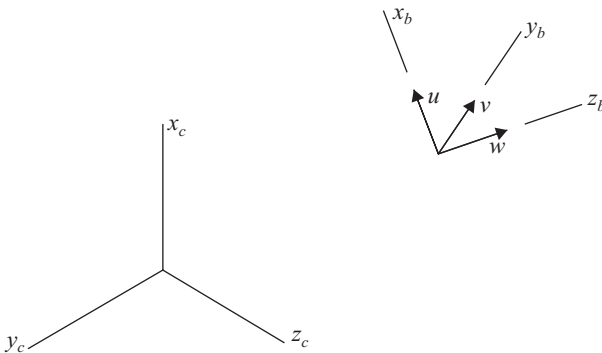


Figure P3.15

3.16 In order to standardize characterization of the kinematics of the human shoulder, the International Society of Biomechanics (ISB) has recommended the use of the rotation sequence consisting of external rotation (E), followed by upward rotation (U), then followed by posterior tilting (P) as shown in the figure for describing scapular kinematics of the shoulder joint. Show that this rotation sequence results in a different final orientation compared to a sequence consisting of $(P)(U)(E)$. Calculate the norm of the difference and the difference between the two norms of the two orientations, if the mean angles are as follows:

posterior tilting $P = 30^\circ$, upward rotation $U = 60^\circ$, and external rotation $E = 90^\circ$.

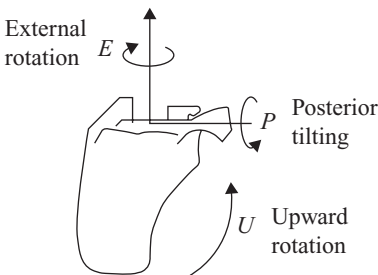


Figure P3.16. Human shoulder model.

3.17 Consider a robot end-effector with two coordinate systems attached to it as shown in Figure P3.17a. One coordinate system is attached to the end-effector with

its origin at the wrist center point and the other is attached to the tip of the end-effector. The kinematic structure of the wrist is a spherical linkage and is illustrated in Figure P3.17b.

Show that if we use the coordinate system attached to the end-effector at the wrist center point, the order in which we perform the roll, pitch, and yaw rotations is irrelevant; however, if we use the coordinate system attached to the end-effector at its tip, then the order does make a difference unless we are only concerned with differential or instantaneous rotations.

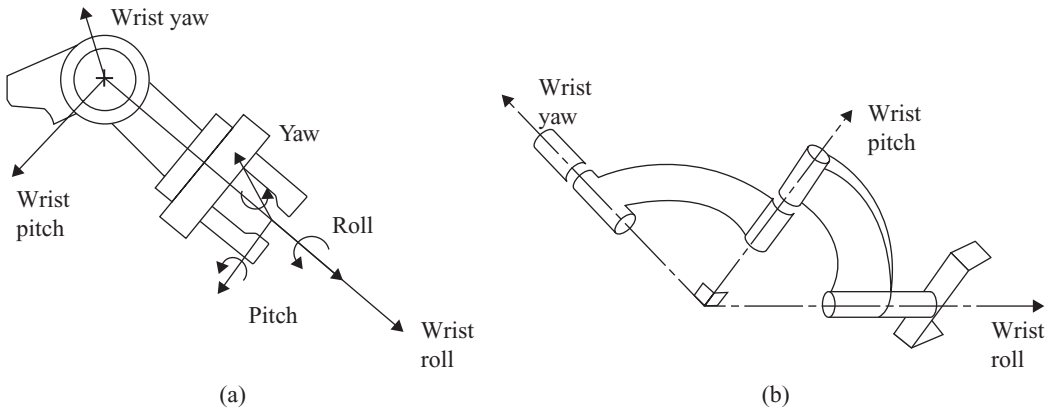
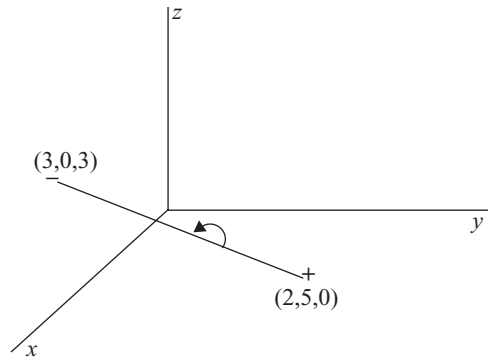


Figure P3.17 (a) (b)

3.18 In an industrial application, a part is to rotate 30° in the counterclockwise direction about the rod shown in Figure P3.18 and move left along the rod by eight inches. Determine the new coordinates of a point whose original coordinates were (x, y, z).

Figure P3.18.



3.19 The rotation matrix as given by Eq. (3.37) shows that the rotation angle can be computed from:

$$\theta = \tan^{-1} \left\{ \left\| \begin{array}{l} \Theta(3, 2) - \Theta(2, 3) \\ \Theta(1, 3) - \Theta(3, 1) \\ \Theta(2, 1) - \Theta(1, 1) \end{array} \right\|, (\text{trace}\Theta - 1) \right\}.$$

4

Modeling Mechanisms and Multibody Systems with Transformation Matrices

4.1 Introduction

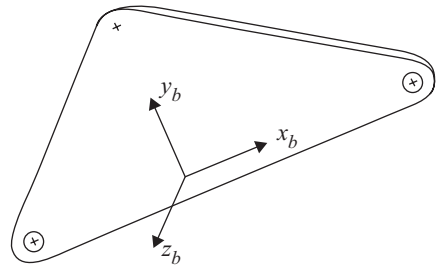
A mechanism or a multibody system consists of several bodies or links that move together in a coordinated fashion based on the nature of the connections between them. The individual bodies or links are usually attached through joints such as in robot manipulators, biomechanical systems, mechanisms and machines, or other clever devices such as in aerospace systems. As a system moves, its posture changes, including displacements of the individual bodies while maintaining the connections through the joints.

The classical formulations of kinematics of rigid bodies discussed in Chapter 3 can be adapted to multibody systems. In order to do this, however, we must keep track of all bodies and their interconnections and make sure that their displacements and motions are described in a fashion that allows us to track the posture of the entire mechanism or multibody system. The matrix method presented in this and subsequent chapters provides a systematic method that allows such a development with no ambiguities. When combined with the methods for topological examination of mechanical systems from Chapter 2, the overall approach provides a powerful tool for computer-aided analysis of mechanisms and multibody systems and for development of general-purpose software tools for such applications.

Now that we are familiar with some of the methods of algebraic geometry and kinematics of rigid bodies, we are ready to start defining a model for our mechanism or multibody system. Of course, our spatial model must start with the definition of proper coordinate systems. We do this in this chapter, where we carefully locate coordinate systems on every link and before and after every joint. We also define appropriate transformation matrices between these coordinate systems. These transformations introduce the required geometric parameters of the bodies and also define the constraints and the motion variables of the joints, which are essential for the motion analysis methods that follow.

4.2 Body Coordinate Systems

Consider the problem of describing the shape of a rigid part of a mechanism or multibody system as pictured in [Figure 4.1](#). From our topological analysis in Chapter 2, we have already assumed that each body has an identifying label b , ($b = 1, 2, \dots, \ell$);

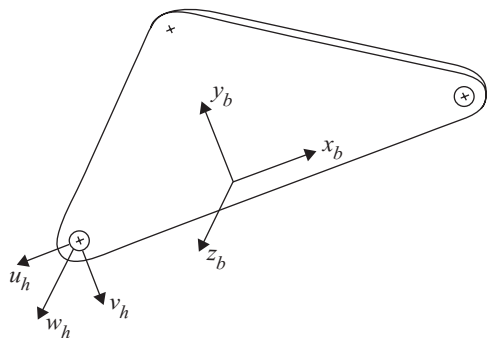
Figure 4.1. Body coordinate system.

we now find it convenient to use these same identifying labels here. To allow a precise definition of what we mean by “shape,” we assume that a right-hand Cartesian coordinate system x_b, y_b, z_b is chosen for each body of our system and is rigidly attached to that body at some convenient, but arbitrary, posture.

The posture of a body is described by the transformation matrix describing the posture of its body coordinate system with respect to some overall fixed global or world coordinate system. Once we find the posture of the body’s coordinate system, we know all that is necessary to define the location and orientation of the body and all its features. How to do this will be shown later; at this time, we only note that such a coordinate system must be defined for each body, and that it is the primary coordinate system for locating other items such as points or lines attached to that body. We note that this coordinate system can also be used as the reference for specifying the shape of the body, the joint element postures, the center of mass location, the mass moments of inertia, and so on, and that, as long as bodies are considered rigid, each such geometric feature remains constant in its own body coordinate system as the system moves.

4.3 Joint and Auxiliary Coordinate Systems

It was pointed out in section 1.4 that the primary function that a rigid body serves is to ensure that the relative postures of its joint elements (and other geometric features) do not change; that is, the purpose of a machine part is to hold its joint elements and other features in constant geometric relationships. To reflect this, we define another right-hand Cartesian coordinate system at each of the joint elements, defined to be aligned conveniently with the natural motion axes of that joint. Such an auxiliary coordinate system may also be convenient for locating an important shape feature of a body such as a hole or a keyway, as shown in [Figure 4.2](#), or for modeling a force or torque applied to the body.

Figure 4.2. Joint coordinate system attached to a body.

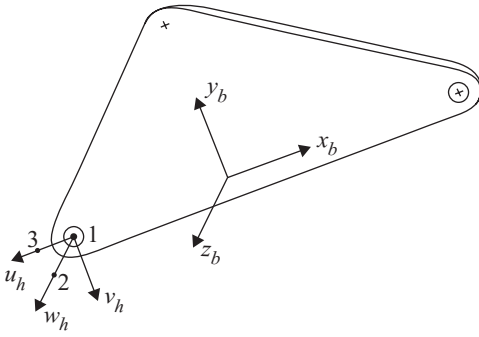


Figure 4.3. Specifying data to locate a coordinate system.

More will be said in later sections about how joint coordinate systems are chosen to locate and orient each type of joint element; however, we need not be concerned with such conventions at present. Here we only note the need to develop a technique for specifying data for the initial posture of each joint or auxiliary coordinate system and storing this data in a form that can be conveniently used in later parts of the analysis.

4.4 Specifying Data for a Coordinate System

Information about the shapes of the bodies of a mechanism or multibody system is usually available in the form of xyz Cartesian coordinate data for certain key points or features. This comes about because, at least in early design stages, a sketch or layout drawing is usually the primary source of geometric data for a mechanism or multibody system. In a biomechanical system, such data may be provided by measurement of a predefined set of target points on the system under study for which detailed imaging data can be obtained. At other times, such as in the analysis of an existing machine, perhaps the detailed drawings or computer-aided design (CAD) data files for individual parts may provide a more convenient source of dimensional data. In any case, it is usually quite easy to obtain xyz coordinate data for strategic points of the bodies.

Here we assume that data is specified in a convenient *measurement coordinate system* whose axes are labeled xyz and that we wish to locate another coordinate system whose axes are labeled uvw . We want to establish a procedure for determining the posture of uvw with respect to xyz ; that is, we wish to find data for the transformation matrix of the equation

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} T(1,1) & T(1,2) & T(1,3) & T(1,4) \\ T(2,1) & T(2,2) & T(2,3) & T(2,4) \\ T(3,1) & T(3,2) & T(3,3) & T(3,4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}, \quad (4.1)$$

where the symbolism $T(i, j)$ refers to the element in row i , column j , of the transformation matrix T .

To establish a general procedure, we require that the following data be specified, all measured along the chosen set of xyz measurement axes:

- (a) Coordinates $x_1y_1z_1$ of the origin of the uvw coordinate system. This is shown as the point labeled 1 in [Figure 4.3](#).

- (b) Coordinates $x_2y_2z_2$ of a point lying on the positive w axis, such as point 2 in Figure 4.3.
- (c) Coordinates $x_3y_3z_3$ of a point lying on the positive u axis, such as point 3 in Figure 4.3.

This measurement coordinate system is chosen for convenient measurement of all data for this body. It need not be the same as that used for specifying data for other bodies because, except for the base (frame), its location is not kept for later analysis.

From the definition of the first of these data points (point 1) we set $u = v = w = 0$, $x = x_1$, $y = y_1$, and $z = z_1$ in Eq. (4.1). Doing this shows that $T(1, 4) = x_1$, $T(2, 4) = y_1$, and $T(3, 4) = z_1$ as we should expect. Procedurally, this means that the data for the first point can be entered directly into the fourth column of the transformation matrix without change.

We next use the difference between $x_2y_2z_2$ and $x_1y_1z_1$ and their definitions to show that

$$\begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \\ 0 \end{bmatrix} = \begin{bmatrix} T(1, 1) & T(1, 2) & T(1, 3) & x_1 \\ T(2, 1) & T(2, 2) & T(2, 3) & y_1 \\ T(3, 1) & T(3, 2) & T(3, 3) & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ w_2 \\ 0 \end{bmatrix} = w_2 \begin{bmatrix} T(1, 3) \\ T(2, 3) \\ T(3, 3) \\ 0 \end{bmatrix}.$$

In this matrix, the elements of the third column describe a unit vector along the w axis, *measured in the coordinate system x, y, z* . Therefore,

$$\begin{aligned} w_2 &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \\ w^x &= T(1, 3) = (x_2 - x_1)/w_2, \\ w^y &= T(2, 3) = (y_2 - y_1)/w_2, \\ w^z &= T(3, 3) = (z_2 - z_1)/w_2. \end{aligned}$$

Procedurally, this shows that we may fill the third column of our matrix with the differences between the data for the second and first points, and that we should then normalize this column to form a unit vector. This should be no surprise because the third column of the transformation denotes the direction for the unit vector along the w axis as measured in the xyz system.

In a similar fashion, we use the difference between points $x_3y_3z_3$ and $x_1y_1z_1$ and their definitions (see Figure 4.3) to show that

$$\begin{aligned} u_3 &= \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}, \\ u^x &= T(1, 1) = (x_3 - x_1)/u_3, \\ u^y &= T(2, 1) = (y_3 - y_1)/u_3, \\ u^z &= T(3, 1) = (z_3 - z_1)/u_3. \end{aligned}$$

Once the first and third columns of the matrix in Eq. (4.1), namely the \mathbf{u} and \mathbf{w} unit vectors, are found and normalized, we can find the entries of the second column, unit vector \mathbf{v} , by taking the vector cross product $\mathbf{v} = \mathbf{w} \times \mathbf{u}$ between unit vectors

along the w axis (column 3) and the u axis (column 1) to form a unit vector along the v axis (column 2). Thus, we form

$$v^x = T(1, 2) = T(2, 3)T(3, 1) - T(3, 3)T(2, 1),$$

$$v^y = T(2, 2) = T(3, 3)T(1, 1) - T(1, 3)T(3, 1),$$

$$v^z = T(3, 2) = T(1, 3)T(2, 1) - T(2, 3)T(1, 1).$$

This completes entries for the entire transformation matrix. However, now it is wise to question the quality of the result in a situation where inaccurate data might be encountered. The previous procedure ensures that the origin of the uvw system is precisely located to match the data for point 1. Similarly, the u and w axis directions will exactly match those implied by the data points given. In addition, even though the data points are not required to be separated by unit distances, columns 3 and 1 have been normalized to represent unit vectors for the w and u axes.

So where might faulty data cause a problem? First, if the data for points 2 and 1 show them to be coincident, or nearly so, then the value computed for w_2 yields a zero and the procedure must terminate (or produce a division by zero during normalization). Similarly, the value of u_3 must be tested for zero to protect against nearly coincident data for points 3 and 1.

Second, if either w_2 or u_3 is smaller than some acceptable tolerance value, the orientation of the w axis or u axis is of questionable accuracy. This also shows why it is unwise to choose a measurement coordinate system x, y, z that is located very distant from the points for which data are measured. Because a computer has finite precision, it sacrifices accuracy (requiring small differences between large values) to choose the global coordinate system of a car, for example, when specifying data for the internal workings of the glove compartment lock.

Third, there is the possibility that the data for all three points prove them to be collinear, or nearly so. This would result in all entries of the second column of the matrix being unacceptably small or perhaps even zero. This must also be tested and treated as an error in the data given.

Finally, it is likely that the three given data points, even though not collinear, are probably *not* situated to form an exact right angle for points 3-1-2. In this case, the u and w axes each become unit vectors, but are not exactly perpendicular to each other. To guard against this, the second column of the final matrix is also normalized after it is found. By normalization of the second column, we guarantee that the v axis is a unit vector perpendicular to both the u and w axes. We then recalculate an *adjusted* u axis that is perpendicular to both v and w by forming another vector cross product *overwriting* the data for the u vector previously stored in column 1:

$$u^x = T(1, 1) = T(2, 2)T(3, 3) - T(3, 2)T(2, 3),$$

$$u^y = T(2, 1) = T(3, 2)T(1, 3) - T(1, 2)T(3, 3),$$

$$u^z = T(3, 1) = T(1, 2)T(2, 3) - T(2, 2)T(1, 3).$$

In effect, this last step revises the original convention, stated as (c) near the start of this section. To account for possible inaccuracy, the convention for choosing the data for point 3 is now restated as follows:

(c) The coordinates $x_3y_3z_3$ of a point lying in the half-plane defined by the w axis and the positive u axis such as point 3 shown in [Figure 4.3](#).

4.5 Modeling Dimensional Characteristics of a Body

Let us now review our progress so far. In Chapter 2, we learned how a computer program can be written to accept information on the kinematic architecture of a mechanism or multibody system, and to discover any kinematic loops and the paths from the fixed frame to each and every other body. In finding these, we assigned identifying numbers (labels) to every body and to every joint.

We now assume that there exists an agreed upon *absolute* Cartesian coordinate system $x_0y_0z_0$ that is assumed *stationary* and becomes the primary coordinate system in which results are expected. In most applications, this coordinate system is referred to as the *base* or *world* or *global* coordinate system.

In section 4.2, we noted that each body of a mechanism or multibody system carries a *body coordinate system* $x_by_bz_b$, where b is the identifying label of the body. Data for the initial posture of each of the body coordinate systems may be supplied by the methods shown in section 4.4. For the initial position of a point of body b , we have

$$R_b = T_{0b}r_b, \quad b = 1, 2, \dots, \ell, \quad (4.2)$$

where r_b shows the homogeneous coordinates of a point attached to body b , measured with respect to $x_by_bz_b$, and R_b shows the homogeneous coordinates of the same point with respect to the absolute system, $x_0y_0z_0$. Thus, for each part of the mechanism or multibody system, we assume that data is supplied as described by which we find initial numeric values for each of the (4×4) transformation matrices T_{0b} for the body coordinate frames with respect to the fixed frame. This assumes that it is convenient to gather the initial data in the global coordinate system, as from a layout drawing. These data, however, are all given for only one posture of the mechanism or multibody system, and change when the system moves.

In section 4.3, we saw that there is also need for several *joint* and *auxiliary* coordinate systems attached to the various bodies, to define joint element postures, for example. Data for the initial postures of these may also be supplied in the same way. If we consider a point of the joint or auxiliary coordinate system $u_hv_hw_h$, then we have:

$$R_h = T_{0h}r_h, \quad h = 1, 2, \dots, n, \quad (4.3)$$

where subscript h identifies the label of a joint or auxiliary coordinate system, r_h shows the homogeneous coordinates of the point measured in coordinate system $u_hv_hw_h$, and the (4×4) homogeneous transformation matrix T_{0h} represents the posture of joint coordinate frame $u_hv_hw_h$ with respect to the global reference frame $x_0y_0z_0$; (see [Figure 4.4](#)).

For the case of joint or auxiliary coordinate systems, however, we do *not* choose to store the T_{0h} data as such. By setting Eq. (4.2) equal to Eq. (4.3) we find

$$T_{0b}r_b = T_{0h}r_h.$$

Rearranging this, we find

$$r_b = S_{bh}r_h, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (4.4)$$

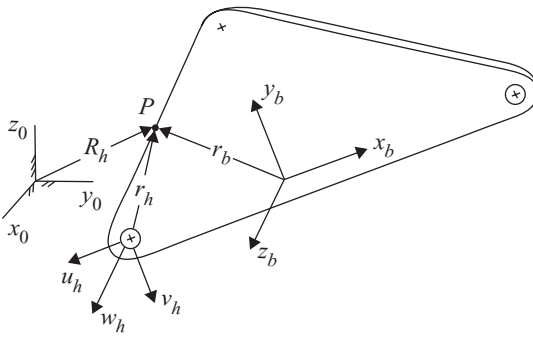


Figure 4.4. Geometry of body and joint coordinates.

where

$$S_{bh} = T_{0b}^{-1} T_{0h}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n. \end{array} \quad (4.5)$$

This S_{bh} matrix is called the *shape matrix* for joint or auxiliary coordinate system h of body b . This shape matrix is the data that we store for later calculations because this matrix remains constant as the system moves. Keeping the shape matrix constant during the motion is our technique for enforcing the rigid body assumption for body b ; this ensures that each joint element and each auxiliary coordinate system on body b retains a constant geometric posture with respect to its own body coordinate system.

If we prefer to enter data for a body from a detailed drawing or a digital image of that body alone, comparing Eq. (4.1) and Eq. (4.4) shows that the procedure of section 4.4 can be used to find the S_{bh} matrix directly, by using the body b coordinate system as the measurement coordinate system. If not, we can collect the T_{0h} data for the joint or auxiliary coordinate systems with respect to a convenient measurement reference frame from the layout drawing or digital image as shown in Eq. (4.3). Then, once the body coordinate systems have been located and the T_{0b} data are complete, Eq. (4.5) can be used to find the shape matrices.

The “measurement” coordinate system used for data collection need not necessarily match the global or world coordinate system used during the final simulation of the mechanism or multibody system. We need only be consistent until each shape matrix is found. Because this is independent of the coordinate system in which the data was gathered for finding it, we can use a different measurement coordinate system for finding another shape matrix, or in later analysis.

By one of these procedures, we assume that a shape matrix S_{bh} is found for the posture of each joint element and each auxiliary coordinate system with respect to its body axes before any further analysis is attempted for the system. These form the primary dimensional data for our computational model of the mechanism or multibody system.

Once we have coordinate data for any chosen point of the system with respect to its body axes, or with respect to one of the joint or auxiliary coordinate systems, from which we can use Eq. (4.4) to find its position with respect to its body axes, we can then use Eq. (4.2) to find its absolute global position. As the system moves, the T_{0b} matrices change, showing the movement of each body; these same T_{0b} matrices

may then be used to find the changed global positions of all points attached to body b .

We also note that very detailed shape models of the bodies might be used, perhaps finite element models for stress analysis, or solid models for animated picture generation or for interference detection between the bodies. In each case, no matter how complex the shape model, each mechanical part's data can be stored unambiguously with respect to its body coordinate system, and remains constant with respect to that coordinate system as the system moves (as long as we accept the rigid body assumption for part shapes). All movement of a mechanical body in a mechanism or multibody system is simulated by modifying the T_{0b} matrix for the body, and this *implicitly* changes the global positions of all points of that body model simultaneously. Thus, we have a very general and widely applicable procedure for measuring and storing all critical dimensional parameters of the moving bodies of a mechanism or multibody system.

4.6 Modeling Joint Characteristics

The descriptions of the body shapes, however, do not tell the whole story. As pointed out in Chapter 1, the primary purpose of a mechanism or multibody system is to transform *motion*, and this is done by virtue of movement within the joints. Because the mechanism or multibody system consists of a collection of rigid bodies, as is assumed in this text, the shape matrices and local coordinate data for all points of interest on the parts are treated as constants. We must now provide a means for characterizing the *motions* allowed in the mechanism or multibody system; that is, we must find a convenient matrix description for the relative motions allowed and the constraints provided by the *joints* of the mechanism or multibody system.

In a mechanism or multibody system, joints are connections between adjacent bodies. A joint, therefore, has mating elements on the two adjacent bodies that it connects. The motion(s) taking place within a joint are fully described if we write equations for the relative motion between that joint's two elemental surfaces. We let each of the two mating joint elements carry one of the Cartesian joint coordinate systems discussed in section 4.3. These are the same joint coordinate systems that were used for finding the shape matrices defined in section 4.5. We have assumed for each joint h that a joint coordinate system $u_h v_h w_h$ is rigidly attached to the "preceding" joint element and another $u'_h v'_h w'_h$ is rigidly attached to the "following" joint element, where the "preceding" and "following" elements are distinguished by the orientation defined for the joint as explained in section 2.2. We see here the necessity of defining a joint such that it connects two and *only two* joint elements. These coordinate axes for a helical joint are shown, as an example, in [Figure 4.5](#).

Because the transformation matrix of Eq. (3.24) can be adapted to describe the relative posture of *any* two coordinate systems, we can formulate such an equation by relating coordinate system $u'_h v'_h w'_h$ to $u_h v_h w_h$ and it will result in a transformation matrix. If we give this new transformation matrix the symbol Φ_h the transformation for joint h will be of the form

$$r_h = \Phi_h(\phi_h)r'_h, \quad h = 1, 2, \dots, n, \quad (4.6)$$

where the relative posture is parameterized in terms of a vector ϕ_h that contains the variables associated with the relative motion(s) within the joint. Such variables are called *joint variables* and the total number of joint variables in a joint corresponds to the number of relative degrees of freedom of the joint. Because a joint is movable, the elements of the transformation $\Phi_h(\phi_h)$ change value depending on the posture of the joint. The remaining task, therefore, is to derive the form of the joint transformation matrix $\Phi_h(\phi_h)$ for each type of joint that we expect to encounter in our analysis. In fact, in each class of multibody system, although we may encounter a large number of shapes for the bodies, the types of joints encountered are typically more limited. This indicates that deriving and cataloging the form of the joint transformation matrices for a class of multibody systems can provide a basis for developing a framework for a general-purpose computer code for analysis of such systems. Here we consider the class of joint transformation matrices for the mechanical link works described in Chapter 1.

Note that only two major limitations have been placed on our methods so far. We have assumed that the system to be analyzed (1) consists entirely of *rigid* bodies, and that (2) all of its joint types fall among those for which an *explicit* form of the $\Phi_h(\phi_h)$ matrix can be written and programmed for digital computation.

We observed in Chapter 1 that the helical joint is the parent of both the revolute and the prismatic joints, and that the remaining lower pairs may be simulated as kinematically equivalent combinations of these. Aside from the fact that helical, revolute, and prismatic joints each have a single degree of freedom, the other feature common to these three is that each has an easily identified axis for its relative motion. These *joint axes* provide a convenient starting place for the mathematical description of joint motions because the choice of the variables for these joint types can be based on the relative posture of the joint axes.

4.6.1 Helical Joint

A helical or screw joint is shown in Figure 4.5 where it can be seen that the relative motion consists of a rotation about and a translation along its helical axis. This joint,

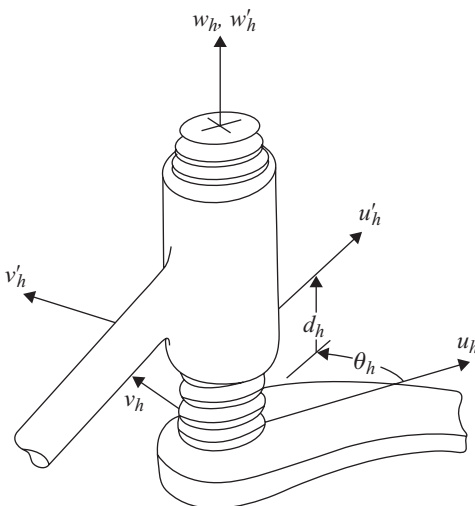


Figure 4.5. Coordinate systems for a helical joint.

however, has only a single relative degree of freedom because its rotational and translational motions are related by the pitch of the helix.

We impose the following conventions on the choice of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ axes for a helical (screw) joint:

1. The w_h and w'_h axes must be chosen along the joint motion axis; they must be collinear and must have the same positive sense.
2. The u_h and u'_h axes must be chosen such that they coincide at the position where the joint variable is zero.

The translational motion in a helical joint is designated d_h and is the distance from u_h to u'_h measured along positive w_h, w'_h . The accompanying rotational motion is designated θ_h , the angle from positive u_h to positive u'_h , and is measured counter-clockwise about positive w_h, w'_h .

There are two possible choices for the joint variable of a helical joint. Only one joint variable is permissible, however, because a helical joint only exhibits one degree of freedom in relative motion, and the two variables d_h and θ_h are related by the pitch σ_h of the screw:

$$\phi_h = d_h = \sigma_h \theta_h. \quad (4.7)$$

We define the sliding distance d_h to be the joint variable ϕ_h because this leaves no ambiguity when the rotation θ_h extends beyond a full revolution.

If we now consider the $u_h v_h w_h$ system as fixed and $u'_h v'_h w'_h$ as a moving coordinate system, we see (using the notation of section 3.7) that the w_h, w'_h axis is the helical axis for which we can write: $\mathbf{P} = \mathbf{0}$ and $\mathbf{w} = [0 \ 0 \ 1]^t$. The screw parameters are $d = \phi_h$ and $\theta = \phi_h/\sigma_h$. The elements of the (4×4) transformation matrix for the helical joint can now be found using Eqs. (3.40) and (3.41) as follows:

$$\begin{aligned} \Theta(1, 1) &= [(w^x)^2 - 1](1 - \cos \theta) + 1 = [0 - 1](1 - \cos \theta_h) + 1 = \cos(\phi_h/\sigma_h), \\ \Theta(1, 2) &= w^y w^x (1 - \cos \theta) - w^z \sin \theta = 0 - 1 \sin \theta_h = -\sin \theta_h = -\sin(\phi_h/\sigma_h), \\ \Theta(1, 3) &= w^z w^x (1 - \cos \theta) + w^y \sin \theta = 0 + 0 = 0, \\ \Theta(2, 1) &= w^x w^y (1 - \cos \theta) + w^z \sin \theta = 0 + 1 \sin \theta_h = \sin(\phi_h/\sigma_h), \\ \Theta(2, 2) &= [(w^y)^2 - 1](1 - \cos \theta) + 1 = [0 - 1](1 - \cos \theta_h) + 1 = \cos(\phi_h/\sigma_h), \\ \Theta(2, 3) &= w^z w^y (1 - \cos \theta) - w^x \sin \theta = 0 - 0 = 0, \\ \Theta(3, 1) &= w^x w^z (1 - \cos \theta) - w^y \sin \theta = 0 - 0 = 0, \\ \Theta(3, 2) &= w^y w^z (1 - \cos \theta) + w^x \sin \theta = 0 + 0 = 0, \\ \Theta(3, 3) &= [(w^z)^2 - 1](1 - \cos \theta) + 1 = [1 - 1](1 - \cos \theta_h) + 1 = 1, \end{aligned}$$

and

$$\begin{aligned} d^x &= \phi_j w^x - [\Theta(1, 1) - 1] P^x - \Theta(1, 2) P^y - \Theta(1, 3) P^z = \phi_h 0 - 0 - 0 - 0 = 0, \\ d^y &= \phi_j w^y - \Theta(2, 1) P^x - [\Theta(2, 2) - 1] P^y - \Theta(2, 3) P^z = \phi_h 0 - 0 - 0 - 0 = 0, \\ d^z &= \phi_j w^z - \Theta(3, 1) P^x - \Theta(3, 2) P^y - [\Theta(3, 3) - 1] P^z = \phi_h 1 - 0 - 0 - 0 = \phi_h. \end{aligned}$$

Therefore, the transformation matrix for a helical joint becomes:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos(\phi_h/\sigma_h) & -\sin(\phi_h/\sigma_h) & 0 & 0 \\ \sin(\phi_h/\sigma_h) & \cos(\phi_h/\sigma_h) & 0 & 0 \\ 0 & 0 & 1 & \phi_h \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.8)$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the value of the joint variable may be found from

$$\phi_h = \Phi_h(3, 4), \quad (4.9)$$

where $\Phi_h(3, 4)$ refers to row 3, column 4 of the matrix Φ_h .

4.6.2 Revolute Joint

A revolute joint is shown in [Figure 4.6](#).

The following conventions are imposed on the choice of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ axes for a revolute joint:

1. The w_h and w'_h axes must be chosen along the joint motion axis; they must be collinear and must have the same positive sense.
2. The origins of the two coordinate systems must be coincident.

These conventions are defined such that, in the case of a two-dimensional application, the motion lies in the common $u_h v_h$ and $u'_h v'_h$ planes.

The joint variable for a revolute joint is designated ϕ_h and is the counterclockwise angle measured *from* positive u_h to positive u'_h about the positive w_h, w'_h axes.

The transformation matrix $\Phi_h(\phi_h)$ is easily found by restricting Eq. (4.8) according to the conditions stated. Note that for a revolute joint with these joint coordinate systems, the rotation about the joint axis is $\theta = \phi_h$ and the translation d along the joint axis is zero. The resulting transformation matrix is:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos \phi_h & -\sin \phi_h & 0 & 0 \\ \sin \phi_h & \cos \phi_h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.10)$$

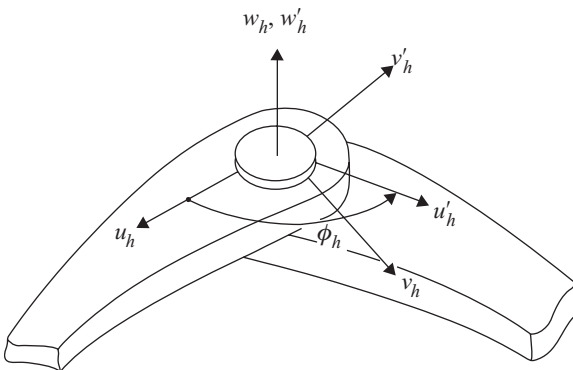


Figure 4.6. Coordinate systems for a revolute joint.

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$ of a revolute joint, then the value of the joint variable may be found as follows:

$$\phi_h = \tan^{-1} \left[\frac{\Phi_h(2, 1) - \Phi_h(1, 2)}{\Phi_h(1, 1) + \Phi_h(2, 2)} \right], \quad (4.11)$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . The signs of the numerator and denominator must be considered separately to resolve the proper quadrant for ϕ_h . Also, the possible division by zero in Eq. (4.11) must be avoided.¹

4.6.3 Prismatic Joint

A prismatic joint is shown in Figure 4.7

The conventions imposed on the choice of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ axes for a prismatic joint are:

1. The u_h and u'_h axes must be parallel to the joint motion axis; they must be collinear and must have the same positive sense.
2. The v_h and v'_h axes must be parallel and must have the same positive sense.

Within these restrictions, the axes may be chosen at will. The motion axis of a prismatic joint is not unique. Any convenient axis parallel to the direction of the relative joint motion may be chosen. The axis conventions are chosen here so that, for a two-dimensional application, u_h, v_h and u'_h, v'_h may lie in the plane of motion, as they do also for a revolute joint.

The joint variable for a prismatic joint is designated ϕ_h and is measured from v_h to v'_h in the direction of positive u_h, u'_h .

For a prismatic joint, the screw axis is u_h, u'_h and is located by the vectors: $\mathbf{P} = \mathbf{0}$ and $\mathbf{w} = [1, 0, 0]^t$. The screw parameters are $\theta = 0$ and $d = \phi_h$ and, using Eqs. (3.40)

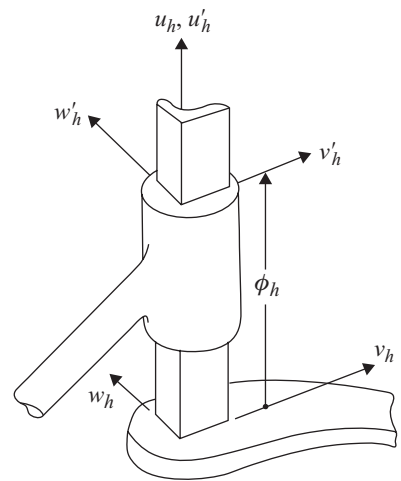


Figure 4.7. Coordinate systems for a prismatic joint.

¹ When programming for digital computation, the $\text{atan2}(-, -)$ function serves both of these purposes.

and (3.41) as we did for the helical joint, the transformation matrix for a prismatic joint becomes:

$$\Phi_h(\phi_h) = \begin{bmatrix} 1 & 0 & 0 & \phi_h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.12}$$

It should be noted that the transformation matrix for the prismatic joint can also be easily obtained by inspection because the relative orientations of the coordinate systems $u_h v_h w_h$ and $u'_h v'_h w'_h$ remain identical during the motion of this joint. If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$ of a prismatic joint, then the value of the joint variable may be found from

$$\phi_h = \Phi_h(1, 4),$$

where $\Phi_h(1,4)$ refers to row 1, column 4 of the matrix Φ_h .

4.6.4 Cylindric Joint

A cylindric joint is shown in Figure 4.8.

The conventions imposed on the choice of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ axes for a cylindric joint are:

1. The w_h and w'_h axes must be chosen along the common joint motion axis.
2. They must be collinear and must have the same positive sense.

There are two joint variables

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \end{bmatrix}. \tag{4.13}$$

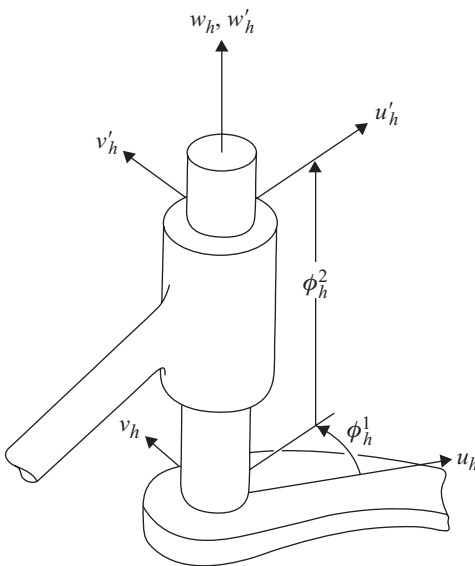


Figure 4.8. Coordinate systems for a cylindric joint.

The first joint variable is the angle ϕ_h^1 from positive u_h to positive u'_h measured counterclockwise about positive w_h, w'_h . The second joint variable is the distance ϕ_h^2 from u_h to u'_h measured in the direction of positive w_h, w'_h .

With $\theta = \phi_h^1$ and $d = \phi_h^2$ for screw parameters, the transformation matrix for a cylindric joint under these conventions is:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos\phi_h^1 & -\sin\phi_h^1 & 0 & 0 \\ \sin\phi_h^1 & \cos\phi_h^1 & 0 & 0 \\ 0 & 0 & 1 & \phi_h^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.14)$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$ of a cylindric joint, then the values of the joint variables may be found as follows:

$$\phi_h^1 = \tan^{-1} \left[\frac{\Phi_h(2, 1) - \Phi_h(1, 2)}{\Phi_h(1, 1) + \Phi_h(2, 2)} \right], \quad (4.15)$$

$$\phi_h^2 = \Phi_h(3, 4),$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix. The signs of the numerator and denominator must be considered separately to resolve the proper quadrant for ϕ_h^1 ; also, the possible division by zero must be avoided.²

4.6.5 Spheric Joint

A spheric joint, sometimes called a ball-and-socket joint, is shown in Figure 4.9. The only condition imposed on the placement of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ coordinate axes for a spheric joint is that they must be chosen such that their origins are coincident and are located at the center of the relative rotation of the spheric joint.

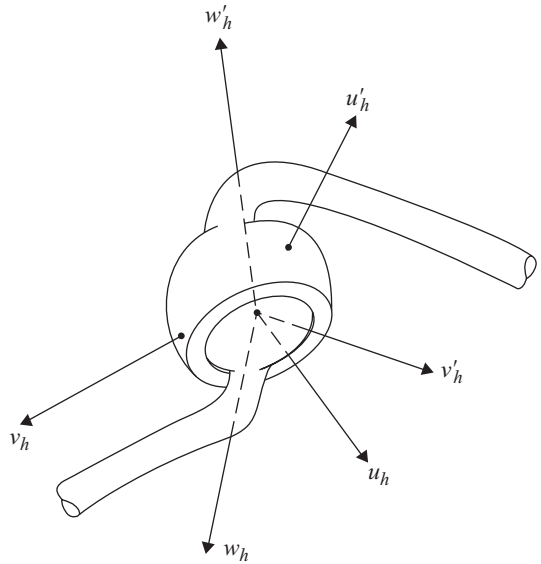


Figure 4.9. Coordinate systems for a spheric joint.

² When programming for digital computation, the `atan2(-,-)` function serves both of these purposes.

Euler-Rodrigues parameters are used to describe the spatial rotation between the two spheric joint coordinate systems. The choice of Euler-Rodrigues parameters rather than Cardan or other Euler angles avoids the difficulty of a “gimbal lock” singularity, as discussed in section 3.7. Therefore, using Euler-Rodrigues parameters, the spheric joint has four joint variables

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \\ \phi_h^4 \end{bmatrix}, \tag{4.16}$$

with the additional constraint that

$$(\phi_h^1)^2 + (\phi_h^2)^2 + (\phi_h^3)^2 + (\phi_h^4)^2 = 1. \tag{4.17}$$

The transformation matrix for a spheric joint under these conventions is:

$$\Phi_h(\phi_h) = \begin{bmatrix} (\phi_h^1)^2 - (\phi_h^2)^2 - (\phi_h^3)^2 + (\phi_h^4)^2 & 2\phi_h^1\phi_h^2 - 2\phi_h^3\phi_h^4 & 2\phi_h^1\phi_h^3 + 2\phi_h^2\phi_h^4 & 0 \\ 2\phi_h^1\phi_h^2 + 2\phi_h^3\phi_h^4 & -(\phi_h^1)^2 + (\phi_h^2)^2 - (\phi_h^3)^2 + (\phi_h^4)^2 & 2\phi_h^2\phi_h^3 - 2\phi_h^1\phi_h^4 & 0 \\ 2\phi_h^1\phi_h^3 - 2\phi_h^2\phi_h^4 & 2\phi_h^2\phi_h^3 + 2\phi_h^1\phi_h^4 & -(\phi_h^1)^2 - (\phi_h^2)^2 + (\phi_h^3)^2 + (\phi_h^4)^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.18}$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$ of a spheric joint, then the values of the Euler-Rodrigues parameters may be found as shown in section 3.8. First, the fourth Euler-Rodrigues parameter ϕ_h^4 is found from Eq. (3.51). Then, depending on its magnitude, the values ϕ_h^1 , ϕ_h^2 , and ϕ_h^3 can be found from Eqs. (3.52) and (3.53) if $|\phi_h^4| > \epsilon$, or from Eqs. (3.54) and (3.55) if $|\phi_h^4| \leq \epsilon$.

4.6.6 Flat Joint

A flat or planar joint is shown in Figure 4.10.

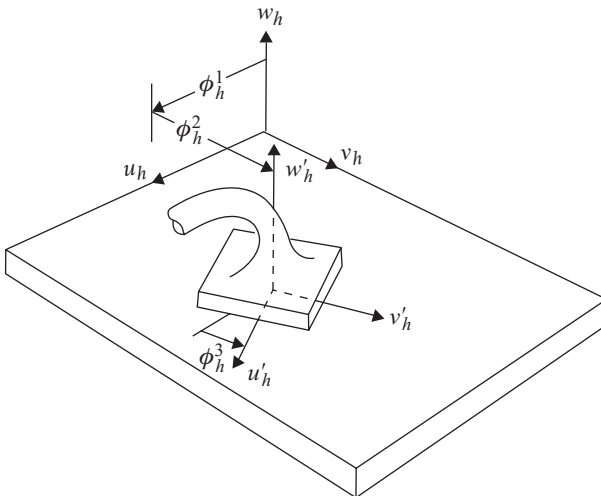


Figure 4.10. Coordinate systems for a flat joint.

The conventions for the assignment of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ coordinate axes for a flat joint are:

1. The $u_h v_h$ and $u'_h v'_h$ planes must coincide and must be parallel to the plane of motion.
2. The w_h and w'_h axes must both be normal to and positively directed to the same side of this common plane.

The three joint variables for a flat joint are the two rectilinear translation coordinates, ϕ_h^1 measured along u_h and ϕ_h^2 measured along v_h , respectively, from w_h to w'_h , and ϕ_h^3 – the angle from positive u_h to positive u'_h measured counterclockwise about the positive w'_h axis:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \end{bmatrix}. \quad (4.19)$$

A flat joint can be thought of as a serial combination of three joints, each with one degree of freedom. The first and the second would be prismatic joints with translations along their u_h and v_h axes, respectively, and the third would be a revolute joint with an axis of w'_h . The transformation matrix for the flat joint can then be obtained as the product of three joint transformation matrices. Using Eqs. (4.12) and (4.10), we would write:

$$\Phi_h(\phi_h) = \begin{bmatrix} 1 & 0 & 0 & \phi_h^1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \phi_h^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi_h^3 & -\sin \phi_h^3 & 0 & 0 \\ \sin \phi_h^3 & \cos \phi_h^3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The resulting transformation matrix is:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos \phi_h^3 & -\sin \phi_h^3 & 0 & \phi_h^1 \\ \sin \phi_h^3 & \cos \phi_h^3 & 0 & \phi_h^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.20)$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$ of a flat joint, then the values of the joint variables may be found as follows:

$$\begin{aligned} \phi_h^1 &= \Phi_h(1, 4), \\ \phi_h^2 &= \Phi_h(2, 4), \\ \phi_h^3 &= \tan^{-1} \left[\frac{\Phi_h(2, 1) - \Phi_h(1, 2)}{\Phi_h(1, 1) + \Phi_h(2, 2)} \right], \end{aligned} \quad (4.21)$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . The signs of the numerator and denominator must be considered separately to resolve the proper quadrant for ϕ_h^3 ; also, the possible division by zero must be avoided.³

³ When programming for digital computation, the $\text{atan2}(-, -)$ function serves both of these purposes.

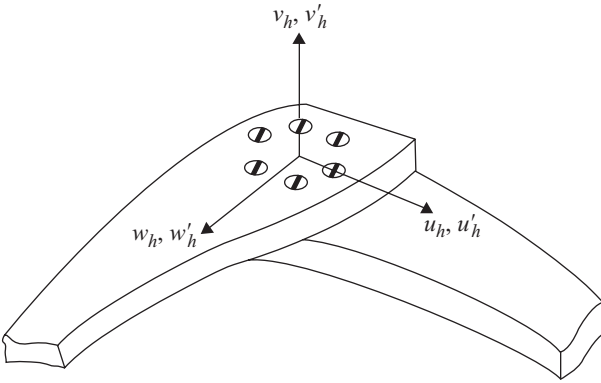


Figure 4.11. Coordinate systems for a rigid joint.

4.6.7 Rigid Joint

A rigid joint is shown in [Figure 4.11](#).

The single convention for the assignment of the u_h, v_h, w_h and u'_h, v'_h, w'_h coordinate axes for a rigid joint is that they must coincide.

The rigid joint allows no relative motion between the connected bodies; therefore, it has no joint variables. The rigid joint transformation matrix is as follows:

$$\Phi_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1. \end{bmatrix} = I \tag{4.22}$$

4.6.8 Open Joint

An open joint is shown in [Figure 4.12](#).

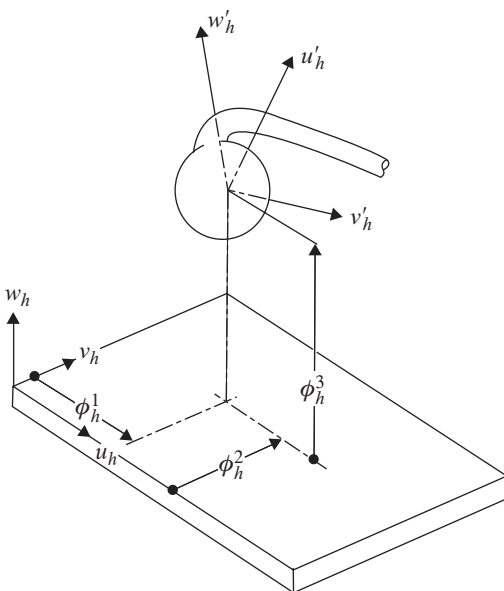


Figure 4.12. Coordinate systems for an open joint.

There are no restrictions on the assignment of the $u_h v_h w_h$ and $u'_h v'_h w'_h$ coordinate axes for an open joint; they may each be chosen completely arbitrarily. Convenience for definition of motion input or interpretation of results, particularly force results, may give advantages to a particular choice of axes; see Chapter 17 for details.

Because we choose Euler-Rodrigues parameters to describe the relative spatial orientation between the two coordinate systems, the open joint has seven joint variables. The first three joint variables for the open joint are the three rectilinear coordinates defining the translations *from* the $u_h v_h w_h$ origin *to* the $u'_h v'_h w'_h$ origin, measured along the $u_h v_h w_h$ axes, respectively. These are followed by the four Euler-Rodrigues parameters defining the relative orientation between the joint axes of the “connected” bodies. The total vector of joint variables is

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \\ \phi_h^4 \\ \phi_h^5 \\ \phi_h^6 \\ \phi_h^7 \end{bmatrix}, \quad (4.23)$$

with the additional constraint

$$(\phi_h^4)^2 + (\phi_h^5)^2 + (\phi_h^6)^2 + (\phi_h^7)^2 = 1. \quad (4.24)$$

Here, as with the spheric joint, the choice of Euler-Rodrigues parameters rather than Cardan or other Euler angles for the rotational joint variables avoids the possible “gimbal lock” singularity discussed in section 3.7.

Under these conventions, the open joint transformation matrix is as follows:

$$\Phi_h(\phi_h) = \begin{bmatrix} (\phi_h^4)^2 - (\phi_h^5)^2 - (\phi_h^6)^2 + (\phi_h^7)^2 & 2\phi_h^4\phi_h^5 - 2\phi_h^6\phi_h^7 & 2\phi_h^4\phi_h^6 + 2\phi_h^5\phi_h^7 & \phi_h^1 \\ 2\phi_h^4\phi_h^5 + 2\phi_h^6\phi_h^7 & -(\phi_h^4)^2 + (\phi_h^5)^2 - (\phi_h^6)^2 + (\phi_h^7)^2 & 2\phi_h^5\phi_h^6 - 2\phi_h^4\phi_h^7 & \phi_h^2 \\ 2\phi_h^4\phi_h^6 - 2\phi_h^5\phi_h^7 & 2\phi_h^5\phi_h^6 + 2\phi_h^4\phi_h^7 & -(\phi_h^4)^2 - (\phi_h^5)^2 + (\phi_h^6)^2 + (\phi_h^7)^2 & \phi_h^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.25)$$

Because there are six degrees of freedom and seven joint variables with one constraint, it is clear that the open joint places no constraints on the relative motions between the bodies.

If accurate numeric values are known for the elements of $\Phi_h(\phi_h)$, then the values of the translational joint variables may be found as follows:

$$\begin{aligned} \phi_h^1 &= \Phi_h(1, 4), \\ \phi_h^2 &= \Phi_h(2, 4), \\ \phi_h^3 &= \Phi_h(3, 4), \end{aligned} \quad (4.26)$$

where $\Phi_h(i, 4)$ refers to row i , column 4 of the matrix Φ_h . The values of the remaining four joint variables, the Euler-Rodrigues parameters for the rotation,

may be found as discussed in section 3.8. First, the fourth Euler-Rodrigues parameter ϕ_h^7 is found from Eq. (3.51). Then, depending on its magnitude, the values of ϕ_h^4 , ϕ_h^5 , and ϕ_h^6 are found from Eqs. (3.52) and (3.53) if $|\phi_h^7| > \varepsilon$, or from Eqs. (3.54) and (3.55) if $|\phi_h^7| \leq \varepsilon$.

4.6.9 Parallel-Axis Gear Joint

A parallel-axis gear joint is shown in Figure 4.13.

This joint type is chosen as an example to illustrate how the proper transformation matrix may be derived for a higher pair. In concept, the same general procedure can be applied to other higher pairs as the need arises, no matter what the joint type. It is true that the relative motion in certain higher pairs may be difficult to describe analytically. However, once this relative motion is modeled as a transformation between coordinate systems attached to the two mating elements, then the transformation matrix provides a standard canonical representation for this type of joint and the many methods of the coming chapters become immediately applicable.

The parallel-axis gear joint has four invariant parameters necessary for its characterization: the two pitch-circle radii R_h and R'_h , the transverse pressure angle α_h , and the helix angle β_h . The gear ratio is $\zeta_h = R'_h/R_h$ and the nominal center-to-center distance is $(R_h + R'_h) = (1 + \zeta_h)R_h$.

For a parallel-axis gear joint, the variety of tooth forms modeled here include spur gears (with pressure angle α_h), helical gears (with helix angle β_h), and herringbone gears (by setting $\beta_h = 0$), each having an involute tooth profile. A circular disk rolling without slip against another can also be modeled by treating both the pressure angle and helix angle as zero.

The helix angle is defined such that positive axial sliding movement of the $u'_h v'_h w'_h$ gear produces positive change in the angle of the $u'_h v'_h w'_h$ gear with respect to the $u_h v_h w_h$ gear.

The coordinate systems for a parallel-axis gear joint are chosen as shown in Figure 4.13. The conventions are:

1. The w_h and w'_h axes must lie along the respective rotation axes of the two mating gears; they must be parallel, and must have the same positive sense.

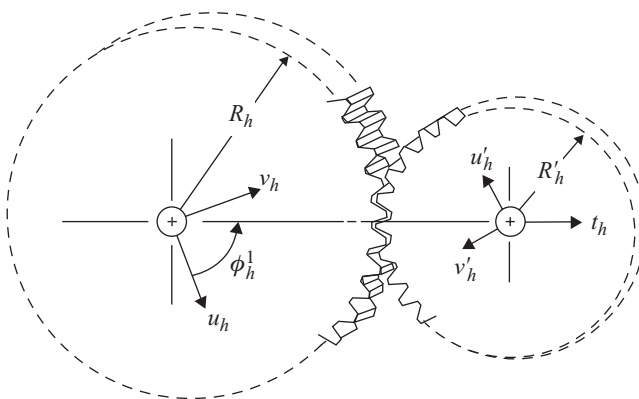


Figure 4.13. Coordinate systems for a parallel-axis gear joint.

2. The common perpendicular directed from w_h to w'_h is designated positive t_h . There must be one position of the joint ($\phi_h^1 = 0$) for which the positive u_h and positive u'_h axes are both directed along positive t_h .

It is *not* assumed that the two gears lie in a common plane, nor that the center-to-center distance of the two gears is equal to the sum of the pitch radii. These constraints may be provided by other joints and body shapes of the system, those that control the postures of the two shafts on which the gears are mounted. However, it is assumed that the two gear axes are parallel. These assumptions are made to avoid complexity in deriving the transformation matrix.

It is also assumed that the gears do not lose contact with each other and that the teeth of one gear do not reach the bottom of the tooth space of the mating gear. If either of these were to happen, the transformation matrix shown would not accurately represent the relative motion of the joint.

The parallel-axis gear joint has three joint variables:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \end{bmatrix}. \quad (4.27)$$

The first joint variable is the angle ϕ_h^1 from positive u_h to positive t_h and is measured counterclockwise about positive w_h . The second joint variable ϕ_h^2 is the possible change in center-to-center distance beyond its nominal value, measured from w_h to w'_h along positive t_h . The third joint variable ϕ_h^3 is the distance measured from the $u_h v_h$ plane to the $u'_h v'_h$ plane along positive w_h, w'_h . The second and third joint “variables” defined here are usually constant parameters, but treating them as variables allows changes in center distance and axial adjustment during installation or operation. Defining these as constants would result in the calculation of statically indeterminate force components in these directions. See Chapter 17 for details.

Because of the assumption of involute tooth profiles, even though the center-to-center distance may not exactly equal the nominal value ($R_h + R'_h$), or the axial plane-offset distance ϕ_h^3 may not equal zero, the relative rotation θ of the gear attached to $u'_h v'_h w'_h$ with respect to the $u_h v_h w_h$ axes can be calculated

$$\theta = \left[(R_h + R'_h) \phi_j^1 + \operatorname{sgn}(F) \left\{ \sqrt{[(R_h + R'_h + \phi_h^2)]^2 - [(R_h + R'_h) \cos \alpha_h]^2} - (R_h + R'_h) \sin \alpha_h \right\} + \phi_h^3 \tan \beta_h \right] / R'_h,$$

where F is positive when the tangential force transmitted between the gear teeth tends to cause the counterclockwise rotation of the $u'_h v'_h w'_h$ gear to increase, negative if it tends to cause a decrease.

The term including $\operatorname{sgn}(F)$ may be ignored in the calculation of θ with very little error, probably less than the manufacturing tolerances. This is a *great* saving in the difficulty of kinematic analysis when forces need not be analyzed. The term is included rather than ignored in these equations, however, because it *cannot* be ignored in the calculation of fictitious derivatives and becomes very important in finding the tooth force transmitted through the gear mesh. This term and its derivatives cause the tooth force to be applied at the proper pressure angle, and ignoring

it causes the parallel axis gear joint to transmit force as high-friction disks with zero pressure angle. This can be seen by studying the methods of Chapter 17.

These conventions yield the following transformation matrix for a parallel-axis gear joint:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & (R_h + R'_h + \phi_h^2)\cos\phi_h^1 \\ \sin\theta & \cos\theta & 0 & (R_h + R'_h + \phi_h^2)\sin\phi_h^1 \\ 0 & 0 & 1 & \phi_h^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.28)$$

This transformation matrix is also valid for the case of an internal gear; however, in such a case, the pitch radius of the internal gear, either R_h or R'_h , must be treated as negative.

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the values of the joint variables may be found as follows:

$$\begin{aligned} \phi_h^1 &= \tan^{-1} [\Phi_h(2, 4)/\Phi_h(1, 4)], \\ \phi_h^2 &= \sqrt{\Phi_h(1, 4)^2 + \Phi_h(2, 4)^2} - (R_h + R'_h), \\ \phi_h^3 &= \Phi_h(3, 4), \end{aligned} \quad (4.29)$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . To resolve the proper quadrant for ϕ_h^1 , the signs of the numerator and denominator must be considered separately; also the possible division by $\Phi_h(1,4) = 0$ must be avoided.⁴

4.6.10 Involute Rack-and-Pinion Joint

An involute rack-and-pinion joint is shown in Figure 4.14.

Because the transformation matrix for a parallel-axis gear joint is valid for both internal and external gears, we might expect that it could also be used to describe a rack and pinion. However, this is not the case; this would require locating the coordinate system for the rack element at an infinitely remote center of curvature, and would provide a very inconvenient description of such a joint.

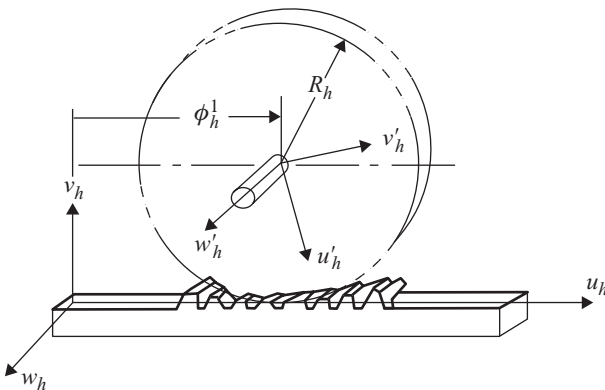


Figure 4.14. Coordinate systems for an involute rack-and-pinion joint.

⁴ When programming for digital computation, the atan2(–,–) function serves both of these purposes.

For the involute rack-and-pinion joint, the orientation must be chosen *from* the rack *to* the pinion. This must be considered when setting the joint orientation, before finding the kinematic loops and paths as discussed in Chapter 2. The coordinate systems are chosen as shown in Figure 4.14. The conventions are:

1. The $u_h v_h w_h$ axes are fixed to the rack and the $u'_h v'_h w'_h$ axes are fixed to the pinion; the $u_h v_h$ and $u'_h v'_h$ planes are parallel. The axes are chosen such that u_h lies along the pitch surface of the rack, in the chosen direction of positive relative motion of the pinion center.
2. The w'_h axis lies along the pinion axis of rotation. The w_h axis, attached to the rack, is parallel to w'_h , and has the same positive sense.
3. There is a position of the joint, where $\phi_h^1 = 0$, for which u_h and u'_h are parallel.

The involute rack-and-pinion joint has three joint variables:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \end{bmatrix}. \quad (4.30)$$

The three values $-\phi_h^1$, $(R'_h + \phi_h^2)$, and ϕ_h^3 – are the relative coordinates of the $u'_h v'_h w'_h$ origin measured along the $u_h v_h w_h$ axes, respectively.

The joint also has three invariant parameters necessary for its complete characterization, the transverse pressure angle α_h , the helix angle β_h (that may be zero), and the pitch radius R'_h of the pinion. The pitch radius is positive when the pinion is on the positive v_h side of the rack, or negative when the pinion lies on the negative v_h side of the rack. The helix angle β_h is defined such that positive axial sliding motion ϕ_h^3 of the pinion produces positive rotational change in the angle of the pinion with respect to the rack. For spur gear teeth on the pinion, the helix angle is zero; for herringbone teeth, a helix angle of $\beta_h = 0$ can be used.

The variety of tooth forms modeled include spur, helical, and herringbone teeth. Each is assumed to have an involute tooth profile. A circular disk rolling without slip against a flat surface may also be modeled by treating both the pressure angle and helix angle as zero.

These conventions do *not* define that the pinion center offset distance from the pitch surface of the rack must match the pitch radius of the pinion, or that the central planes of the pinion and rack must be coplanar. These constraints can be provided by other joints and body shapes of the system, those that control the posture of the rack with respect to the pinion shaft. However, it is assumed that the central planes of the rack and pinion are parallel, and that the teeth remain in proper mesh. The second and third joint “variables” are usually small, nearly constant parameters, but they allow small changes in center distance and axial adjustment during installation or operation. If defined as constants, they result in the calculation of statically indeterminate force components. See Chapter 17 for details.

Because of the involute tooth profile, even though the pinion center offset distance $(R'_h + \phi_h^2)$ may not exactly match the nominal pitch radius, the rotation angle θ of the pinion u'_h axis with respect to the u_h axis can be calculated

$$\theta = -[\phi_h^1 + \text{sgn}(F)\phi_h^2 \tan \alpha_h - \phi_h^3 \tan \beta_h] / R'_h,$$

where F is positive when the tangential force transmitted between the gear teeth tends to cause counterclockwise rotation of the u'_h axis to increase, negative if it tends to cause a decrease.

The term including $\text{sgn}(F)$ may be ignored in the calculation of θ with very little error, probably less than the error of ignoring manufacturing tolerances. This is a great saving in the difficulty of kinematic analysis where forces need not be analyzed. The term is included rather than ignored in these equations, however, because it *cannot* be ignored in the calculation of fictitious derivatives and becomes critical in finding the tooth force transmitted through the mesh. This term and its derivatives cause the tooth force to be applied at the proper pressure angle and ignoring it causes the rack and pinion joint to transmit force as would happen with a high-friction disk with zero pressure angle. This can be seen by studying the methods of Chapter 17.

These conventions for placement of the coordinate axes yield the following transformation matrix for the involute rack-and-pinion joint:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & \phi_h^1 \\ \sin\theta & \cos\theta & 0 & R'_h + \phi_h^2 \\ 0 & 0 & 1 & \phi_h^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.31)$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the values of the joint variables may be found as follows:

$$\begin{aligned} \phi_h^1 &= \Phi_h(1, 4), \\ \phi_h^2 &= \Phi_h(2, 4) - R'_h, \\ \phi_h^3 &= \Phi_h(3, 4), \end{aligned} \quad (4.32)$$

where $\Phi_h(i, 4)$ refers to the element of matrix Φ_h in row i , column 4.

4.6.11 Straight-Tooth Bevel-Gear Joint

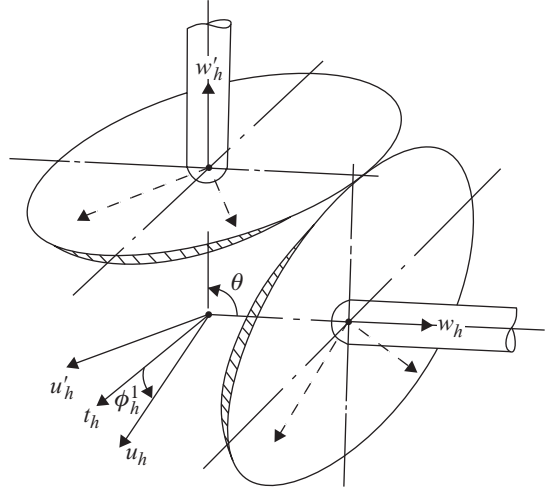
A straight-tooth bevel-gear joint is shown in [Figure 4.15](#).

The straight-tooth bevel-gear joint has two invariant parameters, the two pitch-cone half-angles γ_h and γ'_h , respectively.

For a straight-tooth involute bevel-gear joint, the coordinate systems are chosen as shown in [Figure 4.15](#). The conventions are:

1. The w_h and w'_h axes intersect at the common apex for the two pitch cones and lie along the respective rotation axes of the two bevel gears. Each is positively directed outward from the apex toward one of the two gears.
2. The common perpendicular to both w_h and w'_h is given the symbol t_h with positive direction chosen in the sense of the vector cross product $t_h = w_h \times w'_h$. There is one position, where $\phi_h^1 = 0$, such that the positive u_h and positive u'_h axes are coincident and both are aligned along positive t_h .

Figure 4.15. Coordinate systems for a straight-tooth bevel-gear joint.



The straight-tooth bevel-gear joint has two joint variables:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \end{bmatrix}. \quad (4.33)$$

The first joint variable is the angle ϕ_h^1 measured *from* positive t_h to positive u_h , measured counterclockwise about positive w_h . The second joint variable ϕ_h^2 is the possible increase in the shaft intersection angle beyond the sum of the two pitch-cone half-angles, γ_h and γ'_h , measured counterclockwise *from* w_h to w'_h about the positive t_h axis.

These conventions do *not* define that the angle θ between the intersecting shaft-axes match the sum of the two pitch-cone half-angles. This constraint may be provided by other joints and body shapes of the system, those that control the postures of the two shafts on which the bevel gears are mounted. It is assumed that the gears do not come out of contact and that the teeth of one do not touch the bottom of the tooth-space of the mating gear. If either of these were to occur, the transformation matrix shown would not accurately represent the relative motion of the joint.

Because of the assumption of involute tooth profiles, the rotation of the bevel gear attached to $u'_h v'_h w'_h$ with respect to the t_h axis may be calculated. This yields the following transformation matrix for a straight-tooth bevel-gear joint:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos \phi_h^1 \cos \phi' - \sin \phi_h^1 \cos \theta \sin \phi' & -\cos \phi_h^1 \sin \phi' + \sin \phi_h^1 \cos \theta \cos \phi' & -\sin \phi_h^1 \sin \theta & 0 \\ -\sin \phi_h^1 \cos \phi' + \cos \phi_h^1 \cos \theta \sin \phi' & \sin \phi_h^1 \sin \phi' + \cos \phi_h^1 \cos \theta \cos \phi' & -\cos \phi_h^1 \sin \theta & 0 \\ \sin \theta \sin \phi' & \sin \theta \cos \phi' & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.34)$$

where

$$\begin{aligned} \phi' &= (\tan \gamma_h / \tan \gamma'_h) \phi_h^1, \\ \theta &= \gamma_h + \gamma'_h + \phi_h^2. \end{aligned}$$

This transformation matrix is also valid for a joint with a straight-tooth *internal* bevel-gear; but, in such a case, the pitch-cone half-angle of the internal gear, either γ_h or γ'_h , must be treated as negative.

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the values of the joint variables may be found as follows:

$$\begin{aligned}\phi_h^1 &= \frac{\tan \gamma'_h}{\tan \gamma_h} \tan^{-1} \left[\frac{\Phi_h(3, 1)}{\Phi_h(3, 2)} \right], \\ \phi_h^2 &= \cos^{-1} [\Phi_h(3, 3)] - \gamma_h - \gamma'_h,\end{aligned}\tag{4.35}$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . The signs of the numerators and denominators must be considered separately to resolve the proper quadrant for ϕ_h^1 . Also, division by $\Phi_h(3, 2) = 0$ must be avoided.⁵

4.6.12 Point on a Planar-Curve Joint

A variety of different kinds of joints are treated together here because their formulations and transformations are similar. Two typical cases are shown in [Figure 4.16](#). The name for this type of joint comes from their common characteristic. In each case, the designer's intent is to cause a chosen "point" of the "following" element of the joint to follow a planar "curve" defined by the shape of the "preceding" element. Typical examples include a pin constrained to follow a slot, or a circular roller center to follow the pitch curve of a disk cam.

For this type of joint, the coordinate systems are chosen as shown. The conventions are:

1. The $u_h v_h$ and $u'_h v'_h$ axes are attached to the joint elements containing the "curve" and the "point," respectively. Both are chosen in planes parallel to the plane containing the "curve." The w_h and w'_h axes must be parallel and both must be normal to and positively directed to the same side of the plane of the "curve." This must be considered when setting the joint orientation, before finding the kinematic loops and paths as discussed in Chapter 2.
2. The "curve" is defined by two specified functions, $u(\phi_h^1)$ and $v(\phi_h^1)$. These functions are continuous, single-valued, and at least twice-differentiable functions of the same independent variable, ϕ_h^1 . As a practical matter, these functions must be programmed for digital computation. Rather than program anew for each new curve, it is highly advantageous to choose a standard form for the functions that can be programmed once and is widely applicable.⁶

A point on a planar-curve joint has three joint variables:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \end{bmatrix}.\tag{4.36}$$

⁵ When programming for digital computation, the $\text{atan2}(-, -)$ function serves both of these purposes.

⁶ Fourier series, for example, is chosen as the standard form in the IMP software.

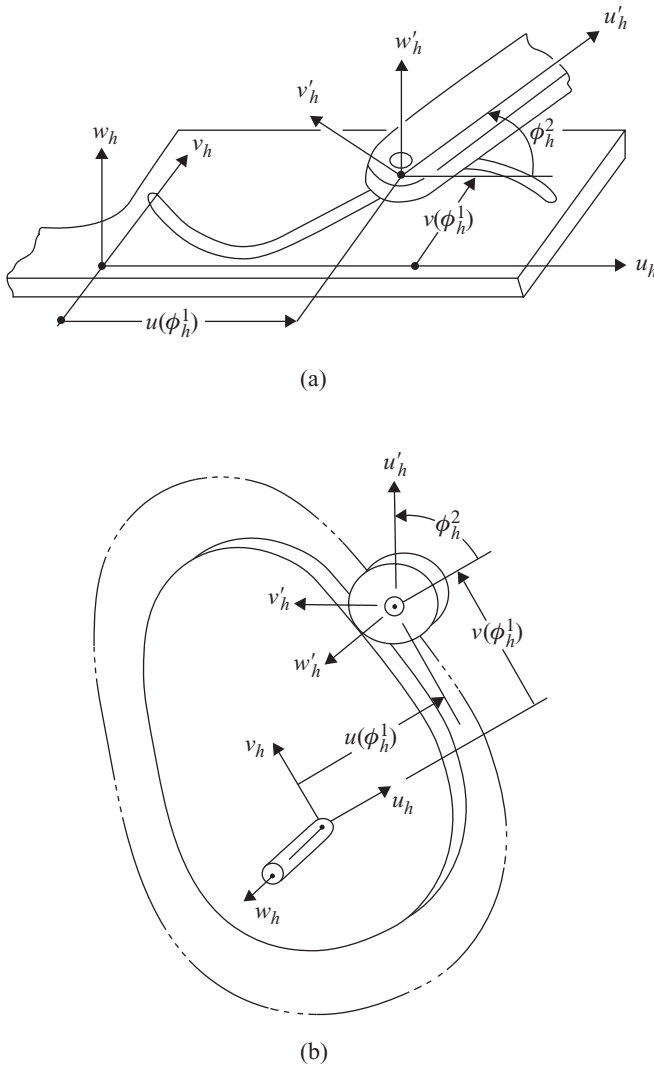


Figure 4.16. Typical situations modeled as a point on a planar-curve joint. Note placement of coordinate systems. Note also in (b) that the u'_h, v'_h, w'_h axes would not be attached to the roller but to the follower arm, as shown in (a).

The first joint variable ϕ_h^1 is the independent parameter for the functions, $u(\phi_h^1)$ and $v(\phi_h^1)$, defining the Cartesian coordinates of the curve. The second joint variable ϕ_h^2 is the angle from the positive u_h axis to the positive u'_h axis, measured counterclockwise about the positive w'_h axis. The third joint variable is the distance ϕ_h^3 , from the $u_h v_h$ plane to the $u'_h v'_h$ plane measured along the w'_h axis.

These conventions do *not* define that the $u_h v_h$ and the $u'_h v'_h$ planes must be coplanar. This constraint may be provided by other joints and body shapes of the system, those that control the relative posture of the bodies to which the “point” and the “curve” are attached. However, it is assumed that the planes of the “point” and “curve” bodies are parallel, resulting in the third joint “variable” remaining nearly constant. If this third joint variable were defined as a constant, however, this

would result in the calculation of a statically indeterminate force component in the w_h direction. See Chapter 17 for details.

These conventions yield the following transformation matrix for a point on a planar-curve joint:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos \phi_h^2 & -\sin \phi_h^2 & 0 & u(\phi_h^1) \\ \sin \phi_h^2 & \cos \phi_h^2 & 0 & v(\phi_h^1) \\ 0 & 0 & 1 & \phi_h^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.37}$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the values of the joint variables may be found as follows:

$$\begin{aligned} \phi_h^1 &= f^{-1} [\Phi_h(1, 4), \Phi_h(2, 4)], \\ \phi_h^2 &= \tan^{-1} \left[\frac{\Phi_h(2, 1) - \Phi_h(1, 2)}{\Phi_h(1, 1) + \Phi_h(2, 2)} \right], \\ \phi_h^3 &= \Phi_h(3, 4), \end{aligned} \tag{4.38}$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . The reason that a closed-form solution for ϕ_h^1 is not shown results from the unspecified parametric form of the functions defining the curve shape, and thus the inability to describe their inverses. This solution is required in the software, however, to provide an initial value for this variable, or at least a reasonable estimate, at the modeling posture. Also, the signs of the numerators and denominators must be considered separately to resolve the proper quadrant for ϕ_h^2 , and division by zero must be avoided.⁷

4.6.13 Line Tangent to a Planar-Curve Joint

A line tangent to a planar-curve joint is shown in Figure 4.17. The name for this type of joint comes from its relative motion characteristic. Here the designer’s intent is to constrain a chosen straight “line” of the “following” element to remain tangent to a

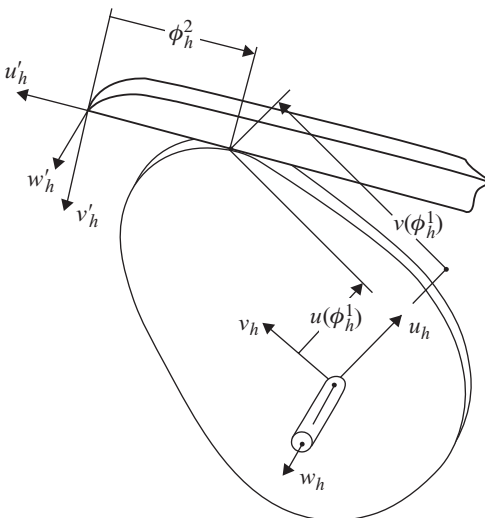


Figure 4.17. A typical situation modeled by a line tangent to a planar-curve joint. Note placement of the coordinate systems. Note also that, as pictured, the value of ϕ_h^2 is negative.

⁷ When programming for digital computation, the atan2(–,–) function serves both of these needs.

planar “curve” defined by the shape of the “preceding” element. A typical example is a flat surface constrained to remain in contact with a curved surface as exemplified by a flat-faced follower sliding on the pitch curve of a disk cam.

For this type of joint, the coordinate systems are chosen as shown in Figure 4.17. The conventions are:

1. The $u_h v_h$ and $u'_h v'_h$ axes are attached to the joint elements containing the “curve” and the “line,” respectively. Both are chosen in planes parallel to the plane of the “curve.” The w_h and w'_h axes must be parallel and must be normal to and positively directed to the same side of the plane of the “curve.” This choice must be considered when setting the joint orientation, before finding the kinematic loops and paths as discussed in Chapter 2.
2. The u'_h axis is defined along the “line” that is to remain tangent to the “curve.”
3. The Cartesian coordinates of the “curve” are defined by two specified functions, $u(\phi_h^1)$ and $v(\phi_h^1)$. These functions are continuous, single-valued, and at least triply-differentiable functions of the same independent variable, ϕ_h^1 . As a practical matter, these functions must be programmed for digital computation. Rather than program anew for each new curve, it is highly advantageous to choose a standard form for the functions that can be programmed once and is widely applicable.⁸

The line tangent to a planar-curve joint has three joint variables:

$$\phi_h = \begin{bmatrix} \phi_h^1 \\ \phi_h^2 \\ \phi_h^3 \end{bmatrix}. \quad (4.39)$$

The first joint variable ϕ_h^1 is the value of the independent variable for the “curve.” The second joint variable ϕ_h^2 is the distance along the positive u'_h axis to the point of tangency. The third joint variable is the distance ϕ_h^3 from the $u_h v_h$ plane to the $u'_h v'_h$ plane measured along w'_h .

The assumptions do *not* define that the planes of the “line” and the “curve” must be coplanar. This constraint may be provided by other joints and body shapes of the system, those that control the postures of the bodies to which the “line” and the “curve” are attached. However, it is assumed that the planes of the “line” and “curve” are parallel, resulting in the third joint “variable” remaining nearly constant. If this third joint variable is defined as a constant, however, this results in the calculation of a statically indeterminate force component in the w_h direction. See Chapter 17 for details.

These conventions yield the following transformation matrix for a line tangent to a planar-curve joint:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & u(\phi_h^1) - \phi_h^2 \cos\theta \\ \sin\theta & \cos\theta & 0 & v(\phi_h^1) - \phi_h^2 \sin\theta \\ 0 & 0 & 1 & \phi_h^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.40)$$

⁸ Fourier series, for example, is chosen as the standard form in the IMP software.

where

$$\theta = \tan^{-1} \left[\frac{dv(\phi_h^1)/d\phi_h^1}{du(\phi_h^1)/d\phi_h^1} \right].$$

If accurate numeric values are known for the elements of the matrix $\Phi_h(\phi_h)$, then the values of the joint variables may be found as follows:

$$\begin{aligned} \theta &= \tan^{-1} \left[\frac{\Phi_h(2, 1) - \Phi_h(1, 2)}{\Phi_h(1, 1) + \Phi_h(2, 2)} \right], \\ \phi_h^1 &= f^{-1} [\Phi_h(1, 4), \Phi_h(2, 4)], \\ \phi_h^2 &= [u(\phi_h^1) - \Phi_h(1, 4)] \cos \theta + [v(\phi_h^1) - \Phi_h(2, 4)] \sin \theta, \\ \phi_h^3 &= \Phi_h(3, 4), \end{aligned} \tag{4.41}$$

where $\Phi_h(i, k)$ refers to row i , column k of the matrix Φ_h . The signs of the numerator and denominator must be considered separately to resolve the proper quadrant for θ , and division by zero must be avoided.⁹

The reason that a closed-form solution for ϕ_h^1 is not shown results from the unspecified parametric form of the functions defining the curve shape, and thus the inability to describe their inverses. This solution is required in the software, however, to provide an initial condition value for this variable, or at least a close estimate.

PROBLEMS

4.1 Knowing that the translation from body axes xyz to joint axes uvw is by distance d , find the transformation matrix S_{bh} shown by **Figure 4.3**.

4.2 Consider a robotic wrist mechanism consisting of three revolute joint with perpendicular axes as shown in **Figure P4.2**. Use joint transformation matrices to show that this mechanism is kinematically equivalent to a single spherical joint positioned at point O of the figure.

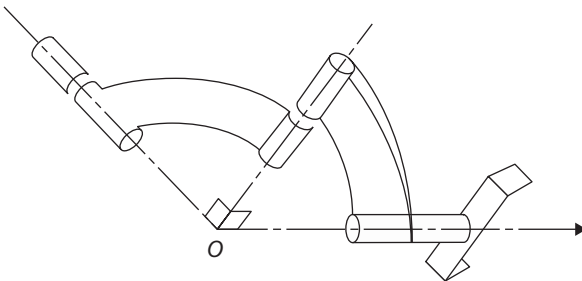


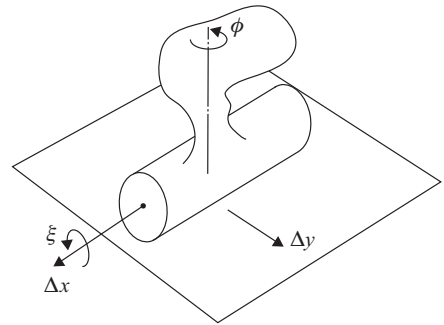
Figure P4.2. Robotic wrist mechanism.

4.3 When the relative motion of two bodies in close proximity to one another is described in terms of a coincidence relationship between geometric features of one body and geometric features of the other, the two bodies are said to have a kinematic bond or coupling between them. For each of the two following kinematic couplings shown, find a single equivalent joint transformation equation describing the corresponding relative motion.

⁹ When programming for digital computation, the $\text{atan2}(-, -)$ function serves both of these purposes.

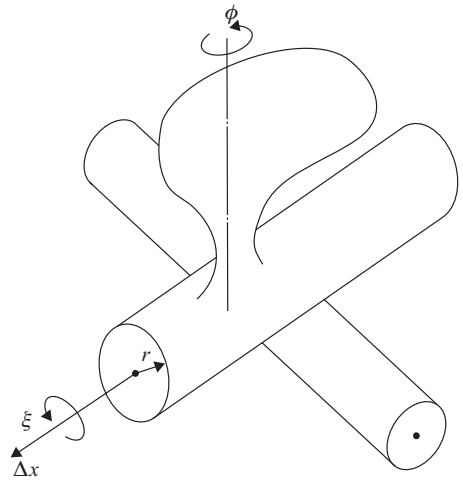
- a) A cylinder moving in contact with a plane. (Note that the coupling is only defined over a relatively small range of motion.)

Figure P4.3a. Cylinder in contact with a plane.



- b) A cylinder rolling and sliding on another cylinder without tilting. (Note again that the coupling is only defined over a relatively small range of motion.)

Figure P4.3b. Cylinder in contact with a cylinder.



4.4 Figure P4.4 shows a slider-crank linkage to be used in the design of a weighing device. Enough information is shown to uniquely locate and orient all body axes and joint axes. Modeling and analysis of this mechanism starts here and continues in several chapters that follow. Here, in Chapter 4, we are concerned only with forming certain elements of the model.

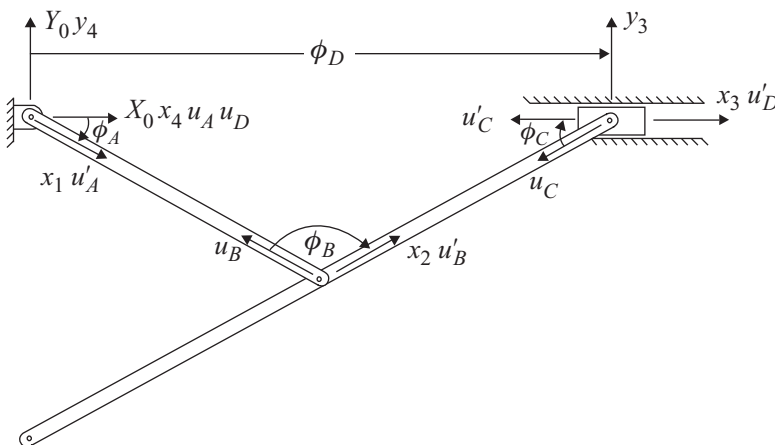


Figure P4.4. Linkage for a nonlinear weighing device.

- a) Form all shape matrices S_{bh} assuming that the lengths of the parts are equal to some known length l .
- b) Form the joint matrices Φ_h for each joint in symbolic form.

4.5 Figure P4.5 shows an exploded diagram of an Oldham shaft coupling with enough information to locate all body and joint axes, which are all coincident at the center of the coupling. The Oldham coupling is a well-known design for torque transmission between two parallel shafts that may be slightly eccentric. The coupling accommodates this eccentricity (e) while maintaining the same rotational speed for the two shafts.

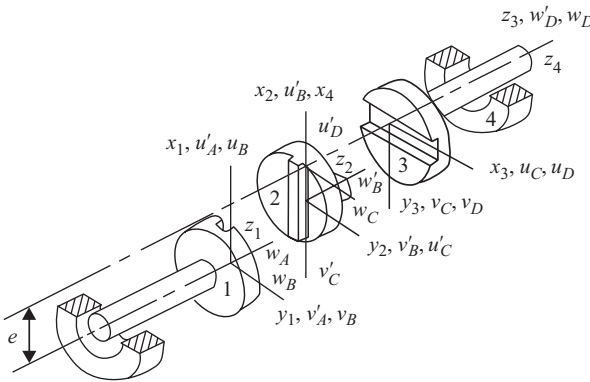


Figure P4.5. Oldham coupling for parallel but misaligned shafts.

- a) Form all shape matrices S_{bh} assuming that the shafts, slots and sliders are centered on each of the circular parts, and that all axes are parallel and are positive in the same directions. The orientations of the joints are chosen such that the loop and path matrices of Chapter 2 are:

$$L = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

- b) Form the joint matrices Φ_h for each joint in symbolic form.

5 Posture Analysis by Kinematic Equations

5.1 Introduction

One common means of describing the posture of a mechanism or multibody system is by specifying a vector of coordinates listing the positions of a set of independent “input” joint variables; that is, by giving the values of a number of joint variables ϕ equal to f , the mobility of the system. These independent degrees of freedom are called *generalized coordinates* and they are often used to specify the posture of the system because, in most cases, their values uniquely determine the values of all other joint variables and the position of each point of every body of the system. That is to say, there is usually only one unique configuration or posture of the system corresponding to a given set of generalized coordinates.

We will denote a generalized coordinate by the symbol ψ_j and note that each is equivalent to one of the joint variable values; that is, $\psi_j = \phi_h$ for some j and h . Listing the full vector of f generalized coordinates in a chosen order, we define the *generalized position vector* that identifies the posture or the configuration of the system:

$$\psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_f \end{bmatrix}. \quad (5.1)$$

What allows only a subset of the joint variables to become generalized coordinates, whereas others are dependent on these for their values? Well, for example, we saw that for each spheric joint and each open joint, where Euler-Rodrigues parameters were used to describe orientation, we have a constraint equation that makes one of the joint variables dependent on others. We also saw in Chapter 2 that some mechanisms and multibody systems contain closed loops of bodies interconnected by joints, and we will see in section 5.5 how each of these closed loops leads to a set of “*loop-closure*” constraint equations. Each situation – where these or other constraint equations allow the determination of some of the joint variable position values from others – reduces the number of joint variables that remain independent; that is, the number of independent generalized coordinates, f , that represents the mobility of the system.

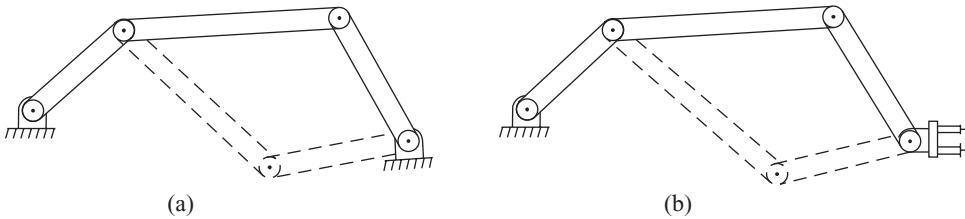


Figure 5.1. (a) Multiple closures of a planar four-bar linkage with the same input angle; (b) multiple postures of a planar manipulator with the same end-effector posture.

This, however, is not the complete story. In some multibody systems, as is also the case for some closed-loop mechanisms, the bodies can be configured in more than one way, even for identical values of the generalized coordinates. In some open-loop systems, such as robot manipulators, a point (or gripper posture) can be reached by more than one system posture and depending on which posture is used, the positions of some bodies may be quite different. Such a case is exemplified by the closed-loop planar four-bar linkage shown in Figure 5.1a where broken lines show a second possible configuration of the same system with the same setting for the independent input angle. Also, in Figure 5.1b, an open-loop planar manipulator has its end-effector reaching the same posture with two different configurations of the system, as illustrated by broken lines.

In the different cases, each configuration is referred to as a “closure” of the system. They result when the nonlinear posture equations have more than one set of real solutions. In order to precisely describe a configuration of such a system, it is necessary to specify additional information, over and above the generalized coordinates, to distinguish one closure of the system from other possible closures. Therefore, we define the posture of a system by a vector ϕ that explicitly includes *all* of the joint variables rather than just the vector of generalized coordinates ψ . It should be remembered here that some joints may require more than one joint variable, and all joint variables are included in the ϕ vector:

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix}. \quad (5.2)$$

These coordinates, of course, are not always independent. They may be related by Euler-Rodrigues constraints or by kinematic loop-closure equations, as will be explained later.

5.2 Consecutive Transformations

Just as in the construction of a real machine, we start by manufacturing the individual parts before joining them together. Also, in the construction of a mathematical model for a multibody system, we have begun by deriving analytic descriptions of the individual components, the bodies and the joints. We will now join these pieces together and show that, in combination, they provide a mathematical model that

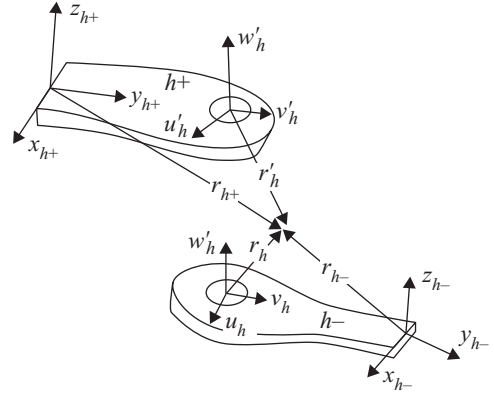


Figure 5.2. Two adjacent bodies, $h-$ and $h+$, connected by joint h .

simulates the real system. Then, in later chapters, we will be in a position to use this model to perform the various phases of mechanical analysis.

Let us start by showing how, in terms of our transformation matrices, we can join two bodies by means of a joint with label h . We recall from Eq. (4.6) that

$$r_h = \Phi_h(\phi_h)r'_h$$

where r_h and r'_h refer to the $u_h v_h w_h$ and $u'_h v'_h w'_h$ coordinates of the same point measured in two different joint coordinate systems: those attached to the mating joint elements of the two bodies connected by joint h .

Because we have allowed labeling of both bodies and joints in arbitrary orders, we now define the ligatures $h-$ and $h+$ to represent the labels of the two bodies that immediately precede (negatively incident with) and follow (positively incident with) the joint with label h , respectively, as specified by the chosen joint orientation. This not very elegant notation is necessary to avoid confusion because, by symbolism, joint label h does not infer a value for a body label, such as b or c , but does when written as $h-$ or $h+$. It is also important to emphasize that although h is a *joint* label, $h-$ and $h+$ are *body* labels. This is shown in Figure 5.2 where joint and body coordinate axes are attached to the joint elements and to each adjacent body using the notation of Chapter 4.

Now, applying Eq. (4.4) to each of the mating joint elements, we see that

$$r_{h-} = S_{h-,h}r_h,$$

and

$$r_{h+} = S_{h+,h}r'_h.$$

These are two more representations for the same point in the coordinate systems of the two bodies joined by joint h . It should be noted that the shape matrix $S_{h-,h}$ is the transformation between the coordinate systems $u_h v_h w_h$ and $x_{h-} y_{h-} z_{h-}$ and $S_{h+,h}$ is the transformation between the coordinate systems $u'_h v'_h w'_h$ and $x_{h+} y_{h+} z_{h+}$. This means that each body may contain more than one shape matrix, with one corresponding to each of its joint axes. Referring to Figure 5.2, for example, the body $h-$ may have one shape matrix $S_{h-,h}$ corresponding to joint h and another shape matrix $S_{h-,(h-1)}$ corresponding to a previous joint with label $(h - 1)$.

Once we recognize that r_{h-} and r_{h+} are merely different descriptions for the same point (see Figure 5.2) with no restriction on what point is chosen, and noting that r_h and r'_h are related by the joint transformation matrix $\Phi_h(\phi_h)$; that is, with $r_h = \Phi_h(\phi_h)r'_h$, we can combine these three equations into a single transformation as follows:

$$r_{h-} = S_{h-,h}\Phi_h S_{h+,h}^{-1}r_{h+}, \quad h = 1, 2, \dots, n. \quad (5.3)$$

Note here that the order of the matrix multiplications is important. Matrix multiplication is not commutative. Also, in general, a rotation of coordinate axes followed by a translation, for example, is not equal to the same translation followed by the same rotation; that is, $\Phi S \neq S\Phi$.

Equation (5.3) says that if we know the position vector of an arbitrary point in the $x_{h+}y_{h+}z_{h+}$ coordinate system of the body following joint h , we can find its position coordinates in the $x_{h-}y_{h-}z_{h-}$ coordinate system of the body preceding joint h . This concept of transforming to coordinates in the axes of a different body is extremely important, and is fundamental to the coming chapters. In other words, this equation represents the kinematic relationship between two adjacent bodies in a mechanism or a multibody system.

Before going further with our development, let us review our progress so far, and perhaps tie together a few loose ends. Recall that in Chapter 2 we went to great lengths to establish the path of each loop through the mechanism or the multibody system and, in so doing, we started by assigning an identifying label to each body and another to each joint, not necessarily in any particular order. We then used these labels in an algorithm for determining the kinematic loops of the system, and the kinematic paths from the fixed body to each of the moving bodies.

Then in Chapter 4 we defined a number of coordinate systems. Each body was assigned a body coordinate system $x_b y_b z_b$ that had a subscript b identifying the *body* label. Several joint coordinate systems $u_h v_h w_h$ were also defined and each of these carries a subscript h identifying the *joint* label to which it applies. In developing the transformation matrices, we found the joint matrix $\Phi_h(\phi_h)$ that describes the relative position of each joint and each of these has a subscript h denoting the label of the joint it describes. However, the shape matrix S_{bh} , which describes geometric characteristics of a body, rather than being subscripted by the body label alone, carries a double subscript; it shows first the *body* label b to which it belongs, and then the *joint* label h to which it applies. Also, a body has as many shape matrices as it has joint elements.

In Eq. (4.4), we have a means to transform the position coordinates of a point from any joint or auxiliary coordinate system to its appropriate body coordinate system. Then, in Eq. (5.3), we have a means to transform these to the position coordinates of the same point in the coordinate system of the preceding body of the kinematic loop or path. However, because this is possible, we can also apply the procedure recursively and transform to position coordinates on *any* previous body in the loop or path. If we presume for the moment that the bodies and joints are labeled consecutively along the paths and if, for example, we have position coordinates r_5 of some point attached to body 5 and wish to transform them to position coordinates

measured in body coordinate system 1, then by repeated applications of Eq. (5.3) we find that

$$r_1 = S_{1A} \Phi_A S_{2A}^{-1} S_{2B} \Phi_B S_{3B}^{-1} S_{3C} \Phi_C S_{4C}^{-1} S_{4D} \Phi_D S_{5D}^{-1} r_5.$$

Although the bodies and joints of a multibody system are usually not labeled in consecutive order, this equation shows in concept how, knowing the position vector of a point in the coordinate system of its own body, where it can be easily measured, we can find its position vector in the coordinate system of any other body by repetitive matrix transformations.

We notice in the above equations the continuing pattern of products of shape and joint and inverse shape transformation matrices. These products appear throughout our remaining work and are given their own symbol. For each joint h , connecting and oriented *from* body number $h-$ to body number $h+$, we define

$$T_{h-,h+} = S_{h-,h} \Phi_h S_{h+,h}^{-1}, \quad h = 1, 2, \dots, n. \quad (5.4)$$

However, we must be careful here. This definition can only be applied consistent with the defined orientation of each joint. If the opposite orientation is desired, then $T_{h+,h-} = T_{h-,h+}^{-1}$ must be used as shown in Eq. (5.15) below. This is not the same as $S_{h+,h} \Phi_h S_{h-,h}^{-1}$, but is equal to $S_{h+,h} \Phi_h^{-1} S_{h-,h}^{-1}$.

Then, for consecutive products of these transformations along the kinematic paths, we define

$$T_{bc} = T_{b,b+1} T_{b+1,b+2} \cdots T_{c-1,c}, \quad b, c = 0, 1, \dots, \ell. \quad (5.5)$$

In applying these equations, however, we must remember that the subscripts do not usually occur in numeric order as implied by the subscripts “ $b+1$,” “ $b+2$,” and “ $c-1$,” “ c ,” but in the order of appearance of the body labels in the kinematic paths and loops through the system as were determined in Chapter 2. This may not be the most elegant mathematical notation; however, the meaning is clear and comfortable to use for hand computation. For digital computation, a programming language should be chosen that has provision for hierarchical data structures. Body and joint data records may then be organized in linked lists (queues) in the order of the kinematic loops and paths.

We note that Eq. (5.5) also gives us a very convenient and compact notation because it shows that

$$T_{bc} T_{cd} = T_{bd}, \quad b, c, d = 0, 1, \dots, \ell. \quad (5.6)$$

Thus, if we know the transformation T_{bc} from body b to body c and the transformation T_{cd} from body c to body d , then their product gives the transformation from body b to body d . Even though the two paths may each include several bodies and joints, the total transformation – consisting of a product of several factors – is naturally collapsed into a single digraph. With this notation, the transformation between any two body (xyz) coordinate systems in the multibody system can be expressed by the simple equation

$$r_b = T_{bd} r_d, \quad b, d = 0, 1, \dots, \ell. \quad (5.7)$$

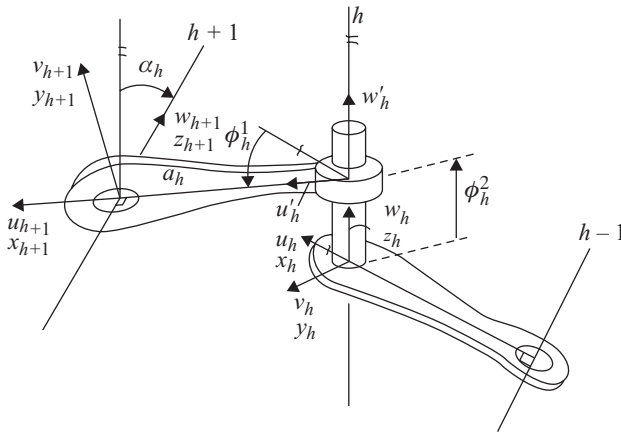


Figure 5.3. Notation for the Denavit-Hartenberg transformation.

5.3 Denavit-Hartenberg Transformations

A special form of the transformation matrix T_{bc} can be found if we choose particular postures for the joint and body coordinate systems in transforming from one joint to the next in a serial multibody system. This special transformation, named after its esteemed originators, is called the Denavit-Hartenberg (D-H) transformation [1, 2, 3]. It combines a consecutive joint matrix and shape matrix into a single (4×4) transformation and is well known in the mechanisms and robotics literature. It describes the geometry of a joint and the subsequent body by a set of four parameters known as Denavit-Hartenberg parameters. Here, we choose a cylindric joint to illustrate the concept because the most commonly used joints in robotic devices, namely revolute and prismatic joints, can be considered special cases of a cylindric joint. In the case of a cylindric joint the D-H parameters are the joint angle ϕ_h^1 , the joint offset ϕ_h^2 , the link length a_h , and the link twist angle α_h .

Consider a cylindric joint connecting two adjacent bodies of a mechanism as shown in Figure 5.3. The joint angle ϕ_h^1 and the joint offset ϕ_h^2 for joint h in this figure are the two joint variables for the cylindric joint. In the case of a revolute joint, the joint angle ϕ_h^1 is the joint variable and the joint offset $\phi_h^2 = d$ is a constant parameter, whereas for a prismatic joint, the joint angle $\phi_h^1 = \theta$ is a constant parameter and the joint offset ϕ_h^2 is the joint variable. In addition, the link shape requires two more constant parameters to uniquely define its kinematic characteristics. One of these constant parameters is a_h – the length of the common perpendicular between the two joint axes w_h and w_{h+1} . It should be noted that the length a_h is not always the physical length of the link, but is the length of the common perpendicular. In the D-H notation, the u_{h+1} axis is chosen as the extension of this common perpendicular; the v_{h+1} axis is then added to complete the coordinate system, $u_{h+1}v_{h+1}w_{h+1}$. The other constant parameter is the link twist α_h , which is the angle from joint axis w_h to joint axis w_{h+1} , and is positive when counterclockwise as viewed from the positive end of the u_{h+1} axis. When the coordinate systems $u_hv_hw_h$ and $u_{h+1}v_{h+1}w_{h+1}$ are chosen in this fashion, the $x_hy_hz_h$ body axes are chosen coincident with the $u_hv_hw_h$ axes. Two joint parameters, ϕ_h^1 and ϕ_h^2 , describe the relative geometry of axis x_{h+1}

with respect to axis x_h ; similarly, two constant link parameters, a_h and α_h , describe the relative geometry of axis z_{h+1} with respect to axis z_h .

The corresponding transformation between the coordinate systems $x_h y_h z_h$ and $x_{h+1} y_{h+1} z_{h+1}$ written in terms of the Denavit-Hartenberg (D-H) parameters is called the Denavit-Hartenberg transformation. Historically, this transformation has played a key role in the development of methods for kinematic analysis of three-dimensional mechanisms and robotic manipulators, but it can be considered a special case of the more general method developed earlier in this chapter.

For the two consecutive links shown in Figure 5.3, let us consider the joint coordinate systems $u_h v_h w_h$, $u'_h v'_h w'_h$, and $u_{h+1} v_{h+1} w_{h+1}$ as shown. The D-H transformation relates the postures of the coordinate systems $x_h y_h z_h$ and $x_{h+1} y_{h+1} z_{h+1}$. Here we have chosen the body coordinate systems for bodies h and $h + 1$ to be coincident with the joint coordinate systems $u_h v_h w_h$ and $u_{h+1} v_{h+1} w_{h+1}$, respectively, so that $x_h y_h z_h = u_h v_h w_h$ and $x_{h+1} y_{h+1} z_{h+1} = u_{h+1} v_{h+1} w_{h+1}$. In this manner, $S_{h-,h} = I$ and the D-H transformation can be derived using Eq. (5.4) as:

$$\mathbb{T}_{h-,h+} = S_{h-,h} \Phi_h S_{h+,h}^{-1} = I \Phi_h S_{h+,h}^{-1} = \Phi_h S_{h+,h}^{-1}.$$

In this last equation, the term Φ_h is the transformation for a cylindric joint. The transformation $S_{h+,h}^{-1}$ is the coordinate transformation between coordinate systems $x_{h+1} y_{h+1} z_{h+1}$ and $u'_h v'_h w'_h$ or the inverse of the transformation between the coordinate systems $u'_h v'_h w'_h$ and $x_{h+1} y_{h+1} z_{h+1}$.

The coordinate transformation for a cylindric joint was derived in Eq. (4.14) as:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos \phi_h^1 & -\sin \phi_h^1 & 0 & 0 \\ \sin \phi_h^1 & \cos \phi_h^1 & 0 & 0 \\ 0 & 0 & 1 & \phi_h^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In order to derive the transformation $S_{h+,h}^{-1}$ in terms of the D-H parameters, we use a screw displacement from the coordinate system $u'_h v'_h w'_h$ to the coordinate system $x_{h+1} y_{h+1} z_{h+1}$. Referring to Figure 5.3, the screw axis for this displacement is the u'_h axis and the screw parameters are $d = a_h$ and $\theta = \alpha_h$. Now, using Eqs. (3.40) and (3.41), we find:

$$S_{h+,h}^{-1} = \begin{bmatrix} 1 & 0 & 0 & a_h \\ 0 & \cos \alpha_h & -\sin \alpha_h & 0 \\ 0 & \sin \alpha_h & \cos \alpha_h & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The resulting total D-H transformation is given by:

$$\mathbb{T}_{h-,h+} = \begin{bmatrix} \cos \phi_h^1 & -\sin \phi_h^1 & 0 & 0 \\ \sin \phi_h^1 & \cos \phi_h^1 & 0 & 0 \\ 0 & 0 & 1 & \phi_h^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_h \\ 0 & \cos \alpha_h & -\sin \alpha_h & 0 \\ 0 & \sin \alpha_h & \cos \alpha_h & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

or

$$\mathbb{T}_{h-,h+} = \begin{bmatrix} \cos \phi_h^1 & -\sin \phi_h^1 \cos \alpha_h & \sin \phi_h^1 \sin \alpha_h & a_h \cos \phi_h^1 \\ \sin \phi_h^1 & \cos \phi_h^1 \cos \alpha_h & -\cos \phi_h^1 \sin \alpha_h & a_h \sin \phi_h^1 \\ 0 & \sin \alpha_h & \cos \alpha_h & \phi_j^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.8)$$

It should be noted that this D-H matrix is a (4×4) homogenous transformation and acts like the transformation defined in Eq. (5.4) relating points in the coordinate system of one body with respect to the adjacent body. This means that transformations along the kinematic path defined by Eq. (5.5) can be established in the same manner using $\mathbb{T}_{h-,h+}$ matrices for each joint connecting adjacent bodies of a multibody system. This type of matrix is commonly used in much of the robotics literature. In this book, however, we utilize the more general matrix notation of Eqs. (5.4) and (5.5). This not only provides a more general approach, with much less need for conventions on the placement of axes, but also avoids difficulties that sometimes arise in use of the D-H matrices. Such difficulties occur when consecutive joint axes intersect or are parallel and, therefore, allow arbitrary choices for some of the coordinate axes, or when adjacent joint axes are nearly parallel causing difficulty in locating common perpendiculars and causing some joint offset values to become very large. Ambiguity in notation also occurs when applying the D-H transformation to multi-loop systems [12].

5.4 Absolute Position

The absolute position of a point on a moving body is often of major importance in a design situation. Obviously, this can be found by applying Eq. (5.7) to transform the local coordinates of the point, measured in the coordinate system of the body to which the point is attached. However, to which coordinate system should the position vector be transformed? Which xyz coordinate system is the primary frame of reference?

To retain maximum flexibility, we choose a coordinate system $x_0y_0z_0$ that we define to be the *absolute* or *world* coordinate system. The only restriction we place on the choice of this system is that it be an *inertial* coordinate system if the analysis is to include dynamic effects. Otherwise, it is chosen totally arbitrarily to fit the problem. This absolute coordinate system, carrying the subscript zero, is the one in which all results are found. Positions measured in this coordinate system are called *absolute positions*.

Reviewing section 4.5 reminds us that we reserved the symbol R_b to stand for the absolute position vector of a point of the body with label b :

$$R_b = \begin{bmatrix} R_b^{x_0} \\ R_b^{y_0} \\ R_b^{z_0} \\ 1 \end{bmatrix}, \quad b = 1, 2, \dots, \ell, \quad (5.9)$$

and, from Eq. (4.2),

$$R_b = T_{0b}r_b, \quad b = 1, 2, \dots, \ell. \quad (5.10)$$

Recalling that the frame member of our multibody system is numbered body ℓ as recommended in Chapter 2, we have already collected data for the transformation matrix $T_{0\ell}$ while we were collecting data for the shape matrices. If body ℓ is stationary, then the $T_{0\ell}$ matrix remains constant as the system moves. If not, as in Figure 2.8, it is variable; however, we have data for it at the initial position, and we may wish to update $T_{0\ell}$ at later values of time. In either case, we can write

$$R_b = T_{0\ell}T_{\ell b}r_b, \quad b = 0, 1, \dots, \ell, \quad (5.11)$$

where $T_{\ell b}$ can be computed as shown in Eq. (5.5) according to the path found in Chapter 2 from body ℓ to body b .

5.5 The Loop-closure Equation (Kinematic Equation for Position Analysis)

Because Eq. (5.7) holds for the position of a point on bodies b and c , suppose that we choose to apply this to a multibody system containing one or more closed kinematic loops such as those discovered in the loop matrix of Chapter 2. Suppose also that we choose b to represent a certain body in loop i , then trace through the bodies and joints in sequence until we meet body c . If we continue around the loop until it closes on itself, we can write Eq. (5.7) with bodies c and b being the same body. In this case we have

$$r_b = T_{bb}r_b, \quad i = 1, 2, \dots, NL,$$

or, transforming to the absolute coordinate system,

$$R_b = T_{0b}T_{bb}T_{b0}R_b = T_{00}R_b, \quad i = 1, 2, \dots, NL,$$

where the matrix T_{00} is the product of transformations from ground to the first body of the loop, then around the loop, and then back to ground.

Because we have made no restrictions on the choice of the point R_b , this equation must hold for all possible choices of the point. This condition can only be satisfied in general if $T_{bb} = I$ when we work in the coordinate system of some arbitrary body b , or

$$T_{00} = T_{0b}T_{bb}T_{b0} = I, \quad i = 1, 2, \dots, NL, \quad (5.12)$$

when we work in global coordinates.

This equation is the transformation matrix form of the loop-closure or kinematic equation for position analysis. It shows that no matter which body is chosen as a starting body, when consecutive transformations are made around a closed kinematic loop, finally returning to the original coordinate system, this matrix product must equal the identity transformation – the (4×4) identity matrix.

This loop-closure equation inherently contains all of the kinematic constraint conditions that must be satisfied by a multibody system containing closed kinematic loops such that the loops remain closed for all possible postures of the system. As the system moves and the joint variables change, they must all change in unison such

that this equation is always satisfied for each and every closed loop. This extremely powerful equation can be used, as will be shown in this and subsequent chapters, to derive all the kinematic position information possible for the multibody system.

In the case of an open-loop multibody system, such as many serial robotic manipulators, the development follows in a similar fashion. In this case, however, the starting body is typically chosen to be the world or base coordinate system and then consecutive transformations are made until reaching the end-effector coordinate system. For a robotic manipulator with six joints and six moving links numbered in sequence, the resulting transformation can be written:

$$T_{01}T_{12}T_{23}T_{34}T_{45}T_{56}T_{67} = T_{07}. \quad (5.13)$$

The matrix T_{07} is the transformation describing the posture of the end-effector coordinate system. In Chapter 3, we discussed several ways for specifying this T_{07} transformation. One example was to use direction cosines of three mutually orthogonal coordinate axes in the end-effector as was discussed in Eq. (3.26). Using this equation, we have:

$$T_{07} = \begin{bmatrix} u^x & v^x & w^x & R_7^{x_0} \\ u^y & v^y & w^y & R_7^{y_0} \\ u^z & v^z & w^z & R_7^{z_0} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is useful to point out that if we take the open-loop system described by kinematic Eq. (5.13) and convert it into a closed-loop system by connecting its last body or end-effector to ground by some kinematic joint such as an open joint – making it into a closed-loop system with seven joints instead of an open-loop system with six joints – then the loop-closure or kinematic equation becomes:

$$T_{01}T_{12}T_{23}T_{34}T_{45}T_{56}T_{67}T_{70} = I. \quad (5.14)$$

Comparing Eqs. (5.14) and (5.13), it is easy to see that $T_{70} = T_{07}^{-1}$. In other words, the kinematics of the two systems are governed by similar equations. This is why the kinematics of a six-revolute serial robotic manipulator is considered to be similar to that of a seven-joint closed kinematic chain or mechanism. Another way of stating this is to say that when we specify the posture of the final body of an open-loop multibody system (for example, the end-effector of a robotic manipulator) and require it to perform a specified motion, then, in a sense, we are defining a constraint that is similar to closing a loop in the system.

Another very useful identity can be obtained from the loop-closure equation by taking into account the recursive nature of the products of the transformation matrices. From Eq. (5.6) we have

$$T_{bc}T_{cb} = T_{bb} = I.$$

By pre-multiplying this equation by the inverse of T_{bc} , we obtain

$$T_{cb} = T_{bc}^{-1}, \quad b, c = 0, 1, \dots, \ell. \quad (5.15)$$

Before continuing, it should be pointed out that, because of the special properties of rigid body transformations, the inverse of such a transformation matrix is

extremely simple to form. If we choose the following notation for the translation and rotation submatrices,

$$T_{bc} = \begin{bmatrix} \Theta_{bc} & r_{O_c} \\ 0 & 1 \end{bmatrix},$$

then, because of the orthogonality of the rotation submatrix, it is easily verified that the inverse matrix is of the form

$$T_{bc}^{-1} = \begin{bmatrix} \Theta_{bc}^t & -\Theta_{bc}^t r_{O_c} \\ 0 & 1 \end{bmatrix}. \quad (5.16)$$

This explicit form of the inverse is very easily evaluated, even by hand computation. It is also easily programmed for digital evaluation, and operates much more economically than a general matrix inversion technique.

The Denavit-Hartenberg transformation of Eq. (5.8) also enjoys the same property and, in a similar fashion, its inverse is found to be

$$T_{h^-,h^+}^{-1} = \begin{bmatrix} \cos \phi_h^1 & \sin \phi_h^1 & 0 & -a_h \\ -\sin \phi_h^1 \cos \alpha_h & \cos \phi_h^1 \cos \alpha_h & \sin \alpha_h & -\phi_h^2 \sin \alpha_h \\ \sin \phi_h^1 \sin \alpha_h & -\cos \phi_h^1 \sin \alpha_h & \cos \alpha_h & \phi_h^2 \cos \alpha_h \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.17)$$

5.6 Closed-form Solution of Kinematic Equations for Joint-variable Positions

According to the previous discussion, we see that in order to solve for the values of the joint variables ϕ of a mechanism or a multibody system, we must first specify the generalized coordinates ψ – the values of the f independent joint variables. We may also be required to specify additional information if necessary to distinguish between multiple closures or postures of the system.

Those with experience in applying classical graphic techniques may consider that determining the posture of a system is a trivial problem because, once a layout drawing is completed, any of the joint variables can be determined directly from the drawing. For example, see [13], especially the first paragraph of section 3.7, p. 52, where this is stated explicitly.

On the contrary, as we will come to appreciate in the remainder of this chapter, the determination of the position values of the dependent joint variables for a given set of generalized coordinates is usually the most difficult problem of kinematic analysis. Once the dependent position values are known, the remaining problems of kinematics are relatively easily solved. This observation holds true for most analytic methods and comes about primarily as a result of the nonlinear nature of the constraints. The rigid-body condition, that two points of a body remain a constant distance apart, as well as the rotational motions found in many types of joints, invariably leads to quadratic and/or trigonometric equations. Because there must be at least as many of these equations as unknown joint variables, their simultaneous solution can be quite challenging.

However, there are some systems – usually open-loop or single-loop systems, where the numbers of bodies and joints are small – for which it is possible to obtain closed-form solutions. We shall present a few examples here to illustrate the approach. In general, the procedure is always the same; the necessary simultaneous equations are found by performing the matrix multiplications indicated in the loop-closure equation, Eq. (5.12), and these are then solved by whatever algebraic techniques the analyst is able to muster. More on methods for finding the solution of the kinematic position equations is given in section 5.7.

The matrix loop-closure equation produces all of the simultaneous equations necessary for the solution of a given problem and the method of formulation or the number of equations is not left to chance. None of the essential equations is overlooked, as might happen if a given problem is formulated intuitively. However, there is no assurance that their solution by hand calculation is feasible.

EXAMPLE 5.1: CARDAN/HOOKE UNIVERSAL SHAFT COUPLING We choose the universal shaft coupling as a first example because solutions known from other methods allow comparison of results. The Cardan/Hooke universal shaft coupling, shown in Figure 5.4, is a spherical linkage consisting of four bodies connected by four revolute joints. It also has two other distinguishing features: first, that all four revolute axes intersect at a common point, and second, that three of the four fixed angles at these intersections are right angles. This mechanism is known both as a Cardan coupling and as a Hooke universal joint. An Italian mathematician, physician, and astrologer, Girolamo Cardano (1501–76) was the first to describe this joint, although it is unclear whether he ever constructed one. An English architect and natural philosopher, Robert Hooke (1635–1703) first applied it to the transmission of rotary motion.

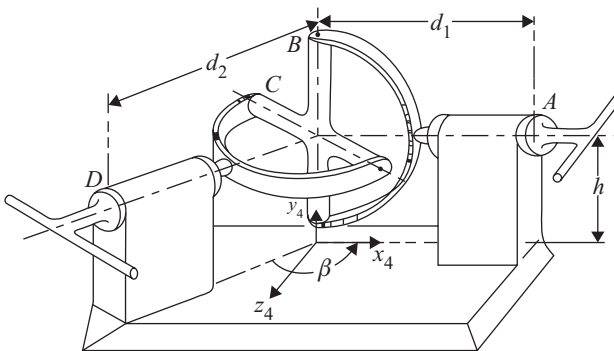


Figure 5.4. Example 5.1: Cardan/Hooke universal shaft coupling.

It is assumed here that body 3, mounted at joint D , is the input shaft and its rotation $\psi = \phi_D$ is a given function of time. The output shaft, body 1, has rotation ϕ_A , whose functional relationship to ψ must be determined. We also note that solutions for the dependent joint variables ϕ_B and ϕ_C , although not of primary concern, may also be useful for predicting wear or choosing proper bearings.

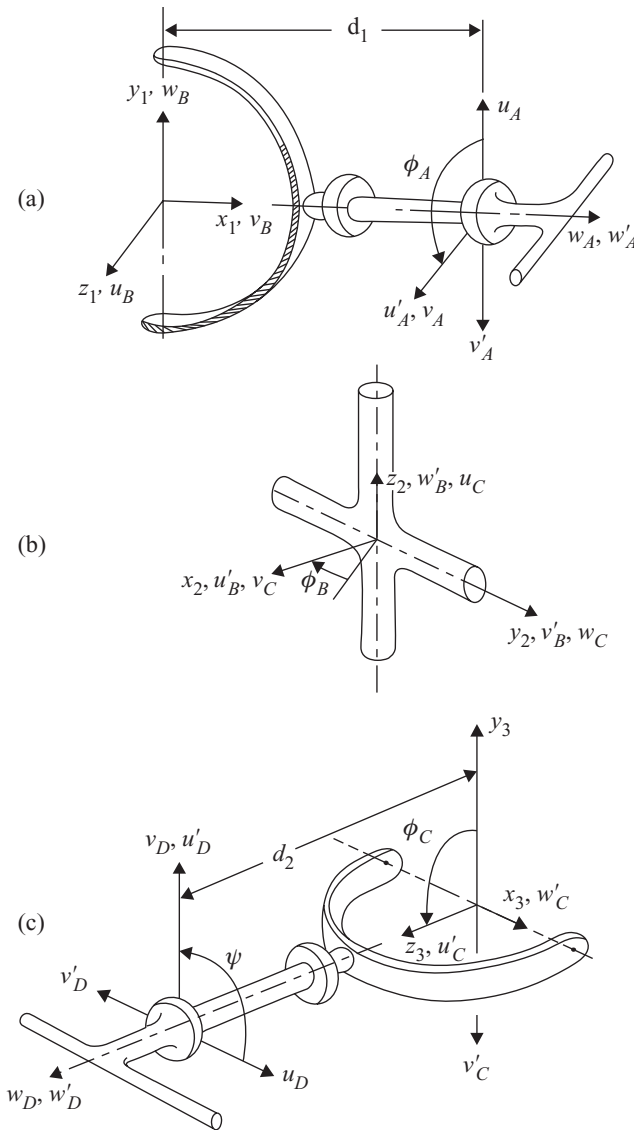


Figure 5.5. Example 5.1: Placement of body and joint coordinate systems for (a) body 1, (b) body 2, (c) body 3.

In this example it is not necessary to perform a formal topological analysis, as discussed in Chapter 2, because the single loop is easily determined by inspection. The loop matrix is

$$L = \begin{bmatrix} 1 & 1 & 1 & 1 \\ A & B & C & D \end{bmatrix}.$$

With the body and joint axes chosen as shown in Figure 5.4 and Figure 5.5, the shape matrices can be found, either directly by inspection of the figures or

by the methods of section 4.4. They are:

$$T_{04} = I, S_{4,A} = \begin{bmatrix} 0 & 0 & 1 & d_1 \\ 1 & 0 & 0 & h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{1,A} = \begin{bmatrix} 0 & 0 & 1 & d_1 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S_{1,B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{2,B} = I,$$

$$S_{2C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{3C} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$S_{3D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{4D} = \begin{bmatrix} 0 & -\sin\beta & \cos\beta & d_2\cos\beta \\ 1 & 0 & 0 & h \\ 0 & \cos\beta & \sin\beta & d_2\sin\beta \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Because all four joints are revolute, Eq. (4.10) can be used to evaluate each of the joint matrices:

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos\phi_h & -\sin\phi_h & 0 & 0 \\ \sin\phi_h & \cos\phi_h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad h = A, B, C, D.$$

Next, following Eq. (5.4), we evaluate the four matrix products. Notice that joint D is defined here to be oriented from body 3 to body 4. Thus, the T_{34} matrix is formulated rather than T_{43} . See example 5.2 for contrast:

$$T_{41} = S_{4A}\Phi_A S_{1A}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin\phi_A & \cos\phi_A & h \\ 0 & -\cos\phi_A & \sin\phi_A & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{12} = S_{1B}\Phi_B S_{2B}^{-1} = \begin{bmatrix} \sin\phi_B & \cos\phi_B & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cos\phi_B & -\sin\phi_B & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{23} = S_{2C}\Phi_C S_{3C}^{-1} = \begin{bmatrix} 0 & -\cos\phi_C & \sin\phi_C & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \sin\phi_C & \cos\phi_C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{34} = S_{3D} \Phi_D S_{4D}^{-1} = \begin{bmatrix} \sin \psi \sin \beta & \cos \psi & -\sin \psi \cos \beta & -h \cos \psi \\ -\cos \psi \sin \beta & \sin \psi & \cos \psi \cos \beta & -h \sin \psi \\ \cos \beta & 0 & \sin \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Based on these four transformations, we can write the loop-closure equation from Eq. (5.12):

$$T_{41} T_{12} T_{23} T_{34} = I.$$

However, in this particular problem, it is more convenient to rewrite this in the form

$$T_{34} T_{41} = (T_{12} T_{23})^{-1},$$

because T_{12} and T_{23} have no translation terms and, therefore, the matrix inversion can be done very easily by Eq. (5.16).

Taking advantage of this, the previous equation becomes

$$\begin{bmatrix} \sin \psi \sin \beta & \cos \psi \sin \phi_A + \sin \psi \cos \phi_A \cos \beta & \cos \psi \cos \phi_A - \sin \psi \sin \phi_A \cos \beta & 0 \\ -\cos \psi \sin \beta & \sin \psi \sin \phi_A - \cos \psi \cos \phi_A \cos \beta & \sin \psi \cos \phi_A + \cos \psi \sin \phi_A \cos \beta & 0 \\ \cos \beta & -\cos \phi_A \sin \beta & \sin \phi_A \sin \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \phi_B & 0 & -\sin \phi_B & 0 \\ -\sin \phi_B \cos \phi_C & \sin \phi_C & -\cos \phi_B \cos \phi_C & 0 \\ \sin \phi_B \sin \phi_C & \cos \phi_C & \cos \phi_B \sin \phi_C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Because both sides of this equation must be equal, we may now equate various individual terms to discover the relationships between the input variable ψ and the remaining dependent joint variables ϕ_j . The input-output relationship, for example, is found by equating the elements of the first row, second column. This gives

$$\phi_A = \tan^{-1} (-\tan \psi \cos \beta).$$

The value of ϕ_B is found directly from row one, column one

$$\phi_B = \cos^{-1} (\sin \psi \sin \beta),$$

and dividing the elements of row three, column one by those of row two, column one we find an equation for ϕ_C in terms of ψ

$$\phi_C = \tan^{-1} \left(\frac{1}{\cos \psi \tan \beta} \right).$$

As discussed earlier, we can group these solutions into a single vector that describes the posture (configuration) of the system as a function of the single generalized coordinate ψ

$$\phi = \begin{bmatrix} \phi_A \\ \phi_B \\ \phi_C \\ \phi_D \end{bmatrix} = \begin{bmatrix} \tan^{-1}(-\tan\psi \cos\beta) \\ \cos^{-1}(\sin\psi \sin\beta) \\ \tan^{-1}\{1/(\cos\psi \tan\beta)\} \\ \psi \end{bmatrix}.$$

Although we may be tempted to stop here, thinking this to be a complete solution, we should note that the quadrants of ϕ_A , ϕ_B , and ϕ_C are not yet determined. We conclude, however, that because the Cardan/Hooke universal joint can only be rotated without interference in situations where β is an obtuse angle, $\cos\beta$ is always negative. Also, we note from [Figure 5.5b](#) that $\sin\phi_B$ always remains negative. Using this information, we can determine from the remaining elements of the previous matrix equation that $\sin\phi_C$ is always positive and that $\cos\phi_A$ always carries the same sign as $\cos\psi$. Thus, all quadrants can be determined.

The reason that the quadrants could not be discovered directly from the equations and that the figure had to be consulted for clues is that there is more than one way in which this system can be assembled. The same equations also describe the other possible closure, where the cross is rotated 180° about its y_2 axis and the output shaft is rotated 180° about y_1 . In order to remove this ambiguity – to specify which closure is being analyzed – it is necessary to note that $\sin\phi_B$ remains negative.

The foregoing analysis only applies to a “perfect” Cardan/Hooke universal joint; that is, one in which the joint axes truly intersect and do so in exact right angles. If the angles are not exactly 90° , the situation is not too bad because the only effect on the previous equations is small. If, however, the axes do not truly intersect, the linkage is no longer spherical. Such a case might happen, for example, if the distances d_1 and d_2 appearing in the shape matrices of the frame were slightly different than those in the shape matrices of the output and input shafts. The analysis would proceed in the same fashion; however, the elements of the fourth column of the matrix products would not produce exact identities. At best, they might be equal for only discrete values of ψ rather than for a continuous range of input motion. Also, because real machine parts are always designed with tolerances because of imperfect manufacture, the real universal joint (as manufactured) might not exactly match the dimensions used in the previous equations. Yet it might still be movable, if only because of looseness in the revolute joint bearings and slight flexing of the “rigid” parts.

EXAMPLE 5.2: DISK-CAM SYSTEM As a second example, let us choose the disk-cam with reciprocating flat-faced follower system shown in [Figure 5.6](#).

Again the loops and paths are obvious by inspection. The loop matrix is

$$L = \begin{bmatrix} 1 & 1 & -1 \\ A & B & C \end{bmatrix}.$$

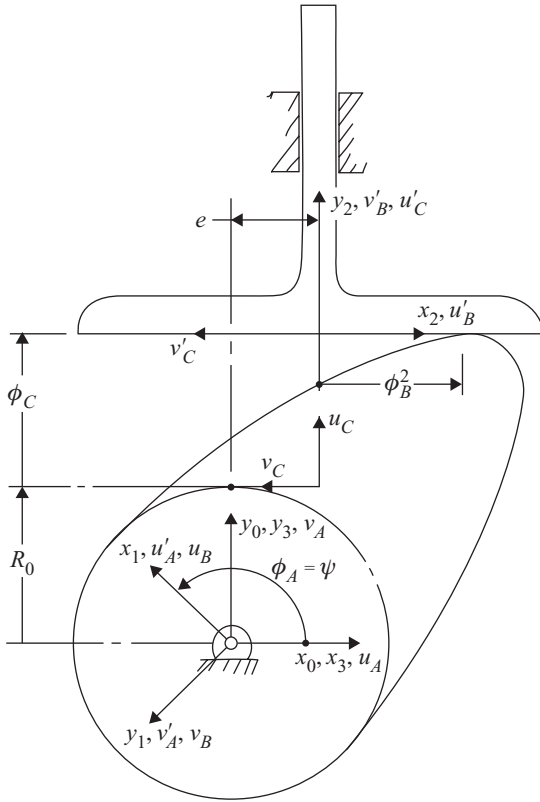


Figure 5.6. Example 5.2: Disk-cam with reciprocating flat-faced follower.

The coordinate systems are chosen as shown in [Figure 5.6](#). The shape matrices, also obtained by inspection, are as follows:

$$T_{03} = S_{1A} = S_{1B} = S_{2B} = S_{3A} = I,$$

$$S_{2C} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{3C} = \begin{bmatrix} 0 & -1 & 0 & e \\ 1 & 0 & 0 & R_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The shape of the cam is defined by the following functions:

$$u(\phi_B^1) = \begin{cases} \left[R_0 + \frac{L}{2\pi} (2\phi_B^1 - \sin 2\phi_B^1) \right] \sin \phi_B^1 + \frac{L}{\pi} (1 - \cos 2\phi_B^1) \cos \phi_B^1, & 0 \leq \phi_B^1 < \pi, \\ \left[R_0 + \frac{L}{2\pi} (4\pi - 2\phi_B^1 + \sin 2\phi_B^1) \right] \sin \phi_B^1 + \frac{L}{\pi} (1 - \cos 2\phi_B^1) \cos \phi_B^1, & \pi \leq \phi_B^1 < 2\pi, \end{cases}$$

$$v(\phi_B^1) = \begin{cases} \left[R_0 + \frac{L}{2\pi} (2\phi_B^1 - \sin 2\phi_B^1) \right] \cos \phi_B^1 - \frac{L}{\pi} (1 - \cos 2\phi_B^1) \sin \phi_B^1, & 0 \leq \phi_B^1 < \pi, \\ \left[R_0 + \frac{L}{2\pi} (4\pi - 2\phi_B^1 + \sin 2\phi_B^1) \right] \cos \phi_B^1 - \frac{L}{\pi} (1 - \cos 2\phi_B^1) \sin \phi_B^1, & \pi \leq \phi_B^1 < 2\pi. \end{cases}$$

From these, using Eq. (4.40) to model cam joint B as a line tangent to a planar curve joint, we find, after some effort, that

$$\theta = \tan^{-1} \left[\frac{\frac{d}{d\phi_B^1} v(\phi_B^1)}{\frac{d}{d\phi_B^1} u(\phi_B^1)} \right] = \tan^{-1} (-\tan \phi_B^1) = -\phi_B^1,$$

and

$$\Phi_B(\phi_B) = \begin{bmatrix} \cos \phi_B^1 & \sin \phi_B^1 & 0 & u(\phi_B^1) - \phi_B^2 \cos \phi_B^1 \\ -\sin \phi_B^1 & \cos \phi_B^1 & 0 & v(\phi_B^1) + \phi_B^2 \sin \phi_B^1 \\ 0 & 0 & 1 & \phi_B^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Joints A and C are revolute and prismatic and are modeled by Eqs. (4.10) and (4.12), respectively.

Because the cam is driven by $\psi = \phi_A$, we can now formulate the following matrix products:

$$T_{31} = S_{3A} \Phi_A S_{1A}^{-1} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{12} = S_{1B} \Phi_B S_{2B}^{-1} = \begin{bmatrix} \cos \phi_B^1 & \sin \phi_B^1 & 0 & u(\phi_B^1) - \phi_B^2 \cos \phi_B^1 \\ -\sin \phi_B^1 & \cos \phi_B^1 & 0 & v(\phi_B^1) + \phi_B^2 \sin \phi_B^1 \\ 0 & 0 & 1 & \phi_B^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{23} = T_{32}^{-1} = S_{2C} \Phi_C^{-1} S_{3C}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -e \\ 0 & 1 & 0 & -(R_0 + \phi_C) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Notice here that joint B is oriented from body 1 to body 2 and the matrix Φ_B is used to find T_{12} . However, joint C is oriented from body 3 to body 2 and Φ_C^{-1} is required in finding T_{23} . We might, instead, find T_{32} using Φ_C and invert it. Either way, an inversion is required in one or the other because the orientations of joints B and C are not both consistent with the loop orientation. Another approach would be to reorganize the loop-closure equation itself, as was done in example 5.1; here we show the alternative approach.

The loop-closure equation, (5.12), gives

$$T_{33} = T_{31}T_{12}T_{23} = I,$$

$$\begin{bmatrix} \cos(\psi - \phi_B^1) & -\sin(\psi - \phi_B^1) & 0 & u(\phi_B^1)\cos\psi - v(\phi_B^1)\sin\psi - (\phi_B^2 + e) \\ & & \cos(\psi - \phi_B^1) + (R_0 + \phi_C)\sin(\psi - \phi_B^1) & \\ \sin(\psi - \phi_B^1) & \cos(\psi - \phi_B^1) & 0 & u(\phi_B^1)\sin\psi - v(\phi_B^1)\cos\psi - (\phi_B^2 + e) \\ & & \sin(\psi - \phi_B^1) - (R_0 + \phi_C)\cos(\psi - \phi_B^1) & \\ 0 & 0 & 1 & \phi_B^3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I.$$

By equating the elements of rows 1 and 2, column 1, with those of the identity matrix, we find

$$\cos(\psi - \phi_B^1) = 1, \quad \sin(\psi - \phi_B^1) = 0, \quad \phi_B^1 = \psi.$$

Then, equating the elements of column 4 and simplifying gives

$$\begin{aligned} u(\phi_B^1)\cos\psi - v(\phi_B^1)\sin\psi - (\phi_B^2 + e) &= 0, \\ u(\phi_B^1)\sin\psi + v(\phi_B^1)\cos\psi - (R_0 + \phi_C) &= 0, \\ \phi_B^3 &= 0. \end{aligned}$$

Finally, rearranging and further simplifying, the solutions for the joint variables are

$$\phi = \begin{bmatrix} \phi_A \\ \phi_B^1 \\ \phi_B^2 \\ \phi_B^3 \\ \phi_C \end{bmatrix} = \begin{bmatrix} \psi \\ \psi \\ \frac{L}{\pi}(1 - \cos 2\psi) - e \\ 0 \\ \begin{cases} \frac{L}{2\pi}(2\psi - \sin 2\psi), & 0 \leq \psi < \pi \\ \frac{L}{2\pi}(4\pi - 2\psi + \sin 2\psi), & \pi \leq \psi < 2\pi \end{cases} \end{bmatrix}.$$

Taking these solutions back to Eq. (5.11), we find the transformation matrices that show the absolute positions of the two moving bodies. These are

$$T_{01} = T_{03}T_{31} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$T_{02} = T_{03} T_{31} T_{12} = \begin{bmatrix} 1 & 0 & 0 & e \\ 0 & 1 & 0 & \begin{cases} R_0 + \frac{L}{2\pi} (2\psi - \sin 2\psi) & 0 \leq \psi < \pi \\ R_0 + \frac{L}{2\pi} (4\pi - 2\psi + \sin 2\psi) & \pi \leq \psi < 2\pi \end{cases} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

EXAMPLE 5.3: SCARA ROBOT For the next example, we choose the Selective Compliant Articulated Robot for Assembly (SCARA robot), shown in Figure 5.7.



Figure 5.7. Example 5.3: Selective Compliant Articulated Robot for Assembly (SCARA). Adept model Cobra 600 (Courtesy of Adept Technology, Inc., Livermore, CA).

The body and joint labels and the coordinate systems chosen for this robot are shown in Figure 5.8. Note that this problem has no closed loops and that, contrary to our usual convention, the fixed body here is labeled body 1. The shape matrices, obtained by inspection, are as follows:

$$T_{01} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{1A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{2A} = I,$$

$$S_{2B} = \begin{bmatrix} -1 & 0 & 0 & d_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{3B} = \begin{bmatrix} -1 & 0 & 0 & d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{3C} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

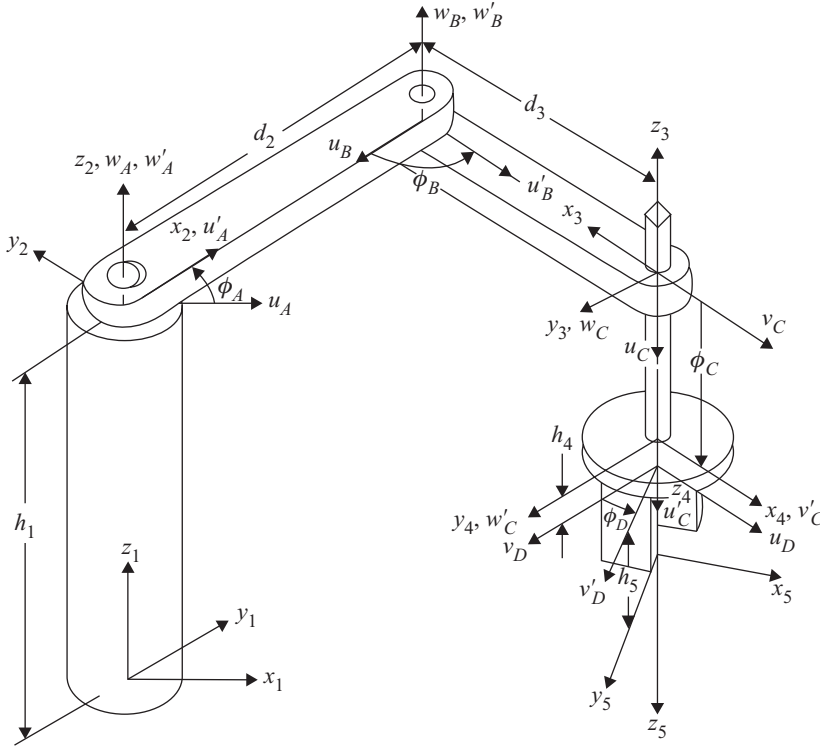


Figure 5.8. Example 5.3: Placement of body and joint coordinate systems for SCARA robot.

$$S_{4C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{4D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{5D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Recognizing that joint C is prismatic and all others are revolute, we formulate the matrix products

$$T_{12} = S_{1A} \Phi_A S_{2A}^{-1} = \begin{bmatrix} \cos \phi_A & -\sin \phi_A & 0 & 0 \\ \sin \phi_A & \cos \phi_A & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{23} = S_{2B} \Phi_B S_{3B}^{-1} = \begin{bmatrix} \cos \phi_B & -\sin \phi_B & 0 & d_2 + d_3 \cos \phi_B \\ \sin \phi_B & \cos \phi_B & 0 & d_3 \sin \phi_B \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{34} = S_{3C} \Phi_C S_{4C}^{-1} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -\phi_C \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{45} = S_{4D} \Phi_D S_{5D}^{-1} = \begin{bmatrix} \cos \phi_D & -\sin \phi_D & 0 & 0 \\ \sin \phi_D & \cos \phi_D & 0 & 0 \\ 0 & 0 & 1 & h_4 + h_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Next, multiplying these together along the path, we get

$$T_{02} = T_{01} T_{12} = \begin{bmatrix} \cos \phi_A & -\sin \phi_A & 0 & x_0 \\ \sin \phi_A & \cos \phi_A & 0 & y_0 \\ 0 & 0 & 1 & z_0 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{03} = T_{02} T_{23} = \begin{bmatrix} \cos(\phi_A + \phi_B) & -\sin(\phi_A + \phi_B) & 0 & x_0 + d_2 \cos \phi_A + d_3 \\ & & & \cos(\phi_A + \phi_B) \\ \sin(\phi_A + \phi_B) & \cos(\phi_A + \phi_B) & 0 & y_0 + d_2 \sin \phi_A + d_3 \\ & & & \sin(\phi_A + \phi_B) \\ 0 & 0 & 1 & z_0 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{04} = T_{03} T_{34} = \begin{bmatrix} -\cos(\phi_A + \phi_B) & -\sin(\phi_A + \phi_B) & 0 & x_0 + d_2 \cos \phi_A + d_3 \\ & & & \cos(\phi_A + \phi_B) \\ -\sin(\phi_A + \phi_B) & \cos(\phi_A + \phi_B) & 0 & y_0 + d_2 \sin \phi_A + d_3 \\ & & & \sin(\phi_A + \phi_B) \\ 0 & 0 & -1 & z_0 + h_1 - \phi_C \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{05} = T_{04} T_{45}$$

$$= \begin{bmatrix} -\cos(\phi_A + \phi_B - \phi_D) & -\sin(\phi_A + \phi_B - \phi_D) & 0 & x_0 + d_2 \cos \phi_A + d_3 \\ & & & \cos(\phi_A + \phi_B) \\ -\sin(\phi_A + \phi_B - \phi_D) & \cos(\phi_A + \phi_B - \phi_D) & 0 & y_0 + d_2 \sin \phi_A + d_3 \\ & & & \sin(\phi_A + \phi_B) \\ 0 & 0 & -1 & z_0 + h_1 - (\phi_C \\ & & & + h_4 + h_5) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This last matrix tells the absolute posture, including orientation, of the gripper coordinate system $x_5 y_5 z_5$, with respect to the $x_0 y_0 z_0$ coordinate system. If, for example, the tip of the tool held by the gripper is located at the position

$$r_5 = \begin{bmatrix} 0 \\ 0 \\ 4.0 \text{ in} \\ 1 \end{bmatrix},$$

relative to body 5, then its absolute position is given by

$$R_5 = T_{05}r_5 = \begin{bmatrix} x_0 + d_2 \cos \phi_A + d_3 \cos(\phi_A + \phi_B) \\ y_0 + d_2 \sin \phi_A + d_3 \sin(\phi_A + \phi_B) \\ z_0 + h_1 - (\phi_C + h_4 + h_5 + 4.0 \text{ in}) \\ 1 \end{bmatrix}.$$

We notice that, in this problem, there is no closed loop. All joint variables are independent; that is, the system has mobility four. Once four joint motions are specified as functions of time, then all other position information is found as shown here. This is the problem that the robotics community refers to as the *direct* or *forward kinematics* problem.

EXAMPLE 5.4: INVERSE KINEMATICS OF THE SCARA ROBOT For our next example, let us consider the problem of programming the SCARA robot of the previous example to perform a specified motion. This is the problem that roboticists term the *inverse kinematics* problem.

Suppose, for example, that we wish the robot of example 5.3 to cause the tip of the tool held by the gripper to travel *without rotation* along a circular path in an oblique plane. That is to say, suppose that we wish to cause the robot tool motion, including its orientation, to be that specified by

$$T_{05}(t) = \begin{bmatrix} 1 & 0 & 0 & 5 \cos \pi t \\ 0 & -1 & 0 & 4 \sin \pi t \\ 0 & 0 & -1 & 3 \sin \pi t \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Our task is to find the proper functions of time for each of the joint variables to achieve this motion specification.

Setting the elements of this desired motion specification matrix equal to those of T_{05} found in example 5.3, we find

$$\begin{aligned} -\cos(\phi_A + \phi_B - \phi_D) &= 1, \\ -\sin(\phi_A + \phi_B - \phi_D) &= 0, \\ x_0 + d_2 \cos \phi_A + d_3 \cos(\phi_A + \phi_B) &= 5 \cos \pi t, \\ y_0 + d_2 \sin \phi_A + d_3 \sin(\phi_A + \phi_B) &= 4 \sin \pi t, \\ z_0 + h_1 - (\phi_C + h_4 + h_5) &= 3 \sin \pi t. \end{aligned}$$

If we have a robot for which the dimensions are

$$\begin{aligned} x_0 &= -12'', & y_0 &= 0, & z_0 &= 0, \\ d_2 &= 12'', & d_3 &= 10'', & h_1 &= 8'', & h_4 &= 1'', & h_5 &= 1''. \end{aligned}$$

Then, solving the equations for the joint variables, we find

$$\phi = \begin{bmatrix} \phi_A \\ \phi_B \\ \phi_C \\ \phi_D \end{bmatrix} = \begin{bmatrix} 2 \tan^{-1} \left(\frac{32 \sin \pi t + \sqrt{5616 + 2240 \cos \pi t - 1432 \cos^2 \pi t - 240 \cos^3 \pi t - 9 \cos^4 \pi t}}{164 + 80 \cos \pi t + 3 \cos^2 \pi t} \right) \\ -\cos^{-1} \left(\frac{-28 + 40 \cos \pi t + 3 \cos^2 \pi t}{80} \right) \\ 6 - 3 \sin \pi t \\ 2 \tan^{-1} \left(\frac{40 \sin \pi t - \sqrt{45744 + 20160 \cos \pi t - 8088 \cos^2 \pi t - 2160 \cos^3 \pi t - 81 \cos^4 \pi t}}{356 + 220 \cos \pi t + 9 \cos^2 \pi t} \right) \end{bmatrix}.$$

Note that this is the solution for positive ϕ_A ; there is another solution with negative ϕ_A .

We see here that to meet the requirement that the gripper follow the specified path without rotation, it is necessary to specify and solve the “rotation” equations of the $T_{05}(t)$ path. We find after solution that ϕ_D must be controlled according to this equation to prevent the gripper from rotating.

As with other loop-closure problems, the task of finding closed-form position solutions can be awkward, as shown by the form of the previous solutions. We note also that a new solution must be found for each new command path to be followed.

EXAMPLE 5.5: FERGUSON'S PARADOX For our next example, let us consider the epicyclic gear train shown in [Figure 5.9](#). This gear train, called Ferguson's paradox, was first published as [5] in 1764 by James Ferguson (1710–76), a Scottish physicist, instrument maker, astronomer, and fellow of the Royal Society. In this device, sometimes called an *orrery*, sun gears 3, 4, and 5 are all in mesh with the same planet gear 2. Sun gear 5 is fixed. The input is the rotation of the planet carrier 1, whereas sun gears 3 and 4 provide two different output rotations.

The loops and paths for this problem, found by the methods of Chapter 2, are

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{matrix} A & B & C & D & E & F & G \end{matrix}$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

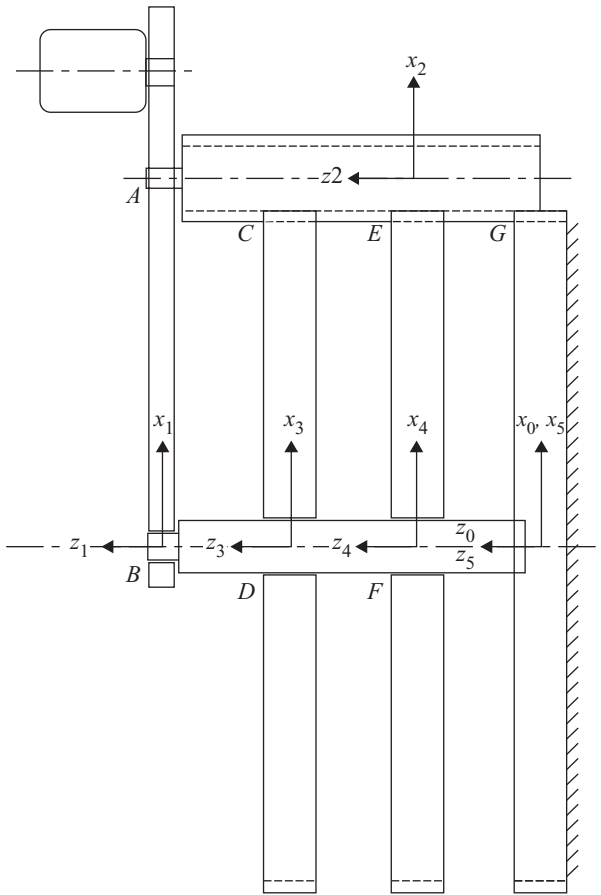


Figure 5.9. Example 5.5: Ferguson's paradox.

The gears all have standard full-depth involute-spur ($\beta_j = 0$) gear teeth with pressure angle $\alpha_j = 20^\circ$ and diametral pitch $P = 20$ teeth per inch.¹ The numbers of teeth on the gears are:

$$N_2 = 10, \quad N_3 = 101, \quad N_4 = 99, \quad N_5 = 100,$$

and, from these, as shown in many texts, we find the pitch circle radii. See, for example, [16, chapter 7]:

$$R'_C = R'_E = R'_G = 0.250'', \quad R_C = 2.525'', \quad R_E = 2.475'', \quad R_G = 2.500''.$$

¹ American Gear Manufacturers' Association (AGMA) and American National Standards Institute (ANSI) have published standards for interchangeable involute gear tooth forms. Contact American Gear Manufacturers' Association, 1001 N. Fairfax Street, Fifth Floor, Alexandria, VA 22314-1587; www.agma.org or email: website@agma.org.

From Figure 5.9, the shape matrices are:

$$T_{05} = S_{1B} = S_{3C} = S_{3D} = S_{2E} = S_{4E} = S_{4F} = S_{5G} = I,$$

$$S_{1A} = \begin{bmatrix} 1 & 0 & 0 & 2.750'' \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{2A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$S_{5B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3'' \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{2C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$S_{5D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2'' \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{5F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1'' \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S_{2G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1'' \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Notice that, as defined in matrix $S_{1A}(1,4)$, the center-to-center distance of pinion 2 from the shaft of gears 3, 4, and 5 are all $2.750''$. This produces a correct mesh at G ; however, the meshes at C and E are not mounted at their standard pitch-circle distances. This explains how the three gear meshes, with different pitch radii, all coexist. The actual meshes at C and E are not mounted properly and are not operating at their nominal 20° pressure angles; notice in the following matrices that the solutions for ϕ_C^2 and ϕ_E^2 do not result in zeroes, but ϕ_G^2 does.

From Eq. (4.28) we find the following joint matrices for the gear meshes. Because we have no information about the direction of rotation or loading, we assume that the $\text{sgn}(F)$ terms are negligible:

$$\Phi_C(\phi_C) = \begin{bmatrix} \cos(11.1\phi_C^1) & -\sin(11.1\phi_C^1) & 0 & (2.775'' + \phi_C^2) \cos \phi_C^1 \\ \sin(11.1\phi_C^1) & \cos(11.1\phi_C^1) & 0 & (2.775'' + \phi_C^2) \sin \phi_C^1 \\ 0 & 0 & 1 & \phi_C^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\Phi_E(\phi_E) = \begin{bmatrix} \cos(10.9\phi_E^1) & -\sin(10.9\phi_E^1) & 0 & (2.725'' + \phi_E^2) \cos \phi_E^1 \\ \sin(10.9\phi_E^1) & \cos(10.9\phi_E^1) & 0 & (2.725'' + \phi_E^2) \sin \phi_E^1 \\ 0 & 0 & 1 & \phi_E^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\Phi_G(\phi_G) = \begin{bmatrix} \cos(11.0\phi_G^1) & -\sin(11.0\phi_G^1) & 0 & (2.750'' + \phi_G^2) \cos \phi_G^1 \\ \sin(11.0\phi_G^1) & \cos(11.0\phi_G^1) & 0 & (2.750'' + \phi_G^2) \sin \phi_G^1 \\ 0 & 0 & 1 & \phi_G^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Having these, we now evaluate the matrix products for each joint:

$$T_{12} = S_{1A} \Phi_A S_{2A}^{-1} = \begin{bmatrix} \cos \phi_A & -\sin \phi_A & 0 & 2.750'' \\ \sin \phi_A & \cos \phi_A & 0 & 0 \\ 0 & 0 & 1 & -2'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{51} = S_{5B} \Phi_B S_{1B}^{-1} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 3'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{32} = S_{3C} \Phi_C S_{2C}^{-1} = \begin{bmatrix} \cos(11.1\phi_C^1) & -\sin(11.1\phi_C^1) & 0 & (2.775'' + \phi_C^2) \cos \phi_C^1 \\ \sin(11.1\phi_C^1) & \cos(11.1\phi_C^1) & 0 & (2.775'' + \phi_C^2) \sin \phi_C^1 \\ 0 & 0 & 1 & \phi_C^3 - 1'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{53} = S_{5D} \Phi_D S_{3D}^{-1} = \begin{bmatrix} \cos \phi_D & -\sin \phi_D & 0 & 0 \\ \sin \phi_D & \cos \phi_D & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{42} = S_{4E} \Phi_E S_{2E}^{-1} = \begin{bmatrix} \cos(10.9\phi_E^1) & -\sin(10.9\phi_E^1) & 0 & (2.725'' + \phi_E^2) \cos \phi_E^1 \\ \sin(10.9\phi_E^1) & \cos(10.9\phi_E^1) & 0 & (2.725'' + \phi_E^2) \sin \phi_E^1 \\ 0 & 0 & 1 & \phi_E^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{54} = S_{5F} \Phi_F S_{4F}^{-1} = \begin{bmatrix} \cos \phi_F & -\sin \phi_F & 0 & 0 \\ \sin \phi_F & \cos \phi_F & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{52} = S_{5G} \Phi_G S_{2G}^{-1} = \begin{bmatrix} \cos(11.0\phi_G^1) & -\sin(11.0\phi_G^1) & 0 & (2.750'' + \phi_G^2) \cos \phi_G^1 \\ \sin(11.0\phi_G^1) & \cos(11.0\phi_G^1) & 0 & (2.750'' + \phi_G^2) \sin \phi_G^1 \\ 0 & 0 & 1 & \phi_G^3 + 1'' \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Guided by the non-zero entries of the loop matrix L , we now formulate the three loop-closure equations. By using Eq. (5.15), we see that negative entries in the loop matrix always lead to inverse matrices in the loop-closure

equation products. This is a direct result of the original choices of orientation in defining the joints, and could be avoided by reversal of those choices. However, reversing the choices of joint orientation affects the sign convention of the joint variable(s) found and also the definitions and signs of the force components found in Chapter 17. As shown, the three loop-closure equations are

$$T_{55} = T_{51}T_{12}T_{52}^{-1} = I$$

$$= \begin{bmatrix} \cos(\psi + \phi_A - 11\phi_G^1) & \sin(\psi + \phi_A - 11\phi_G^1) & 0 & 2.750'' \cos \psi - (2.750'' + \phi_G^2) \\ \sin(\psi + \phi_A - 11\phi_G^1) & \cos(\psi + \phi_A - 11\phi_G^1) & 0 & 2.750'' \sin \psi - (2.750'' + \phi_G^2) \\ 0 & 0 & 1 & \phi_G^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{55} = T_{51}T_{12}T_{42}^{-1}T_{54}^{-1} = I$$

$$= \begin{bmatrix} \cos(\psi + \phi_A - 10.9\phi_E^1 - \phi_F) & \sin(\psi + \phi_A - 10.9\phi_E^1 - \phi_F) & 0 & 2.750'' \cos \psi - (2.725'' + \phi_E^2) \\ -\sin(\psi + \phi_A - 10.9\phi_E^1 - \phi_F) & \cos(\psi + \phi_A - 10.9\phi_E^1 - \phi_F) & 0 & 2.750'' \sin \psi - (2.725'' + \phi_E^2) \\ 0 & 0 & 1 & \phi_E^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{55} = T_{51}T_{12}T_{32}^{-1}T_{53}^{-1} = I$$

$$= \begin{bmatrix} \cos(\psi + \phi_A - 11.1\phi_C^1 - \phi_D) & \sin(\psi + \phi_A - 11.1\phi_C^1 - \phi_D) & 0 & 2.750'' \cos \psi - (2.775'' + \phi_C^2) \\ -\sin(\psi + \phi_A - 11.1\phi_C^1 - \phi_D) & \cos(\psi + \phi_A - 11.1\phi_C^1 - \phi_D) & 0 & 2.750'' \sin \psi - (2.775'' + \phi_C^2) \\ 0 & 0 & 1 & \phi_C^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Setting the elements of these loop-closure equation products to those of the identity matrix, we get the following independent equations:

$$\begin{aligned} \psi + \phi_A - 11\phi_G^1 &= 0, \\ 2.750'' \cos \psi - (2.750'' + \phi_G^2) \cos(\psi + \phi_A - 10\phi_G^1) &= 0, \\ 2.750'' \sin \psi - (2.750'' + \phi_G^2) \sin(\psi + \phi_A - 10\phi_G^1) &= 0, \\ \phi_G^3 &= 0, \\ \psi + \phi_A - 11.1\phi_C^1 - \phi_D &= 0, \end{aligned}$$

$$\begin{aligned}
 2.750'' \cos \psi - (2.775'' + \phi_C^2) \cos (\psi + \phi_A - 10.1\phi_C^1) &= 0, \\
 2.750'' \sin \psi - (2.775'' + \phi_C^2) \sin (\psi + \phi_A - 10.1\phi_C^1) &= 0, \\
 \phi_C^3 &= 0, \\
 \psi + \phi_A - 10.9\phi_E^1 - \phi_F &= 0, \\
 2.750'' \cos \psi - (2.725'' + \phi_E^2) \cos (\psi + \phi_A - 9.9\phi_E^1) &= 0, \\
 2.750'' \sin \psi - (2.725'' + \phi_E^2) \sin (\psi + \phi_A - 9.9\phi_E^1) &= 0, \\
 \phi_E^3 &= 0,
 \end{aligned}$$

and, from these, we can solve for the joint variables

$$\phi = \begin{bmatrix} \phi_A \\ \phi_B \\ \phi_C^1 \\ \phi_C^2 \\ \phi_C^3 \\ \phi_D \\ \phi_E^1 \\ \phi_E^2 \\ \phi_E^3 \\ \phi_F \\ \phi_G^1 \\ \phi_G^2 \\ \phi_G^3 \end{bmatrix} = \begin{bmatrix} 10\psi \\ \psi \\ (100/101)\psi \\ -0.025'' \\ 0 \\ (1/101)\psi \\ (100/99)\psi \\ 0.025'' \\ 0 \\ -(1/99)\psi \\ \psi \\ 0 \\ 0 \end{bmatrix}.$$

Finally, guided by the entries of the path matrix, we now express the posture of each body as a function of the independent variable ψ :

$$\begin{aligned}
 T_{01} = T_{05}T_{51} &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 3'' \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{02} = T_{01}T_{12} &= \begin{bmatrix} \cos (11\psi) & -\sin (11\psi) & 0 & 2.750'' \cos \psi \\ \sin (11\psi) & \cos (11\psi) & 0 & 2.750'' \sin \psi \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 T_{03} = T_{05}T_{53} &= \begin{bmatrix} \cos (\psi/101) & -\sin (\psi/101) & 0 & 0 \\ \sin (\psi/101) & \cos (\psi/101) & 0 & 0 \\ 0 & 0 & 1 & 2'' \\ 0 & 0 & 0 & 1 \end{bmatrix},
 \end{aligned}$$

Table 5.1. Example 5.5: Ferguson's paradox comparison of gear meshes

Joint	Nominal center distance	Actual center distance	Actual pressure angle	Actual contact ratio
<i>C</i>	2.775"	2.750"	21.373°	1.612
<i>E</i>	2.725"	2.750"	18.502°	1.610
<i>G</i>	2.750"	2.750"	20.000°	1.611

$$T_{04} = T_{05}T_{54} = \begin{bmatrix} \cos(\psi/99) & \sin(\psi/99) & 0 & 0 \\ -\sin(\psi/99) & \cos(\psi/99) & 0 & 0 \\ 0 & 0 & 1 & 1'' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{05} = I.$$

In the final analysis, we find that a single input crank turns by an angle ψ carrying a thick planet gear 2 around the periphery of fixed sun gear 5. This also causes two other almost identical-looking sun gears, 3 and 4, to rotate. However, sun gear 3 rotates by $(1/101)\psi$ in the same direction as the input crank, whereas sun gear 4 rotates by $(1/99)\psi$ in the opposite direction.

It was pointed out previously that the center-to-center distance of 2.750" is not equal to the sum of the pitch radii for all three gear meshes. Therefore, all three cannot properly mesh according to their design specifications. A comparison is presented in [Table 5.1](#).

5.7 General Styles for Closed-Form Solutions of Kinematic Equations

In general, finding closed-form solutions for joint variable positions of closed- or open-loop multibody systems is difficult because of the fact that the describing equations are typically nonlinear, coupled, and trigonometric. If we take derivatives of these equations, however, the resulting system of equations becomes linear in the derivative variables and is much more easily solved. However, this produces a solution for velocities rather than positions. We will show, in Chapter 6, how linearized equations can be used to provide numeric solutions to the nonlinear kinematic equations. In the mechanisms and robotics literature, other powerful and general techniques have also been developed for solving such trigonometric equations. All such methods first transform the trigonometric equations to algebraic form using identities such as writing sines and cosines in terms of tangents of half-angles. They then exploit methods for solving systems of polynomial equations. Among the most powerful of these is the method of Raghavan and Roth [10, 11], in which they introduce a special symbolic elimination method for finding the solution to the general six-revolute robot manipulator, with later application to other problems of kinematic position analysis. Another class of powerful methods is those that are based on the use of homotopy methods of numerical algebraic geometry [14]. These homotopy-based methods were first developed in kinematics by the pioneering work of Freudenstein and Roth [6], and later works of Tsai and Morgan [15] and then Wampler, Morgan, and Sommese [17].

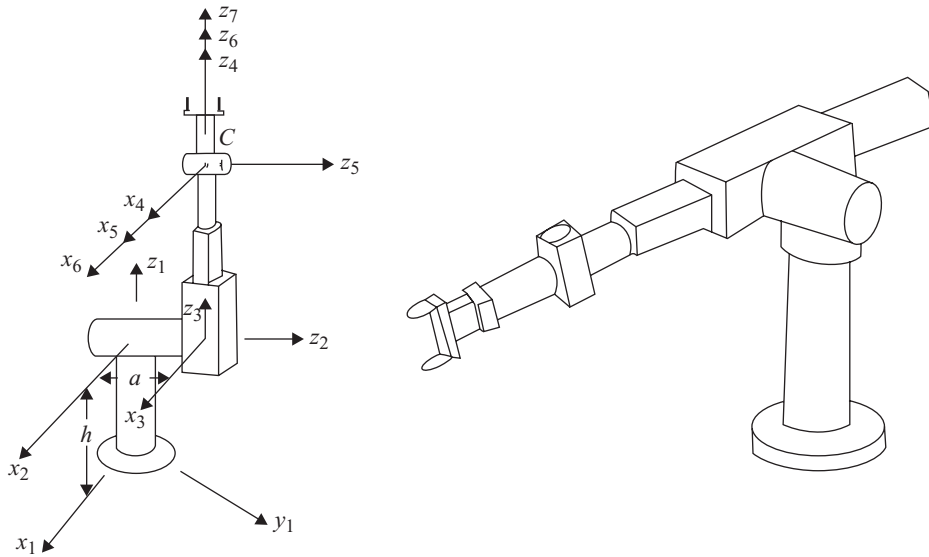


Figure 5.10. The Stanford manipulator showing body coordinate systems.

Here, we present a method that can be used in cases where there is a particular arrangement of joint axes, typical in open serial-kinematic chains such as an open-loop robotic manipulator. For a multibody system such as a robotic manipulator consisting of six revolute joints, if three consecutive joint axes intersect, then we can use their point of intersection to partition the kinematic equations, which leads to simpler equations from which a closed-form solution can usually be found. Such mechanisms are said to have a “solvable” kinematic architecture. Discussion of other solvable kinematic architectures for a serial robot manipulator can be found in [8, 9]. It should also be pointed out that, although the main emphasis in this book is on developing general methods suitable for computer applications, we sometimes depart from this main goal and, as in this section, present more specialized techniques to provide a balance between generality and developing intuitive understanding of the problems being addressed.

Let us consider as an example, the robotic system with six serial joints shown in Figure 5.10. This robot is called the Stanford manipulator [8]. We consider a configuration of this robotic system that has its last three joints located such that their axes intersect at a point referred to here as point C , the wrist center point. For such a robot, we can write the following general kinematic equation:

$$T_{12}T_{23}T_{34}T_{45}T_{56}T_{67} = T_{17},$$

where T_{17} is the matrix describing the posture of the end-effector, which can be written as:

$$T_{17} = \begin{bmatrix} u_7^{x_1} & v_7^{x_1} & w_7^{x_1} & R_{o_7}^{x_1} \\ u_7^{y_1} & v_7^{y_1} & w_7^{y_1} & R_{o_7}^{y_1} \\ u_7^{z_1} & v_7^{z_1} & w_7^{z_1} & R_{o_7}^{z_1} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This matrix, which we assume is specified, describes the desired posture for the end-effector in terms of the three unit vectors describing the orientation of the end-effector, and the position of the origin of the end-effector coordinate system, namely:

$$u_7^1 = \begin{bmatrix} u_7^{x_1} \\ u_7^{y_1} \\ u_7^{z_1} \\ 0 \end{bmatrix}, \quad v_7^1 = \begin{bmatrix} v_7^{x_1} \\ v_7^{y_1} \\ v_7^{z_1} \\ 0 \end{bmatrix}, \quad w_7^1 = \begin{bmatrix} w_7^{x_1} \\ w_7^{y_1} \\ w_7^{z_1} \\ 0 \end{bmatrix}, \quad \text{and} \quad R_{O_7} = \begin{bmatrix} R_{o_7}^{x_1} \\ R_{o_7}^{y_1} \\ R_{o_7}^{z_1} \\ 1 \end{bmatrix}.$$

The challenge now is to find values of the six joint variables corresponding to this specified posture of the end-effector. Note that the body coordinate systems are chosen with the origins of the fourth, fifth, sixth, and seventh coordinate systems coincident at the wrist center point C . Therefore, the kinematic equations can be partitioned into two sets: one set describing the position of the wrist center point C , which is only dependant on the first three joint variables, and a second set describing the orientations of the unit vectors of the coordinate system of the end-effector. The two partitioned sets of kinematic equations are:

$$R_{C_1} = T_{12}T_{23}T_{34}r_{C_4}, \quad (5.18)$$

and (recalling section 3.5)

$$\begin{aligned} u_7^1 &= T_{12}T_{23}T_{34}T_{45}T_{56}T_{67}u_7, \\ v_7^1 &= T_{12}T_{23}T_{34}T_{45}T_{56}T_{67}v_7, \\ w_7^1 &= T_{12}T_{23}T_{34}T_{45}T_{56}T_{67}w_7. \end{aligned} \quad (5.19)$$

Because the origins of the fourth and seventh coordinate systems are coincident at the wrist center point C , then

$$R_{C_1} = R_{O_7} = \begin{bmatrix} R_{o_7}^{x_1} \\ R_{o_7}^{y_1} \\ R_{o_7}^{z_1} \\ 1 \end{bmatrix}, \quad \text{and} \quad r_{C_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

It should be noted that Eq. (5.18) only involves the first three joint variables. By pre-multiplying it with T_{bc}^{-1} matrices, it can be written to form three kinematic equations with the three unknown joint variables more evenly distributed on both sides of the equation as follows:

$$\begin{aligned} T_{12}^{-1}(\phi_1)R_{C_1} &= T_{23}(\phi_2)T_{34}(\phi_3)r_{C_4}, \\ T_{23}^{-1}(\phi_2)T_{12}^{-1}(\phi_1)R_{C_1} &= T_{34}(\phi_3)r_{C_4}, \\ T_{34}^{-1}(\phi_3)T_{23}^{-1}(\phi_2)T_{12}^{-1}(\phi_1)R_{C_1} &= r_{C_4}. \end{aligned}$$

Once the first three joint variables are found from these equations, then the matrix T_{14} can be formulated explicitly and Eqs. (5.19) can be rewritten as

$$\begin{aligned} u_7^4 &= T_{14}^{-1}u_7^1, & u_7^4 &= T_{45}T_{56}T_{67}u_7, \\ v_7^4 &= T_{14}^{-1}v_7^1, & \text{and} \quad v_7^4 &= T_{45}T_{56}T_{67}v_7, \\ w_7^4 &= T_{14}^{-1}w_7^1, & w_7^4 &= T_{45}T_{56}T_{67}w_7. \end{aligned} \quad (5.20)$$

Again, this last set of equations can be pre-multiplied by T_{bc}^{-1} matrices to form three kinematic equations where the three remaining unknown joint variables are more evenly distributed on both sides of the equation, and can possibly be solved in closed form:

$$\begin{aligned} T_{45}^{-1}(\phi_4)u_7^4 &= T_{56}(\phi_5)T_{67}(\phi_6)u_7, & T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)u_7^4 &= T_{67}(\phi_6)u_7, \\ T_{45}^{-1}(\phi_4)v_7^4 &= T_{56}(\phi_5)T_{67}(\phi_6)v_7, & T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)v_7^4 &= T_{67}(\phi_6)v_7, \\ T_{45}^{-1}(\phi_4)w_7^4 &= T_{56}(\phi_5)T_{67}(\phi_6)w_7, & T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)w_7^4 &= T_{67}(\phi_6)w_7, \end{aligned}$$

and

$$\begin{aligned} T_{67}^{-1}(\phi_6)T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)u_7^4 &= u_7, \\ T_{67}^{-1}(\phi_6)T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)v_7^4 &= v_7, \\ T_{67}^{-1}(\phi_6)T_{56}^{-1}(\phi_5)T_{45}^{-1}(\phi_4)w_7^4 &= w_7. \end{aligned} \quad (5.21)$$

EXAMPLE 5.6 In applying this method to the Stanford manipulator, we choose the body and joint coordinate systems such that the transformation matrices T_{bc} are in the form of D-H matrices. These matrices become:

$$T_{12}(\phi_1) = \begin{bmatrix} \cos\phi_1 & 0 & -\sin\phi_1 & 0 \\ \sin\phi_1 & 0 & \cos\phi_1 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{23}(\phi_2) = \begin{bmatrix} \cos\phi_2 & 0 & \sin\phi_2 & 0 \\ \sin\phi_2 & 0 & -\cos\phi_2 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{34}(\phi_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \phi_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{45}(\phi_4) = \begin{bmatrix} \cos\phi_4 & 0 & -\sin\phi_4 & 0 \\ \sin\phi_4 & 0 & \cos\phi_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{56}(\phi_5) = \begin{bmatrix} \cos\phi_5 & 0 & \sin\phi_5 & 0 \\ \sin\phi_5 & 0 & -\cos\phi_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{67}(\phi_6) = \begin{bmatrix} \cos\phi_6 & -\sin\phi_6 & 0 & 0 \\ \sin\phi_6 & \cos\phi_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The (4×4) matrix describing the command posture of the end-effector is as follows:

$$T_{17} = \begin{bmatrix} u_7^x & v_7^x & w_7^x & x_7^1 \\ u_7^y & v_7^y & w_7^y & y_7^1 \\ u_7^z & v_7^z & w_7^z & z_7^1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now, using the first set of partitioned kinematic equations, we can write:

$$\mathbb{T}_{23}^{-1}\mathbb{T}_{12}^{-1}R_C = \mathbb{T}_{34}r_C^4,$$

$$\begin{bmatrix} \cos\phi_2 & \sin\phi_2 & 0 & 0 \\ 0 & 0 & 1 & -a \\ \sin\phi_2 & -\cos\phi_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi_1 & \sin\phi_1 & 0 & 0 \\ 0 & 0 & -1 & h \\ -\sin\phi_1 & \cos\phi_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_7^1 \\ y_7^1 \\ z_7^1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \phi_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

This results in three equations with the first three joint variables as the unknowns as follows:

$$\begin{aligned} \cos\phi_2(\cos\phi_1x_7^1 + \sin\phi_1y_7^1) - \sin\phi_2(z_7^1 - h) &= 0, \\ -\sin\phi_1x_7^1 + \cos\phi_1y_7^1 - a &= 0, \\ \phi_3 &= \sin\phi_2(\cos\phi_1x_7^1 + \sin\phi_1y_7^1) + \cos\phi_2(z_7^1 - h). \end{aligned}$$

The second of these equations can be solved first for ϕ_1 and then the remaining two joint variables, ϕ_2 and ϕ_3 , can be found sequentially from the first and the third of these equations. Next, we can calculate the transformation matrix $T_{14} = T_{12}T_{23}T_{34}$ because ϕ_1 , ϕ_2 , and ϕ_3 are now known.

The vectors u_7^4 , v_7^4 , and w_7^4 can now be computed from the first set of Eqs. (5.20). Then the three additional kinematic equations describing the orientation parameters of the end-effector can be obtained using the second set of Eqs. (5.20) as follows:

$$\mathbb{T}_{56}^{-1}\mathbb{T}_{45}^{-1}w_7^4 = \mathbb{T}_{67}w_7, \quad \mathbb{T}_{56}^{-1}\mathbb{T}_{45}^{-1}v_7^4 = \mathbb{T}_{67}v_7, \quad \text{and} \quad \mathbb{T}_{56}^{-1}\mathbb{T}_{45}^{-1}u_7^4 = \mathbb{T}_{67}u_7. \quad (5.22)$$

The first of these equations results in:

$$\begin{aligned} &\begin{bmatrix} \cos\phi_5\cos\phi_4 & \cos\phi_5\sin\phi_4 & -\sin\phi_5 & 0 \\ -\sin\phi_4 & \cos\phi_4 & 0 & 0 \\ \sin\phi_5\cos\phi_4 & \sin\phi_5\sin\phi_4 & \cos\phi_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_7^{x_4} \\ w_7^{y_4} \\ w_7^{z_4} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi_6 & -\sin\phi_6 & 0 & 0 \\ \sin\phi_6 & \cos\phi_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \end{aligned}$$

From the second row, we obtain the following equation: $-\sin\phi_4w_7^{x_4} + \cos\phi_4w_7^{y_4} = 0$ that results in the solution for the fourth joint variable ϕ_4 . Once ϕ_4 is known, the third row allows us to find ϕ_5 . The second or the third set of Eqs. (5.22) can then be used to find ϕ_6 .

REFERENCES

1. J. Denavit, “A Symbolic Approach to Mechanisms Leading to Electrical Computation Methods,” *MS Thesis*, Department of Mechanical Engineering, Northwestern University, Evanston, IL, 1953.
2. ———, “Description and Displacement Analysis of Mechanisms Based on (2×2) Dual Matrices,” *PhD Dissertation*, Department of Mechanical Engineering, Northwestern University, Evanston, IL, 1956.
3. J. Denavit and R. S. Hartenberg, “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,” *ASME Transactions, J. of Applied Mechanics*, vol. **22**, no. 2, 1955.
4. ———, “Approximate Synthesis of Spatial Linkages,” *ASME Transactions, J. of Applied Mechanics*, vol. **27**, no. 1, 1960, pp. 201–06.
5. J. Ferguson, *The Description and Use of a New Machine Called the Mechanical Paradox*, London, 1764.
6. F. Freudenstein and B. Roth, “Numerical Solution of Systems of Nonlinear Equations,” *J. of Association of Computing Machinery*, vol. **10**, 1963, pp. 550–56.
7. R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA, 1981.
8. D. Pieper, “The Kinematics of Manipulators Under Computer Control,” *PhD Dissertation*, Department of Mechanical Engineering, Stanford University, Stanford, CA, 1968.
9. D. Pieper and B. Roth, “The Kinematics of Manipulator Under Computer Control,” *Proc. 2nd International Congress for the Theory of Machines and Mechanisms*, Zakopane, Poland, vol. **2**, 1969, pp. 159–68.
10. M. Raghavan and B. Roth, “Inverse Kinematics of the General 6R Manipulator and Related Linkages,” *ASME Transactions, J. of Mechanical Design*, vol. **115**, no. 3, 1993, pp. 502–08.
11. ———, “Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators,” *ASME Transactions, J. of Mechanical Design*, Special 50th Anniversary Issue, vol. **117**, June 1995, pp. 71–79.
12. P. N. Sheth and J. J. Uicker, Jr., “A Generalized Symbolic Notation for Mechanisms,” *ASME Transactions, J. of Engineering for Industry*, vol. **93**, 1971, pp. 102–12.
13. J. E. Shigley, *Kinematic Analysis of Mechanisms*, 2nd ed., McGraw-Hill Book Co., New York, 1969.
14. A. J. Sommese and C. W. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*, World Scientific Publishing Co., 2005.
15. L. W. Tsai and A. P. Morgan, “Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods,” *ASME Transactions, J. of Mechanisms, Transmissions, and Automation in Design*, vol. **107**, June 1985, pp. 189–95.
16. J. J. Uicker, Jr., G. E. Pennock, and J. E. Shigley, *Theory of Machines and Mechanisms*, 4th ed., Oxford University Press, New York, 2011.
17. C. W. Wampler, A. P. Morgan, and A. J. Sommese, “Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics,” *ASME Transactions, J. of Mechanical Design*, vol. **112**, no. 1, 1990, pp. 59–68.

PROBLEMS

5.1 Find a set of Denavit-Hartenberg parameters to describe the mechanism of problem 4.4.

5.2 Form the transformation matrices $T_{h,h+1}$ for each link in problem 5.1.

5.3 Continue from the results of problem 4.4 to find the following:

- a) Form the transformation matrices T_{0b} for all four bodies.
- b) Form the loop-closure equation $T_{00} = I$ in symbolic form.

- c) Solve the loop-closure equation for closed-form expressions for each of the joint variables as functions of the independent variable $\psi = \phi_A$.
- d) Substitute the results of c) into a) and form each of the T_{0b} matrices as a function of ψ alone.

5.4 Verify the inverse transformation matrix shown in Eq. (5.16) at the end of section 5.5.

5.5 Continue from the results of problem 4.5 to find the following:

- a) Form the transformation matrices T_{0b} for all four bodies.
- b) Form the loop-closure equation $T_{00} = I$ in symbolic form.
- c) Solve the loop-closure equation for closed-form expressions for each of the joint variables as functions of the independent variable $\psi = \phi_A$.
- d) Substitute the results of c) into a) and form each of the T_{0b} matrices as a function of ψ alone.

5.6 Consider a robot manipulator as shown in Figure P5.6. The kinematic structure of this robotic arm is very similar to that of the Stanford manipulator studied in example 5.6 except that it has an offset (a) between the base and the shoulder (the first two) joint axes. For this robotic arm, derive and solve the kinematic position equations using shape and joint matrices.

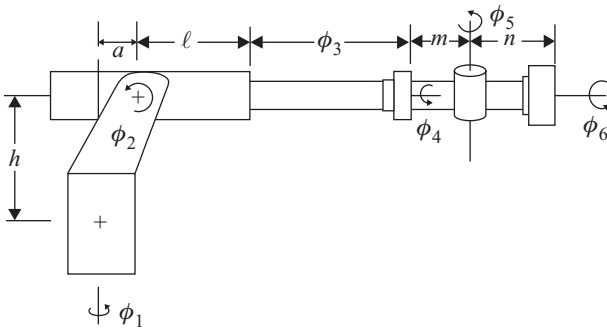
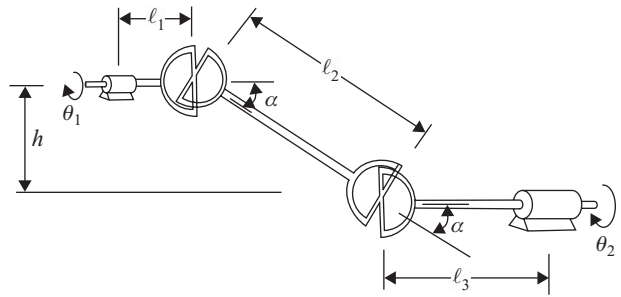


Figure P5.6

5.7 For the robot manipulator of problem 5.6, derive the kinematic position equations using Denavit-Hartenberg transformation matrices and find the solution to these equations using the partitioning method of section 5.7.

5.8 A Cardan/Hooke universal shaft coupling was studied in example 5.1. This coupling is usually used as a shaft coupling to transmit power between two shafts whose center lines intersect at an angle. The variations in angular displacements of the two shafts connected by such a coupling can be eliminated if two of such couplings are used in series in a symmetrical fashion as shown in Figure P5.8. This results in a constant velocity shaft coupling and is used in many applications such as in an automobile drive train. For this symmetrical arrangement of the two Cardan/Hooke's couplings, derive the kinematic loop equations. The notation used in example 5.1 can be adopted for this problem.

Figure P5.8. Double universal shaft coupling.



5.9 The DaVinci robot presently used in some medical applications involving surgeries is basically a SCARA-type robot such as the one studied in example 5.3 with the exception that the last joint has a specialized articulated wrist attached to it as shown in **Figure P5.9**. This articulated wrist adds five more degrees of freedom to the system consisting of four revolute and one prismatic joints as shown. Derive the kinematic equation for the posture of this robot.

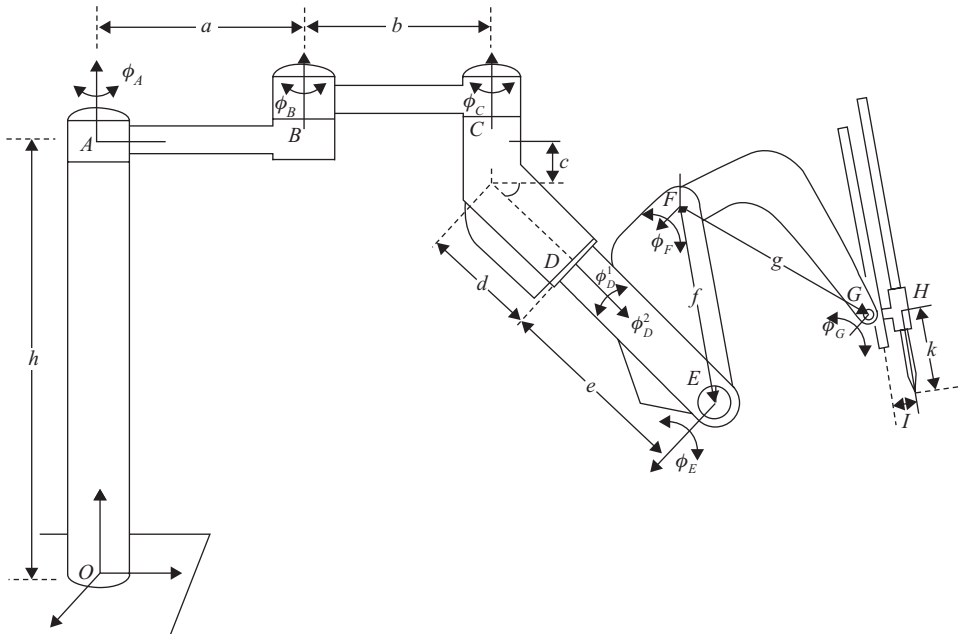


Figure P5.9. DaVinci robot.

6.1 Introduction

In Chapter 5 we studied how the postures of some mechanisms and multibody systems can be found analytically using hand calculations to find closed-form solutions. Typically, this requires forming the necessary transformation matrices, and ensuring that all dependent position variables are made consistent with the constraints expressed by the loop-closure equations. In Chapter 5 we solved several example problems, in both 2-D and 3-D, to illustrate the process, but we also found that the calculations quickly became burdensome, even for problems with only a few unknown joint variables. In principle the methods look powerful, but in practice they quickly reach a limit on practicality.

Does this mean that the methods are not adequate? Not exactly; rather, it means that we are in need of a better means of calculating. Perhaps these tedious computations should be automated for solution by numeric methods using a computer.

Let us reflect on the nature of the problem of posture analysis of a multibody system. In general, the number of bodies (ℓ) is usually reasonably small, typically limited by cost and the desire for simplicity and reliability to tens of moving parts or less. The number of joints (n) is of the same order. The number of closed loops (NL) is usually much smaller. The number of joint variables (ϕ) is of the same order as the number of joints. However, the number of independent variables (ψ) is almost always very small. After all, the whole point of our multibody system is to control the movements of the parts to only those required for proper function of the system. Thus, the mobility (f) is often only one, and is very rarely as many as ten.

Let us say, for example, that a complex industrial machine is comprised of a single assembly and that it has mobility of $f = 8$; let us also say that it contains only revolute and prismatic joints and has $NL = 10$ closed kinematic loops. With these given, the topological formulae of Chapter 2 show that the number of bodies is $\ell = 59$ and the number of joint variables is $n = 68$. These are fair estimates for a quite complex mechanism or multibody system.

Now let us consider the nature of the constraint equations for such a system. The loop-closure equations are products of transformation matrices containing trigonometric functions of ten or so joint variables each. Thus, each constraint equation is a polynomial of approximately tenth degree in sines and cosines of unknown joint

angles, and there are $6NL = 60$ such independent equations. However, once we use trigonometric identities to reduce all sines and cosines to tangents of half-angles, we have 60 equations of degree 20 in our 60 unknown tangents of joint half-angles. Eliminating one unknown from this set of equations doubles the degree of those remaining; thus, we have only 59 equations in 59 unknowns, but of degree 40; then 58 equations of degree 80; 57 equations of degree 160; and so on. Are we getting a message here?

When (and if) we finally get our long-sought single equation in one unknown, its degree is horrendous; what will we do with it then to solve for that one unknown? Because it is a polynomial of extremely high degree, we will take it to the computer to find its many, many roots, won't we? What else can we do? After all, a general polynomial of degree higher than four *cannot* be solved in closed form [6, section 1.6.3]. Then we will take each of these roots in turn and return to the previous equation to solve for the second-last unknown; again doing root extraction of a very high-degree polynomial, collecting even more combinations of roots found. Then the third-last unknown, and so on. Sure we will!

Fortunately, with the use of computers, there is a simpler way – namely seeking a numeric solution from the very beginning. Because most numeric solutions require iteration or incremental improvement, and with multibody systems we are usually dealing with continuous movements, we need to investigate the *differential kinematics* of such systems. Differential or infinitesimal kinematics is a rich subject and includes velocity and acceleration analyses that will be discussed in subsequent chapters. In this chapter, we start with a simple introduction of differential kinematics dealing only with first-order differentials, and introduce the important concepts of derivative operator matrices and the instantaneous screw axis before we present a systematic approach for numeric solution of the kinematic posture equations.

6.2 Differential Kinematics of a Helical Joint

In studying the differential kinematics of a multibody system, we start by studying the movement of the helical joint because, based on Chasles' theorem, it is the general parent of all other single degree-of-freedom motions. From Chapter 4, we know that for a helical joint connecting two bodies, the following body's joint coordinate system, $u'_h v'_h w'_h$, moves with respect to the preceding body's joint coordinate system, $u_h v_h w_h$, with the helical motion of the joint itself. This movement is represented by Eq. (4.8)

$$\Phi_h(\phi_h) = \begin{bmatrix} \cos(\phi_h/\sigma_j) & -\sin(\phi_h/\sigma_j) & 0 & 0 \\ \sin(\phi_h/\sigma_j) & \cos(\phi_h/\sigma_j) & 0 & 0 \\ 0 & 0 & 1 & \phi_h \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{a})$$

Starting from an arbitrary value of ϕ_h as the reference position, let us consider a differential (infinitesimal) displacement of the joint of size $\delta\phi_h$. After this displacement, the altered joint matrix becomes

$$\Phi_h(\phi_h + \delta\phi_h) \approx \Phi_h(\phi_h) + \left[\frac{\partial}{\partial \phi_h} \Phi_h(\phi_h) \right] \delta\phi_h. \quad (\text{b})$$

The partial derivative of the helical joint matrix of Eq. (a) is

$$\frac{\partial}{\partial \phi_h} \Phi_h(\phi_h) = \begin{bmatrix} -1/\sigma_h \sin(\phi_h/\sigma_h) & -1/\sigma_h \cos(\phi_h/\sigma_h) & 0 & 0 \\ 1/\sigma_h \cos(\phi_h/\sigma_h) & 1/\sigma_h \sin(\phi_h/\sigma_h) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

However, the right-hand side of this equation can be factored into two matrices as follows:

$$\frac{\partial}{\partial \phi_h} \Phi_h(\phi_h) = \begin{bmatrix} 0 & -1/\sigma_h & 0 & 0 \\ 1/\sigma_h & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi_h/\sigma_h) & -\sin(\phi_h/\sigma_h) & 0 & 0 \\ \sin(\phi_h/\sigma_h) & \cos(\phi_h/\sigma_h) & 0 & 0 \\ 0 & 0 & 1 & \phi_h \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Observing that the second of these two matrices is identical with the original form of the joint matrix for a helical joint, we define a symbol Q_h for the first matrix

$$Q_h = \begin{bmatrix} 0 & -1/\sigma_h & 0 & 0 \\ 1/\sigma_h & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.1)$$

that allows us to write Eq. (b) as

$$\Phi_h(\phi_h + \delta\phi_h) \approx \Phi_h(\phi_h) + [Q_h \delta\phi_h] \Phi_h(\phi_h),$$

where the scalar factor $\delta\phi_h$ has been moved inside the bracket. The equation for the differential displacement of a helical joint then reduces to

$$\delta\Phi_h = \Phi_h(\phi_h + \delta\phi_h) - \Phi_h(\phi_h) \approx [Q_h \delta\phi_h] \Phi_h(\phi_h). \quad (6.2)$$

It is left as an exercise for the reader to verify, by a parallel development, that the same Q_h matrix can be used with the inverse of the Φ_h matrix as follows

$$\delta\Phi_h^{-1} \approx \Phi_h^{-1} [-Q_h \delta\phi_h]. \quad (6.3)$$

From Eq. (6.2), we can now write an equation for the helical joint transformation matrix after a differential displacement:

$$\Phi_h(\phi_h + \delta\phi_h) \approx [I + Q_h \delta\phi_h] \Phi_h, \quad (6.4)$$

or, for the inverse matrix,

$$\Phi_h^{-1}(\phi_h + \delta\phi_h) \approx \Phi_h^{-1} [I - Q_h \delta\phi_h]. \quad (6.5)$$

From the beginning of this section, we have been working in the $u_h v_h w_h$ coordinate system, aligned with the “preceding” element of the *joint*. However, it is frequently more convenient to express the equations in the coordinate system of the preceding *body*. Suppose, for example, that we identify a point attached to the body labeled b that follows joint h and has coordinates r_b with respect to that body. It may appear that b is just another label for body $h+$. In the beginning, this is true. It is advantageous, however, to carry two labels because we will see shortly that b can be the label of *any* body “following” joint h ; that is, any body for which joint h is in the path to body b .

If we now shift our viewpoint and imagine that we stand on the body labeled $h-$ immediately preceding joint h , then the position of our point, viewed from coordinate system $x_{h-}y_{h-}z_{h-}$, is given by Eq. (5.3)

$$r_b(\phi_h) = S_{h-,h} \Phi_h S_{h+,h}^{-1} r_b,$$

where the symbol $r_b(\phi_h)$ has been chosen to remind us that the point is attached to body b , but is seen from coordinate system $h-$ and moves when the joint variable ϕ_h changes. After a displacement of the joint variable the position of the point changes to

$$r_b(\phi_h + \delta\phi_h) = S_{h-,h} \Phi_h(\phi_h + \delta\phi_h) S_{h+,h}^{-1} r_b,$$

or, if the joint is oriented in the reverse sense, then body b to which the point is attached is body $h+$, but it now comes after joint h on the path from ground and it is now seen from coordinate system of body $h+$:

$$r_b(\phi_h + \delta\phi_h) = S_{h+,h} \Phi_h^{-1}(\phi_h + \delta\phi_h) S_{h-,h}^{-1} r_b.$$

Because $\delta\phi_h$ is assumed small, Eqs. (6.4) and (6.5) may be used to approximate these two equations by

$$r_b(\phi_h + \delta\phi_h) \approx S_{h-,h} [I + Q_h \delta\phi_h] \Phi_h S_{h+,h}^{-1} r_b,$$

or, for the inverse matrix,

$$r_b(\phi_h + \delta\phi_h) \approx S_{h+,h} \Phi_h^{-1} [I - Q_h \delta\phi_h] S_{h-,h}^{-1} r_b.$$

In these two equations, the square-bracketed expressions are both written in the *joint* coordinate system preceding joint h ; that is, in the coordinate system $u_h v_h w_h$. By insertion of appropriate identity factors, they may be expressed in the coordinate system of the *body* preceding joint h on the path to body b . This yields

$$r_b(\phi_h + \delta\phi_h) \approx [I + S_{h-,h} Q_h \delta\phi_h S_{h-,h}^{-1}] S_{h-,h} \Phi_h S_{h+,h}^{-1} r_b, \quad (6.6)$$

or, when joint h has reversed orientation,

$$r_b(\phi_h + \delta\phi_h) \approx [I - S_{h+,h} \Phi_h^{-1} Q_h \delta\phi_h \Phi_h S_{h+,h}^{-1}] S_{h+,h} \Phi_h^{-1} S_{h-,h}^{-1} r_b. \quad (6.7)$$

Continuing to work in the coordinate system of the body preceding joint h on the path to body b , we recognize the small displacement of our point of body b as

$$\delta r_b \approx [S_{h-,h} Q_h \delta\phi_h S_{h-,h}^{-1}] r_b(\phi_h),$$

or

$$\delta r_b \approx -[S_{h+,h} \Phi_h^{-1} Q_h \delta\phi_h \Phi_h S_{h+,h}^{-1}] r_b(\phi_h),$$

depending on the orientation of joint h .

We should notice here that body b is no longer required to be the body that is immediately adjacent to joint h , as long as the displacement δr_b is caused solely by the displacement $\delta\phi_h$ of joint h . These equations are valid for a point attached to *any* body b that is displaced by the movement of joint h where we recognize that r_b and

dr_b are expressed in the coordinate system of the body that precedes joint h along the path to body b . If the displacement of joint h does not affect body b , then these equations do not pertain. To keep this clear in the equation itself and to simplify the treatment of joint orientation, we can take advantage of the topological path matrix P of section 2.5 as follows:

$$\delta r_b \approx P(b, h) [S_{h-,h} Q_h \delta \phi_h S_{h-,h}^{-1}] r_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (6.8)$$

or

$$\delta r_b \approx P(b, h) [S_{h+,h} \Phi_h^{-1} Q_h \delta \phi_h \Phi_h S_{h+,h}^{-1}] r_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (6.9)$$

where $P(b, h)$ symbolizes the entry in row b , column h of path matrix P . This modification not only unifies the signs of the two equations with different joint orientations, it directly yields zero displacement when joint h is not on the path to the point in question.

We can perform an exactly parallel development while picturing ourselves using the absolute coordinate system $x_0 y_0 z_0$. Here the position of the point of body b is expressed by

$$R_b = T_{0h-} S_{h-,h} \Phi_h S_{h+,h}^{-1} T_{h+,b} r_b = T_{0b} r_b,$$

or if joint h happens to be oriented the other way, by

$$R_b = T_{0h+} S_{h+,h} \Phi_h^{-1} S_{h-,h}^{-1} T_{h-,b} r_b = T_{0b} r_b.$$

Because we assume (for now) that all motion comes from the displacement of helical joint h , a small displacement of the joint results in

$$R_b(\phi_h + \delta \phi_h) \approx T_{0h-} S_{h-,h} [I + Q_h \delta \phi_h] \Phi_h S_{h+,h}^{-1} T_{h+,b} r_b,$$

or

$$R_b(\phi_h + \delta \phi_h) \approx T_{0h+} S_{h+,h} \Phi_h^{-1} [I - Q_h \delta \phi_h] S_{h-,h}^{-1} T_{h-,b} r_b.$$

Now, by insertion of appropriate identity factors, these become

$$R_b(\phi_h + \delta \phi_h) \approx T_{0h-} S_{h-,h} [I + Q_h \delta \phi_h] (T_{0h-} S_{h-,h})^{-1} (T_{0h-} S_{h-,h}) \Phi_h S_{h+,h}^{-1} T_{h+,b} r_b,$$

or

$$\begin{aligned} R_b(\phi_h + \delta \phi_h) \approx & T_{0h+} S_{h+,h} \Phi_h^{-1} (S_{h-,h}^{-1} S_{h-,h}) [I - Q_h \delta \phi_h] \\ & \cdot (T_{0h+} S_{h+,h} \Phi_h^{-1} (S_{h-,h}^{-1} S_{h-,h}))^{-1} (T_{0h+} S_{h+,h} \Phi_h^{-1}) S_{h-,h}^{-1} T_{h-,b} r_b. \end{aligned}$$

Remembering that $h-$ is the body that precedes joint h and body $h+$ follows it in the originally defined joint orientation, we find that both orientations are now expressed by the single equation

$$R_b(\phi_h + \delta \phi_h) \approx [I + P(b, h) (T_{0h-} S_{h-,h}) Q_h \delta \phi_h (T_{0h-} S_{h-,h})^{-1}] R_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (6.10)$$

from which we find

$$\delta R_b \approx P(b, h) [(T_{0h-} S_{h-,h}) Q_h \delta \phi_h (T_{0h-} S_{h-,h})^{-1}] R_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (6.11)$$

and the displacement of body b caused by the small displacement of joint h is

$$\delta T_{0b} \approx P(b, h) [(T_{0h-} S_{h-,h}) Q_h \delta \phi_h (T_{0h-} S_{h-,h})^{-1}] T_{0b}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n. \end{array} \quad (6.12)$$

We can notice in all of these equations that the movement comes entirely from the small displacement $\delta \phi_h$ of joint h . The Q_h matrix carries the information about the axis and motion capability of joint h and is naturally defined in the joint h coordinate system. We see in Eqs. (6.6) and (6.7) and Eqs. (6.8) and (6.9) how this axis and motion information is transformed into the preceding body $h-$ coordinate system or, in Eqs. (6.10), (6.11), and (6.12), into the absolute coordinate system. In all cases, this is done by similarity transformations. Thus, in all of these equations, the terms in the square brackets of each reflect the same information, the motion freedom and axis information of Q_h and the size of the motion increment $\delta \phi_h$, but each is transformed to a suitable coordinate system for the object being displaced.

We also notice that body index b is not referenced in the square bracketed operators, and that joint index h and body index $h-$ are *only* referenced in the square brackets, not in the items being displaced. This makes perfect sense; joint index h must appear in each square bracket operator because that joint is the only item being displaced, and body index $h-$ is sometimes used as the convenient coordinate system for expressing its axis.

The conclusion from these observations is that when joint h is the only joint being displaced, the previous formulae can be used for the small movement of an item associated with any body b that is displaced as a result of the small movement of joint h that is for any body b for which joint h is in the path to that body from the ground. Of course, when multiple joint variables along such a path are displaced simultaneously, the aforementioned formulae only predict the displacement of the item associated with body b coming from the displacement of the single joint h . Combinations with other simultaneous joint displacements will be discussed later.

6.3 Derivative Operator Matrices

Continuing our thinking with infinitesimally small – that is, differential – displacements, it is clear, for example, that if we wish to find the differential displacement of joint h , we can start from Eqs. (6.2) and (6.3). If we seek the displacement of a point of body b resulting from the differential displacement of joint h , we can start from Eq. (6.11).

If we look for the rate of change of the posture of body $h+$ with respect to body $h-$ resulting from the movement of the variable of helical joint h , then we are looking for the derivative of Φ_h with respect to ϕ_h . Using Eq. (6.2), this can be found directly from its definition:

$$\frac{\partial}{\partial \phi_h} \Phi_h = Q_h \Phi_h, \quad h = 1, 2, \dots, n. \quad (6.13)$$

If we seek the rate of change of the absolute position of a point attached to body b with respect to the change in position of joint h , then we are looking for the derivative of R_b with respect to ϕ_h . Using Eq. (6.11), this becomes

$$\frac{\partial}{\partial \phi_h} R_b = P(b, h) [(T_{0h} S_{h-,h}) Q_h (T_{0h} S_{h-,h})^{-1}] R_b,$$

and, similarly, for the rate of change of the absolute posture of body b itself with respect to the change in position of joint h , we have from Eq. (6.12),

$$\frac{\partial}{\partial \phi_h} T_{0b} = P(b, h) [(T_{0h} S_{h-,h}) Q_h (T_{0h} S_{h-,h})^{-1}] T_{0b}.$$

Let us now define the following derivative operator matrix

$$D_h = (T_{0h} S_{h-,h}) Q_h (T_{0h} S_{h-,h})^{-1}, \quad h = 1, 2, \dots, n. \quad (6.14)$$

We must be careful in interpreting the notation of this definition. Here the body $h-$ is the body immediately preceding joint h as it was originally defined, irrespective of its orientation in the path to body b .

This definition immediately reduces the previous two equations to

$$\frac{\partial}{\partial \phi_h} R_b = P(b, h) D_h R_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n, \end{array} \quad (6.15)$$

and

$$\frac{\partial}{\partial \phi_h} T_{0b} = P(b, h) D_h T_{0b}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n. \end{array} \quad (6.16)$$

Of Eqs. (6.13) through (6.16), Eq. (6.13) is considered the most fundamental because the others are a direct consequence of this one and the definition of Eq. (6.14). If, for example, Eq. (6.13) is used to differentiate Eq. (5.10) with respect to joint variable ϕ_h , and Eq. (6.14) is then used to simplify the notation, Eq. (6.15) is a direct result.

It should be pointed out that partial differentiation symbolism is used in these equations to remind us that they account only for displacements resulting from changes in the single joint variable ϕ_h , and do not include possible additional simultaneous displacements resulting from other joint variable changes along the path to body b . Similarly, the factor $P(b, h)$ reminds us that joint h must be on the path from ground to body b ; if joint h is not on this path, then T_{0b} , for example, is not a function of ϕ_h and Eq. (6.16) yields zero.

We also stress here that Eqs. (6.13), (6.15), and (6.16) are not approximations; they yield *exact* values of the derivatives! The approximation of infinitesimally small displacement used in Eqs. (6.2) and (6.12) become exact when we pass to the limit of $\delta\phi_h \rightarrow 0$. This observation is extremely important and will be raised again in evaluating numeric accuracy in coming chapters. This ability to differentiate precisely is a unique advantage of the matrix methods presented here over other numeric methods where derivatives are usually replaced by approximate finite differences and can lead to numeric error.

Reviewing this section reminds us that we started from the assumption of the previous sections that our displacement is that of a helical joint. However, the same

concepts can be applied to any single joint freedom, ϕ_h . Returning to Eq. (6.13), let us generalize the idea. Let us simply define a derivative operator matrix, Q_h , as follows:

$$Q_h = \left[\frac{\partial}{\partial \phi_h} \Phi_h \right] \Phi_h^{-1}, \quad h = 1, 2, \dots, n. \quad (6.17)$$

If joint h has more than one joint variable, then we define a derivative operator matrix Q_h^g for differentiation with respect to each of the joint variables, ϕ_h^g .

Once we define Q_h in this way, Eq. (6.13) must hold true for any joint type or any joint variable, helical or not. Also, Eqs. (6.14), (6.15), and (6.16) must hold true as well. As usual, there is an attendant penalty; we must work out and program the appropriate form of the Q_h matrix for each type of joint and for each variable of each type of joint. However, using Eq. (6.17), these are not hard to find and all of the joint types of Chapter 4 are listed as follows.

6.3.1 Helical Joint

$$Q_h = \begin{bmatrix} 0 & -1/\sigma_h & 0 & 0 \\ 1/\sigma_h & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.18)$$

6.3.2 Revolute Joint

$$Q_h = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.19)$$

6.3.3 Prismatic Joint

$$Q_h = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.20)$$

6.3.4 Cylindric Joint

$$Q_h^1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.21)$$

6.3.5 Spheric Joint

$$\begin{aligned}
 Q_h^1 &= \begin{bmatrix} 2\phi_h^1 & 2\phi_h^2 & 2\phi_h^3 & 0 \\ -2\phi_h^2 & 2\phi_h^1 & -2\phi_h^4 & 0 \\ -2\phi_h^3 & 2\phi_h^4 & 2\phi_h^1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^2 &= \begin{bmatrix} 2\phi_h^2 & -2\phi_h^1 & 2\phi_h^4 & 0 \\ 2\phi_h^1 & 2\phi_h^2 & 2\phi_h^3 & 0 \\ -2\phi_h^4 & -2\phi_h^3 & 2\phi_h^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^3 &= \begin{bmatrix} 2\phi_h^3 & -2\phi_h^4 & -2\phi_h^1 & 0 \\ 2\phi_h^4 & 2\phi_h^3 & -2\phi_h^2 & 0 \\ 2\phi_h^1 & 2\phi_h^2 & 2\phi_h^3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^4 &= \begin{bmatrix} 2\phi_h^4 & 2\phi_h^3 & -2\phi_h^2 & 0 \\ -2\phi_h^3 & 2\phi_h^4 & 2\phi_h^1 & 0 \\ 2\phi_h^2 & -2\phi_h^1 & 2\phi_h^4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.22)
 \end{aligned}$$

with the additional constraint equation among the first-order differential displacements that

$$2\phi_h^1\delta\phi_h^1 + 2\phi_h^2\delta\phi_h^2 + 2\phi_h^3\delta\phi_h^3 + 2\phi_h^4\delta\phi_h^4 = 0.$$

6.3.6 Flat Joint

$$Q_h^1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^3 = \begin{bmatrix} 0 & -1 & 0 & \phi_h^2 \\ 1 & 0 & 0 & -\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.23)$$

6.3.7 Rigid Joint

The rigid joint has no joint variables. Therefore, no derivative operator matrices are defined for it.

6.3.8 Open Joint

$$\begin{aligned}
 Q_h^1 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^3 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^4 &= \begin{bmatrix} 2\phi_h^4 & 2\phi_h^5 & 2\phi_h^6 & -2\phi_h^1\phi_h^4 - 2\phi_h^2\phi_h^5 - 2\phi_h^3\phi_h^6 \\ -2\phi_h^5 & 2\phi_h^4 & -2\phi_h^7 & 2\phi_h^1\phi_h^5 - 2\phi_h^2\phi_h^4 + 2\phi_h^3\phi_h^7 \\ -2\phi_h^6 & 2\phi_h^7 & 2\phi_h^4 & 2\phi_h^1\phi_h^6 - 2\phi_h^2\phi_h^7 - 2\phi_h^3\phi_h^4 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^5 &= \begin{bmatrix} 2\phi_h^5 & -2\phi_h^4 & 2\phi_h^7 & -2\phi_h^1\phi_h^5 + 2\phi_h^2\phi_h^4 - 2\phi_h^3\phi_h^7 \\ 2\phi_h^4 & 2\phi_h^5 & 2\phi_h^6 & -2\phi_h^1\phi_h^4 - 2\phi_h^2\phi_h^5 - 2\phi_h^3\phi_h^6 \\ -2\phi_h^7 & -2\phi_h^6 & 2\phi_h^5 & 2\phi_h^1\phi_h^7 + 2\phi_h^2\phi_h^6 - 2\phi_h^3\phi_h^5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.24)
 \end{aligned}$$

$$Q_h^6 = \begin{bmatrix} 2\phi_h^6 & -2\phi_h^7 & -2\phi_h^4 & -2\phi_h^1\phi_h^6 + 2\phi_h^2\phi_h^7 + 2\phi_h^3\phi_h^4 \\ 2\phi_h^7 & 2\phi_h^6 & -2\phi_h^5 & -2\phi_h^1\phi_h^7 - 2\phi_h^2\phi_h^6 + 2\phi_h^3\phi_h^5 \\ 2\phi_h^4 & 2\phi_h^5 & 2\phi_h^6 & -2\phi_h^1\phi_h^4 - 2\phi_h^2\phi_h^5 - 2\phi_h^3\phi_h^6 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^7 = \begin{bmatrix} 2\phi_h^7 & 2\phi_h^6 & -2\phi_h^5 & -2\phi_h^1\phi_h^7 - 2\phi_h^2\phi_h^6 + 2\phi_h^3\phi_h^5 \\ -2\phi_h^6 & 2\phi_h^7 & 2\phi_h^4 & 2\phi_h^1\phi_h^6 - 2\phi_h^2\phi_h^7 - 2\phi_h^3\phi_h^4 \\ 2\phi_h^5 & -2\phi_h^4 & 2\phi_h^7 & -2\phi_h^1\phi_h^5 + 2\phi_h^2\phi_h^4 - 2\phi_h^3\phi_h^7 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

with the additional constraint equation among the first-order differential displacements that

$$2\phi_h^4\delta\phi_h^4 + 2\phi_h^5\delta\phi_h^5 + 2\phi_h^6\delta\phi_h^6 + 2\phi_h^7\delta\phi_h^7 = 0.$$

6.3.9 Parallel-axis Gear Joint

$$Q_h^1 = \frac{R_h + R'_h}{R'_h} \begin{bmatrix} 0 & -1 & 0 & R_h \left(1 + \frac{\phi_h^2}{R_h + R'_h}\right) \sin \phi_h^1 \\ 1 & 0 & 0 & -R_h \left(1 + \frac{\phi_h^2}{R_h + R'_h}\right) \cos \phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^2 = \frac{\operatorname{sgn}(F) (R_h + R'_h + \phi_h^2)}{\sqrt{[R_h + R'_h + \phi_h^2]^2 - [(R_h + R'_h) \cos \alpha_h]^2}} \begin{bmatrix} 0 & -1 & 0 & \sin \phi_j^1 \\ 1 & 0 & 0 & -\cos \phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.25)$$

$$Q_h^3 = \frac{\tan \beta_h}{R'_h} \begin{bmatrix} 0 & -1 & 0 & (R_h + R'_h + \phi_h^2) \sin \phi_h^1 \\ 1 & 0 & 0 & -(R_h + R'_h + \phi_h^2) \cos \phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

6.3.10 Involute Rack-and-Pinion Joint

$$Q_h^1 = \frac{-1}{R'_h} \begin{bmatrix} 0 & -1 & 0 & \phi_h^2 \\ 1 & 0 & 0 & -\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \frac{\operatorname{sgn}(F) \tan \alpha_h}{R'_h} \begin{bmatrix} 0 & -1 & 0 & -(R'_h + \phi_h^2) \\ 1 & 0 & 0 & 1 - \phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (6.26)$$

$$Q_h^3 = \frac{-\tan \beta_h}{R'_h} \begin{bmatrix} 0 & -1 & 0 & R'_h + \phi_h^2 \\ 1 & 0 & 0 & \phi_h^1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

6.3.11 Straight-tooth Bevel-gear Joint

$$\theta = \gamma_h + \gamma'_h + \phi_h^2,$$

$$Q_h^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{\tan \gamma_h}{\tan \gamma'_h} \begin{bmatrix} 0 & -\cos \theta & -\cos \phi_h^1 \sin \theta & 0 \\ \cos \theta & 0 & \sin \phi_h^1 \sin \theta & 0 \\ \cos \phi_h^1 \sin \theta & -\sin \phi_h^1 \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^2 = \begin{bmatrix} 0 & 0 & -\sin \phi_h^1 & 0 \\ 0 & 0 & -\cos \phi_h^1 & 0 \\ \sin \phi_h^1 & \cos \phi_h^1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.27)$$

6.3.12 Point on a Planar-Curve Joint

$$u' = \frac{\partial}{\partial \phi_h^1} u(\phi_h^1), \quad v' = \frac{\partial}{\partial \phi_h^1} v(\phi_h^1),$$

$$Q_h^1 = \begin{bmatrix} 0 & 0 & 0 & u' \\ 0 & 0 & 0 & v' \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^2 = \begin{bmatrix} 0 & -1 & 0 & v(\phi_h^1) \\ 1 & 0 & 0 & -u(\phi_h^1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

(6.28)

6.3.13 Line Tangent to a Planar-Curve Joint

$$u' = \frac{\partial}{\partial \phi_h^1} u(\phi_h^1), \quad v' = \frac{\partial}{\partial \phi_h^1} v(\phi_h^1), \quad u'' = \frac{\partial}{\partial \phi_h^1} u', \quad v'' = \frac{\partial}{\partial \phi_h^1} v',$$

$$\theta = \tan^{-1} \left(\frac{v'}{u'} \right), \quad \theta' = \frac{u'v'' - v'u''}{(u')^2 - (v')^2},$$

$$Q_h^1 = \begin{bmatrix} 0 & -\theta' & 0 & u' + \theta'v' \\ \theta' & 0 & 0 & v' - \theta'u' \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^2 = \begin{bmatrix} 0 & 0 & 0 & -\cos \theta \\ 0 & 0 & 0 & -\sin \theta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

(6.29)

6.4 Screw Axes and Ball Vectors for Differential Displacements

Considering an instantaneous or a differential screw displacement, we are now interested in finding its differential operator matrix. Because a differential displacement is the limiting case of a finite displacement, we can start with the equation for finite screw displacement with coordinate systems uvw and $u'v'w'$ arranged such that we can write:

$$r = \Phi r' = \begin{bmatrix} \Theta & d \\ 0 & 1 \end{bmatrix} r'.$$

For a differential displacement,

$$\delta r = \delta \Phi r' = \begin{bmatrix} \delta \Theta & \delta d \\ 0 & 0 \end{bmatrix} r'.$$

However, because

$$r' = \Phi^{-1} r = \begin{bmatrix} \Theta^{-1} & -\Theta^{-1}d \\ 0 & 1 \end{bmatrix} r,$$

and, remembering that θ is the rotational screw parameter, the equation for a differential displacement becomes

$$\delta r = \delta \Phi r' = \left(\frac{\partial \Phi}{\partial \theta} \right) \delta \theta \Phi^{-1} r = \left(\frac{\partial \Phi}{\partial \theta} \Phi^{-1} \right) \delta \theta r. \quad (6.30)$$

Considering the definition of derivative operator matrices Q , as given in Eq. (6.17), it is clear that the derivative operator matrix for a one-degree-of-freedom screw displacement is

$$Q = \left(\frac{\partial \Phi}{\partial \theta} \Phi^{-1} \right) = \begin{bmatrix} \frac{\partial \Theta}{\partial \theta} & \frac{\partial d}{\partial \theta} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta' & -\Theta' d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\partial \Theta}{\partial \theta} \Theta' & \frac{\partial d}{\partial \theta} - \frac{\partial \Theta}{\partial \theta} \Theta' d \\ 0 & 0 \end{bmatrix}.$$

Recall from Chapter 3, Eq. (3.37) that $\Theta = I + \tilde{w} \sin \theta + \tilde{w}^2 (1 - \cos \theta)$; therefore,

$$\frac{\partial \Theta}{\partial \theta} = \tilde{w} \cos \theta + \tilde{w}^2 \sin \theta \quad \text{and} \quad \Theta' = I - \tilde{w} \sin \theta + \tilde{w}^2 (1 - \cos \theta).$$

Observing that w is a unit vector and \tilde{w} is skew symmetric, we can write:

$$\tilde{w}' = -\tilde{w}; \quad \tilde{w}^3 = -\tilde{w}; \quad \text{and} \quad \tilde{w}^4 = \tilde{w}^3 \tilde{w} = -\tilde{w}^2.$$

We now find that $(\partial \Theta / \partial \theta) \Theta'$ simplifies to $\tilde{w}(\sin^2 \theta + \cos^2 \theta) = \tilde{w}$. Therefore, the derivative operator matrix for our one degree-of-freedom screw displacement becomes:

$$Q = \left(\frac{\partial \Phi}{\partial \theta} \Phi^{-1} \right) = \begin{bmatrix} \tilde{w} & \frac{\partial d}{\partial \theta} - \tilde{w} d \\ 0 & 0 \end{bmatrix}. \quad (6.31)$$

Looking over section 6.3 for the different forms that the Q_h derivative operator matrices take for different joints and joint variables, we notice an interesting pattern. We see that the bottom row is always zero and the upper-left (3×3) submatrix is always skew-symmetric, which is the same as the form we used in Chapter 3,

Eq. (3.14), to represent a pair of Plücker vectors in matrix form. We can therefore rewrite Eq. (6.30) as follows:

$$\delta \mathbf{r} = \begin{bmatrix} \tilde{\Omega} & \Omega \\ 0 & 0 \end{bmatrix} \mathbf{r} \delta \theta, \quad (6.32)$$

where $\tilde{\Omega} = \tilde{w}$ and $\Omega = \partial d / \partial \theta - \tilde{w}d$.

Using vector notation, Eq. (6.32) reduces to:

$$\delta \mathbf{r} = [\hat{\Omega} \times \mathbf{r} + \Omega] \delta \theta. \quad (6.33)$$

Eqs. (6.33) and (6.32) are respectively the vector and matrix forms of what can be considered a fundamental equation for differential or instantaneous displacements written in terms of the two vectors $(\Omega, \hat{\Omega})$.

Let us now find the components of Eq. (6.33) in the direction of the vector $\hat{\Omega}$ by taking the dot product with that vector; that is, by pre-multiplying the equation by $\hat{\Omega}'$:

$$\hat{\Omega}' \delta \mathbf{r} = \hat{\Omega}' \tilde{\Omega} \mathbf{r} \delta \theta + \hat{\Omega}' \Omega \delta \theta$$

and because the vector $\hat{\Omega}$ is perpendicular to the vector $\tilde{\Omega} \mathbf{r}$, this reduces to

$$\hat{\Omega}' \delta \mathbf{r} = \hat{\Omega}' \Omega \delta \theta.$$

However, this equation says that the component of the differential displacement $\delta \mathbf{r}$ in the direction of $\hat{\Omega}$ is *independent of the choice of the point, r*. This is to say that all points of the body have equal displacements in the $\hat{\Omega}$ direction. Remembering from Chasles' theorem that we expect the pattern to have the properties of a screw, we can now identify that the $\hat{\Omega}$ unit vector shows the *orientation of the screw axis*. We can also recognize the pitch of the screw as the rate of displacement in the direction of the screw axis per unit rotation of the screw:

$$\hat{\Omega}' \frac{\delta \mathbf{r}}{\delta \theta} = \hat{\Omega}' \Omega = \sigma. \quad (6.34)$$

This is the instantaneous pitch of the screw. Because we are dealing with the displacement of an arbitrary joint type, not necessarily a true helical joint, the pitch may change as the joint moves.

Let us next find the location of a point P that is on the screw axis, thus uniquely locating the axis. Remembering that a point on the screw axis experiences *only* the displacement along the axis, from Eq. (6.33) we write

$$\frac{\delta \mathbf{P}}{\delta \theta} = \Omega + \tilde{\Omega} \mathbf{P} = \sigma \hat{\Omega}.$$

Let us now take the vector cross product of the screw axis direction $\hat{\Omega}$ with all terms of this equation:

$$\tilde{\Omega} \frac{\delta \mathbf{P}}{\delta \theta} = \tilde{\Omega} \Omega + \tilde{\Omega} (\tilde{\Omega} \mathbf{P}) = \tilde{\Omega} (\sigma \hat{\Omega}) = \mathbf{0}.$$

When we replace this triple-vector product by the vector identity

$$\widehat{\Omega} \times (\widehat{\Omega} \times P) = (\widehat{\Omega} \cdot P)\widehat{\Omega} - (\widehat{\Omega} \cdot \widehat{\Omega})P,$$

which in matrix notation reads

$$\widetilde{\Omega}(\widetilde{\Omega}P) = (\widehat{\Omega}'P)\widehat{\Omega} - (\widehat{\Omega}'\widehat{\Omega})P,$$

then our equation for point P on the screw axis becomes

$$\widetilde{\Omega}\Omega + (\widehat{\Omega}'P)\widehat{\Omega} - (\widehat{\Omega}'\widehat{\Omega})P = 0.$$

This equation, of course, fits all points on the screw axis and, therefore, does not yield a unique solution for P . Because we seek a single point to locate the axis, let us choose the particular point P on the screw axis for which the vector P from the origin is perpendicular to the screw axis. This is the point P for which $(\widehat{\Omega}'P) = 0$. The solution for this particular point P is

$$P = \frac{\widetilde{\Omega}\Omega}{\widehat{\Omega}'\widehat{\Omega}}. \quad (6.35)$$

If we denote the components of the two vectors $(\Omega, \widehat{\Omega})$ as follows:

$$\Omega = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \widehat{\Omega} = \begin{bmatrix} d \\ e \\ f \end{bmatrix}, \quad (6.36)$$

then the expression for P in terms of these six parameters becomes

$$P = \frac{1}{d^2 + e^2 + f^2} \begin{bmatrix} 0 & -f & e \\ f & 0 & -d \\ -e & d & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \quad (6.37)$$

Thus, overall, we have shown that the Q differentiation operator matrices carry the full information of the screw axis for the corresponding differential displacement of a single joint variable. The screw can now be identified because Eq. (6.36) gives its orientation, Eq. (6.34) gives its pitch, and Eqs. (6.35) and (6.37) give a point on its axis.

We notice that the (4×4) matrix form of Q is convenient for use as a differentiation operator. However, there are only six independent parameters in the operator and these can equally well be kept in the form of the two 3-D Cartesian vectors Ω and $\widehat{\Omega}$ of Eq. (6.36). These vectors are perhaps more convenient for geometric identification of the instantaneous screw axis. They were developed by Sir Robert Stawell Ball (1840–1913), Lowndean Professor of Astronomy and Geometry, Cambridge University, and are called the *Ball vectors* of the screw [2].

The same information can also be stored as a single six-dimensional vector, called *screw coordinates*:

$$\hat{\Omega} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}. \quad (6.38)$$

This format may be best for computer storage, and will appear in later chapters. No matter which format is chosen, the six parameters identify the same information: the instantaneous screw axis and the pitch of the differential motion.

Throughout this section we have treated the Q_h differentiation operator matrices as expressed in the $u_h v_h w_h$ coordinate system of a joint. However, we have already seen similar forms such as $S_{h-,h} Q_h S_{h-,h}^{-1}$, which is expressed in the coordinate system of body $h-$, immediately preceding joint h , and

$$D_h = (T_{0h-} S_{h-,h}) Q_h (T_{0h-} S_{h-,h})^{-1},$$

which is expressed in the absolute coordinate system. Each of these is related to Q_h by a similarity transformation. Therefore, each is of the form

$$TQT^{-1} = \begin{bmatrix} \Theta & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\Omega} & \Omega \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta^t & -\Theta^t d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \Theta \tilde{\Omega} \Theta^t & \Theta \Omega - \Theta \tilde{\Omega} \Theta^t d \\ 0 & 0 \end{bmatrix}.$$

Here we see that these other forms of differentiation operator matrices also have a bottom row of zeroes. The upper-left (3×3) rotation submatrix is of the form $\Theta \tilde{\Omega} \Theta^t$ and because $\tilde{\Omega}^t = -\tilde{\Omega}$, we find that $(\Theta \tilde{\Omega} \Theta^t)^t = -(\Theta \tilde{\Omega} \Theta^t)$. Therefore, the upper-left (3×3) submatrix remains skew-symmetric after the similarity transformation. Therefore, these transformed differentiation operator matrices are of the same characteristic form assumed in Eq. (6.36), and any of these can be used for identifying an instantaneous screw axis or a set of Ball vectors or screw coordinates. Equations (6.34), (6.35), (6.36), and (6.37) apply equally to all of these differentiation operators, no matter in which coordinate system they happen to be expressed. The resulting screw axis is found in coordinates corresponding to the form that is used.

The Ball vectors or screw coordinates discussed in this section uniquely identify the axis and pitch of a screw, along and about which a moving body is displaced to a new position with respect to a reference body. Thus, they each identify a unique screw. It will be noticed, however, that they do not, in general, identify the magnitude of the displacement taken on this screw. The six parameters $\hat{\Omega} = (a, b, c, d, e, f)^t$ form a homogeneous set in the sense that the set can be multiplied by any nonzero constant and still identifies the same screw. Thus, they identify the screw, but not its displacement. Also, because the set can be scaled by an arbitrary constant, only five of the six parameters are independent.

Sometimes we may wish to specify the magnitude of the displacement on the screw, a sixth independent value; we may choose to scale the screw coordinates to express this magnitude. They will then identify the same screw and also the displacement experienced, all within the same six-dimensional vector, which is no longer homogeneous. Such a helical displacement of a given magnitude is called a *twist*, and

a twist can be identified by properly scaled Ball vectors or screw coordinates. The differential twist associated with our operator matrix Q_h , for example, has scaled Ball vectors of $\Omega_h d\phi$ and $\hat{\Omega}_h d\phi$.

EXAMPLE 6.1 As an illustrative example, let us find the Ball vectors, the screw coordinates, and the screw parameters for the differential displacement of the first joint variable of a parallel-axis gear joint as defined by Eq. (6.25).

Assuming exact mounting of the gear joint, such that $\phi_h^2 = \phi_h^3 = 0$, we have

$$Q_h^1 = \frac{(R_h + R'_h)}{R'_h} \begin{bmatrix} 0 & -1 & 0 & R_h \sin \phi_h^1 \\ 1 & 0 & 0 & -R_h \cos \phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

From Eqs. (6.36) and (6.38), we can immediately write the Ball vectors and the screw coordinates for this joint variable:

$$\Omega_h = \begin{bmatrix} R_h \sin \phi_h^1 \\ -R_h \cos \phi_h^1 \\ 0 \end{bmatrix}, \quad \hat{\Omega}_h = \begin{bmatrix} R_h \sin \phi_h^1 \\ -R_h \cos \phi_h^1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\hat{\Omega}_h = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

Equation (6.34) gives $\sigma_h = 0$ for the pitch, which makes perfect sense because ϕ_h^1 allows only rotation. Equation (6.37) gives one point on the screw axis

$$P = \begin{bmatrix} R_h \cos \phi_h^1 \\ R_h \sin \phi_h^1 \\ 0 \end{bmatrix},$$

and we recognize that the locus of this point as ϕ_h^1 changes is the pitch circle, specified in u_h, v_h, w_h coordinates, whereas the orientation of the screw axis as specified by $\hat{\Omega}_h$ is parallel to the w_h joint axis.

6.5 Numeric Solution of Kinematic Posture Equations

Once a multibody system has been modeled on a computer, an analysis of the kinematic architecture of the system is performed in accordance with the methods of Chapter 2 to identify the complete topology, including numbers of bodies, joints, joint types and joint variables, number of assemblies, and to identify any kinematic loops and all kinematic paths. A data structure is then formulated in computer memory that reflects the architecture of the system modeled.

Numeric data is entered next to specify the exact shapes, sizes, and initial postures of all bodies or components of the system. Of course, this data entry stage requires many modeling decisions of the user. Not the least of these is the choice of

an initial configuration, a single modeling posture of the system that is used for the specification of numeric data. If a layout drawing of the design exists, the posture shown in that drawing serves very nicely. If not, it may be worthwhile to create one, either on paper or in a CAD system. An actual hardware system or a 3-D physical scale model can serve here as long as we choose a specific position for each generalized coordinate. The important thing is to ensure that there is a source of accurate geometric data, all captured at one consistent configuration of the system and known to fit together into a real machine, at least at the initial posture.

Choices of postures of coordinate systems are also made at this time. Not only must there be an agreed upon global coordinate system $x_0y_0z_0$, but also a body coordinate system $x_b y_b z_b$ must be chosen for each body, two joint coordinate systems, $u_h v_h w_h$ and $u'_h v'_h w'_h$, must be identified for each joint, and other auxiliary coordinate systems may also be specified.

Next, the postures of all coordinate systems are entered and transformation matrices are formulated numerically, perhaps by the methods of section 4.4. All required shape matrices $S_{h-,h}$ and $S_{h+,h}$ are found and stored for later usage. Simultaneously, initial values of joint matrices $\Phi_h = S_{h-,h}^{-1} T_{0h-}^{-1} T_{0h+} S_{h+,h}$ are formulated numerically for each joint at this initial posture. From these, the methods of section 4.6 are used to extract initial numeric position values for all joint variables ϕ_h , each depending on its own joint type. Some of these are, of course, the initial (modeling) position values of the generalized coordinates ψ_j . As all of this numeric data is being collected, pertinent tests are made to ensure that the numeric values of each joint matrix Φ_h are consistent with the assumptions made in section 4.6 with respect to the placement and orientation of joint coordinate systems for the corresponding joint types. When discrepancies are discovered between modeling assumptions and the numeric values received, the computer software can warn the user with an appropriate message and allow the interactive correction of the model until all data are consistent with these modeling assumptions.

Finally, when the data entry phase of analysis is completed, the computer memory contains a validated model of the mechanical system at the initial modeling posture. This computer model may or may not match the real mechanical system intended by the user. It may or may not be movable or, if it is, the motion may or may not represent the true motion of the real machine. However, in any case, it is a model of a possible mechanical system.

6.5.1 Solution for a Nearby Posture

Once all data are entered and conformity with assumptions is verified, accurate numeric values are known for all joint variables ϕ_h at the initial posture of the system. Suppose, however, that we wish to move the model to another nearby posture. Suppose that the new desired posture is specified by the user by changing some or all of the generalized coordinates ψ_j to new position values that we assume are “close” to their initial position values. The problem that we wish to solve now is to find a way to update the other joint variable values to be consistent with the new generalized coordinate values, thus moving the system model to this new posture. The original research for this and subsequent sections, before the many extensions shown here, was first published as [12].

Because we assume that the new posture is close to the previously known posture, we now assume that the unknown values of the dependent joint variables are also close to their preceding known values. That is, we assume that the unknown values are of the form

$$\phi_h^* = \phi_h + \delta\phi_h, \quad h = 1, 2, \dots, n, \quad (6.39)$$

where $\delta\phi_h$ represent small unknown changes from the known position values. For joints with more than one variable there will be as many ϕ_h^s and $\delta\phi_h^s$ values as there are variables in each joint.

We must now consider how these small changes can be made consistent with the constraints of the loop-closure equations. Because we have changed the generalized coordinate position values ψ_j , we must recalculate the numeric values of the joint matrices Φ_h according to their joint types and the formulae in section 4.6 and the values of the individual transformation matrices of Eq. (5.4)

$$T_{h-,h+} = S_{h-,h} \Phi_h S_{h+,h}^{-1}.$$

As we calculate new T_{0b} products of transformation matrices along the paths of the model and products of transformation matrices around the kinematic loops, we find that the T_{00} products around the loops are no longer equal to the identity transformation because some joint variables have been changed to new position values and others have not. Because of this, the loop-closure equations are not satisfied. Instead of the identity transformation, the products T_{00} for each loop are in error by small amounts that we represent by the matrix E_i :

$$T_{00} = I + E_i, \quad i = 1, 2, \dots, NL.$$

Of course, if we knew the small changes $\delta\phi_h$ of Eq. (6.39), all calculations could be redone and the errors E_i would not exist. Our problem, therefore, is to find the $\delta\phi_h$ corrections necessary to eliminate the errors E_i in the loop-closure constraint equations.

Let us expand the loop-closure equations in Taylor series in the neighborhood close to the known values of ϕ_h . To first order this gives

$$T_{00} + \sum_h \frac{\partial T_{00}}{\partial \phi_h} \delta\phi_h + \dots = I, \quad i = 1, 2, \dots, NL,$$

where there is one term in this summation for each joint variable ϕ_h that appears in the loop being considered.

If we take advantage of the D_h differentiation operator matrices, then consistent with Eq. (6.16), our Taylor series becomes

$$T_{00} + \sum_{h=1}^n L(i, h) D_h T_{00} \delta\phi_h + \dots = I, \quad i = 1, 2, \dots, NL,$$

where $L(i, h)$ symbolizes the loop matrix and is used to provide sign information in conformity with the existence and orientation of each joint h in each loop i .

Post-multiplying this series by T_{00}^{-1} gives

$$I + \sum_{h=1}^n L(i, h) D_h \delta\phi_h + \dots = T_{00}^{-1}, \quad i = 1, 2, \dots, NL.$$

Then, dropping all quadratic and higher-order terms and rearranging this equation slightly, it becomes

$$\sum_{h=1}^n L(i, h) D_h \delta \phi_h \approx T_{00}^{-1} - I = E_i, \quad i = 1, 2, \dots, NL, \quad (6.40)$$

which is a set of NL (4×4) matrix equations relating the n unknown joint variable corrections, $\delta \phi_h$.

Realizing that each matrix has only six independent entries, we can, without loss of information, replace each of these matrices by its screw coordinates. Therefore, referring to Eqs. (6.36) and (6.38) for form, we define the following screw coordinate vectors:

$$\hat{D}_h = \begin{bmatrix} D_h(1, 4) \\ D_h(2, 4) \\ D_h(3, 4) \\ D_h(3, 2) \\ D_h(1, 3) \\ D_h(2, 1) \end{bmatrix}, \quad h = 1, 2, \dots, n, \quad \hat{E}_i = \begin{bmatrix} E_i(1, 4) \\ E_i(2, 4) \\ E_i(3, 4) \\ E_i(3, 2) \\ E_i(1, 3) \\ E_i(2, 1) \end{bmatrix}, \quad i = 1, 2, \dots, NL. \quad (6.41)$$

With these definitions, Eq. (6.40) reduces to

$$\sum_{h=1}^n L(i, h) \hat{D}_h \delta \phi_h \approx \hat{E}_i, \quad i = 1, 2, \dots, NL, \quad (6.42)$$

where 6-D screw coordinate vectors have replaced the (4×4) matrices. There is an equation of this form for each of the NL loops of the system, and they relate the n error correction unknowns, $\delta \phi_h$.

In addition to these equations, we must include an additional equation for each joint that employs Euler-Rodrigues parameters for its joint variables. From a first-order Taylor series expansion of Eq. (4.17) we have a constraint equation of the form

$$2\phi_h^1 \delta \phi_h^1 + 2\phi_h^2 \delta \phi_h^2 + 2\phi_h^3 \delta \phi_h^3 + 2\phi_h^4 \delta \phi_h^4 = 1 - (\phi_h^1)^2 - (\phi_h^2)^2 - (\phi_h^3)^2 - (\phi_h^4)^2, \quad (6.43)$$

for each spheric joint and, from a similar expansion of Eq. (4.24), we have a constraint equation of the form

$$2\phi_h^4 \delta \phi_h^4 + 2\phi_h^5 \delta \phi_h^5 + 2\phi_h^6 \delta \phi_h^6 + 2\phi_h^7 \delta \phi_h^7 = 1 - (\phi_h^4)^2 - (\phi_h^5)^2 - (\phi_h^6)^2 - (\phi_h^7)^2, \quad (6.44)$$

for each open joint of the system.

We now wish to display these equations in a more standard form; however, the process is better shown by example. Suppose we take the gear train problem of example 5.5, Figure 5.9; the loop matrix for this three-loop example was found to be

$$L = \begin{bmatrix} A & B & C & D & E & F & G \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

Developing the explicit form of Eq. (6.42) for this example, we find that it is

$$\begin{bmatrix} \hat{D}_A & \hat{D}_B & 0 & 0 & 0 & 0 & -\hat{D}_G \\ \hat{D}_A & \hat{D}_B & 0 & 0 & -\hat{D}_E & -\hat{D}_F & 0 \\ \hat{D}_A & \hat{D}_B & -\hat{D}_C & -\hat{D}_D & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\phi_A \\ \delta\phi_B \\ \delta\phi_C \\ \delta\phi_D \\ \delta\phi_E \\ \delta\phi_F \\ \delta\phi_G \end{bmatrix} = \begin{bmatrix} \hat{E}_1 \\ \hat{E}_2 \\ \hat{E}_3 \end{bmatrix}.$$

The procedure shown by this example can be generalized for any rigid-body mechanical system. It always results in a set of linear equations of the form

$$\mathcal{J}\delta\phi = \hat{E}. \quad (6.45)$$

For a system with NL loops and n joint variables, the coefficient matrix \mathcal{J} of this set of equations has $(6NL + NC)$ rows and n columns, where NC is the number of constraint equations for the Euler-Rodrigues parameters of spheric and open joints in the system. The coefficient matrix of Eq. (6.45) is called the *Jacobian*. The determinant of this matrix was studied in depth by the Prussian mathematician, Carl Gustav Jacob Jacobi (1804–51), and was presented in [4].

This Jacobian matrix is always of the same form as the loop matrix L found in Chapter 2 for the same problem, except that each non-zero entry of L that had a value of ± 1 is now replaced by a (6×1) screw coordinate vector of $\pm \hat{D}_h$ in the column of \mathcal{J} corresponding to the joint variable for joint h . The zero entries of L are each replaced by columns of six zeroes. When a joint h has more than one joint variable, then the corresponding screw vectors $\pm \hat{D}_h^s$ for each joint variable are entered into successive columns of \mathcal{J} . The coefficients of the constraint equations of the form of Eqs. (6.43) and (6.44) relating the Euler-Rodrigues parameters of each spheric and open joint are entered as NC additional rows of \mathcal{J} using these same columns. The column vector \hat{E} is filled with the screw coordinate vectors of Eq. (6.41) showing the errors in closure for each of the loops, with additional entries from the right-hand sides of Eqs. (6.43) and (6.44) for the NC additional constraint equations. The unknowns of this set of equations (6.45) are the error corrections $\delta\phi_h$ (or $\delta\phi_h^s$) for the joint variables, taken in the same order as the columns of \mathcal{J} .

If our application had been a robotic manipulator with body number e being the end-effector, then let us suppose that T_{0e}^* represents the desired posture of the end-effector. In this case, an exactly parallel development leads to the following equation in place of Eq. (6.40):

$$\sum_{h=1}^n P(e, h) D_h \delta\phi_h \approx T_{0e}^* T_{0e}^{-1} - I = E_e, \quad (6.46)$$

that gives

$$\sum_{h=1}^n P(e, h) \hat{D}_h \delta\phi_h \approx \hat{E}_e, \quad (6.47)$$

instead of Eq. (6.42) when put into screw coordinate form. This leads to a Jacobian matrix that has either $[6+NC]$ or $[6(NL+1)+NC]$ rows, depending on whether the

manipulator also includes any closed loops. Otherwise, a manipulator is identical to other mechanism or multibody applications.

Of course, we started this section by assuming that one or more of the joint variables had intentionally been changed to a nearby position; those joint variable(s) already have new values and do not need “corrections.” We may as well delete these columns from \mathcal{J} , and the corresponding “corrections” from the column of unknowns. Although we will show a better procedure later, for now, let us consider these eliminated.

If the \mathcal{J} matrix is square and non-singular, then Eq. (6.45) can be solved directly by matrix inversion

$$\delta\phi = \mathcal{J}^{-1}\hat{E}.$$

More will be said later about problems in which \mathcal{J} is either singular or is not square.

Once the $\delta\phi$ error corrections are found, they are added to the previous values of the joint variables according to Eq. (6.39) giving improved values for the dependent joint variables. However, because our equations were linearized by dropping higher-order terms of the Taylor series, these may still not be of sufficient accuracy. If not, the process is repeated iteratively until the accuracy is acceptable.

To repeat the process means accepting the improved joint variables to replace the previous values of ϕ_h , recomputing the joint matrices Φ_h using these improved values, finding new transformation matrices and products T_{0b} and derivative operator matrices D_h , new coefficient and error matrices \mathcal{J} and E , and then new corrections $\delta\phi_h$.

With each iteration of this process, the values of the joint variables improve to better fit the loop-closure constraints and the T_{00} matrix products more closely approximate the identity matrix. Therefore, the E matrix entries, showing errors in loop-closure, become smaller and the corresponding \hat{E}_i columns of values also become smaller, leading to smaller error corrections $\delta\phi_h$. Ultimately, when all E matrix entries and $\delta\phi_h$ corrections become smaller than an agreed upon tolerance value, the process is declared finished. Much more will be said on this iteration process and its convergence in the sections to follow. However, after convergence, the new updated values of all joint variables and all transformation matrices and derivative operator matrices are in conformance with the loop-closure constraints at the new posture of the system.

6.5.2 Avoiding Convergence to a False Solution

Once the iteration process of the previous section converges to the specified tolerance level, are we assured that this numeric solution represents a valid posture for the real mechanical system? Unfortunately, there are conditions when this may not be the case. Let us look more closely at what our numeric procedure has and has not assured.

We have continued to monitor and are assured that the column of constants \hat{E}_i has become zero within our agreed tolerance. Comparing this with Eqs. (6.42) and (6.41) and remembering from section 6.4 that the bottom row of each D_h matrix is always zero assures us that all off-diagonal terms of each E_i matrix have become zero

in Eq. (6.40). Thus, remembering the orthogonality conditions of all transformation matrices, we are assured for each loop that

$$T_{00} = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

However, our iteration equation does nothing to ensure that the three diagonal elements of the rotation submatrices have converged to *positive* unit values to match the identity matrix.

We do know, in addition, from the properties of our transformations that the determinant of each T_{00} matrix is always positive unity. However, it is still conceivable that our iteration process could converge with a T_{00} matrix having *two negative* unit values on its main diagonal. Geometrically, this would mean a rotation error of 180° in a loop about one of the major global axes without correction by our iteration process. Worse yet, experience has proven that this can and sometimes *has* happened in practice, even for very simple systems moving through small displacements.

Fortunately, once this problem was discovered, a very simple correction was found. The definition of \hat{E}_i in Eq. (6.41) is, therefore, modified as follows:

$$\hat{E}_i = \begin{bmatrix} E_i(1, 4) \\ E_i(2, 4) \\ E_i(3, 4) \\ E_i(3, 3) + E_i(3, 2) + E_i(2, 2) \\ E_i(1, 1) + E_i(1, 3) + E_i(3, 3) \\ E_i(2, 2) + E_i(2, 1) + E_i(1, 1) \end{bmatrix}, \quad i = 1, 2, \dots, NL. \quad (6.48)$$

This modification is equivalent to adding two of the diagonal terms to each of the off-diagonal rotation terms in Eq. (6.40) before the screw coordinates are extracted. However, because the D_h matrices have zeroes on their diagonals, this causes no changes in the D_h values on the left side of the equation or in the \hat{D}_h vectors or the \mathcal{J} matrix. It only affects the E_i matrices on the right side of Eq. (6.40). This change in definition of \hat{E}_i has no effect on the convergence or accuracy of the iteration process when converging toward a *valid* solution because the changes are in the quadratic and higher-order terms of the Taylor series. The diagonal terms of a helical transformation matrix are unity and $\cos \theta$ where, hopefully, θ is a small angle. However, the Taylor series of $\cos \theta$ for small angles has no first-order term.

However, with this modification, if the process comes even remotely close to one of the false solutions, a *very* large correction is made that *prevents* convergence toward such a solution. This heuristic modification has been tested in software, has totally eliminated the problem, and has shown no further problems in over forty years of extensive use.

6.5.3 Numeric Solution of the Loop-closure Equation

In section 6.5.1, Eq. (6.45) was solved by inversion of the \mathcal{J} matrix of coefficients on the temporary assumption that it was square and non-singular. This strategy was

used to simplify the explanation and reduce confusion for the reader; however, it is not the full story.

Let us now consider, for example, the simulation of a simple four-bar linkage after the input crank angle has been set to a new position. We discover that there are only three dependent joint variables for which $\delta\phi_h$ corrections need be found. Yet there are six equations. In that case, as for all planar linkages, the out-of-plane components of the screw coordinate vectors are zeroes, thus leaving only three nontrivial equations in the three unknowns. Similarly, if we consider a spherical linkage, such as the universal shaft coupling of example 5.1, only the three rotational components of the screw coordinates carry values and all translation equations are null. Therefore, only three unknowns can be found per loop for planar or spherical problems.

From these and many other situations with special geometry we can see that there are not always six *useful* equations per loop. Often, some of the equations become null or simple identities and cannot yield solutions for unknowns. The \mathcal{J} matrix may carry more rows than columns, yet still be meaningful. Still, it is not clear at the time the software is being programmed *which* equations carry meaningful information for a given problem, and which may be null or trivial identities.

Another circumstance that can lead to difficulty is when special geometric situations, such as parallelism or intersection of multiple joint axes, lead to some screw coordinates that are linear combinations of others. Again, this gives a \mathcal{J} matrix that has no inverse, either because it is not square or because it is rank deficient. The \mathcal{J} matrix of Eq. (6.45) results from an algorithm that assures that it always has $(6NL+NC)$ equations, but there is no guarantee that these equations are independent and nontrivial. Similarly, there is no guarantee that there are $(6NL+NC)$ joint variables that need corrections.

We can say with certainty that the $(6NL+NC)$ equations contained in \mathcal{J} represent *all* of the kinematic constraints that must be enforced between the unknown joint variables. There are no other kinematic constraints. No more than $(6NL+NC)$ joint variables can be found from the constraint equations. However, this does not assure that $(6NL+NC)$ joint variable corrections can always be found because some equations may be either trivial or redundant or linearly dependent.

It is also possible that a mechanism or multibody problem can be posed wherein the \mathcal{J} matrix has more than $(6NL+NC)$ unknown joint variables. This happens, for example, in problems of higher mobility if less than the full number of generalized coordinates are identified and given values by the user. In such a case, the additional joint variables of the undiscovered degrees of freedom cannot be found from the loop-closure constraints. The positions of those joint variables must be found in some other way. They cannot be found from Eqs. (6.45).

In the general case, therefore, the \mathcal{J} matrix may not be square and also it may be singular in the sense that it may have rank less than $(6NL+NC)$. How then are we to solve Eq. (6.45) for its unknown joint variable corrections $\delta\phi_h$ if standard matrix inversion software cannot be used?

Historically, the earliest solution to this dilemma in this application [12] was put forward for the case where \mathcal{J} has more rows than columns; that is, where there are more equations than unknown joint variables. The argument was made that because the process is iterative, the best r.m.s. (root-mean-squared) approximation

to all $(6NL+NC)$ equations could be accepted, and would then be corrected again, if necessary, in later iterations. Therefore, Eq. (6.45) was approximated by

$$(\mathcal{J}^t \mathcal{J})\delta\phi = \mathcal{J}^t \hat{E},$$

and because the coefficient matrix was then square, the solution

$$\delta\phi = (\mathcal{J}^t \mathcal{J})^{-1} \mathcal{J}^t \hat{E}$$

was accepted. In general, this approach did give acceptable solutions for many cases. However, later study showed that it sometimes converges more slowly to a solution, thus requiring more iterations. Much worse, however, is the robustness of this approach when the system moves into or near a posture of poor mechanical advantage. Near such a posture, the determinant of \mathcal{J} (if it is square) becomes small and the determinant of $(\mathcal{J}^t \mathcal{J})$ becomes quadratically smaller. Thus, the accuracy of joint variable corrections becomes poor near such a posture and convergence to a solution is sometimes in doubt.

Fortunately, a much more robust algorithm has been found [10] for the solution of Eq. (6.45). It is the Gauss-Jordan method of elimination, [6, Sec. 10.4], named after German mathematician Carl Friedrich Gauss (1777–1855) and German geodesist Wilhelm Jordan (1842–99), with a special variation of complete pivoting. Other methods could be similarly adapted if the modified pivoting is tailored to fit. Cholesky's method or Crout's method of lower-upper matrix factorization would be good alternative choices. However, these have not been tested by these authors.

This algorithm starts by assuming that the joint variables are arranged such that those representing known generalized coordinates ψ_j carry the largest identification labels and, therefore, are represented in the right-most columns of the \mathcal{J} matrix. Next the \mathcal{J} matrix is filled with screw coordinate vectors as explained in section 6.5.1 and then augmented on the right by a $[(6NL + NC) \times (6NL + NC)]$ identity matrix, where we wish to develop the "inverse" of \mathcal{J} . Subdividing the columns of \mathcal{J} into those for the unknown joint variables and those associated with the specified generalized coordinates, we have

$$[\mathcal{J}_1 \mid \mathcal{J}_3 \mid I].$$

Next, we proceed with the Gauss-Jordan elimination algorithm except for a small but important variation. We seek the largest possible pivot element, but *searching only the \mathcal{J}_1 submatrix*. That is, we search only in those columns that do not represent specified generalized coordinates. Finding the pivot element, we switch this row to the top and switch this column to the left. Then we divide all elements of the pivot row by that element, thus making the pivot element unity. Then, by adding or subtracting correct multiples of that modified pivot row to each of the other rows in succession, we zero all other elements in the pivot column, thus eliminating that variable from all equations except the first.

We then identify the largest remaining pivot element, again *searching only the remaining portion of the modified \mathcal{J}_1 submatrix*. Switching that row and that column to second, we also place it on the main diagonal. We then normalize that row and eliminate that variable from all other equations by subtracting proper multiples from each of the others. We continue in this same manner and the augmented \mathcal{J} matrix evolves as we do this.

The Gauss-Jordan elimination operations are, of course, all valid matrix row operations. Except for the reordering of rows and columns, these combined operations are equivalent to pre-multiplying the original augmented \mathcal{J} matrix as follows

$$\begin{aligned} & \left[\begin{array}{ccc|c} \frac{\mathcal{J}_{11}^{-1}}{} & & & 0 \\ -\frac{\mathcal{J}_{21}}{\mathcal{J}_{11}} & \frac{\mathcal{J}_{11}^{-1}}{} & & I \end{array} \right] \left[\begin{array}{ccc|c} \frac{\mathcal{J}_{11}}{} & \frac{\mathcal{J}_{12}}{} & \frac{\mathcal{J}_{13}}{} & I \ 0 \\ \frac{\mathcal{J}_{21}}{} & \frac{\mathcal{J}_{22}}{} & \frac{\mathcal{J}_{23}}{} & 0 \ I \end{array} \right] \\ & = \left[\begin{array}{ccc|c} I & & & \\ 0 & \frac{\mathcal{J}_{22} - \frac{\mathcal{J}_{21}\mathcal{J}_{11}^{-1}\mathcal{J}_{12}}{} & \frac{\mathcal{J}_{23} - \frac{\mathcal{J}_{21}\mathcal{J}_{11}^{-1}\mathcal{J}_{13}}{} & -\frac{\mathcal{J}_{21}\mathcal{J}_{11}^{-1}}{} \ 0 \\ & & & I \end{array} \right]. \quad (6.49) \end{aligned}$$

Here the different submatrices of the original augmented \mathcal{J} matrix are shown separated by horizontal and vertical dashed lines. A second row of symbols has been added to distinguish rows above and below the current pivot element; also, the original \mathcal{J}_1 columns have been separated into columns to the left (\mathcal{J}_{i1}) and to the right (\mathcal{J}_{i2}) of the current pivot element.

Thus, as we continue with each new elimination step, we search out a new pivot element in the lower submatrix containing $[\mathcal{J}_{22} - \mathcal{J}_{21}\mathcal{J}_{11}^{-1}\mathcal{J}_{12}]$; we then switch this pivot element to the top-left of this submatrix by changing rows and columns. When we eliminate multiples of this row from all other rows, we again reach the state shown, except that the identity matrix at the top-left has increased in size by one, and the size of the area to be searched for the next pivot has decreased by one row and column.

As we continue, we finally reach a state where either (a) we eliminate all rows of the original \mathcal{J} matrix, or (b) the largest pivot element found in the ever-shrinking $[\mathcal{J}_{22} - \mathcal{J}_{21}\mathcal{J}_{11}^{-1}\mathcal{J}_{12}]$ submatrix is either zero or essentially zero (less than a given tolerance). The tolerance used here may be set to the round-off tolerance of the computer system used, but need not be smaller than the strictest dimensional tolerance of the manufactured part dimensions.* When such a state is reached, the \mathcal{J} matrix stored in computer memory has, within this tolerance, reached the form

$$\left[\begin{array}{ccc|c} I & \frac{\mathcal{J}_{11}^{-1}\mathcal{J}_{12}}{} & \frac{\mathcal{J}_{11}^{-1}\mathcal{J}_{13}}{} & 0 \\ 0 & 0 & 0 & -\frac{\mathcal{J}_{21}\mathcal{J}_{11}^{-1}}{} \ I \end{array} \right].$$

Notice that the entries in the rows below $\mathcal{J}_{11}^{-1}\mathcal{J}_{13}$ are also shown as zeroes. These remainders should be checked by the software. If they are not small – that is, if non-negligible size elements still remain in this area – it is a signal that the user has set as a *specified generalized coordinate* the position of a joint variable that can (and should) be determined from others. Such a situation should be treated as an error in the user's input data.

Once the modified Gauss-Jordan elimination process finishes, the form of the original Eq. (6.45),

$$\mathcal{J} \delta\phi = \hat{E},$$

* The IMP software uses a default tolerance of 10^{-10} length units or radians.

has been reduced to

$$\begin{bmatrix} I & \mathcal{J}_{11}^{-1} \mathcal{J}_{12} & \mathcal{J}_{11}^{-1} \mathcal{J}_{13} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\phi_1 \\ \delta\phi_2 \\ \delta\phi_3 \end{bmatrix} = \begin{bmatrix} \mathcal{J}_{11}^{-1} & 0 \\ -\mathcal{J}_{21} \mathcal{J}_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} \hat{E}_1 \\ \hat{E}_2 \end{bmatrix}, \quad (6.50)$$

where \hat{E}_1 and \hat{E}_2 refer to portions of the original \hat{E} vector, but shown subdivided to fit the subdivision of the preceding matrix. $\delta\phi_1$ refers to the corrections just found. $\delta\phi_3$ are “corrections” whose labels refer to the specified generalized coordinates ψ_3 and because these are already set to “correct” positions, $\delta\phi_3$ is now set to zero. $\delta\phi_2$ is a set of “corrections” for joint variables not yet converged, but that cannot be solved from the loop-closure constraints. Lacking further information, these corrections are also set to zero. Therefore, the set of equations becomes

$$\begin{bmatrix} \delta\phi_1 \\ \delta\phi_2 \\ \delta\phi_3 \end{bmatrix} = \begin{bmatrix} \mathcal{J}_{11}^{-1} \hat{E}_1 \\ 0 \\ 0 \end{bmatrix}, \quad (6.51)$$

with the additional condition that the equations require

$$\hat{E}_2 = \mathcal{J}_{21} \mathcal{J}_{11}^{-1} \hat{E}_1.$$

This condition may not be satisfied exactly, particularly in early iterations. However, when the iteration process converges, both \hat{E}_1 and \hat{E}_2 approach zero and the condition is satisfied automatically.

In applying Eq. (6.51) we must remember that multiple row and column changes take place during the pivoting steps of the modified Gauss-Jordan algorithm. While this is happening the original row and column identification labels must be tracked so that when Eq. (6.51) is applied, the modified ordering of the original \mathcal{J} column numbers can be used to tell which joint variables are to be corrected by the rearranged elements of $\delta\phi_1$.

The astute reader might legitimately ask why the \mathcal{J} matrix was augmented with a full identity matrix, rather than just augmenting \mathcal{J} with the single column \hat{E} matrix. Similarly, why were the columns of \mathcal{J}_3 kept if we are to set $\delta\phi_3 = 0$ at the end? The answers to these questions will become clear in the next and later chapters where we will see that the information found in these columns at the completion of the iteration process is extremely useful and meaningful, and comes as a “free” by-product of the process.

6.6 Identification of Generalized Coordinates

One of the most valuable by-products of the iterative solution process described in section 6.5.3 is that it *automatically* discovers the mobility of our mechanical system and selects a suitable choice of independent generalized coordinates.

As the modified Gauss-Jordan algorithm is applied to the inversion of the \mathcal{J} matrix of the loop-closure equation coefficients, the modified pivoting algorithm avoids choosing any of the user-specified generalized coordinates and avoids allowing them to be “corrected” by the constraint equations. However, having said this, the pivoting algorithm does use the largest available coefficients in the remaining

part of \mathcal{J} , in the order found, to solve for other joint variables where possible. The choice of the largest available elements of \mathcal{J} for use as pivot elements assures the best possible numeric accuracy and minimizes roundoff and truncation errors.

As pivot elements are chosen, the choices are made implicitly of which joint variables are to be corrected; that is, which are to be the dependent joint variables of the system. All joint variables that are solved from the Euler-Rodrigues or loop-closure constraints become *dependent* variables and those that are not solvable in this way must be *independent* variables; that is, they are additional generalized coordinates.

The positions of the generalized coordinates, as we have just seen, are not solved from the kinematic constraints – the loop-closure equations. Their positions must be set by other factors. They may be independently set by the user, or they may take on positions as a result of static and dynamic forces imposed on them. These are subjects for later chapters. Either way, the joint variables in the group $\delta\phi_2$ are independent and are additional generalized coordinates ψ_j not originally identified by the user. This explains why we set their “corrections” $\delta\phi_2$ to zero, thus leaving them in their previously known positions.

It may seem arbitrary to choose generalized coordinates for our mechanical system based on numeric convenience. However, it becomes clear in later chapters that the choice of the largest pivot elements also selects those joint variables that have the best mechanical advantage and are, in this sense, not only suitable, but optimal choices for generalized coordinates.

We should also take careful note that the final decision on the *mobility* of our mechanical system is a by-product of our iterative kinematic analysis, not a declaration made by the user; thus, it is not subject to errors of human intuition or judgment. The mobility of our mechanical system is

$$f = n - \text{rank}(\mathcal{J}). \quad (6.52)$$

Also, the generalized coordinates for the system come in two groups. Some of these ($\delta\phi_3$), which we call *specified generalized coordinates (SGC)*, have positions specified by the user. We give the symbol NS to the number of SGC joint variables. Others ($\delta\phi_2$), which we call *free generalized coordinates (FGC)*, numbering NF joint variables, are discovered by the system during the previous iteration process. Together they give $f = NF + NS$ total generalized coordinates – the total mobility of the system.

Note also that the number NF of free generalized coordinates, and the choice of joint variables representing them, are discovered anew each time our system seeks a new posture. There is no assurance either that the number NF or the choice of variables will be the same from one system posture to another. The choice of variables for the FGCs may change as the system posture changes. As the system moves through a “dead-center” or other special geometry, it is very possible that even the mobility may change because a screw axis may become linearly dependent on others, and there may result an additional (instantaneous) degree of freedom (an FGC) at such a posture that did not exist at the previous posture and that may not exist at the next. This is the nature of a dead-center posture. We must be watchful for this possibility in later phases of the analysis where it can complicate our techniques. However, this is the physical nature of such a system; such complications can not be avoided.

6.7 Scaling Internal Length Units

So far in this text no mention has been made of units for the various physical quantities. This is fitting because the laws of science must hold true in spite of the choice of units used as long as the units chosen form a consistent set. Thus, there has been no mention of whether distance units are to be measured in inches or millimeters or feet or meters or even light years; it should not matter as long as we are consistent for all distance measurements.

Yet experience shows that our choice of distance unit can have a major impact on the convergence rate and numeric accuracy of the iterative solution of the loop-closure equations. Perhaps this is most easily understood by considering the tolerance used in testing for convergence. Suppose, as an example, that an automotive suspension system is being simulated and that, in view of manufacturing tolerances, we hope to achieve solutions accurate to ± 0.050 mm. Length and distance data are entered in millimeters and vary from quite small to perhaps 5 000 mm. Thus, we already see five orders of magnitude difference between what we consider “large” and “small” distances for such a problem.

However, far more insidious, suppose that the same suspension problem contains a rotating arm of length 500 mm. To what accuracy must we calculate the arm’s orientation to ensure our required distance accuracy of ± 0.050 mm? Remember that angular quantities are always treated in radians in all higher-level programming languages; the programmer has little choice. This implies an accuracy of approximately ± 0.0001 radians for calculation of angular quantities; if the arm were longer, the angular accuracy requirement would be even more stringent. Recalling that our iterative solution of the loop-closure equations is used to solve for a combination of both distance and angular unknowns among the elements of $\delta\phi$, what accuracy should be chosen in testing for convergence?

We see that there can easily be about five orders of magnitude difference between the magnitudes of distances and angular parameters even in a very reasonable problem. Yet a typical digital computer with 32-bit accuracy in single-precision calculations can only carry accuracy of about six significant figures. Does this mean that we must, or even that we should, use double-precision calculations? *No!* The real problem here is the difference in magnitude of distance and angular numeric values and the fact that they mix together in the position equations. This is a problem of scaling, and should be treated as such. It is the opinion of these authors that most double-precision software is not required, but is the result of lack of care by the analyst in problem formulation. Usually, in simulating problems of the mechanical world, numeric ill-conditioning comes from poor problem formulation, not from the laws of mechanics.

If, when all data are first entered, we seek out the maximum and minimum values of all posture origins and point-position coordinate data in the problem, we can discover the limiting dimensions of the rectangular volume in which the problem is defined. Suppose that we take the diagonal distance across this volume and define *this* distance to be *one internal length unit*. Then, using this distance as a scale factor, all distance units for other data can be scaled to this distance as an internal unit of length. This ensures that all distance values in the numeric model are less than unity, as are typical angular values that are always in radians. All numeric values are now of

the same general magnitude. On a 32-bit computer, all calculations are now carried out to precision of about six significant figures of meaningful accuracy with respect to each other. When results are printed, the internal distance units are converted back to the user's preferred units by inverse use of the same scale factor. J. Angeles presents an extensive review [1] of many authors' attempts to find a "characteristic length," and then he proves that such a length is not possible for the general case because it must approach zero. He concludes that a problem-dependent engineering definition must be used instead.

With this scaling procedure in place, experience shows that position solutions can be iterated to an accuracy tolerance of about $\pm 0.000\ 01$ internal units with only a few iterations for a typical problem. For our suspension example, this is equivalent to accuracy of approximately $\pm 0.000\ 57^\circ$ for angular values and ± 0.050 mm for distances. With this scaling procedure, the accuracy is suited to the problem at hand without the penalties on speed and memory caused by double-precision calculations.

6.8 Quality Index

In section 6.6 we discussed how the mobility of a particular problem may change if the system reaches a posture where some of its screw coordinates become linearly dependent on others. We noted that a new FGC may appear as we pass through such a posture, and then disappear again once the dependency no longer exists. Such phenomena do happen in real mechanism and multibody equipment simulation. Such a posture might be called a "dead-center" posture in one application. In another, the problem might be said to have a poor "pressure angle." In another, we may have reached the "end of stroke," and cannot move the input any further. In yet another, we may say that the system has reached a "toggle" posture. In many different applications we may hear that a machine has either good or poor mechanical advantage. We are not yet ready to give a precise definition to the term mechanical advantage; this must wait for our study of force analysis. Even now, however, we can define a general measure of how well a system is likely to serve its intended purpose.

Reviewing the Gauss-Jordan elimination method of section 6.5.3, we see that as each pivot element is identified, these can be multiplied together and, at completion, this product gives the determinant of the J_{11} portion of the Jacobian, the submatrix that is actually inverted. We define the absolute value of this determinant as Q , which we call the *quality index* of the mechanical system being simulated:

$$Q = |\det(J_{11})|. \quad (6.53)$$

As the name implies, this value is a general indicator of how well a system is suited to performing its function. Its definition allows the quality index to be found for any kinematic system at a chosen posture with a chosen set of input variables, and it describes a qualitative measure of performance similar to many of the application-specific terms previously named (dead-center, pressure angle, toggle, etc.).

As a system moves from one posture to another, the quality index changes. When its value is high, that is good; when it becomes small, that is an indication that the system is approaching a special geometry (singularity) posture where the mobility ($N F$) may change. By itself, the value of the quality index Q means nothing, and it depends very much on scaling. However, if this value is monitored as the system

moves, it is a good indicator of increasing or decreasing functionality of the system for the task simulated. Note that the quality index has units of internal length units to a power that depends on the number of distance pivot-elements selected in \mathcal{J}_{11} . Experience shows, however, that its value is far more stable from one problem to another than without scaling, and, perhaps in the future, an acceptable range of values may be found.

We will see in coming chapters that calculation of any derivatives for our mechanical system is done by inverting the same Jacobian matrix. When its determinant becomes small, this implies that all derivative equations become nearly ill-conditioned. When the quality index becomes small, all derivatives of the system become doubtful together. Thus, all aspects of the system's performance (velocity ratios, force transmission, influence of friction, positioning accuracy, effects of manufacturing tolerances, etc.) deteriorate. When one of these degenerates, they all decay together. This is the meaning of a dramatically decreasing value of the quality index.

6.9 Convergence and Robustness

The iteration procedure described in this chapter was first proposed in 1963 in [12] by the first author of this text, and it has been discussed in depth by many for a wide variety of mechanical systems. That author has been party to many discussions and inquiries, and has supported the use of this algorithm in commercial software since 1970; he has shared the experiences of literally hundreds of real-world applications. Still, there is little agreement on the merits and disadvantages of the approach. In fact, there is so much disinformation based on opinions of those with no first-hand experience that this issue has become one motivation for this text. The primary concerns of critics are on issues of computational speed, rate of convergence, robustness of the algorithm, and on the size of the largest displacement for which it will converge.

The first of these questions, that of computational speed and rate of convergence, is often raised by a critic who *assumes* that *any* "iterative" technique must take a "large number" of iterations to converge to an acceptable precision and will therefore be slow and impractical. That criticism simply does not apply in this case. Numerical analysis texts show that the Newton-Raphson iteration technique, which is the technique used here, has quadratic convergence and is often the method of choice for solution of nonlinear equations in multiple variables because of its rapid convergence [11, pp.100–02]. This root-finding procedure was first published by Joseph Raphson (1648–1715) in [9]. However, historians have shown that he had become aware of private notes of Sir Isaac Newton (1643–1727) that were written in 1671, but were not published until almost fifty years later as part of [7].

The original publication on the application of this method to mechanisms [12] reports an example set of calculations for a single-loop 3-D linkage with six dependent joint variables. The rate of convergence, taken from that example, is shown in Table 6.1.

The convergence rate shown by this example is not a fluke; it is *typical* of the authors' fifty years of experience, even for problems with many loops and many more unknown variables. The first iteration usually corrects all variables so that errors are

Table 6.1 Convergence rate for an iteration process example

Joint\Iteration	1	2	3	4
2 (deg)	-5.150 0	-0.163	0.000 596	0.000 001 6
3 (in)	0.001 2	-0.112	0.000 740	0.000 000 1
4 (deg)	0.302 0	0.255	-0.000 569	0.000 000 5
5 (in)	-0.863 3	0.170	-0.000 748	0.000 000 2
6 (deg)	-5.560 0	-0.236	0.000 537	0.000 000 1
7 (in)	-0.061 5	-0.151	0.000 796	0.000 000 1

all of the same general order-of-magnitude. Later iterations improve the accuracy of all variables simultaneously at a rate of two or three orders-of-magnitude with each iteration. Typical problems require three or four or (rarely) five iterations to converge to accuracy close enough for all engineering applications.

Experience shows that a problem *never* iterates more than six times and still reaches a solution. Either it converges in fewer iterations or it fails to converge at all. If it fails to converge, then either (a) it reaches the limit of the computer's accuracy without achieving the requested tolerance (which is probably unnecessarily small), or (b) it diverges quickly and dramatically, giving huge correction terms that become larger rather than smaller with each iteration.

The first difficulty – decreasing size corrections that do not reach the requested tolerance – indicates that the requested tolerance limit is too strict for the accuracy of the computer. The only known choices under these conditions are to relax the requested tolerance, or to change the software to use double-precision calculations.

The second problem – divergence of the iteration process – indicates that (a) the posture sought cannot be reached, perhaps because it is past the limit of travel that the input can physically achieve, or that (b) the new posture is not “nearby” to the posture from which the iteration process was started, thus defying the assumption of our Taylor series expansion. When this is the case, an intermediate posture can be chosen and two (or more) increments can usually be used with success.

This, of course, raises the question of how close is “nearby.” How large a change in posture can be found with success? How near to the requested posture is required for the estimates to assure convergence? This question is not easy to answer with a simply applied criterion.

Once a problem is modeled at some initial posture, it is probably unwise to try to turn an input crank by 180° or more in a single step; however, it is surprising how often this may be successful. It is the authors' experience that it is much more likely that a solution will be found than that the iteration process will diverge. However, when a large displacement is taken in a single step, it is also possible that a valid numeric solution may be found, but not necessarily the solution the user expected or intended. We must remember that multiple real solutions may exist for the nonlinear loop-closure equations. There may be more than one configuration in which the same multibody system can be arranged for the requested input values. However, even if the process converges to one of these alternate solutions, this does not assure that the real system can move continuously from one solution to the other. If the iteration process were to converge to an alternate configuration, then it may be necessary to disassemble the physical system and reassemble it to reach the later posture. This

other closure may be a clearly valid solution to the loop-closure equations, but not one that the user expected or intended.

Fortunately, the iteration process, as presented, does not switch from one closure to another as long as the new posture is in the neighborhood of the starting posture. This still does not answer the question, however, of “how close is nearby?” The author has no comprehensive answer for this. It is amazingly “far” when the system is in a “nice” posture with a good-sized value for the quality index. However, as the quality index decreases, it is wise to reduce to smaller displacement increments if simulated motion is to continue.

Probably the most comforting advice the authors can offer from experience is that if a real system – one that functions well in the physical world – is being simulated, then the increment at which the user wishes to see the results reported – that is, the step size for printing or graphing results – will not cause a problem. This is not very scientific, but it is an honest report of many years of experience.

The most extensive study of convergence and robustness of this iteration process was done in 1993 by Olsen in his study [8] of its applicability to the control of a six degree of freedom serial manipulator. His purpose was to test the validity of the following premise [3], expressed in different ways in various texts on robotics:

We will split all proposed manipulator solution strategies into two broad classes: *closed-form solutions* and *numerical solutions*. Because of their iterative nature, numerical solutions generally are much slower than the corresponding closed-form solution; in fact, so much so that for most uses, we are not interested in the numerical approach to solution of kinematics.

Olsen’s work was done specifically on a Cincinnati-Milacron model T3-726 robotic manipulator; however, he used Denavit-Hartenberg parameters with the iterative solution explained here so that his software would fit any serial, mobility-six robotic arm without reprogramming, solely by changing data.

His tests were done by starting from a variety of postures, scattered throughout the robot workspace, and giving commands to move to other specified postures, also chosen haphazardly. In each case, he recorded the accuracy with which the commanded displacement was achieved after one, then two, then three iterations. Finally, he plotted the resulting displacement accuracy versus the requested displacement distance after each iteration of the algorithm. The results of these experiments are shown in the log-log plot of [Figure 6.1](#).

The figure shows the expected scatter that comes from experimental work. It also shows scatter coming from differences in the quality index (conditioning of the equations) in different parts of the workspace. However, it shows amazing consistency in spite of very large differences in the displacement distances requested.

[Figure 6.2](#), also taken from Olsen, shows the same information as [Figure 6.1](#), but with the data curve fit to reduce scatter and make the graph easier to interpret. The three shaded boundaries show limitations stemming from the application itself. The vertical shaded boundary on the right shows the full reach of the robot workspace. The robot physically cannot move to the right of this boundary. The horizontal shaded boundary near the bottom of the figure shows the smallest errors distinguishable by the sensors of the robot. This boundary represents a positioning

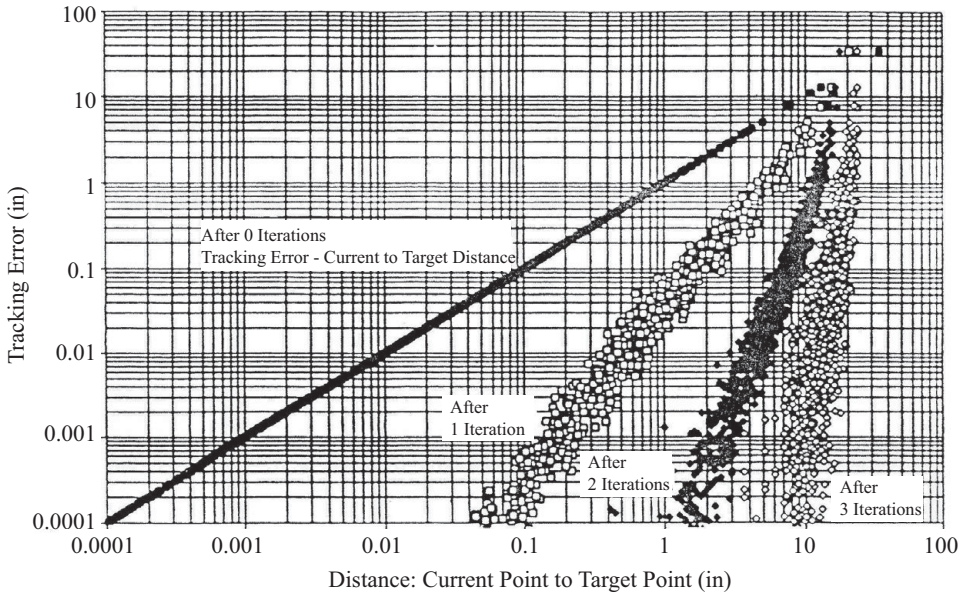


Figure 6.1. Displacement error versus point-to-point displacement distance for different algorithm iterations (copied from Olsen [8]).

accuracy limit of the robot actuators. The third (slanted) boundary shows an accuracy limit that comes from the design of the controller on that robot and the fact that it interpolates instructions in joint coordinates rather than in Cartesian coordinates. This boundary was established experimentally and the solid boundary shows

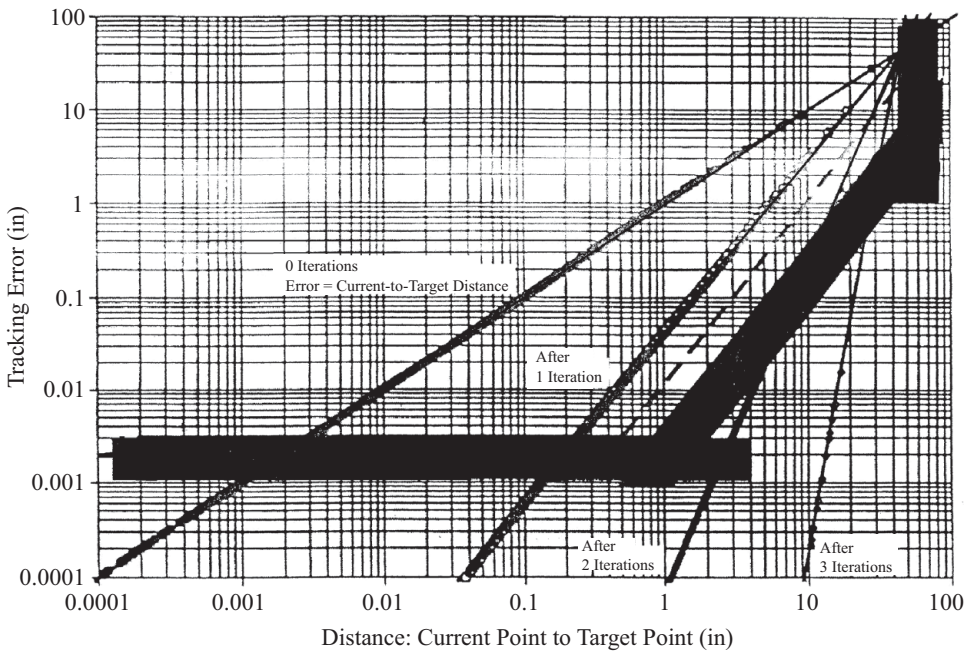


Figure 6.2. Displacement error versus point-to-point displacement distance for different algorithm iterations with application limits (copied from Olsen [8]).

the best-case data; the dashed line represents the worst-case data for this type of error. Overall, these boundaries set some limits on what accuracy can reasonably be expected or is meaningful for this application.

Overall, the conclusion strongly suggested by Figure 6.2 is that even though the algorithm is iterative, the application limitations are such that it seldom, if ever, makes sense to iterate! When we consider the context in which the algorithm is expected to be used, we don't need (or even want) an algorithm that takes 30-inch displacement increments. In analysis applications, we usually want results printed or plotted at much smaller intervals. In a control system, the controller and its algorithm usually requires much smaller increments. Thus, in general, our "iterative" algorithm will usually not need to iterate to achieve an appropriate accuracy for the situation.

Olsen also addresses the question of computational speed, and whether this provides a real limitation on the use of the numeric solution algorithm. He quotes [3] as follows:

In many path control schemes . . . it is necessary to calculate the inverse kinematics of a manipulator at fairly high rates, for example 30 Hz or faster. Therefore, computational efficiency is an issue. These speed requirements rule out the use of numerical solution techniques which are iterative in nature, and for this reason, we have not considered them here.

Using this figure of 30 Hz as a guide, Olsen timed his own (iterative) software running on different microprocessors of that time (1993) with different clock speeds. Using a computer based on the Intel 386 microprocessor chip with a clock speed of 20 MHz, his algorithm ran at 67 Hz for a single iteration per posture, or 34 Hz when two iterations per posture were used. Using the same code on a computer with an Intel 486 chip and a clock speed of 66 MHz, these rates became 620 Hz for a single iteration per posture, and 310 Hz for two iterations per posture. We note that these tests were done on computer chips of the early 1990s and all were faster than the rate cited in [3] at that time; some were more than an order of magnitude faster.

Although Olsen's experiments and conclusions are all based on a particular application – the control of a specific manipulator through the performance of a chosen task – his arguments and his conclusions are applicable to an amazing range of problems. Critics of the iterative approach to kinematic analysis have claimed for many years that such a method suffers from problems in convergence and is too inefficient for effective engineering use. Yet criticisms always come from those who have no first-hand experience and no counter-examples are offered with data comparable to Olsen's study. Software operating on desktop microprocessor systems today show that iteration can be a very effective method, and operates far faster than the analyst can read or digest results.

REFERENCES

1. J. Angeles, "Is There a Characteristic Length of a Rigid-Body Displacement?" *Mechanism and Machine Theory*, vol. 41, no. 8, August 2006, pp. 884–96.
2. R. S. Ball, *A Treatise on the Theory of Screws, 1900*, Cambridge University Press, Cambridge, 544 pp., reprinted in 1998.

3. J. J. Craig, *Introduction to Robotics Mechanics and Control*, 2nd. ed., Addison-Wesley Publishing Company, Inc., Cambridge, MA, 1989.
4. C. G. J. Jacobi, "De Determinantibus Functionalibus (On Determinant Functions)," *Crelle's Journal für die reine und angewandte Mathematik (Crelle's Journal for Pure and Applied Mathematics)*, vol. **22**, pp. 318–52, Leipzig, 1841.
5. M. L. James, G. M. Smith, and J. C. Wolford, *Applied Numerical Methods for Digital Computation with FORTRAN and CSMP*, 2nd ed., IEP-A Dun-Donnelley Publisher, New York, 1977, pp. 100–02.
6. K. S. Kaiser, *Numerical Analysis*, McGraw-Hill Book Company, Inc., 1957.
7. G. A. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Book Company, Inc., New York, 1961.
8. I. Newton, *De methodis fluxionum et serierum infinitorum (The method of fluxions and infinite series)*, London, 1736.
9. W. J. Olsen, "On the Real Time Inverse Kinematics Solution for the General Six Degree of Freedom Manipulator," *MS Thesis*, University of Wisconsin – Madison, Madison WI, 1994.
10. J. Raphson, *Analysis aequationum universalis (The universal analysis equation)*, London, 1690.
11. P. N. Sheth, "Improved Iterative Techniques for the (4×4) Matrix Method of Kinematic Analysis," *MS Thesis*, University of Wisconsin – Madison, Madison, 1968.
12. J. J. Uicker, Jr., J. Denavit, and R. S. Hartenberg, "An Iterative Method for the Displacement Analysis of Spatial Mechanisms," *Journal of Applied Mechanics, ASME Transactions*, June 1964, pp. 309–14.

PROBLEMS

- 6.1** Continue from the results of problems 4.4 and 5.3 to find the following:
 - a) Form the Q_h derivative operator matrix for each joint.
 - b) Form the D_h derivative operator matrix for each joint as a function of ψ .
- 6.2** Continue from the results of problem 6.1 as follows:
 - a) Form the Jacobian matrix \mathcal{J} as a function of ψ .
 - b) Rearrange the rows and columns of the Jacobian matrix \mathcal{J} and identify \mathcal{J}_{11} and \mathcal{J}_{12} .
 - c) Find the determinant and inverse of \mathcal{J}_{11} .
- 6.3** Prove Eq. (6.3) for the small displacement of the inverse transformation matrix as shown in section 6.2.
- 6.4** Continue from the results of problems 4.5 and 5.5 to find the following:
 - a) Form the Q_h derivative operator matrix for each joint.
 - b) Form the D_h derivative operator matrix for each joint as a function of ψ .
- 6.5** Continue from the results of problem 6.4 as follows:
 - a) Form the Jacobian matrix \mathcal{J} as a function of ψ .
 - b) Rearrange the rows and columns of the Jacobian matrix \mathcal{J} and identify \mathcal{J}_{11} and \mathcal{J}_{12} .
 - c) Find the determinant and inverse of \mathcal{J}_{11} .
- 6.6** For the Stanford manipulator of example 5.6, derive the Jacobian matrix relating differential joint displacements to the differential displacement of an end-effector coordinate system attached at the wrist center point.

7.1 Introduction

In Chapter 3 we defined the words position and posture as the terms that tell “where” an item is. Depending on the “item,” we find it convenient to use a Cartesian coordinate system as a global reference and we choose homogeneous coordinates to define the position of a point. We use the (4×4) transformation matrix T_{0b} to represent the posture of a rigid body, and we refer to “posture” to emphasize that we include both the orientation of the body as well as the location of a reference point. The posture of a mechanism or multibody system can usually be described by a vector of generalized coordinates ψ equal in number to the mobility of the system. However, because it is sometimes possible that a system can be assembled in more than one way for identical values of the generalized coordinates, we choose to represent the posture of a system by the vector ϕ that explicitly includes all of the joint variables.

In Chapter 3 we defined the term displacement as the change in position or posture of a point, a joint variable, a rigid body, or a system. Then, in Chapter 6, we showed how the concept of differential displacement leads naturally to the derivative of position or posture. We defined the very powerful derivative operator matrices, Q_h and D_h , to make the process of numeric differentiation both easy and precise. However, a quick review shows that derivatives are taken first with respect to a changing joint variable value because these are the variables on which the joint transformation matrices explicitly depend. Little is said about the fact that, in most mechanisms or multibody systems, many joint variables change simultaneously.

Because we wish to study motion, perhaps we should now identify what is meant by the word *motion*. A motion is not just any haphazard set of positions or postures. In a mechanical system, a motion is a systematic or orderly sequence of positions or postures of a point, joint, body, or system that results in accordance with the laws of mechanics.

From the beginning we have recognized the simultaneous change of many variables and we carefully identify the mobility f of our system as its number of independent degrees of freedom. We even choose a particular subset of the n joint variables ϕ , the generalized coordinates ψ , which we agree are to represent the independent

variables. Still, we say little about the ordering or about the relative sizes of changes in these generalized coordinates.

Ultimately, we must also ask where time enters into our problem or, for that matter, what is the definition and meaning of time? This is a deep question that has challenged philosophers throughout history. However, for the purposes of this text, we say that time is defined by the symbol t and it is the ultimate independent variable by which the order of events in nature is enumerated. It is a continuous scalar variable that is always positive and that increases as events occur. Time is physically measured by a clock and, in multibody and mechanical system dynamics, is usually measured in units of seconds.

7.2 Definition of Velocity

The term *velocity* is defined as the time rate of change of position or posture. Just as the positions or postures of different items have different sets of coordinates, so too do their velocities.

The velocity of a point, for example, is the time rate of change of its position. Because we find it convenient to use homogeneous coordinates to express the position of a point, we define the absolute velocity of a point by taking the time derivative of Eq. (5.9)

$$\dot{R} = \frac{dR}{dt} = \begin{bmatrix} \frac{dR^{x_0}}{dt} \\ \frac{dR^{y_0}}{dt} \\ \frac{dR^{z_0}}{dt} \\ 0 \end{bmatrix}, \quad (7.1)$$

where R^{x_0} , R^{y_0} , R^{z_0} are the point's global Cartesian coordinates.

The velocity of body b is given by

$$\dot{T}_{0b} = \frac{dT_{0b}}{dt}, \quad (7.2)$$

which may require the time derivative of its Ball vectors or screw coordinates. More will be said about this shortly.

The (relative) velocities of joint variables are found from the time derivative of Eq. (5.2)

$$\dot{\phi} = \frac{d\phi}{dt} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \vdots \\ \dot{\phi}_n \end{bmatrix}, \quad (7.3)$$

where the velocities within any individual joint are given by the time derivatives of the individual variables of the joint:

$$\dot{\phi}_h = \frac{d\phi_h}{dt} = \begin{bmatrix} \dot{\phi}_h^1 \\ \dot{\phi}_h^2 \\ \vdots \\ \dot{\phi}_h^f \end{bmatrix}, \quad h = 1, 2, \dots, n. \quad (7.4)$$

The velocity of the total system is fully defined by its vector of generalized velocities that are the time derivatives of the vector of generalized coordinates. Taking the time derivative of Eq. (5.1) we find

$$\dot{\psi} = \frac{d\psi}{dt} = \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \vdots \\ \dot{\psi}_f \end{bmatrix}. \quad (7.5)$$

It is true that $\dot{\phi}$ of Eq. (7.3) also fully defines the velocity of the system, and may better fit as the system velocity definition because we chose ϕ to represent the system position. This is not necessary, however, because there is no ambiguity in velocity for a chosen closure. We will have need for both $\dot{\psi}$ and $\dot{\phi}$.

Once the independent motions are known, the motions of the dependent joint variables are dictated by the motions of the generalized coordinates. They are explicit functions of the generalized coordinates

$$\phi = \phi(\psi), \quad (7.6)$$

and through these, they become implicit functions of time. Similarly, the motions of bodies and points are explicit functions of the generalized coordinates and only implicitly, through them, become functions of time. It is important to understand this hierarchy as we proceed.

Of course, for each of these velocity quantities we will need a good operational strategy for numerically finding the indicated derivatives. These methods, which first appeared in [1], are the subject of this chapter. Before going into detail, however, we should recognize that, ultimately, time is the underlying independent variable for all. The f generalized coordinates, both the NS specified generalized coordinates and the NF free generalized coordinates, are explicit functions of time:

$$\psi = \psi(t). \quad (7.7)$$

The SGC motions are directly specified by the analyst, and the FGC motions are found by the laws of mathematics and mechanics, but all are explicit functions of time. Finding the FGC motions becomes the primary objective of Chapters 12, 13, 14, and 16.

7.3 First Geometric Derivatives of Joint Variables

Thinking of this hierarchy of dependency, let us first find derivatives of the joint variables with respect to the generalized coordinates, on which they depend explicitly. We call these *first geometric derivatives*, and we use a prime notation with subscripts as the symbolism. For a joint that has more than a single joint variable, a superscript may also be required:

$$\phi'_{h,j} = \frac{\partial \phi_h}{\partial \psi_j}, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ j = 1, 2, \dots, f. \end{array} \quad (7.8)$$

How will we find these first geometric derivatives? Well, if we differentiate the kinematic loop-closure equations with respect to generalized coordinate ψ_j , we recognize that the loops depend directly on the joint variables and we write

$$\frac{\partial T_{00}}{\partial \psi_j} = \sum_{h=1}^n \frac{\partial T_{00}}{\partial \phi_h} \phi'_{h,j} = 0, \quad j = 1, 2, \dots, f.$$

Next, from Eq. (6.16), we utilize the D_h derivative operator matrices to get

$$\sum_{h=1}^n L(i, h) D_h \phi'_{h,j} T_{00} = 0, \quad \begin{array}{l} i = 1, 2, \dots, NL, \\ j = 1, 2, \dots, f. \end{array}$$

In the case of a multi-variable joint h , multiple terms and superscripts are required. If joint h is a spheric joint, for example, then

$$D_h \phi'_{h,j} = D_h^1 \phi_{h,j}^1 + D_h^2 \phi_{h,j}^2 + D_h^3 \phi_{h,j}^3 + D_h^4 \phi_{h,j}^4.$$

Now we recognize that $T_{00} = I$ and we eliminate redundancy by replacing each (4×4) D_h matrix by its equivalent (6×1) vector of screw coordinates

$$\sum_{h=1}^n L(i, h) \hat{D}_h \phi'_{h,j} = 0, \quad \begin{array}{l} i = 1, 2, \dots, NL, \\ j = 1, 2, \dots, f, \end{array} \quad (7.9)$$

with an additional constraint equation of the form

$$2\phi_h^1 \phi_{h,j}^1 + 2\phi_h^2 \phi_{h,j}^2 + 2\phi_h^3 \phi_{h,j}^3 + 2\phi_h^4 \phi_{h,j}^4 = 0, \quad j = 1, 2, \dots, f, \quad (7.10)$$

for each spheric joint in the system, and another of the form

$$2\phi_h^4 \phi_{h,j}^4 + 2\phi_h^5 \phi_{h,j}^5 + 2\phi_h^6 \phi_{h,j}^6 + 2\phi_h^7 \phi_{h,j}^7 = 0, \quad j = 1, 2, \dots, f, \quad (7.11)$$

for each open joint in the system – a total of NC constraint equations, which come from taking derivatives of Eqs. (4.17) and (4.24) with respect to generalized coordinate ψ_j .

Because these equations are of the same form as Eqs. (6.42), (6.43), and (6.44), we can now write, similar to the form of Eq. (6.45), that

$$\mathcal{J} \phi' = 0, \quad (7.12)$$

where \mathcal{J} is the same $[(6NL + NC) \times n]$ Jacobian matrix that resulted at convergence of the numeric iteration process of section 6.5, and ϕ' is the $(n \times f)$ matrix of first geometric derivatives $\phi'_{h,j}$ defined in Eq. (7.8).

Reviewing the numeric iteration process of section 6.5.3, we recall that the modified pivoting scheme used there has reordered the joint variables so that all dependent joint variables are numbered first, followed by the FGC variables, and finally by the SGC variables. Recognizing that both the FGC and the SGC variables have been accepted as the generalized coordinates ψ , we see that they are independent of each other by definition. Therefore, the ϕ' matrix must be of the form

$$\phi' = \begin{bmatrix} \frac{\partial\phi_1}{\partial\psi_1} & \frac{\partial\phi_1}{\partial\psi_2} & \cdots & \frac{\partial\phi_1}{\partial\psi_f} \\ \frac{\partial\phi_2}{\partial\psi_1} & \frac{\partial\phi_2}{\partial\psi_2} & \cdots & \frac{\partial\phi_2}{\partial\psi_f} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial\phi_{n-f}}{\partial\psi_1} & \frac{\partial\phi_{n-f}}{\partial\psi_2} & \cdots & \frac{\partial\phi_{n-f}}{\partial\psi_f} \\ \hline 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \phi'_{dep} \\ I \end{bmatrix}, \quad (7.13)$$

where ϕ'_{dep} is an $[(n-f) \times f]$ submatrix of the dependent first geometric derivative values.

Once we recognize this form for ϕ' , we return to Eq. (7.12) and subdivide the Jacobian into compatible size submatrices:

$$[\mathcal{J}_{dep} \mid \mathcal{J}_{ind}] \begin{bmatrix} \phi'_{dep} \\ - \\ I \end{bmatrix} = 0.$$

However, remembering Eq. (6.50), we see that this is the same subdivision that resulted from the modified Gauss-Jordan elimination process of section 6.5.3. Reverting to the notation of Eq. (6.50), we have

$$\begin{bmatrix} \mathcal{J}_{11} & \mathcal{J}_{12} & \mathcal{J}_{13} \\ \mathcal{J}_{21} & \mathcal{J}_{22} & \mathcal{J}_{23} \end{bmatrix} \begin{bmatrix} \phi'_{dep} \\ - \\ I \end{bmatrix} = 0,$$

and, after the elimination process is complete, these equations are reduced to the form

$$\begin{bmatrix} I & \mathcal{J}_{11}^{-1}\mathcal{J}_{12} & \mathcal{J}_{11}^{-1}\mathcal{J}_{13} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi'_{dep} \\ - \\ I \end{bmatrix} = 0,$$

which has for its solution

$$\phi'_{dep} = -[\mathcal{J}_{11}^{-1}\mathcal{J}_{12} \quad \mathcal{J}_{11}^{-1}\mathcal{J}_{13}]. \quad (7.14)$$

Thus, we see that there is absolutely no calculation left to be done. Once the numeric iteration process of section 6.5 has converged to a solution, the ϕ'_{dep} matrix

of first geometric derivatives of Eq. (7.13) can be copied directly from the top-right corner of the Gauss-Jordan row-reduced form of the \mathcal{J} matrix, requiring only negation and augmentation by an $(f \times f)$ identity matrix. This is a very useful and “free” by-product of our numeric iteration process.

These first geometric derivatives are not velocities, even though they are sometimes referred to in this manner. They are rates of change of the dependent joint variables with respect to changes of the generalized coordinates. Time is not included in them and they typically have units such as radians per radian. They represent totally geometric information and are functions of ψ alone. They are very important, however, and arise frequently in our coming work.

EXAMPLE 7.1 To illustrate the process, let us continue the analysis of the Cardan/Hooke universal joint started in example 5.1. From that example, we have already found solutions for the positions of all joint variables and the transformation matrices of the various bodies. From those, we find the derivative operator matrices for each of the joints. Entries are converted to functions of the generalized coordinate ψ alone by use of the position solutions. After a bit of algebra, these are

$$D_A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_B = \begin{bmatrix} 0 & \frac{\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ \frac{\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_C = \begin{bmatrix} 0 & \cos \beta \sin \psi & \cos \psi & -h \cos \beta \sin \psi \\ -\cos \beta \sin \psi & 0 & -\sin \beta \sin \psi & 0 \\ -\cos \psi & \sin \beta \sin \psi & 0 & -h \sin \beta \sin \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D = \begin{bmatrix} 0 & -\sin \beta & 0 & h \sin \beta \\ \sin \beta & 0 & -\cos \beta & 0 \\ 0 & \cos \beta & 0 & -h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Next, we formulate the terms of Eq. (7.12):

$$\mathcal{J}\phi' = \begin{bmatrix} 0 & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -h \cos \beta \sin \psi & h \sin \beta \\ 0 & 0 & 0 & 0 \\ -h & 0 & -h \sin \beta \sin \psi & -h \cos \beta \\ 1 & 0 & \sin \beta \sin \psi & \cos \beta \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi & 0 \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi & \sin \beta \end{bmatrix} \begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \\ \phi'_{D,1} \end{bmatrix} = 0.$$

Recognizing that the first and third rows of coefficients are simply h multiples of the sixth and negative fourth rows, respectively, and that because $\psi_1 = \phi_D$, we have $\phi'_{D,1} = 1$. Therefore, the first three equations are equivalent to the lower $[\mathcal{J}_{21} \mid \mathcal{J}_{22} \mid \mathcal{J}_{23}]$ portion of the Jacobian and can be dropped. Those remaining can be rearranged into the form

$$\begin{bmatrix} 1 & 0 & \sin \beta \sin \psi \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi \end{bmatrix} \begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \end{bmatrix} = \begin{bmatrix} -\cos \beta \\ 0 \\ -\sin \beta \end{bmatrix}.$$

The resulting solution for the first geometric derivatives is

$$\begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \\ \phi'_{D,1} \end{bmatrix} = \begin{bmatrix} \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{\sin \beta \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{\sin \beta \cos \beta \sin \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{1}{1} \end{bmatrix}.$$

It can also be noted that the quality index, from Eq. (6.53), is

$$Q = |\det(\mathcal{J}_{11})| = \sqrt{1 - \sin^2 \beta \sin^2 \psi},$$

and that its value remains in the range ($|\cos \beta| \leq Q \leq 1.00$). Because the denominators of these geometric derivatives are either Q or Q^2 , there should be no difficulty in the operation of the device or in the numeric evaluation of these geometric derivatives for reasonable values of β .

7.4 Velocities of Joint Variables

The (relative) velocities of the joint variables, as shown by Eq. (7.3), can be found in any of three ways. If we wish to do hand calculations, the expressions found for

the positions of the joint variables can be directly differentiated analytically. This approach, however, is highly susceptible to human error and is not easily adapted to computer solution. The second approach is to differentiate the loop-closure equations with respect to time, as we will see later. The third approach is to take advantage of the first geometric derivatives. Remembering the hierarchy of dependencies, we write

$$\dot{\phi}_h = \frac{d\phi_h}{dt} = \sum_{j=1}^f \frac{\partial \phi_h}{\partial \psi_j} \frac{d\psi_j}{dt} = \sum_{j=1}^f \phi'_{h,j} \dot{\psi}_j, \quad h = 1, 2, \dots, n, \quad (7.15)$$

which can be written in matrix form as

$$\dot{\phi} = \phi' \dot{\psi}. \quad (7.16)$$

Here we see that all dependent joint velocities vary linearly with the generalized velocities. They are nonlinear functions of ψ , however, through the first geometric derivatives.

As previously mentioned, we can differentiate the loop-closure equations with respect to time. From Eq. (6.16) we get

$$\sum_{h=1}^n L(i, h) D_h T_{00} \dot{\phi}_h = 0, \quad i = 1, 2, \dots, NL,$$

where, again, in the case of a multi-variable joint h , multiple terms and superscripts are required. If joint h is a spheric joint, for example, then

$$D_h \dot{\phi}_h = D_h^1 \dot{\phi}_h^1 + D_h^2 \dot{\phi}_h^2 + D_h^3 \dot{\phi}_h^3 + D_h^4 \dot{\phi}_h^4.$$

After recognizing that $T_{00} = I$, we eliminate redundancy by replacing each D_h matrix by its equivalent (6×1) vector of screw coordinates

$$\sum_{h=1}^n L(i, h) \hat{D}_h \dot{\phi}_h = 0, \quad i = 1, 2, \dots, NL, \quad (7.17)$$

with an additional constraint equation of the form

$$2\phi_h^1 \dot{\phi}_h^1 + 2\phi_h^2 \dot{\phi}_h^2 + 2\phi_h^3 \dot{\phi}_h^3 + 2\phi_h^4 \dot{\phi}_h^4 = 0,$$

for each spheric joint, and another of the form

$$2\phi_h^4 \dot{\phi}_h^4 + 2\phi_h^5 \dot{\phi}_h^5 + 2\phi_h^6 \dot{\phi}_h^6 + 2\phi_h^7 \dot{\phi}_h^7 = 0,$$

for each open joint in the system, thereby creating a total of NC constraint equations, which come from taking derivatives of Eqs. (4.17) and (4.24) with respect to time.

Then, just as with first geometric derivatives in Eq. (7.12), like Eq. (6.42), we see that

$$\mathcal{J} \dot{\phi} = 0. \quad (7.18)$$

Recognizing that the highest numbered f of these joint variable velocities are the generalized coordinate velocities, these equations can be written as

$$\begin{bmatrix} \mathcal{J}_{11} & | & \mathcal{J}_{12} & \mathcal{J}_{13} \\ \mathcal{J}_{21} & | & \mathcal{J}_{22} & \mathcal{J}_{23} \end{bmatrix} \begin{bmatrix} \dot{\phi}_{dep} \\ \dot{\psi} \end{bmatrix} = 0$$

and can then be solved as was Eq. (7.12):

$$\dot{\phi}_{dep} = - [\mathcal{J}_{11}^{-1} \mathcal{J}_{12} \quad \mathcal{J}_{11}^{-1} \mathcal{J}_{13}] \dot{\psi} = \phi'_{dep} \dot{\psi}. \quad (7.19)$$

However, this is identical with Eq. (7.16) and shows no new advantages.

EXAMPLE 7.2 Let us continue the analysis of the Cardan/Hooke universal joint of example 7.1. If the input shaft is driven at a rate of $\dot{\phi}_D = \dot{\psi}$, then let us find the velocities of the other joint variables. These are given directly by Eq. (7.16) and the first geometric derivatives found in example 7.1:

$$\dot{\phi} = \begin{bmatrix} \dot{\phi}_A \\ \dot{\phi}_B \\ \dot{\phi}_C \\ \dot{\phi}_D \end{bmatrix} = \begin{bmatrix} \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} \\ \frac{\sin \beta \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \dot{\psi} \\ \frac{\sin \beta \cos \beta \sin \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} \\ \dot{\psi} \end{bmatrix}.$$

7.5 First Geometric Derivatives of Body Postures

As with joint variables, before we find the time derivative of the posture of a body, it is wise to consider the hierarchy of dependencies involved. We recall that the posture of body number b with respect to ground is described by its transformation matrix T_{0b} . Because this is made up of a series of products of shape matrices S_{bh} and joint matrices $\Phi_h(\phi_h)$, the only variables in the T_{0b} matrix are the joint matrices Φ_h , which are functions of the joint variables ϕ_h . These joint variables are functions of the generalized coordinates ψ_j that, in turn, are functions of time, t .

Therefore, let us take derivatives with respect to each level of the hierarchy in turn. From Eq. (6.16), we have

$$\frac{\partial T_{0b}}{\partial \phi_h} = P(b, h) D_h T_{0b}, \quad \begin{matrix} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n. \end{matrix}$$

Next, let us find the derivative of T_{0b} with respect to a generalized coordinate ψ_j . Because there may be several joints on the path from ground to body b ,

$$\frac{\partial T_{0b}}{\partial \psi_j} = \sum_{h=1}^n \frac{\partial T_{0b}}{\partial \phi_h} \frac{\partial \phi_h}{\partial \psi_j}, \quad \begin{matrix} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f. \end{matrix}$$

However, in view of the previous equation and Eq. (7.8), this becomes

$$\frac{\partial T_{0b}}{\partial \psi_j} = \sum_{h=1}^n P(b, h) D_h T_{0b} \phi'_{h,j}, \quad \begin{matrix} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f, \end{matrix}$$

where, in the case of a multi-variable joint h , multiple terms and superscripts are required. If joint h is a spheric joint, for example, then

$$D_h \phi'_{h,j} = D_h^1 \phi_{h,j}^1 + D_h^2 \phi_{h,j}^2 + D_h^3 \phi_{h,j}^3 + D_h^4 \phi_{h,j}^4.$$

This particular form arises so often in coming developments that we find it convenient to define another symbol as follows:

$$W_{b,j} = \sum_{h=1}^n P(b,h) D_h \phi'_{h,j}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f, \end{array} \quad (7.20)$$

so that the previous equation reduces to

$$\frac{\partial T_{0b}}{\partial \psi_j} = W_{b,j} T_{0b}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f. \end{array} \quad (7.21)$$

Studying the form of Eq. (7.20), we see that it is a weighted sum of D_h operator matrices, weighted by the relative rates of change of each joint variable along the path from ground to body b with respect to the change of generalized coordinate ψ_j . One immediate conclusion is that, because $P(b,h)$ and $\phi'_{h,j}$ are both scalar factors, the screw coordinate pattern of the Q_h and D_h matrices is also preserved in the $W_{b,j}$ operator matrices.

The six independent elements of $W_{b,j}$ define the instantaneous helical motion of body b with respect to a change of one generalized coordinate, ψ_j . Note that the six elements can be interpreted geometrically as an instantaneous screw axis, uniquely oriented and located in the global coordinate system as discussed in section 6.4, and scaled in size to show the rate of the helical movement of body b with respect to a change in generalized coordinate ψ_j .

Note how, in Eq. (7.20), $W_{b,j}$ is found by summing the contributions of the relative rates of change of the joint variables along the path from ground to body b . However, once these are summed, we have the rate of change of the posture of body b with respect to ground. Note also that Eq. (7.21) answers our earlier question of how the motions of several joint variables that change simultaneously combine to fully describe the motion of a particular body.

EXAMPLE 7.3 Let us now continue the analysis of the Cardan/Hooke universal joint of example 5.1 and example 7.1, and find the geometric derivative operator matrices $W_{b,j}$ for each of its bodies.

From Example 7.1 we have

$$D_A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_B = \begin{bmatrix} 0 & \frac{\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ \frac{\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_C = \begin{bmatrix} 0 & \cos \beta \sin \psi & \cos \psi & -h \cos \beta \sin \psi \\ -\cos \beta \sin \psi & 0 & -\sin \beta \sin \psi & 0 \\ -\cos \psi & \sin \beta \sin \psi & 0 & -h \sin \beta \sin \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D = \begin{bmatrix} 0 & -\sin \beta & 0 & h \sin \beta \\ \sin \beta & 0 & -\cos \beta & 0 \\ 0 & \cos \beta & 0 & -h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$\begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \\ \phi'_{D,1} \end{bmatrix} = \begin{bmatrix} \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{\sin \beta \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{\sin \beta \cos \beta \sin \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{1}{1} \end{bmatrix}.$$

With these, we use Eq. (7.20) to find

$$W_{1,1} = D_A \phi'_{A,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ 0 & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{2,1} = W_{1,1} + D_B \phi'_{B,1} = \begin{bmatrix} 0 & \frac{\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-h \sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{-\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{3,1} = W_{2,1} + D_C \phi'_{C,1} = \begin{bmatrix} 0 & \sin \beta & 0 & -h \sin \beta \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & -\cos \beta & 0 & h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{4,1} = W_{3,1} + D_D \phi'_{D,1} = 0.$$

Note that $W_{4,1}$ becoming zero confirms that no point of the frame, body 4, moves when the input shaft is rotated.

7.6 Velocities of Bodies

In Eq. (7.2) we defined the velocity of a body with respect to the global reference frame to be the time derivative of its transformation matrix:

$$\dot{T}_{0b} = \frac{dT_{0b}}{dt}, \quad b = 1, 2, \dots, \ell.$$

From Eq. (6.16), we have

$$\frac{\partial T_{0b}(\phi_h)}{\partial \phi_h} = P(b, h)D_h T_{0b}(\phi_h), \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ h = 1, 2, \dots, n. \end{array}$$

Therefore, employing chain-rule differentiation, we write

$$\begin{aligned} \dot{T}_{0b} &= \sum_{h=1}^n \frac{\partial T_{0b}}{\partial \phi_h} \frac{d\phi_h}{dt} \\ &= \sum_{h=1}^n P(b, h)D_h T_{0b} \dot{\phi}_h, \quad b = 1, 2, \dots, \ell. \end{aligned}$$

Following the lead of the previous section, we now define yet another derivative operator matrix

$$\omega_b = \sum_{h=1}^n P(b, h)D_h \dot{\phi}_h, \quad b = 1, 2, \dots, \ell. \quad (7.22)$$

As shown previously, in the case of a multi-variable joint h , multiple terms and superscripts are required. If joint h is a spheric joint, for example, then

$$D_h \dot{\phi}_h = D_h^1 \dot{\phi}_h^1 + D_h^2 \dot{\phi}_h^2 + D_h^3 \dot{\phi}_h^3 + D_h^4 \dot{\phi}_h^4.$$

From these equations we can write

$$\dot{T}_{0b} = \omega_b T_{0b}, \quad b = 1, 2, \dots, \ell. \quad (7.23)$$

By substituting Eq. (7.15) into Eq. (7.22), we find

$$\omega_b = \sum_{h=1}^n P(b, h)D_h \sum_{j=1}^f \phi'_{h,j} \dot{\psi}_j, \quad b = 1, 2, \dots, \ell,$$

which can be rearranged to read

$$\omega_b = \sum_{h=1}^n \sum_{j=1}^f P(b, h)D_h \phi'_{h,j} \dot{\psi}_j, \quad b = 1, 2, \dots, \ell,$$

and, comparing this with Eq. (7.20), we find that

$$\omega_b = \sum_{j=1}^f W_{b,j} \dot{\psi}_j, \quad b = 1, 2, \dots, \ell. \quad (7.24)$$

This latest formula is simply another way of computing the ω_b matrices for the various bodies. Whether Eq. (7.22) or (7.24) should be used depends totally on convenience.

EXAMPLE 7.4 Continuing Example 7.3, the ω_b matrices for each of the bodies of a Cardan/Hooke universal shaft coupling are easily found by Eq. (7.24):

$$\omega_1 = W_{1,1}\dot{\psi} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & 0 \\ 0 & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\omega_2 = W_{2,1}\dot{\psi} = \begin{bmatrix} 0 & \frac{\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & \frac{-h \sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} \\ \frac{-\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & 0 \\ \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\omega_3 = W_{3,1}\dot{\psi} = \begin{bmatrix} 0 & \sin \beta \dot{\psi} & 0 & -h \sin \beta \dot{\psi} \\ -\sin \beta \dot{\psi} & 0 & \cos \beta \dot{\psi} & 0 \\ 0 & -\cos \beta \dot{\psi} & 0 & h \cos \beta \dot{\psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\omega_4 = W_{4,1}\dot{\psi} = 0.$$

7.7 First Geometric Derivatives of Point Positions

We recall from Eq. (4.2) that the global position of a point attached to body b is given by

$$R_b = T_{0b}r_b, \quad b = 1, 2, \dots, \ell.$$

Now, because the point is attached to body b , which remains rigid, the point's local coordinates r_b are constants. The transformation matrix T_{0b} for body b , however, is a function of the joint matrices along its path, which are functions of the joint variables ϕ_h , and these are functions of the generalized coordinates ψ_j that, in turn, are functions of time.

Derivatives of our point's position with respect to a single joint variable on its path are found by Eq. (6.15). In this section, we wish to find the derivative of the global position of our point with respect to the generalized coordinate ψ_j . Using Eq. (7.21) to differentiate the previous position equation and remembering that r_b is constant, we get

$$R'_{b,j} = \frac{\partial R_b}{\partial \psi_j} = W_{b,j}T_{0b}r_b = W_{b,j}R_b, \quad \begin{matrix} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f. \end{matrix} \quad (7.25)$$

7.8 Velocities of Points

Equation (7.1) defines the velocity of a point as the time derivative of its global position. Using chain-rule differentiation we get

$$\dot{R}_b = \frac{dR_b}{dt} = \sum_{j=1}^f \frac{\partial R_b}{\partial \psi_j} \frac{d\psi_j}{dt} = \sum_{j=1}^f R'_{b,j} \dot{\psi}_j, \quad b = 1, 2, \dots, \ell.$$

Then, by Eq. (7.25), this becomes

$$\dot{R}_b = \sum_{j=1}^f W_{b,j} R_b \dot{\psi}_j, \quad b = 1, 2, \dots, \ell, \quad (7.26)$$

and according to Eq. (7.24),

$$\dot{R}_b = \omega_b R_b, \quad b = 1, 2, \dots, \ell. \quad (7.27)$$

Thus, we see that the same derivative operator matrices found for taking either geometric or time derivatives of body postures are also used for differentiating point positions.

REFERENCES

1. J. Denavit, R. S. Hartenberg, R. Razi, and J. J. Uicker, Jr., "Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages," *Journal of Applied Mechanics, ASME Transactions*, 1965, pp. 903–10.

PROBLEMS

7.1 Continue from the results of problems 6.1 and 6.2 as follows:

- a) Form the matrix of first geometric derivatives of the joint variables.
- b) Form the matrix of first time derivatives of the joint variables.
- c) Form the first geometric derivative operator matrix $W_{b,j}$ for each body.
- d) Form the velocity operator matrix ω_b for each body.

7.2 Continue from the results of problems 6.4 and 6.5 as follows:

- a) Form the matrix of first geometric derivatives of the joint variables.
- b) Form the matrix of first time derivatives of the joint variables.
- c) Form the first geometric derivative operator matrix $W_{b,j}$ for each body.
- d) Form the velocity operator matrix ω_b for each body.

7.3 Consider the double Cardan/Hooke joint of problem 5.8. Derive the equation for the velocity ratio of the output to input shafts and show that this mechanism is a constant velocity coupling.

8 Acceleration Analysis

8.1 Definition of Acceleration

The term *acceleration* is defined as a time rate of change of velocity. The acceleration of a point, for example, is the time rate of change of the velocity of that point. Because we find it convenient to use homogeneous coordinates to express the position and velocity of a point, we define the absolute acceleration of a point by taking the time derivative of its absolute velocity from Eq. (7.1):

$$\ddot{\mathbf{R}} = \frac{d^2 \mathbf{R}}{dt^2} = \begin{bmatrix} \frac{d^2 R^{x_0}}{dt^2} \\ \frac{d^2 R^{y_0}}{dt^2} \\ \frac{d^2 R^{z_0}}{dt^2} \\ 0 \end{bmatrix}, \quad (8.1)$$

where R^{x_0} , R^{y_0} , R^{z_0} are the point's global Cartesian coordinates.

Consistent with our description of the velocity of body b in Eq. (7.2), the acceleration of body b is given by

$$\ddot{T}_{0b} = \frac{d^2 T_{0b}}{dt^2}, \quad b = 1, 2, \dots, \ell, \quad (8.2)$$

that may require the second time-derivative of its screw coordinates. More will be said on this shortly.

The accelerations of joint variables are the time derivatives of their velocities. From the time derivative of Eq. (7.3) we have

$$\ddot{\phi} = \frac{d^2 \phi}{dt^2} = \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \vdots \\ \ddot{\phi}_n \end{bmatrix}, \quad (8.3)$$

where the accelerations within an individual joint are given by the second time-derivatives of the individual motion variables of that joint. That is

$$\ddot{\phi}_h = \frac{d^2\phi_h}{dt^2} = \begin{bmatrix} \ddot{\phi}_h^1 \\ \ddot{\phi}_h^2 \\ \vdots \\ \ddot{\phi}_h^{f_h} \end{bmatrix}. \quad (8.4)$$

The acceleration of a complete mechanism or multibody system is defined by its vector of *generalized accelerations* that are the time derivatives of the vector of generalized velocities. From the time derivative of Eq. (7.5) we find

$$\ddot{\psi} = \frac{d^2\psi}{dt^2} = \begin{bmatrix} \ddot{\psi}_1 \\ \ddot{\psi}_2 \\ \vdots \\ \ddot{\psi}_f \end{bmatrix}. \quad (8.5)$$

It is true that $\ddot{\phi}$ of Eq. (8.3) also fully defines the accelerations of a mechanical system, and may be better fit as the system acceleration definition because we defined ϕ to represent the system posture. This is not necessary, however, because there is no ambiguity in accelerations coming from multiple closures. We will have need for both $\ddot{\phi}$ and $\ddot{\psi}$.

Of course, to solve acceleration problems by hand calculation, we can proceed by directly differentiating with respect to time the velocity equations found in Chapter 7. However, recalling that we are seeking a numeric method suited to digital computation, we prefer to recall from section 7.2 that kinematic quantities are found as functions of the joint variables ϕ that are functions of the generalized coordinates ψ , and that these, in turn, are functions of time. The methods of subsequent sections were first published as [1].

8.2 Derivatives of the Q_h Operator Matrices

Because we will use chain-rule differentiation, we first seek the derivatives of our Q_h derivative operator matrices. Reviewing section 6.3, we recall that several of the Q_h matrices contain only zeroes and ones because their instantaneous screw axes and Ball vectors remain constant in the coordinate system of the body preceding the joint. However, for some of the joints, the instantaneous screw axes or Ball vectors are not constant, but vary with changes of the joint variables. Thus, we need to seek out formulae for derivatives of each of the Q_h operators with respect to each of the joint variables on which it depends. In general, remembering that a joint h may have more than one joint variable, these are of the form

$$\frac{\partial Q_h^g}{\partial \phi_h^i} = Q_h^{g,i}, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ g, i = 1, 2, \dots, f_h. \end{array} \quad (8.6)$$

The derivative of Q_h^g with respect to a generalized coordinate ψ_j , therefore, is given by

$$\frac{\partial Q_h^g}{\partial \psi_j} = \sum_{i=1}^{f_h} Q_h^{g,i} \phi_{h,j}^i, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ g = 1, 2, \dots, f_h, \\ j = 1, 2, \dots, f. \end{array} \quad (8.7)$$

However, as we have done above, we will often write this with the reduced symbolism

$$\frac{\partial Q_h^g}{\partial \psi_j} = Q_h^{g'} \phi_{h,j}', \quad \begin{array}{l} h = 1, 2, \dots, n, \\ g = 1, 2, \dots, f_h, \\ j = 1, 2, \dots, f. \end{array}$$

The derivative of Q_h^g with respect to time is

$$\frac{dQ_h^g}{dt} = \sum_{i=1}^{f_h} Q_h^{g,i} \dot{\phi}_h^i, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ g = 1, 2, \dots, f_h. \end{array} \quad (8.8)$$

Again, in the interest of brevity, we will often write this in the reduced symbolism,

$$\frac{dQ_h^g}{dt} = Q_h^{g'} \dot{\phi}_h, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ g = 1, 2, \dots, f_h. \end{array}$$

Because the form of $Q_h^{g,i}$ depends on the type of joint in question, we must seek these out separately for each of the joint types.

8.2.1 Helical (Screw) Joint

A helical joint has only one joint variable and its Q_h operator matrix consists entirely of constants. Thus,

$$Q_h' = 0. \quad (8.9)$$

8.2.2 Revolute Joint

A revolute joint also has only one joint variable and its Q_h operator matrix also consists entirely of zeroes and ones. Thus,

$$Q_h' = 0. \quad (8.10)$$

8.2.3 Prismatic Joint

A prismatic joint also has only one joint variable and its Q_h operator matrix also consists entirely of zeroes and ones. Thus,

$$Q_h' = 0. \quad (8.11)$$

8.2.4 Cylindric Joint

A cylindric joint has two joint variables, and both of its Q_h^i operator matrices consist entirely of zeroes and ones. Thus,

$$Q_h^{1,1} = Q_h^{1,2} = Q_h^{2,1} = Q_h^{2,2} = 0. \quad (8.12)$$

8.2.5 Spheric Joint

A spheric joint has four Euler-Rodrigues parameters as joint variables and the forms of the four Q_h^s operator matrices are given in Eqs. (6.22). Differentiating these, we find

$$\begin{aligned} Q_h^{1,1} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,2} &= \begin{bmatrix} 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{1,3} &= \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,4} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{2,1} &= \begin{bmatrix} 0 & -2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{2,2} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{2,3} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{2,4} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{3,1} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{3,2} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{3,3} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{3,4} &= \begin{bmatrix} 0 & -2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{4,1} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{4,2} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{4,3} &= \begin{bmatrix} 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{4,4} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (8.13)$$

In addition, each spheric joint has a constraint equation relating the second geometric derivatives of its Euler-Rodrigues parameters with respect to generalized coordinates ψ_j and ψ_k .

$$\begin{aligned} & 2\phi_h^1\phi_{h,j,k}''^1 + 2\phi_h^2\phi_{h,j,k}''^2 + 2\phi_h^3\phi_{h,j,k}''^3 + 2\phi_h^4\phi_{h,j,k}''^4 \\ & = -2\phi_{h,j}^1\phi_{h,k}^1 - 2\phi_{h,j}^2\phi_{h,k}^2 - 2\phi_{h,j}^3\phi_{h,k}^3 - 2\phi_{h,j}^4\phi_{h,k}^4. \end{aligned} \quad (8.14)$$

8.2.6 Flat Joint

A flat joint has three joint variables and the forms of the three Q_h^g operator matrices are given in Eqs. (6.23). Differentiating these gives

$$Q_h^{1,1} = Q_h^{1,2} = Q_h^{1,3} = 0, \quad Q_h^{2,1} = Q_h^{2,2} = Q_h^{2,3} = 0,$$

$$Q_h^{3,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^{3,2} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^{3,3} = 0. \quad (8.15)$$

8.2.7 Rigid Joint

A rigid joint has no joint variables and no Q_h operator matrices. Therefore, there are no Q'_h matrices.

8.2.8 Open Joint

An open joint has seven joint variables and the forms of the seven Q_h^g operator matrices are given in Eqs. (6.24). Differentiating these gives:

$$\begin{aligned} Q_h^{1,1} &= Q_h^{1,2} = Q_h^{1,3} = Q_h^{1,4} = Q_h^{1,5} = Q_h^{1,6} = Q_h^{1,7} = 0, \\ Q_h^{2,1} &= Q_h^{2,2} = Q_h^{2,3} = Q_h^{2,4} = Q_h^{2,5} = Q_h^{2,6} = Q_h^{2,7} = 0, \\ Q_h^{3,1} &= Q_h^{3,2} = Q_h^{3,3} = Q_h^{3,4} = Q_h^{3,5} = Q_h^{3,6} = Q_h^{3,7} = 0, \\ Q_h^{4,1} &= Q_h^{4,2} = Q_h^{4,3} = 0, \end{aligned}$$

$$\begin{aligned} Q_h^{4,4} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{4,5} &= \begin{bmatrix} 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ Q_h^{4,6} &= \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{4,7} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned}
Q_h^{5,1} = Q_h^{5,2} = Q_h^{5,3} &= 0, \\
Q_h^{5,4} &= \begin{bmatrix} 0 & -2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{5,5} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
Q_h^{5,6} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{5,7} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
Q_h^{6,1} = Q_h^{6,2} = Q_h^{6,3} &= 0, \\
Q_h^{6,4} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{6,5} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
Q_h^{6,6} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{6,7} &= \begin{bmatrix} 0 & -2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
Q_h^{7,1} = Q_h^{7,2} = Q_h^{7,3} &= 0, \\
Q_h^{7,4} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{7,5} &= \begin{bmatrix} 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
Q_h^{7,6} &= \begin{bmatrix} 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{7,7} &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{8.16}
\end{aligned}$$

In addition, each open joint has a constraint equation relating the second geometric derivatives of its Euler-Rodrigues parameters with respect to generalized coordinates ψ_j and ψ_k :

$$\begin{aligned}
2\phi_h^4\phi_{h,j,k}''^4 + 2\phi_h^5\phi_{h,j,k}''^5 + 2\phi_h^6\phi_{h,j,k}''^6 + 2\phi_h^7\phi_{h,j,k}''^7 \\
= -2\phi_{h,j}^4\phi_{h,k}^4 - 2\phi_{h,j}^5\phi_{h,k}^5 - 2\phi_{h,j}^6\phi_{h,k}^6 - 2\phi_{h,j}^7\phi_{h,k}^7. \tag{8.17}
\end{aligned}$$

8.2.9 Parallel-Axis Gear Joint

A parallel-axis gear joint has three joint variables and the forms of the three Q_h^g operator matrices are given in Eqs. (6.25). Differentiating these, we find

$$Q_h^{1,1} = \frac{R_h(R_h + R'_h + \phi_h^2)}{R'_h} \begin{bmatrix} 0 & 0 & 0 & \cos\phi_h^1 \\ 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^{1,2} = \frac{R_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & -\cos\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^{1,3} = 0,$$

$$Q_h^{2,1} = \frac{\operatorname{sgn}(F) [R_h + R'_h + \phi_h^2]^2}{R'_h \sqrt{[R_h + R'_h + \phi_h^2]^2 - [(R_h + R'_h) \cos\alpha_h]^2}} \begin{bmatrix} 0 & 0 & 0 & \cos\phi_h^1 \\ 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & -\sin\phi_h^1 \\ 0 & 0 & 0 & \cos\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^{2,2} = \frac{\operatorname{sgn}(F) \left\{ [R_h + R'_h + \phi_h^2]^3 - 2 [R_h + R'_h + \phi_h^2] [(R_h + R'_h) \cos\alpha_h]^2 \right\}}{\left\{ [R_h + R'_h + \phi_h^2]^2 - [(R_h + R'_h) \cos\alpha_h]^2 \right\}^{3/2}}$$

$$\times \begin{bmatrix} 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & -\cos\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^{2,3} = 0,$$

$$Q_h^{3,1} = \frac{(R_h + R'_h) \tan\beta_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & \cos\phi_h^1 \\ 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_h^{3,2} = \frac{\tan\beta_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & \sin\phi_h^1 \\ 0 & 0 & 0 & -\cos\phi_h^1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_h^{3,3} = 0. \quad (8.18)$$

8.2.10 Involute Rack-and-Pinion Joint

An involute rack-and-pinion joint has three joint variables and the forms of the three Q_h^g operator matrices are given in Eqs. (6.26). Differentiating these, we find

$$\begin{aligned}
 Q_h^{1,1} &= \frac{1}{R'_h} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,2} &= \frac{1}{R'_h} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,3} &= 0, \\
 Q_h^{2,1} &= \frac{\operatorname{sgn}(F)\tan\alpha_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^{2,2} &= \frac{\operatorname{sgn}(F)\tan\alpha_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{2,3} &= 0, \\
 Q_h^{3,1} &= \frac{\tan\beta_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{3,2} &= \frac{\tan\beta_h}{R'_h} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{3,3} &= 0.
 \end{aligned} \tag{8.19}$$

8.2.11 Straight-Tooth Bevel-Gear Joint

A straight-tooth bevel-gear joint has two joint variables and the forms of its two Q_h^g operator matrices are given in Eqs. (6.27). Differentiating these give

$$\begin{aligned}
 Q_h^{1,1} &= \frac{\tan\gamma_h}{\tan\gamma'_h} \begin{bmatrix} 0 & 0 & \sin\phi_h^1 \sin\theta & 0 \\ 0 & 0 & \cos\phi_h^1 \sin\theta & 0 \\ -\sin\phi_h^1 \sin\theta & -\cos\phi_h^1 \sin\theta & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^{1,2} &= \frac{\tan\gamma_h}{\tan\gamma'_h} \begin{bmatrix} 0 & \sin\theta & -\cos\phi_h^1 \cos\theta & 0 \\ -\sin\theta & 0 & \sin\phi_h^1 \cos\theta & 0 \\ \cos\phi_h^1 \cos\theta & -\sin\phi_h^1 \cos\theta & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^{2,1} &= \begin{bmatrix} 0 & 0 & -\cos\phi_h^1 & 0 \\ 0 & 0 & \sin\phi_h^1 & 0 \\ \cos\phi_h^1 & -\sin\phi_h^1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
 Q_h^{2,2} &= 0.
 \end{aligned} \tag{8.20}$$

8.2.12 Point on a Planar-Curve Joint

A point on a planar curve joint has three joint variables and the forms of the three Q_h^g operator matrices are given in Eqs. (6.28). Differentiating these give

$$\begin{aligned}
 u'' &= \frac{\partial u'}{\partial \phi_h^1}, & v'' &= \frac{\partial v'}{\partial \phi_h^1}, & Q_h^{1,1} &= \begin{bmatrix} 0 & 0 & 0 & u'' \\ 0 & 0 & 0 & v'' \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,2} &= Q_h^{1,3} = 0, \\
 Q_h^{2,1} &= \begin{bmatrix} 0 & 0 & 0 & v' \\ 0 & 0 & 0 & -u' \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{2,2} &= Q_h^{2,3} = 0, \\
 Q_h^{3,1} &= Q_h^{3,2} = Q_h^{3,3} = 0.
 \end{aligned} \tag{8.21}$$

8.2.13 Line Tangent to a Planar-Curve Joint

A line tangent to a planar curve joint has three joint variables and the forms of the three Q_h^g operator matrices are given in Eqs. (6.29). Differentiating these give

$$\begin{aligned}
 u''' &= \frac{\partial u''}{\partial \phi_h^1}, & v''' &= \frac{\partial v''}{\partial \phi_h^1}, & \theta'' &= \frac{[u'v''' - u'''v'][(u')^2 + (v')^2] - 2[u'u'' - v'v'']}{[(u')^2 - (v')^2]^2}, \\
 Q_h^{1,1} &= \begin{bmatrix} 0 & -\theta'' & 0 & u'' + \theta'v'' + \theta''v' \\ \theta'' & 0 & 0 & v'' - \theta'u'' - \theta''u' \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{1,2} &= Q_h^{1,3} = 0, \\
 Q_h^{2,1} &= \begin{bmatrix} 0 & 0 & 0 & \theta' \sin \theta \\ 0 & 0 & 0 & -\theta' \cos \theta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & Q_h^{2,2} &= Q_h^{2,3} = 0, \\
 Q_h^{3,1} &= Q_h^{3,2} = Q_h^{3,3} = 0.
 \end{aligned} \tag{8.22}$$

8.3 Derivatives of the D_h Operator Matrices

Using chain-rule differentiation for finding accelerations, we seek the derivative of the D_h operator matrix, defined in Eq. (6.14),

$$D_h = (T_{0h-}S_{h-h})Q_h(T_{0h-}S_{h-h})^{-1},$$

where we recall that $h-$ is the label of the body that precedes joint h according to its defined orientation. Suppose that we seek the derivative of this D_h operator matrix with respect to one of the joint variables, ϕ_g .

We start by finding the geometric derivative of the inverse transformation matrix T_{0h-}^{-1} . We differentiate the identity

$$T_{0h-}^{-1} T_{0h-} = I$$

as follows:

$$\frac{\partial T_{0h-}^{-1}}{\partial \phi_g} T_{0h-} + T_{0h-}^{-1} \frac{\partial T_{0h-}}{\partial \phi_g} = 0,$$

which we rearrange to read

$$\frac{\partial T_{0h-}^{-1}}{\partial \phi_g} = -T_{0h-}^{-1} \frac{\partial T_{0h-}}{\partial \phi_g} T_{0h-}^{-1}.$$

Eq. (6.16) is now used to reduce this to the form

$$\frac{\partial T_{0h-}^{-1}}{\partial \phi_g} = -T_{0h-}^{-1} P(h-, g) D_g, \quad \begin{array}{l} h- = 1, 2, \dots, \ell, \\ g = 1, 2, \dots, n. \end{array} \quad (8.23)$$

Next, we find the derivative of the D_h operator matrix with respect to the joint variable ϕ_g :

$$\begin{aligned} \frac{\partial D_h}{\partial \phi_g} &= \frac{\partial (T_{0h-} S_{h-,h})}{\partial \phi_g} Q_h (T_{0h-} S_{h-,h})^{-1} + (T_{0h-} S_{h-,h}) \frac{\partial Q_h}{\partial \phi_g} (T_{0h-} S_{h-,h})^{-1} \\ &\quad + (T_{0h-} S_{h-,h}) Q_h \frac{\partial (T_{0h-} S_{h-,h})^{-1}}{\partial \phi_g}, \end{aligned}$$

which, with the aid of Eqs. (6.16) and (8.23), becomes

$$\begin{aligned} \frac{\partial D_h}{\partial \phi_g} &= P(h-, g) D_g (T_{0h-} S_{h-,h}) Q_h (T_{0h-} S_{h-,h})^{-1} + (T_{0h-} S_{h-,h}) Q'_h \delta_{h,g} (T_{0h-} S_{h-,h})^{-1} \\ &\quad - (T_{0h-} S_{h-,h}) Q_h (T_{0h-} S_{h-,h})^{-1} P(h-, g) D_g, \end{aligned}$$

where $\delta_{h,g}$ is the Kronecker delta, signifying that the second term does not appear unless ϕ_h and ϕ_g refer to variables of the same joint; that is, unless $h = g$.

Comparing the first and third terms of this equation with Eq. (6.14), however, and recognizing that $P(h-, g)$ and $\delta_{h,g}$ are scalars, the previous equation simplifies to the form

$$\begin{aligned} \frac{\partial D_h}{\partial \phi_g} &= P(h-, g) [D_g D_h - D_h D_g] \\ &\quad + \delta_{h,g} [(T_{0h-} S_{h-,h}) Q'_h (T_{0h-} S_{h-,h})^{-1}], \quad g, h = 1, 2, \dots, n. \end{aligned} \quad (8.24)$$

To differentiate the D_h operator matrix with respect to the generalized coordinate ψ_j , we write

$$\frac{\partial D_h}{\partial \psi_j} = \sum_{g=1}^n \frac{\partial D_h}{\partial \phi_g} \phi'_{g,j}.$$

Here, again, in the case of a multi-variable joint h , multiple terms will be required. If joint h is a spheric joint, for example, then

$$\frac{\partial D_h}{\partial \phi_g} \phi'_{g,j} = \frac{\partial D_h}{\partial \phi_g^1} \phi'^1_{g,j} + \frac{\partial D_h}{\partial \phi_g^2} \phi'^2_{g,j} + \frac{\partial D_h}{\partial \phi_g^3} \phi'^3_{g,j} + \frac{\partial D_h}{\partial \phi_g^4} \phi'^4_{g,j}.$$

By Eq. (8.24), the equation that follows it becomes

$$\frac{\partial D_h}{\partial \psi_j} = \sum_{g=1}^n [P(h-, g)(D_g D_h - D_h D_g) \phi'_{g,j}] + [(T_{0h-} S_{h-,h}) Q'_h \phi'_{h,j} (T_{0h-} S_{h-,h})^{-1}],$$

and, by Eq. (7.20), this further simplifies to

$$\begin{aligned} \frac{\partial D_h}{\partial \psi_j} &= W_{h-,j} D_h - D_h W_{h-,j} \\ &+ (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,j} (T_{0h-} S_{h-,h})^{-1}, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ j = 1, 2, \dots, f. \end{array} \end{aligned} \quad (8.25)$$

To differentiate the D_h operator matrix with respect to time, we write

$$\frac{dD_h}{dt} = \sum_{j=1}^f \frac{\partial D_h}{\partial \psi_j} \dot{\psi}_j,$$

which, by Eq. (8.25), becomes

$$\frac{dD_h}{dt} = \sum_{j=1}^f [W_{h-,j} D_h - D_h W_{h-,j} + (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,j} (T_{0h-} S_{h-,h})^{-1}] \dot{\psi}_j,$$

and, by Eqs. (7.24) and (7.15), this reduces to

$$\begin{aligned} \frac{dD_h}{dt} &= \omega_{h-} D_h - D_h \omega_{h-} \\ &+ (T_{0h-} S_{h-,h}) Q'_h \dot{\phi}_h (T_{0h-} S_{h-,h})^{-1}, \quad h = 1, 2, \dots, n. \end{aligned} \quad (8.26)$$

8.4 Second Geometric Derivatives of Joint Variables

Remembering the hierarchy of dependency explained in section 7.2, let us now find the second derivatives of the joint variables with respect to the generalized coordinates on which they explicitly depend. We call these *second geometric derivatives* and, as in section 7.3, we use a prime notation with subscripts as the symbolism:

$$\phi''_{h,j,k} = \frac{\partial}{\partial \psi_k} \left(\frac{\partial \phi_h}{\partial \psi_j} \right) = \frac{\partial \phi'_{h,j}}{\partial \psi_k}, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.27)$$

Of course, for a joint that has more than a single joint variable, superscripts may also be required.

How will we find these derivatives? Well, in section 7.3 we differentiated the loop-closure equations with respect to generalized coordinate ψ_j and found that

$$\sum_{h=1}^n L(i, h) D_h \phi'_{h,j} = 0, \quad \begin{array}{l} i = 1, 2, \dots, NL, \\ j = 1, 2, \dots, f, \end{array}$$

is a set of mandatory constraint equations among the first geometric derivatives that ensures that their values change compatibly with the geometric requirements of loop closure. However, a similar requirement also holds among the values of the second geometric derivatives. Therefore, we differentiate again with respect to generalized coordinate ψ_k :

$$\sum_{h=1}^n L(i, h) \left\{ D_h \phi''_{h,j,k} + \frac{\partial D_h}{\partial \psi_k} \phi'_{h,j} \right\} = 0, \quad i = 1, 2, \dots, NL, \\ j, k = 1, 2, \dots, f.$$

Using Eq. (8.25) to take the indicated derivative gives

$$\sum_{h=1}^n L(i, h) \{ D_h \phi''_{h,j,k} + [W_{h-,k} D_h - D_h W_{h-,k} \\ + (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,k} (T_{0h-} S_{h-,h})^{-1}] \phi'_{h,j} \} = 0,$$

which we rearrange to read

$$\sum_{h=1}^n L(i, h) D_h \phi''_{h,j,k} = - \sum_{h=1}^n L(i, h) [W_{h-,k} D_h - D_h W_{h-,k} \\ + (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,k} (T_{0h-} S_{h-,h})^{-1}] \phi'_{h,j}. \quad (8.28)$$

We now define a new (4×4) matrix $C''_{i,j,k}$ for each loop as follows:

$$C''_{i,j,k} = - \sum_{h=1}^n L(i, h) [W_{h-,k} D_h - D_h W_{h-,k} \\ + (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,k} (T_{0h-} S_{h-,h})^{-1}] \phi'_{h,j}, \quad i = 1, 2, \dots, NL, \\ j, k = 1, 2, \dots, f, \quad (8.29)$$

so that Eq. (8.28) reduces to

$$\sum_{h=1}^n L(i, h) D_h \phi''_{h,j,k} = C''_{i,j,k}, \quad i = 1, 2, \dots, NL, \\ j, k = 1, 2, \dots, f. \quad (8.30)$$

We will see the individual terms of the sum included in $C''_{i,j,k}$ again. They will appear in section 8.6 where we seek second geometric derivatives of body postures. We will see that Eq. (8.28) is the specific case that states that the second geometric derivative of the posture of the stationary body is zero. Compare this with the definition of the $A_{i,j,k}$ operator matrices in Eq. (8.47). Some economy can be gained by calculating, summing, and storing the terms of $A_{i,j,k}$ in the order in which joints are met as the loop is traced along the paths to each successive body. Once each loop is completely traced, the $C''_{i,j,k}$ matrix results.

It is left as an exercise to verify that the upper-left (3×3) submatrix of $C''_{i,j,k}$ in Eq. (8.29) is skew-symmetric as were the Q_h , D_h , and W_{h-j} operator matrices. We can also see that this must be true for compatibility with the left-hand side of Eq. (8.30). Once this is recognized, we eliminate redundancy in Eq. (8.30) by putting it into (6×1) screw coordinate form

$$\sum_{h=1}^n L(i, h) \hat{D}_h \phi''_{h,j,k} = \hat{C}''_{i,j,k}, \quad i = 1, 2, \dots, NL, \\ j, k = 1, 2, \dots, f, \quad (8.31)$$

where \hat{D}_h is as defined in Eq. (6.41) and

$$\hat{C}_{i,j,k}'' = \begin{bmatrix} C''_{i,j,k}(1,4) \\ C''_{i,j,k}(2,4) \\ C''_{i,j,k}(3,4) \\ C''_{i,j,k}(3,2) \\ C''_{i,j,k}(1,3) \\ C''_{i,j,k}(2,1) \end{bmatrix}, \quad \begin{matrix} i = 1, 2, \dots, NL, \\ j, k = 1, 2, \dots, f. \end{matrix} \quad (8.32)$$

In addition to these equations, of course, we also have a constraint equation of the form

$$\begin{aligned} & 2\phi_h^1 \phi_{h,j,k}''^1 + 2\phi_h^2 \phi_{h,j,k}''^2 + 2\phi_h^3 \phi_{h,j,k}''^3 + 2\phi_h^4 \phi_{h,j,k}''^4 \\ & = -2\phi_{h,j}^1 \phi_{h,k}^1 - 2\phi_{h,j}^2 \phi_{h,k}^2 - 2\phi_{h,j}^3 \phi_{h,k}^3 - 2\phi_{h,j}^4 \phi_{h,k}^4, \quad j, k = 1, 2, \dots, f, \end{aligned}$$

for each spheric joint ϕ_h in the system, and another

$$\begin{aligned} & 2\phi_h^4 \phi_{h,j,k}''^4 + 2\phi_h^5 \phi_{h,j,k}''^5 + 2\phi_h^6 \phi_{h,j,k}''^6 + 2\phi_h^7 \phi_{h,j,k}''^7 \\ & = -2\phi_{h,j}^4 \phi_{h,k}^4 - 2\phi_{h,j}^5 \phi_{h,k}^5 - 2\phi_{h,j}^6 \phi_{h,k}^6 - 2\phi_{h,j}^7 \phi_{h,k}^7, \quad j, k = 1, 2, \dots, f, \end{aligned}$$

for each open joint ϕ_h in the system, a total of NC constraint equations, which come from taking derivatives of Eqs. (7.10) and (7.11) with respect to generalized coordinate ψ_k .

Because, for any particular choice of j and k , Eq. (8.31) is now of the same form as Eq. (6.42), we can write, from Eq. (6.45), that

$$\mathcal{J} \phi_{j,k}'' = \hat{C}_{j,k}'', \quad j, k = 1, 2, \dots, f, \quad (8.33)$$

where \mathcal{J} is the same $[(6NL + NC) \times n]$ Jacobian matrix that resulted from the numeric iteration process of section 6.5, and $\phi_{j,k}''$ for a given choice of j and k is an $(n \times 1)$ vector of second geometric derivatives that are yet to be determined. The $[(6NL + NC) \times 1]$ column $\hat{C}_{j,k}''$ is composed of the NL (6×1) vectors from the right-hand sides of Eq. (8.31) for each loop, augmented by the NC right-hand sides of Eq. (8.14) for each spheric joint and Eq. (8.17) for each open joint.

Reviewing the numeric iteration process of section 6.5, we recall that the modified pivoting used in the Gauss-Jordan process reordered the joint variables so that all dependent joint variables are numbered first, followed next by the NF joint variables of the FGCs, and finally by the NS joint variables of the SGCs. Recognizing that both the FGC and the SGC joint variables have been accepted as generalized

coordinates ψ , we see that, by definition, they are independent of each other. Therefore, for a particular choice of j and k , the $\phi''_{j,k}$ matrix must be of the form

$$\phi''_{j,k} = \begin{bmatrix} \phi''_{1,j,k} \\ \phi''_{2,j,k} \\ \vdots \\ \phi''_{n-f,j,k} \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \phi''_{dep,j,k} \\ \hline 0 \\ 0 \end{bmatrix}, \quad j, k = 1, 2, \dots, f, \quad (8.34)$$

where $\phi''_{dep,j,k}$ is an $[(n-f) \times 1]$ submatrix of the dependent second geometric derivative values to be found for each combination of j and k .

Once we recognize this form for $\phi''_{dep,j,k}$, we return to Eq. (8.33) and subdivide it into compatible-size submatrices:

$$\begin{bmatrix} \mathcal{J}_{dep} & | & \mathcal{J}_{ind} \end{bmatrix} \begin{bmatrix} \phi''_{dep,j,k} \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \hat{C}''_{j,k} \end{bmatrix}.$$

However, we see that this is the same subdivision that resulted from the modified Gauss-Jordan elimination process. Reverting to the notation of Eq. (6.49), we have

$$\begin{bmatrix} \mathcal{J}_{11} & | & \mathcal{J}_{12} & \mathcal{J}_{13} \\ \mathcal{J}_{21} & | & \mathcal{J}_{22} & \mathcal{J}_{23} \end{bmatrix} \begin{bmatrix} \phi''_{dep,j,k} \\ \hline 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (\hat{C}''_{j,k})_1 \\ \hline (\hat{C}''_{j,k})_2 \end{bmatrix},$$

and after the Gauss-Jordan process is complete, these equations are reduced to the form of Eq. (6.50):

$$\begin{bmatrix} I & | & \mathcal{J}_{11}^{-1} \mathcal{J}_{12} & \mathcal{J}_{11}^{-1} \mathcal{J}_{13} \\ \hline 0 & | & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi''_{dep,j,k} \\ \hline 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{J}_{11}^{-1} & | & 0 \\ \hline -\mathcal{J}_{21} \mathcal{J}_{11}^{-1} & | & I \end{bmatrix} \begin{bmatrix} (\hat{C}''_{j,k})_1 \\ \hline (\hat{C}''_{j,k})_2 \end{bmatrix},$$

which has for a solution

$$\phi''_{dep,j,k} = \mathcal{J}_{11}^{-1} (\hat{C}''_{j,k})_1, \quad j, k = 1, 2, \dots, f, \quad (8.35)$$

with the additional condition that

$$(\hat{C}''_{j,k})_2 - \mathcal{J}_{21} \mathcal{J}_{11}^{-1} (\hat{C}''_{j,k})_1 = 0, \quad j, k = 1, 2, \dots, f. \quad (8.36)$$

Thus, the calculations for finding the set of all second geometric derivatives of the joint variables can proceed in a set of nested loops, each incrementing counters j and k from one to f . Of course, it reduces the computational load by half to notice from Eq. (8.27) that

$$\phi''_{h,k,j} = \phi''_{h,j,k}, \quad \begin{array}{l} h = 1, 2, \dots, n, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.37)$$

For each choice of j and k , the corresponding $C''_{i,j,k}$ matrix is formed by Eq. (8.29) for all loops i and, from them, the $\hat{C}''_{i,j,k}$ vectors are formed by Eq. (8.32). These

are augmented by the NC right-hand sides of Eq. (8.14) for each spheric joint and Eq. (8.17) for each open joint. Finally, Eq. (8.35) is used to provide values of $\phi''_{dep,j,k}$ for Eq. (8.34).

For each combination of j and k , it is also wise to check that Eq. (8.36) is satisfied within a satisfactory numeric tolerance.¹ However, from the authors' forty-plus years of experience, when a problem having physically realizable input data is properly formulated, this test has never been found to fail.

The second geometric derivatives just found are not actual accelerations, even though they are sometimes referred to as such. They are second rates of change of dependent joint variables with respect to changes of generalized coordinates. Time is not included in them and they typically have units such as radians per radian squared. They represent totally geometric information and are functions of the generalized coordinates ψ alone. They are very important, however, and arise frequently in our coming work.

EXAMPLE 8.1 To illustrate the previous methods, let us continue the analysis of the Cardan/Hooke universal joint studied in previous examples. In example 5.1, we found solutions for the positions of all joint variables and the various transformation matrices. In example 7.1, we found the D_h derivative operator matrices, the Jacobian, and the first geometric derivatives for each of the joints. These are

$$D_A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_B = \begin{bmatrix} 0 & \frac{\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ \frac{\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_C = \begin{bmatrix} 0 & \cos \beta \sin \psi & \cos \psi & -h \cos \beta \sin \psi \\ -\cos \beta \sin \psi & 0 & -\sin \beta \sin \psi & 0 \\ -\cos \psi & \sin \beta \sin \psi & 0 & -h \sin \beta \sin \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D = \begin{bmatrix} 0 & -\sin \beta & 0 & h \sin \beta \\ \sin \beta & 0 & -\cos \beta & 0 \\ 0 & \cos \beta & 0 & -h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

¹ The IMP software system uses a default tolerance of 10^{-6} for this test.

$$\mathcal{J} = \begin{bmatrix} 0 & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -h \cos \beta \sin \psi & | & h \sin \beta \\ 0 & 0 & 0 & | & 0 \\ -h & 0 & -h \sin \beta \sin \psi & | & -h \cos \beta \\ \hline 1 & 0 & \sin \beta \sin \psi & | & \cos \beta \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi & | & 0 \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi & | & \sin \beta \end{bmatrix},$$

$$\begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \\ \phi'_{D,1} \end{bmatrix} = \begin{bmatrix} \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{\sin \beta \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{\sin \beta \cos \beta \sin \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ 1 \end{bmatrix}.$$

In example 7.3, we found the $W_{b,1}$ derivative operator matrices

$$W_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ 0 & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{2,1} = \begin{bmatrix} 0 & \frac{\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-h \sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{-\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{3,1} = \begin{bmatrix} 0 & \sin \beta & 0 & -h \sin \beta \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & -\cos \beta & 0 & h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{4,1} = 0.$$

Recognizing that, for this example, all joints are revolute and, therefore, that all Q'_h matrices are zero, we formulate the following matrices:

$$(W_{4,1} D_A - D_A W_{4,1}) \phi'_{A,1} = 0,$$

$$(W_{1,1}D_B - D_B W_{1,1})\phi'_{B,1} = \frac{\sin \beta \cos \beta \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)^2} \begin{bmatrix} 0 & -\cos \beta \sin \psi & -\cos \psi & -h \cos \beta \sin \psi \\ \cos \beta \sin \psi & 0 & 0 & 0 \\ \cos \psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$(W_{2,1}D_C - D_C W_{2,1})\phi'_{C,1} = \frac{\sin \beta \cos \beta \sin \psi}{(1 - \sin^2 \beta \sin^2 \psi)} \begin{bmatrix} 0 & \cos \beta \cos \psi & -\sin \psi & -h \cos \beta \cos \psi \\ -\cos \beta \cos \psi & 0 & -\sin \beta \cos \psi & 0 \\ \sin \psi & \sin \beta \cos \psi & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$(W_{3,1}D_D - D_D W_{3,1})\phi'_{D,1} = 0.$$

From these, we now form the $C''_{1,1,1}$ matrix of Eq. (8.29). Noting that there is only one loop and one degree of freedom, the result is

$$C''_{1,1,1} = \frac{\sin \beta \cos \beta}{(1 - \sin^2 \beta \sin^2 \psi)} \times \begin{bmatrix} 0 & \frac{\sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)} & 1 + \sin^2 \beta \sin^2 \psi & \frac{-h \sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)} \\ \frac{-\sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)} & 0 & \sin \beta \sin \psi \cos \psi & 0 \\ -(1 + \sin^2 \beta \sin^2 \psi) & -\sin \beta \sin \psi \cos \psi & 0 & h \sin \beta \sin \psi \cos \psi \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Next, we formulate the terms of Eq. (8.33):

$$\begin{bmatrix} 0 & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -h \cos \beta \sin \psi & h \sin \beta \\ 0 & 0 & 0 & 0 \\ -h & 0 & -h \sin \beta \sin \psi & -h \cos \beta \\ \hline 1 & 0 & \sin \beta \sin \psi & \cos \beta \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi & 0 \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi & \sin \beta \end{bmatrix} \begin{bmatrix} \phi''_{A,1,1} \\ \phi''_{B,1,1} \\ \phi''_{C,1,1} \\ \phi''_{D,1,1} \end{bmatrix}$$

$$= \frac{\sin \beta \cos \beta}{(1 - \sin^2 \beta \sin^2 \psi)} \begin{bmatrix} \frac{-h \sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)} \\ 0 \\ h \sin \beta \sin \psi \cos \psi \\ \hline -\sin \beta \sin \psi \cos \psi \\ 1 + \sin^2 \beta \sin^2 \psi \\ \frac{-\sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)} \end{bmatrix}.$$

As in example 7.1, we recognize that the second row of the right-hand column is zero and the first and third rows are h multiples of the sixth and negative fourth rows, respectively, and that because $\psi = \psi_1 = \phi_D$, we have $\phi''_{D,1,1} = 0$. Therefore, these equations can be reduced to the form

$$\begin{bmatrix} 1 & 0 & \sin \beta \sin \psi \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi \end{bmatrix} \begin{bmatrix} \phi''_{A,1,1} \\ \phi''_{B,1,1} \\ \phi''_{C,1,1} \end{bmatrix} = \frac{\sin \beta \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \begin{bmatrix} -\sin \beta \sin \psi \cos \psi \\ 1 + \sin^2 \beta \sin^2 \psi \\ \frac{-\sin^2 \beta \cos \beta \sin^3 \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} \end{bmatrix}.$$

The resulting solution for the second geometric derivatives is

$$\begin{bmatrix} \phi''_{A,1,1} \\ \phi''_{B,1,1} \\ \phi''_{C,1,1} \\ \phi''_{D,1,1} \end{bmatrix} = \frac{\sin \beta \cos \beta}{(1 - \sin^2 \beta \sin^2 \psi)^2} \begin{bmatrix} -2 \sin \beta \sin \psi \cos \psi \\ -\sqrt{1 - \sin^2 \beta \sin^2 \psi} \cos \beta \sin \psi \\ (1 + \sin^2 \beta \sin^2 \psi) \cos \psi \\ \hline 0 \end{bmatrix}.$$

From example 7.1 and Eq. (6.53), the quality index is

$$Q = |\det(\mathcal{J}_{11})| = \sqrt{1 - \sin^2 \beta \sin^2 \psi},$$

and its value remains in the range ($\cos \beta \leq Q \leq 1.00$). We see here that the denominators of the second geometric derivatives are the fourth power of the quality index. Thus, as before, there should not be difficulty in the operation of the device or in their numeric evaluation.

8.5 Accelerations of Joint Variables

The (relative) accelerations of the joint variables, as shown by Eq. (8.3) can be found in any of three ways. For hand calculations, the expressions found for the positions of the joint variables can be directly differentiated analytically on a case-by-case basis. This approach, however, is highly susceptible to human error and is not easily adapted to computer solution. The second approach is to differentiate the loop-closure equations, as was done in the last section to find second geometric derivatives. The third approach is to take advantage of the second geometric derivatives. Remembering the hierarchy of dependencies, we can differentiate Eq. (7.15) as follows:

$$\ddot{\phi}_h = \frac{d^2 \phi_h}{dt^2} = \sum_{j=1}^f \phi'_{h,j} \ddot{\psi}_j + \sum_{k=1}^f \sum_{j=1}^f \phi''_{h,j,k} \dot{\psi}_j \dot{\psi}_k, \quad h = 1, 2, \dots, n. \quad (8.38)$$

Here we see that all dependent joint accelerations vary linearly with the generalized accelerations and quadratically with the generalized velocities. They are nonlinear functions of ψ , however, through the geometric derivatives. The double

summation cannot be expressed in matrix form because $\phi''_{h,j,k}$ is a three-dimensional array.

As discussed earlier, we can differentiate the loop-closure equations with respect to time. From section 7.4 we have

$$\sum_{h=1}^n L(i, h) D_h \dot{\phi}_h = 0, \quad i = 1, 2, \dots, NL.$$

Differentiating this again with respect to time gives

$$\sum_{h=1}^n L(i, h) \left[D_h \ddot{\phi}_h + \frac{dD_h}{dt} \dot{\phi}_h \right] = 0, \quad i = 1, 2, \dots, NL.$$

Here again, in the case of a multi-variable joint h , multiple terms will be required. If joint h is a spheric joint, for example, then

$$D_h \dot{\phi}_h = D_h^1 \dot{\phi}_h^1 + D_h^2 \dot{\phi}_h^2 + D_h^3 \dot{\phi}_h^3 + D_h^4 \dot{\phi}_h^4.$$

Along with the use of Eq. (8.26), this can be rearranged to read

$$\sum_{h=1}^n L(i, h) D_h \ddot{\phi}_h = - \sum_{h=1}^n L(i, h) [\omega_{h-} D_h - D_h \omega_{h-} + (T_{0h-} S_{h-,h}) Q'_h \dot{\phi}_h (T_{0h-} S_{h-,h})^{-1}] \dot{\phi}_h, \quad i = 1, 2, \dots, NL, \quad (8.39)$$

with an additional constraint equation of the form,

$$2\phi_h^1 \ddot{\phi}_h^1 + 2\phi_h^2 \ddot{\phi}_h^2 + 2\phi_h^3 \ddot{\phi}_h^3 + 2\phi_h^4 \ddot{\phi}_h^4 = -2(\dot{\phi}_h^1)^2 - 2(\dot{\phi}_h^2)^2 - 2(\dot{\phi}_h^3)^2 - 2(\dot{\phi}_h^4)^2,$$

for each spheric joint in the system, and another of the form,

$$2\phi_h^4 \ddot{\phi}_h^4 + 2\phi_h^5 \ddot{\phi}_h^5 + 2\phi_h^6 \ddot{\phi}_h^6 + 2\phi_h^7 \ddot{\phi}_h^7 = -2(\dot{\phi}_h^4)^2 - 2(\dot{\phi}_h^5)^2 - 2(\dot{\phi}_h^6)^2 - 2(\dot{\phi}_h^7)^2,$$

for each open joint in the system; a total of NC constraint equations, which come from taking the derivatives of Eqs. (4.17) and (4.24) with respect to time.

Parallel to Eq. (8.29), we now define

$$\ddot{C}_i = - \sum_{h=1}^n L(i, h) [\omega_{h-} D_h - D_h \omega_{h-} + (T_{0h-} S_{h-,h}) Q'_h \dot{\phi}_h (T_{0h-} S_{h-,h})^{-1}] \dot{\phi}_h, \quad i = 1, 2, \dots, NL, \quad (8.40)$$

and the corresponding screw coordinate vectors

$$\hat{\ddot{C}}_i = \begin{bmatrix} \ddot{C}_i(1, 4) \\ \ddot{C}_i(2, 4) \\ \ddot{C}_i(3, 4) \\ \ddot{C}_i(3, 2) \\ \ddot{C}_i(1, 3) \\ \ddot{C}_i(2, 1) \end{bmatrix}, \quad i = 1, 2, \dots, NL, \quad (8.41)$$

so that Eq. (8.39) becomes

$$J \ddot{\phi} = \hat{\ddot{C}}. \quad (8.42)$$

After recognizing that $\ddot{\phi}$ has the form

$$\ddot{\phi} = \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \vdots \\ \ddot{\phi}_{n-f} \\ \ddot{\psi}_1 \\ \vdots \\ \ddot{\psi}_f \end{bmatrix} = \begin{bmatrix} \ddot{\phi}_{dep} \\ \ddot{\psi} \end{bmatrix}, \quad (8.43)$$

we see that Eq. (8.42) is of the form

$$\begin{bmatrix} J_{11} & | & J_{12} & J_{13} \\ J_{21} & | & J_{22} & J_{23} \end{bmatrix} \begin{bmatrix} \ddot{\phi}_{dep} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \end{bmatrix},$$

and can be solved in the same manner as for the second geometric derivatives. In its reduced form, we have

$$\begin{bmatrix} I & | & J_{11}^{-1}J_{12} & J_{11}^{-1}J_{13} \\ 0 & | & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\phi}_{dep} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} J_{11}^{-1} & | & 0 \\ -J_{21}J_{11}^{-1} & | & I \end{bmatrix} \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \end{bmatrix}.$$

Therefore, we see that the solution is

$$\ddot{\phi}_{dep} = J_{11}^{-1}(\hat{C}_1 - [J_{12} \ J_{13}]\ddot{\psi}), \quad (8.44)$$

with the additional condition that

$$\hat{C}_2 - J_{21}J_{11}^{-1}\hat{C}_1 = 0. \quad (8.45)$$

EXAMPLE 8.2 Let us continue the Cardan/Hooke universal joint of example 8.1. If the input shaft is driven at a rate of $\dot{\phi}_D = \dot{\psi}$, and if this rate is constant ($\ddot{\phi}_D = \ddot{\psi} = 0$), then let us find the accelerations of the other joint variables. These are given directly by Eq. (8.39) and the second geometric derivatives found in example 8.1:

$$\ddot{\phi} = \begin{bmatrix} \ddot{\phi}_A \\ \ddot{\phi}_B \\ \ddot{\phi}_C \\ \ddot{\phi}_D \end{bmatrix} = \frac{\sin\beta \cos\beta \dot{\psi}^2}{(1 - \sin^2\beta \sin^2\psi)^2} \begin{bmatrix} -2\sin\beta \sin\psi \cos\psi \\ -\sqrt{1 - \sin^2\beta \sin^2\psi} \cos\beta \sin\psi \\ (1 - \sin^2\beta \sin^2\psi) \cos\psi \\ 0 \end{bmatrix}.$$

8.6 Second Geometric Derivatives of Body Postures

Before we find the acceleration of a body or point, let us first find the geometric derivative of the $W_{b,j}$ operator matrices. From Eq. (7.20), we have

$$W_{b,j} = \sum_{h=1}^n P(b,h) D_h \phi'_{h,j}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f. \end{array}$$

Differentiating this with respect to generalized coordinate ψ_k gives the new, second geometric derivative operator matrix $A_{i,j,k}$, defined by

$$A_{b,j,k} = \frac{\partial W_{b,j}}{\partial \psi_k}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j, k = 1, 2, \dots, f, \end{array} \quad (8.46)$$

and, using Eq. (8.25), we find

$$A_{b,j,k} = \sum_{h=1}^n P(b, h) \{ D_h \phi''_{h,j,k} + [W_{h-,k} D_h - D_h W_{h-,k} + (T_{0h-} S_{h-,h}) Q'_h \phi'_{h,k} (T_{0h-} S_{h-,h})^{-1}] \phi'_{h,j} \}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.47)$$

Here again, in the case of a multi-variable joint h , multiple terms will be required. If joint h is a spheric joint, for example, then

$$D_h \phi'_{h,j} = D_h^1 \phi_{h,j}^1 + D_h^2 \phi_{h,j}^2 + D_h^3 \phi_{h,j}^3 + D_h^4 \phi_{h,j}^4.$$

Now, to find the second geometric derivative of the posture of a body, we start with Eq. (7.21)

$$\frac{\partial T_{0b}}{\partial \psi_j} = W_{b,j} T_{0b}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f, \end{array}$$

and differentiate again, with respect to the generalized coordinate ψ_k . Using Eqs. (8.46) and (7.21), we find

$$T''_{0b,j,k} = \frac{\partial}{\partial \psi_k} \left(\frac{\partial T_{0b}}{\partial \psi_j} \right) = (A_{b,j,k} + W_{b,j} W_{b,k}) T_{0b}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.48)$$

We note here that, in general, $A_{b,k,j} \neq A_{b,j,k}$ in spite of the fact that the order of the subscripts implies an order of taking derivatives. Similarly, in general,

$$\frac{\partial W_{b,k}}{\partial \psi_j} \neq \frac{\partial W_{b,j}}{\partial \psi_k}.$$

However, it is true for the general case that

$$A_{b,k,j} + W_{b,k} W_{b,j} = A_{b,j,k} + W_{b,j} W_{b,k}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.49)$$

EXAMPLE 8.3 Let us now continue the analysis of the Cardan/Hooke universal joint of example 8.1 and example 8.2, and find the second geometric derivative operator matrices for each of its bodies. From the previous examples we have

$$D_A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_B = \begin{bmatrix} 0 & \frac{\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ \frac{\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_C = \begin{bmatrix} 0 & \cos \beta \sin \psi & \cos \psi & -h \cos \beta \sin \psi \\ -\cos \beta \sin \psi & 0 & -\sin \beta \sin \psi & 0 \\ -\cos \psi & \sin \beta \sin \psi & 0 & -h \sin \beta \sin \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D = \begin{bmatrix} 0 & -\sin \beta & 0 & h \sin \beta \\ \sin \beta & 0 & -\cos \beta & 0 \\ 0 & \cos \beta & 0 & -h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \phi'_{A,1} \\ \phi'_{B,1} \\ \phi'_{C,1} \\ \phi'_{D,1} \end{bmatrix} = \begin{bmatrix} \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{\sin \beta \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} \\ \frac{\sin \beta \cos \beta \sin \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{1}{1 - \sin^2 \beta \sin^2 \psi} \end{bmatrix},$$

$$W_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ 0 & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{2,1} = \begin{bmatrix} 0 & \frac{\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-h \sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{-\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{3,1} = \begin{bmatrix} 0 & \sin \beta & 0 & -h \sin \beta \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & -\cos \beta & 0 & h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$W_{4,1} = 0,$$

$$Q'_A = Q'_B = Q'_C = Q'_D = 0,$$

$$\begin{bmatrix} \phi''_{A,1,1} \\ \phi''_{B,1,1} \\ \phi''_{C,1,1} \\ \phi''_{D,1,1} \end{bmatrix} = \frac{\sin \beta \cos \beta}{(1 - \sin^2 \beta \sin^2 \psi)} \begin{bmatrix} -2 \sin \beta \sin \psi \cos \psi \\ -\cos \beta \sin \psi \sqrt{1 - \sin^2 \beta \sin^2 \psi} \\ (1 + \sin^2 \beta \sin^2 \psi) \cos \psi \\ 0 \end{bmatrix}.$$

Next, we calculate

$$[W_{4,1}D_A - D_A W_{4,1} + (T_{04}S_{4A})Q'_A \phi'_{A,1} (T_{04}S_{4A})^{-1}] \phi'_{A,1} = 0,$$

$$\begin{aligned} A_{1,1,1} &= D_A \phi''_{A,1,1} + [W_{4,1}D_A - D_A W_{4,1} + (T_{04}S_{4A})Q'_A \phi'_{A,1} (T_{04}S_{4A})^{-1}] \phi'_{A,1} \\ &= \frac{\sin^2 \beta \cos \beta \cos 2\psi}{(1 - \sin^2 \beta \sin^2 \psi)^2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -h \\ 0 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$[W_{1,1}D_B - D_B W_{1,1} + (T_{01}S_{1B})Q'_B \phi'_{B,1} (T_{01}S_{1B})^{-1}] \phi'_{B,1}$$

$$= \frac{\sin \beta \cos \beta \cos \psi}{(1 - \sin^2 \beta \sin^2 \psi)^2} \begin{bmatrix} 0 & -\cos \beta \sin \psi & -\cos \psi & h \cos \beta \sin \psi \\ \cos \beta \sin \psi & 0 & 0 & 0 \\ \cos \psi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_{2,1,1} = A_{1,1,1} + D_B \phi''_{B,1,1} + [W_{1,1}D_B - D_B W_{1,1} + (T_{01}S_{1B})Q'_B \phi'_{B,1} (T_{01}S_{1B})^{-1}] \phi'_{B,1}$$

$$= \frac{\sin \beta \cos \beta}{(1 - \sin^2 \beta \sin^2 \psi)^2} \times \begin{bmatrix} 0 & -\cos \beta \sin 2\psi & -(\cos 2\psi + \sin^2 \beta \sin^2 \psi) & h \cos \beta \sin 2\psi \\ \cos \beta \sin 2\psi & 0 & 0 & 0 \\ \cos 2\psi + \sin^2 \beta \sin^2 \psi & 0 & 0 & h \sin \beta \sin 2\psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$[W_{2,1}D_C - D_C W_{2,1} + (T_{02}S_{2C})Q'_C \phi'_{C,1} (T_{02}S_{2C})^{-1}] \phi'_{C,1}$$

$$= \frac{\sin \beta \cos \beta \sin \psi}{(1 - \sin^2 \beta \sin^2 \psi)} \begin{bmatrix} 0 & \cos \beta \cos \psi & -\sin \psi & h \cos \beta \cos \psi \\ -\cos \beta \cos \psi & 0 & -\sin \beta \cos \psi & 0 \\ \sin \psi & \sin \beta \cos \psi & 0 & -h \sin \beta \cos \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_{3,1,1} = A_{2,1,1} + D_C \phi''_{C,1,1} + [W_{2,1}D_C - D_C W_{2,1} + (T_{02}S_{2C})Q'_C \phi'_{C,1} (T_{02}S_{2C})^{-1}] \phi'_{C,1} = 0,$$

$$[W_{3,1}D_D - D_D W_{3,1} + (T_{03}S_{3D})Q'_D \phi'_{D,1} (T_{03}S_{3D})^{-1}] \phi'_{D,1} = 0,$$

$$A_{4,1,1} = A_{3,1,1} + D_D \phi''_{D,1,1} + [W_{3,1}D_D - D_D W_{3,1} + (T_{03}S_{3D})Q'_D \phi'_{D,1} (T_{03}S_{3D})^{-1}] \phi'_{D,1} = 0.$$

Note the order of the calculations performed in this example; here we see how the calculation of the portions of C'' in example 8.1 is done in parallel with building the $A_{b,j,k}$ matrices. It may be instructive to compare the form of Eq. (8.28) to that of Eq. (8.47). Because, for this problem, each row of the loop matrix L is a possible path through the system back to the frame, body 4, it should be expected that $A_{4,1,1} = 0$.

8.7 Second Geometric Derivatives of Point Positions

To find the second geometric derivative of the position of a point, we start with Eq. (7.25)

$$R'_{b,j} = \frac{\partial R_b}{\partial \psi_j} = W_{b,j} T_{0b} r_b = W_{b,j} R_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f, \end{array}$$

and differentiate again, with respect to the generalized coordinate ψ_k . Using Eqs. (8.46), (7.20), and (7.25) we find

$$R''_{b,j,k} = \frac{\partial}{\partial \psi_k} \left(\frac{\partial R_b}{\partial \psi_j} \right) = (A_{b,j,k} + W_{b,j} W_{b,k}) R_b, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ j, k = 1, 2, \dots, f. \end{array} \quad (8.50)$$

8.8 Accelerations of Bodies

In Eq. (7.22) we defined the velocity operator for a body as

$$\omega_b = \sum_{h=1}^n P(b, h) D_h \dot{\phi}_h, \quad b = 1, 2, \dots, \ell.$$

Before finding the acceleration of a body, let us first define the new acceleration operator α_b

$$\alpha_b = \frac{d\omega_b}{dt}, \quad b = 1, 2, \dots, \ell, \quad (8.51)$$

which can be found as the time derivative of the previous equation,

$$\alpha_b = \sum_{h=1}^n P(b, h) \left\{ D_h \ddot{\phi}_h + \frac{dD_h}{dt} \dot{\phi}_h \right\}, \quad b = 1, 2, \dots, \ell.$$

Again, in the case of a multi-variable joint h , multiple terms will be required. If joint h is a spheric joint, for example, then

$$D_h \dot{\phi}_h = D_h^1 \dot{\phi}_h^1 + D_h^2 \dot{\phi}_h^2 + D_h^3 \dot{\phi}_h^3 + D_h^4 \dot{\phi}_h^4.$$

By Eq. (8.26) the previous equation now expands to

$$\alpha_b = \sum_{h=1}^n P(b, h) \{ D_h \ddot{\phi}_h + [\omega_{h-} D_h - D_h \omega_{h-} + (T_{0h-} S_{h-,h}) Q'_h \dot{\phi}_h (T_{0h-} S_{h-,h})^{-1}] \dot{\phi}_h \},$$

$$b = 1, 2, \dots, \ell, \quad (8.52)$$

for evaluation.

As another alternative, we might start with Eq. (7.24)

$$\omega_b = \sum_{j=1}^f W_{b,j} \dot{\psi}_j, \quad b = 1, 2, \dots, \ell,$$

and use the chain rule to differentiate with respect to time

$$\alpha_b = \sum_{j=1}^f W_{b,j} \ddot{\psi}_j + \sum_{j=1}^f \sum_{k=1}^f A_{b,j,k} \dot{\psi}_j \dot{\psi}_k, \quad b = 1, 2, \dots, \ell. \quad (8.53)$$

This is an entirely separate but equivalent formula for α_b to the previous one. Whether Eq. (8.52) or Eq. (8.53) is preferable depends on the particular situation; we will have future needs for both.

Next, we recall that in Eq. (8.2) we defined the acceleration of a body to be the second derivative with respect to time of its transformation matrix with respect to the global frame:

$$\ddot{T}_{0b} = \frac{d^2 T_{0b}}{dt^2}, \quad b = 1, 2, \dots, \ell.$$

From Eq. (7.23) we have

$$\dot{T}_{0b} = \omega_b T_{0b}, \quad b = 1, 2, \dots, \ell,$$

and, differentiating this again with respect to time, we get

$$\ddot{T}_{0b} = \frac{d\omega_b}{dt} T_{0b} + \omega_b \frac{dT_{0b}}{dt}, \quad b = 1, 2, \dots, \ell,$$

which, by use of Eqs. (8.51) and (7.23), becomes

$$\ddot{T}_{0b} = (\alpha_b + \omega_b \omega_b) T_{0b}, \quad b = 1, 2, \dots, \ell, \quad (8.54)$$

or, by Eqs. (8.53) and (8.54), can be written as

$$\ddot{T}_{0b} = \left[\sum_{j=1}^f W_{b,j} \ddot{\psi}_j + \sum_{j=1}^f \sum_{k=1}^f (A_{b,j,k} + W_{b,j} W_{b,k}) \dot{\psi}_j \dot{\psi}_k \right] T_{0b}, \quad b = 1, 2, \dots, \ell. \quad (8.55)$$

EXAMPLE 8.4 Continuing from example 8.3, the α_i matrices for each of the bodies of a Cardan/Hooke universal shaft coupling are easily found by Eq. (8.53):

$$\alpha_1 = W_{1,1}\ddot{\psi} + A_{1,1,1}\dot{\psi}^2$$

$$= \left(\frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \ddot{\psi} + \frac{\sin^2 \beta \cos \beta \cos 2\psi}{1 - \sin^2 \beta \sin^2 \psi} \dot{\psi}^2 \right) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\alpha_2 = W_{2,1}\ddot{\psi} + A_{2,1,1}\dot{\psi}^2$$

$$= \ddot{\psi} \begin{bmatrix} 0 & \frac{\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-h \sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} \\ \frac{-\sin \beta \cos^2 \psi}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 \\ \frac{-\sin \beta \cos \beta \sin \psi \cos \psi}{1 - \sin^2 \beta \sin^2 \psi} & \frac{-\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} & 0 & \frac{h \cos \beta}{1 - \sin^2 \beta \sin^2 \psi} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \frac{\sin \beta \cos \beta \dot{\psi}^2}{(1 - \sin^2 \beta \sin^2 \psi)^2}$$

$$\times \begin{bmatrix} 0 & -\cos \beta \sin 2\psi & -(\cos 2\psi + \sin^2 \beta \sin^2 \psi) & h \cos \beta \sin 2\psi \\ \cos \beta \sin 2\psi & 0 & 0 & 0 \\ \cos 2\psi + \sin^2 \beta \sin^2 \psi & 0 & 0 & h \sin \beta \sin 2\psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\alpha_3 = W_{3,1}\ddot{\psi} + A_{3,1,1}\dot{\psi}^2$$

$$= \ddot{\psi} \begin{bmatrix} 0 & \sin \beta & 0 & -h \sin \beta \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & -\cos \beta & 0 & h \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{\sin \beta \cos \beta \sin \psi \dot{\psi}^2}{1 - \sin^2 \beta \sin^2 \psi}$$

$$\times \begin{bmatrix} 0 & \cos \beta \cos \psi & -\sin \psi & -h \cos \beta \cos \psi \\ -\cos \beta \cos \psi & 0 & -\sin \beta \cos \psi & 0 \\ \sin \psi & \sin \beta \cos \psi & 0 & -h \sin \beta \cos \psi \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\alpha_4 = W_{4,1}\ddot{\psi} + A_{4,1,1}\dot{\psi}^2 = 0.$$

8.9 Accelerations of Points

Next, we recall that in Eq. (8.1) we defined the acceleration of a point to be the second derivative with respect to time of its global position vector:

$$\ddot{R}_b = \frac{d^2 R_b}{dt^2}, \quad b = 1, 2, \dots, \ell.$$

From Eq. (7.27) we have

$$\dot{R}_b = \omega_b R_b, \quad b = 1, 2, \dots, \ell,$$

and, differentiating this again with respect to time, we get

$$\ddot{R}_b = \frac{d\omega_b}{dt} R_b + \omega_b \frac{dR_b}{dt}, \quad b = 1, 2, \dots, \ell,$$

which, by use of Eqs. (8.51) and (7.27), becomes

$$\ddot{R}_b = (\alpha_b + \omega_b \omega_b) R_b, \quad b = 1, 2, \dots, \ell, \quad (8.56)$$

or, by Eqs. (8.53) and (8.54), can be written as

$$\ddot{R}_b = \left[\sum_{j=1}^f W_{b,j} \ddot{\psi}_j + \sum_{k=1}^f \sum_{j=1}^f (A_{b,j,k} + W_{b,j} W_{b,k}) \dot{\psi}_j \dot{\psi}_k \right] R_b, \quad b = 1, 2, \dots, \ell. \quad (8.57)$$

Thus, we see that the same derivative operator matrices found for taking either second geometric or second time derivatives of body postures are also used for finding second derivatives of point positions.

REFERENCE

1. J. Denavit, R. S. Hartenberg, R. Razi, and J. J. Uicker, Jr., “Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages,” *Journal of Applied Mechanics, ASME Transactions*, 1965, pp. 903–10.

PROBLEMS

8.1 Continue from the results of problems 6.1, 6.2, and 7.1 as follows:

- a) Form the matrix ϕ'' of second geometric derivatives of the joint variables.
- b) Form the second geometric derivative-operator matrix $A_{b,j,k}$ for each body.
- c) Form the set $\ddot{\phi}$ of second time derivatives of the joint variables.
- d) Form the second time derivative (acceleration) operator matrix α_b for each body.

8.2 Verify that the upper-left (3×3) submatrix of $C''_{i,j,k}$ in Eq. (8.29) is skew symmetric.

8.3 Continue from the results of problems 6.4, 6.5, and 7.2 as follows:

- a) Form the matrix ϕ'' of second geometric derivatives of the joint variables.
- b) Form the second geometric derivative-operator matrix $A_{b,j,k}$ for each body.
- c) Form the set $\ddot{\phi}$ of second time derivatives of the joint variables.
- d) Form the second time derivative (acceleration) operator matrix α_b for each body.

Modeling Dynamic Aspects of Mechanisms and Multibody Systems

9.1 Introduction

In the very beginning of this text, section 1.1, we observed that the science of mechanics is composed of two parts called statics and dynamics, first distinguished by Euler in 1765. His advice is, perhaps, worth repeating here [1]:

The investigation of the motion of a rigid body may be conveniently separated into two parts, the one geometrical, the other mechanical. In the first part, the transference of the body from a given position to any other position must be investigated without respect to the causes of the motion, and must be represented by analytical formulae which will define the position of each point of the body after the transference with respect to its initial placement. This investigation will therefore be referable solely to geometry, or rather to stereometry [the art of stone-cutting].

It is clear that by the separation of this part of the question from the other, which belongs properly to Mechanics, the determination of the motion from dynamic principles will be made much easier than if the two parts were undertaken conjointly.

We also noted that dynamics is made up of two major disciplines, later recognized as the distinct sciences of kinematics and kinetics, which treat the motion and the forces producing it, respectively.

As should be evident from the preceding chapters, one predominate challenge in the analysis of multibody systems, particularly those with closed-loop topology, is that of kinematics, and one major emphasis of this book has been on that topic. As Euler advised, the methods of the preceding chapters have been totally derived from geometric principles. The units of all parameters defined up to this point have been solely those of length and time.

Statics and kinetics, however, are also extremely important parts of a complete design analysis. The engineer or designer is vitally concerned with the forces transmitted between the parts of a system so that they can be designed to withstand the stresses induced. The work and energy that the system produces or requires are of significant interest. Therefore, these topics and units, including those of force and energy, are also covered in depth in the coming chapters.

Of course, every student, from their first introduction to Newton's laws, has been taught that the first requirement when analyzing a system to find forces is to

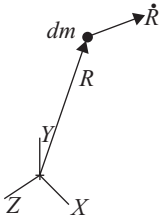


Figure 9.1 Differential particle of mass

“draw a free-body diagram” of the system being studied. This is excellent advice for hand-calculation methods. However, such approaches do not lend themselves to computer evaluation. Computers gain no guidance about the solution of such problems from hand-drawn diagrams. The techniques that we seek here need not depend on such an approach but, instead, will be based on energy. In the energy approach, applied forces can be handled through work-energy principles where the energy contribution of the applied forces are computed from the work performed by such forces.

9.2 Modeling Kinetic Energy

In order to use an energy approach, we will require the formulation of an expression for the kinetic energy of our multibody system. To derive such an equation, let us start by considering a single differential particle of mass dm as shown in [Figure 9.1](#). In terms of the global components of the velocity of this moving particle, its kinetic energy is

$$\begin{aligned} dH &= \frac{1}{2}\{(\dot{R}^X)^2 + (\dot{R}^Y)^2 + (\dot{R}^Z)^2\}dm \\ &= \frac{1}{2}[\dot{R}^X \quad \dot{R}^Y \quad \dot{R}^Z \quad 0] \begin{bmatrix} \dot{R}^X \\ \dot{R}^Y \\ \dot{R}^Z \\ 0 \end{bmatrix} dm \\ &= \frac{1}{2}\dot{R}^t \dot{R} dm. \end{aligned}$$

Let us suppose that this is a single particle of mass on moving body number b and, from Eq. (7.27), substitute our matrix form for its absolute velocity,

$$dH_b = \frac{1}{2}r_b^t T_{0b}^t \omega_b^t \omega_b T_{0b} r_b dm.$$

On integrating this over all particles of body b , we see

$$H_b = \frac{1}{2} \int r_b^t [T_{0b}^t \omega_b^t \omega_b T_{0b}] r_b dm.$$

Unfortunately, this formula presents a significant problem. When we consider which elements of this integral change from one particle to another, we see that every particle of body b uses the same transformation matrix T_{0b} and the same velocity operator matrix ω_b . However, because r_b changes for every particle of body b , and because matrix multiplication is not commutative, we cannot factor the invariants out of the integral. Therefore, if done in this way, the integration must be performed

anew, probably numerically, every time the posture or the velocity of the body changes. This is certainly not an attractive prospect.

Fortunately, there is another approach. Starting again in formulating the expression for the kinetic energy of the particle, and recalling that the trace of a square matrix is defined as the sum of the terms on the principal diagonal, we can write

$$\begin{aligned}
 dH &= \frac{1}{2}\{(\dot{R}^X)^2 + (\dot{R}^Y)^2 + (\dot{R}^Z)^2\}dm \\
 &= \frac{1}{2}\text{trace} \left(\begin{bmatrix} \dot{R}^X \dot{R}^X & \dot{R}^X \dot{R}^Y & \dot{R}^X \dot{R}^Z & 0 \\ \dot{R}^Y \dot{R}^X & \dot{R}^Y \dot{R}^Y & \dot{R}^Y \dot{R}^Z & 0 \\ \dot{R}^Z \dot{R}^X & \dot{R}^Z \dot{R}^Y & \dot{R}^Z \dot{R}^Z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) dm \\
 &= \frac{1}{2}\text{trace} \left(\begin{bmatrix} \dot{R}^X \\ \dot{R}^Y \\ \dot{R}^Z \\ 0 \end{bmatrix} \begin{bmatrix} \dot{R}^X & \dot{R}^Y & \dot{R}^Z & 0 \end{bmatrix} \right) dm \\
 &= \frac{1}{2}\text{trace}(\dot{R}\dot{R}^t)dm.
 \end{aligned}$$

The huge advantage of this seemingly minor change does not become apparent until we substitute Eq. (7.27) for the velocity of the differential particle into our new energy expression. This now gives

$$dH_b = \frac{1}{2}\text{trace}(\omega_b T_{0b} r_b r_b^t T_{0b}^t \omega_b^t) dm, \quad b = 1, 2, \dots, \ell.$$

When we recognize that ω_b and T_{0b} are invariant for every particle of body b , the integral can now be written

$$H_b = \frac{1}{2}\text{trace} \left(\omega_b T_{0b} \left[\int r_b r_b^t dm \right] T_{0b}^t \omega_b^t \right), \quad b = 1, 2, \dots, \ell. \quad (9.1)$$

The integration over all particles of the body no longer depends either on the posture of the body (T_{0b}) or on its velocity (ω_b). It is now an integral over the local coordinates of the particles of mass. It is now possible to perform the integration once, for all postures and all velocities of the body, as will be shown in the next section.

9.3 The Inertia Matrix

Let us define the matrix

$$J_b = \int r_b r_b^t dm, \quad b = 1, 2, \dots, \ell. \quad (9.2)$$

When we expand the terms, we see

$$J_b = \int \begin{bmatrix} (r_b^x)^2 & r_b^x r_b^y & r_b^x r_b^z & r_b^x \\ r_b^y r_b^x & (r_b^y)^2 & r_b^y r_b^z & r_b^y \\ r_b^z r_b^x & r_b^z r_b^y & (r_b^z)^2 & r_b^z \\ r_b^x & r_b^y & r_b^z & 1 \end{bmatrix} dm, \quad b = 1, 2, \dots, \ell.$$

However, because integration is a process of repetitive summation and because matrix sums are performed by summing individual terms, the integral can be performed on a term-by-term basis. Therefore,

$$J_b = \begin{bmatrix} \int (r_b^x)^2 dm & \int r_b^x r_b^y dm & \int r_b^x r_b^z dm & \int r_b^x dm \\ \int r_b^y r_b^x dm & \int (r_b^y)^2 dm & \int r_b^y r_b^z dm & \int r_b^y dm \\ \int r_b^z r_b^x dm & \int r_b^z r_b^y dm & \int (r_b^z)^2 dm & \int r_b^z dm \\ \int r_b^x dm & \int r_b^y dm & \int r_b^z dm & \int dm \end{bmatrix}, \quad b = 1, 2, \dots, \ell.$$

Checking with any of a number of texts on mechanics reminds us that many of these integrals are known from the definitions of the mass distribution parameters of a rigid body, [2]. For example, if we denote the total mass of body b by the symbol m_b and the location of its center of mass, measured with respect to the local body coordinate system by the vector \bar{r}_b , then the definition of the elements of the center of mass location are given by

$$\bar{r}_b^x = \frac{1}{m_b} \int r_b^x dm, \quad \bar{r}_b^y = \frac{1}{m_b} \int r_b^y dm, \quad \bar{r}_b^z = \frac{1}{m_b} \int r_b^z dm, \quad b = 1, 2, \dots, \ell, \quad (9.3)$$

and the integrals in the fourth row and column of the J_b matrix become

$$J_b = \begin{bmatrix} \int (r_b^x)^2 dm & \int r_b^x r_b^y dm & \int r_b^x r_b^z dm & m_b \bar{r}_b^x \\ \int r_b^y r_b^x dm & \int (r_b^y)^2 dm & \int r_b^y r_b^z dm & m_b \bar{r}_b^y \\ \int r_b^z r_b^x dm & \int r_b^z r_b^y dm & \int (r_b^z)^2 dm & m_b \bar{r}_b^z \\ m_b \bar{r}_b^x & m_b \bar{r}_b^y & m_b \bar{r}_b^z & m_b \end{bmatrix}.$$

Checking further with mechanics texts [2] reminds us that the remaining off-diagonal integrals are known as the mass products of inertia of the body. They usually carry symbols such as

$$I_b^{xy} = \int r_b^x r_b^y dm, \quad I_b^{yz} = \int r_b^y r_b^z dm, \quad I_b^{zx} = \int r_b^z r_b^x dm, \quad b = 1, 2, \dots, \ell. \quad (9.4)$$

Different texts do not agree on whether the mass products of inertia should include a minus sign as a part of their definitions. No minus sign is shown here; however, the reader is advised to use care if comparing with different texts.

Using our notation, the mass moments of inertia are usually defined as follows [2]

$$\begin{aligned} I_b^{xx} &= \int \left\{ (r_b^y)^2 + (r_b^z)^2 \right\} dm, & I_b^{yy} &= \int \left\{ (r_b^z)^2 + (r_b^x)^2 \right\} dm, \\ I_b^{zz} &= \int \left\{ (r_b^x)^2 + (r_b^y)^2 \right\} dm, & & b = 1, 2, \dots, \ell. \end{aligned} \quad (9.5)$$

These are sometimes called “polar” moments of inertia and the definition of I^{zz} , for example, integrates the square of the distance of each mass particle from the z axis.

Once we express the aforementioned J_b matrix in terms of these definitions for the mass moments and products of inertia, we obtain an inertia matrix for each body:

$$J_b = \begin{bmatrix} \frac{1}{2}(-I_b^{xx} + I_b^{yy} + I_b^{zz}) & I_b^{xy} & I_b^{zx} & m_b \bar{r}_b^x \\ I_b^{xy} & \frac{1}{2}(I_b^{xx} - I_b^{yy} + I_b^{zz}) & I_b^{yz} & m_b \bar{r}_b^y \\ I_b^{zx} & I_b^{yz} & \frac{1}{2}(I_b^{xx} + I_b^{yy} - I_b^{zz}) & m_b \bar{r}_b^z \\ m_b \bar{r}_b^x & m_b \bar{r}_b^y & m_b \bar{r}_b^z & m_b \end{bmatrix},$$

$$b = 1, 2, \dots, \ell. \quad (9.6)$$

Because these are values that designers may already know about the mass distributions of the individual parts of their devices, or that they might find reasonable to obtain through experiment or through their favorite CAD system, these values form a very suitable model in which the user can be requested to supply the mass distribution data for each moving body [4].

As might have been expected, we note that each J_b matrix is symmetric. This becomes very important in section 10.3 when finding the equations of motion.

The data for the mass distribution of a body is collected totally in the local coordinate system of the moving body. Thus, the J_b matrix is, by its very definition, expressed in that coordinate system. However, when we substitute Eq. (9.2) into Eq. (9.1),

$$H_b = \frac{1}{2} \text{trace}(\omega_b T_{0b} J_b T_{0b}^t \omega_b^t), \quad b = 1, 2, \dots, \ell,$$

we see the matrix product $T_{0b} J_b T_{0b}^t$. After review of this derivation, we recognize that this product, called a similarity transformation, yields the transformation of the inertia matrix to the global coordinate system where it combines with the global coordinate velocity matrix ω_b of the same body. This matrix product is of size (4×4) and all entries have the same arrangement and the same physical interpretation as those of Eq. (9.6) except that they are transformed to the global coordinate system.

Of course, the total kinetic energy of the system is found by summing the kinetic energies of all individual bodies:

$$H = \sum_{b=1}^{\ell} H_b = \frac{1}{2} \sum_{b=1}^{\ell} \text{trace}(\omega_b T_{0b} J_b T_{0b}^t \omega_b^t). \quad (9.7)$$

When the record for storage of the data for a body is formed in computer memory, the software should require that each body record include storage locations for the elements of the inertia matrix, preferably expressed in the local coordinate system of the body. These storage locations can be initialized to zeroes. When data are supplied by the user for the mass of the body, this data can be stored in row four, column four, thus simulating a point mass located at the origin of the body coordinate system. When the body coordinates of the center of mass are given, the elements of the fourth row and column are formed; the mass data then still represents a point mass, but at the newly specified location on the body. Finally, when mass moments and products of inertia are supplied by the user, the remaining elements of the matrix are formed and the inertia matrix no longer represents a point mass, but a distributed mass. Any body for which mass data are not supplied is considered to have negligible (zero) mass when compared to those for which nonzero mass data are supplied.

9.4 Systems of Units

The units of the data in the J_b inertia matrix are also of great importance. As is usual in engineering design analysis, it is probable that time is measured in seconds. Up to this point, it has been assumed that all data for lengths or distances are entered in one self-consistent set of distance units, but no particular system of units has been specified. Lengths might be measured in meters, millimeters, feet, or inches depending on the application and the preference of the analyst. Any of these or other length units can be chosen as long as all lengths are specified in the *same* units, and the user must also be satisfied with these chosen units for all values reported as simulation results involving distances.

As we progress in our study of dynamics, we will also choose a standard unit for force, almost certainly the Newton if the meter or the millimeter is chosen for length, or the pound if the inch or the foot is chosen as the unit of length.

Once these choices are made, then consistency demands that mass data *must* be given in units of force seconds squared per unit length. This is called a gravitational system of units. See, for example, [5] Uicker, *op.cit.*, section 13.3.

With such a system, use of Newtons for forces and meters for lengths requires that mass data be given in Newton seconds squared per meter ($N \cdot s^2/m$), which are named kilograms (kg). However, use of Newtons for forces and millimeters for lengths requires mass data in Newton seconds squared per millimeter ($N \cdot s^2/mm$), which are named *megagrams* (Mg).

Similarly, use of pounds for forces and feet for distances requires mass data in pound seconds squared per foot ($lb \cdot s^2/ft$). Pound seconds squared per foot are named slugs in some texts. However, to the authors' knowledge, this term is not used anywhere outside of academia and, therefore, it is avoided in this text. Use of pounds for forces and inches for distances requires mass data in units of pound seconds squared per inch ($lb \cdot s^2/in$), which have no other name. The term pound seconds squared per inch ($lb \cdot s^2/in$) must be used for mass.

For user friendliness, it may be preferable to allow the user to supply mass and inertia data in terms of weight units and to have the software convert to mass units by dividing by the standard gravitational constant. However, it is *strongly* recommended that this be done as data are entered and that all data for a particular application be stored in computer memory in one consistent system of units as just discussed.

9.5 Modeling Gravitational Effects

The mass distributions described by the J_b inertia matrices can also be used for modeling the effects of gravitational loads in our multibody system. If we define a homogeneous coordinate vector for the local position of the origin of a body coordinate system,

$$r_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (9.8)$$

then, from Eq. (9.6), we see that

$$J_b r_0 = \begin{bmatrix} m_b \bar{r}_b^x \\ m_b \bar{r}_b^y \\ m_b \bar{r}_b^z \\ m_b \end{bmatrix} = m_b \bar{r}_b, \quad b = 1, 2, \dots, \ell.$$

Transforming this homogeneous coordinate vector from local body coordinates to global coordinates, we get

$$T_{0b} J_b r_0 = T_{0b} \begin{bmatrix} m_b \bar{r}_b^x \\ m_b \bar{r}_b^y \\ m_b \bar{r}_b^z \\ m_b \end{bmatrix} = m_b T_{0b} \bar{r}_b = m_b \bar{R}_b, \quad b = 1, 2, \dots, \ell, \quad (9.9)$$

and because it is easily verified that $T_{0b}^t r_0 = r_0$, the equation can also be written in the form

$$T_{0b} J_b r_0 = (T_{0b} J_b T_{0b}^t) r_0, \quad b = 1, 2, \dots, \ell, \quad (9.10)$$

if this proves to be more convenient when coding the software.

Next, we require the computer user to specify the magnitude and direction of the gravitational force field, if any, which acts on the bodies having mass. This vector is defined in homogeneous coordinate form with components along the global axes and is given the symbol g :

$$g = \begin{bmatrix} g^X \\ g^Y \\ g^Z \\ 0 \end{bmatrix}. \quad (9.11)$$

Note that there is no stipulation that the g vector need have the magnitude of standard gravity, although it often may, and it can conveniently default this way in the software. Still, without this constraint, it occasionally becomes possible to use the g vector to model a distributed force field that is not the result of gravity. For example, if a vehicle is rounding a curve of known radius at a known speed, it may be advantageous to model the centrifugal force field by specifying that g includes an appropriately sized component in the outward radial direction in addition to the vertical gravitational component. In this manner the vehicle can be subjected to a centrifugal force field on all parts having mass, even though the simulation may be done in a quasi-static mode.

With the gravitational g vector known, it becomes possible to write an expression for gravitational potential energy. Assuming that a zero-reference position is defined, where the center of mass of body b coincides with the global origin, the gravitational potential energy of body b with respect to this reference is

$$V_b = -m_b g^t \bar{r}_b, \quad b = 1, 2, \dots, \ell.$$

The negative sign recognizes the convention used here that positive represents energy contained within the body and signifies that when a body is moved opposite

to the direction of the gravity vector, work is done and the potential energy of the body is increased.

Using Eqs. (9.9) and (9.10), the gravitational potential energy of body b becomes

$$V_b = -g^j (T_{0b} J_b T_{0b}^t) r_0, \quad b = 1, 2, \dots, \ell.$$

Therefore, the total gravitational potential energy of the system is

$$V = - \sum_{b=1}^{\ell} g^j (T_{0b} J_b T_{0b}^t) r_0. \quad (9.12)$$

9.6 Modeling Joint Stiffness

Sometimes a real spring is designed to act directly with a joint variable, as is the case with the struts of many automotive or aircraft suspensions, or the coil spring of a clock or a wind-up toy. In other situations, the analyst may wish to simulate the stiffness of a motor or control system that acts within a joint. Therefore, we assume that our mechanical system model may include a stiffness value k_h acting directly with the movement of joint variable ϕ_h . Such a spring or stiffness provides a linear restoring force of magnitude $k_h(\phi_h - \phi_{h0})$, where ϕ_{h0} represents the “free position” of joint ϕ_h , at which position the restoring force is zero. Of course, when the movement of joint variable ϕ_h is a rotation, then k_h is a torsional stiffness, ϕ_{h0} is an angle, and $k_h(\phi_h - \phi_{h0})$ is the magnitude of a restoring torque.

The potential energy stored in such a spring or joint stiffness is

$$V_h = \frac{1}{2} k_h (\phi_h - \phi_{h0})^2, \quad h = 1, 2, \dots, n,$$

and the total potential energy stored in all such springs of the model is

$$V = \frac{1}{2} \sum_{h=1}^n k_h (\phi_h - \phi_{h0})^2. \quad (9.13)$$

As previously explained for mass data, the record created in computer memory to store each joint can include memory locations for storing values of a stiffness and a free position associated with each joint variable. These can be initialized to have values of zero. At a later step in the creation of the system model, data may (or may not) be supplied to represent the existence of a nonzero spring rate or stiffness for one or more of these joint variables.

9.7 Modeling Joint Damping

Just as a joint can display stiffness as modeled in the previous section, so too, it can show energy dissipation through damping. Here we assume that this energy dissipation can be modeled as viscous damping, that is, that there can be a resisting force or torque acting against the motion of a joint variable, which is of magnitude $-c_h \dot{\phi}_h$ proportional in size but opposite in sense to the (relative) velocity of the joint variable. Here c_h is the viscous-damping coefficient.

During an infinitesimal displacement $\delta\phi_h$ of this joint variable at velocity $\dot{\phi}_h$, the energy loss from the system through such damping is

$$\delta U_h = -c_h \dot{\phi}_h \delta\phi_h, \quad h = 1, 2, \dots, n.$$

If we express the displacement and velocity of this joint variable in terms of the displacements and velocities of the generalized coordinates, then

$$\delta\phi_h = \sum_{k=1}^f \phi'_{hk} \delta\psi_k, \quad h = 1, 2, \dots, n,$$

and

$$\dot{\phi}_h = \sum_{j=1}^f \phi'_{hj} \dot{\psi}_j, \quad h = 1, 2, \dots, n.$$

Then the infinitesimal energy loss from this joint variable is

$$\delta U_h = - \sum_{k=1}^f \sum_{j=1}^f (\phi'_{hk} c_h \phi'_{hj} \dot{\psi}_j) \delta\psi_k, \quad h = 1, 2, \dots, n.$$

The total energy loss from all viscous joint damping during this small displacement is

$$\delta U = - \sum_{k=1}^f \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hk} c_h \phi'_{hj} \dot{\psi}_j) \delta\psi_k. \quad (9.14)$$

Again, the record created in computer memory to store each joint can include a memory location for storage of a value for a viscous-damping coefficient associated with each joint variable. These can be initialized to have coefficient values of zero. At a later step in the creation of the system model, data may (or may not) be supplied to represent the existence of nonzero viscous-damping coefficients for one or more particular joint variables.

Many other types of energy dissipation have been proposed in the literature beyond the viscous-damping model shown here. Coulomb damping, hysteretic damping, proportional damping, quadratic damping, and others are covered in various texts, [3]. Arguments abound over which is the more appropriate model in a given situation. However, these are not the purpose of this text, and only viscous damping is presented here. Others can be implemented in matrix notation by similar methods if the reader wishes to do so.

9.8 Modeling Point-to-Point Springs

Many multibody systems include one or more springs connected between points of the moving bodies. If we assume that the endpoints of such a spring are the points R_b and R_c of two different bodies labeled b and c , then we can define the symbol R_{bc} to represent the vector between these two points. Note that the digraph bc as a subscript signifies only a single spring, yet it includes a pair of points, the endpoints of the spring, and a pair of integers are referenced. These are the labels of the bodies

containing the two endpoints. Still, there is only a single spring – a single point pair – for each such digraph

$$R_{bc} = R_b - R_c = T_{0b}r_b - T_{0c}r_c, \quad b, c = 1, 2, \dots, \ell. \quad (9.15)$$

The distance between the two points is

$$\ell_{bc} = \sqrt{R_{bc}^t R_{bc}}, \quad b, c = 1, 2, \dots, \ell, \quad (9.16)$$

and the unit vector showing the orientation of the vector between the two points is defined by the symbol

$$u_{bc} = R_{bc}/\ell_{bc}, \quad b, c = 1, 2, \dots, \ell, \quad (9.17)$$

so that

$$R_{bc} = \ell_{bc}u_{bc}, \quad b, c = 1, 2, \dots, \ell.$$

We now assume that our mechanical system model includes a linear spring with rate k_{bc} and free length ℓ_{bc0} , which provides a tensile force of magnitude $k_{bc}(\ell_{bc} - \ell_{bc0})$ directed onto point b from point c and having an equal and opposite reaction force onto point c . The potential energy stored in such a spring is

$$V_{bc} = \frac{1}{2}k_{bc}(\ell_{bc} - \ell_{bc0})^2, \quad b, c = 1, 2, \dots, \ell,$$

and the total potential energy stored in all such springs of the system is

$$V = \frac{1}{2} \sum_{bc} k_{bc}(\ell_{bc} - \ell_{bc0})^2. \quad (9.18)$$

Note that no symbol has been defined for the total number of such springs in the model and no numbering convention or order has been chosen for their identification. In the actual software, the records for these springs will, almost certainly, be implemented in a linked list or queue, and the summation will be accomplished by incrementing through this list.

9.9 Modeling Point-to-Point Dampers

Many multibody systems also include one or more dampers in the form of dashpots connected between points of the moving bodies. As with point-to-point springs, we assume that the two endpoints of such a damper are the points R_c and R_d of two different bodies labeled c and d . We again assume that the energy dissipation can be modeled as viscous friction; that is, that there is a resisting force that is proportional to the relative velocity, $-c_{cd}\dot{\ell}_{cd}u_{cd}$, acting along the line between the two points. During an infinitesimal virtual displacement δR_{cd} between the two points, the energy loss to the system is

$$\delta U_{cd} = -c_{cd}\dot{\ell}_{cd}u_{cd}^t \delta R_{cd}, \quad c, d = 1, 2, \dots, \ell. \quad (9.19)$$

Next, we express the displacement and the relative velocity across this damper in terms of the motions of the generalized coordinates. The displacement is

$$\delta R_{cd} = \sum_{j=1}^f (W_{cj}R_c - W_{dj}R_d)\delta\psi_j,$$

and we define

$$R'_{cdj} = W_{cj}R_c - W_{dj}R_d = R'_{cj} - R'_{dj}, \quad \begin{matrix} c, d = 1, 2, \dots, \ell, \\ j = 1, 2, \dots, f, \end{matrix} \quad (9.20)$$

so that

$$\delta R_{cd} = \sum_{j=1}^f R'_{cdj} \delta \psi_j, \quad c, d = 1, 2, \dots, \ell. \quad (9.21)$$

From Eq. (9.16),

$$\begin{aligned} \ell_{cd}^2 &= R_{cd}^t R_{cd}, \\ 2\ell_{cd} \dot{\ell}_{cd} &= \dot{R}_{cd}^t R_{cd} + R_{cd}^t \dot{R}_{cd} = 2R_{cd}^t \dot{R}_{cd}, \\ \dot{\ell}_{cd} &= u_{cd}^t \dot{R}_{cd}, \\ &= u_{cd}^t \sum_{j=1}^f (W_{cj}R_c - W_{dj}R_d) \dot{\psi}_j, \\ \dot{\ell}_{cd} &= \sum_{j=1}^f u_{cd}^t R'_{cdj} \dot{\psi}_j, \quad c, d = 1, 2, \dots, \ell. \end{aligned} \quad (9.22)$$

Therefore, substituting Eqs. (9.22) and (9.21) into Eq. (9.19), the amount of energy dissipated by such a damper is

$$\delta U_{cd} = - \sum_{j=1}^f \sum_{k=1}^f R_{cdj}^t u_{cd} c_{cd} u_{cd}^t R'_{cdk} \dot{\psi}_k \delta \psi_j,$$

and the total energy dissipated by all such point-to-point dampers during such a motion is

$$\delta U = - \sum_{j=1}^f \sum_{k=1}^f \sum_{cd} R_{cdj}^t u_{cd} c_{cd} u_{cd}^t R'_{cdk} \dot{\psi}_k \delta \psi_j. \quad (9.23)$$

As previously explained, the record created in computer memory to store data for each point-to-point damper can include a memory location for storage of a viscous-damping coefficient associated with that damper. Also aforementioned, many other types of energy dissipation are possible beyond the viscous-damping model shown here. Coulomb damping, hysteretic damping, proportional damping, quadratic damping, and others are covered in various texts, [3]. Others can be implemented by similar methods using matrix notation if the reader wishes to do so.

Notice again that no symbol has been defined for the total number of dampers in the model and no numbering convention or order has been chosen for their identification. In the actual software, the records for such dampers will almost certainly be implemented in a linked list or queue, and the summation will be implemented by incrementing through this list.

9.10 Modeling External Forces and Torques Applied with Joint Variables

For a good simulation tool for multibody dynamic systems, it is also necessary to model applied forces and torques that may act on the system. There are two major

categories of such applied forces or torques; some are external forces or torques that act directly within the joints along with the joint variables whereas the other type consists of external forces or torques applied at specific locations on the moving bodies.

In this section, we will model the first category, where an external force acts directly with a rectilinear joint variable or an external torque acts directly with a rotational joint variable. Such a force might, for example, be the result of the action of an electric or hydraulic motor or actuator within the joint labeled h that is programmed to provide a prescribed force or torque as a given function of time, or the force or torque might be applied by an intricate control system in joint h that provides a force or torque as a known function of system geometry.

In any case, we assume that the functional variation of the force or torque acting with joint variable h is described by a known function $f_h(\phi, t)$.¹ The software may even allow reading a table of numeric data, possibly empirical, and when the data is read, the software can perform a Fourier transform to produce a differentiable periodic function.

We adopt the sign convention that the force or torque is positive when it acts to cause a positive displacement of joint variable ϕ_h . Therefore, the work done by such an applied force or torque onto the system during a small displacement is the product of the force or torque and the displacement of the joint variable:

$$\delta U_h = f_h(\phi, t)\delta\phi_h = \sum_{j=1}^f \phi'_{hj} f_h(\phi, t)\delta\psi_j, \quad h = 1, 2, \dots, n.$$

The total work done on the system by all such applied joint forces or torques during a small displacement of the system is

$$\delta U = \sum_{j=1}^f \sum_{h=1}^n \phi'_{hj} f_h(\phi, t)\delta\psi_j. \quad (9.24)$$

9.11 Modeling External Forces and Torques Applied to Bodies

The other major category of force systems is composed of external forces and torques having specified magnitudes and orientations applied to the moving bodies. In order to achieve a high degree of flexibility in modeling, we assume that each such force is applied at a specified point, having location r_b on the body labeled b ; a torque is applied anywhere on body b . Also, a force or torque is oriented to act along a line directed toward specified point r_c from specified point r_d on bodies labeled c and d , respectively.

Note that the bodies and points b , c , and d may be, but are not required to be, distinct from each other. If, for example, both c and d refer to points on the same body, then the force or torque always has a fixed orientation with respect to that body, even though that body may rotate. If, as another example, c and d both refer to points on the fixed frame while body b is moving, then the global orientation of the force or torque is constant. The variety of modeling possibilities is almost unlimited,

¹ Within the IMP software, there is an extensive library of functions that can be combined by mathematical operators to describe most situations.

and no situation has yet arisen that could not be modeled. The only restriction is that points r_c and r_d never become coincident because this would cause the orientation of the force or torque to become undefined.

As we did in section 9.8, we define

$$R_{cd} = R_c - R_d = T_{0c}r_c - T_{0d}r_d.$$

Then, as in Eq. (9.16), we define the distance between the two points as

$$\ell_{cd} = \sqrt{R_{cd}^t R_{cd}},$$

and, as in Eq. (9.17), we define a unit vector showing the orientation of the line between the two points as

$$u_{cd} = R_{cd} / \ell_{cd}.$$

We assume that the functional description of the magnitude of the external force or torque acting on body b is known and is described by a given function of time and system geometry: $f_{bcd}(\phi, t)$ for a force or $\tau_{bcd}(\phi, t)$ for a torque. We assume that the force is positive when pointed toward the point r_c from the point r_d . Therefore, in global coordinates, the force vector is

$$\mathbf{f}_{bcd}(\phi, t) = u_{cd} f_{bcd}(\phi, t), \quad b, c, d = 1, 2, \dots, \ell. \quad (9.25)$$

As was done in the case of the force, the unit vector u_{cd} for a torque is found from two points on bodies c and d . Therefore, in global coordinates, the externally applied torque vector is

$$\boldsymbol{\tau}_{bcd}(\phi, t) = u_{cd} \tau_{bcd}(\phi, t), \quad b, c, d = 1, 2, \dots, \ell. \quad (9.26)$$

If we consider the combination of a force $\mathbf{f}_{bcd}(\phi, t)$ applied at point b along u_{cd} and a torque $\boldsymbol{\tau}_{bcd}(\phi, t)$ along the same line u_{cd} , this force and torque pair is referred to as a *wrench*.

Now let us consider a rigid body with a system of forces and torques applied to it. We know from elementary statics that, at any general point of the body, such a force and torque system is equivalent to a single resultant force $\mathbf{f}(\phi, t)$ and a single resultant torque $\boldsymbol{\tau}(\phi, t)$ acting at this point, as shown in Figure 9.2. This resultant torque can be resolved into two components: one $\tau_t(\phi, t)$ in the direction of the resultant force, and one $\tau_n(\phi, t)$ perpendicular to it as shown.

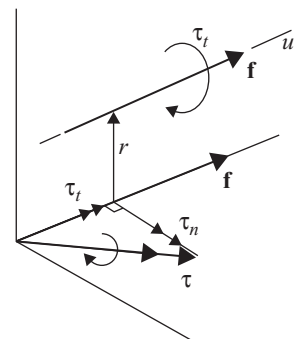


Figure 9.2 Resolving the resultant force (\mathbf{f}) and torque ($\boldsymbol{\tau}$) at a point into an equivalent wrench (\mathbf{f}, τ_t). Note that $\mathbf{r} \times \mathbf{f} = -\tau_n$.

Now an axis \mathbf{u} can be found that is parallel to the line of \mathbf{f} and is in the plane perpendicular to $\boldsymbol{\tau}_n$, but is shifted by a distance r , where $\mathbf{r} \times \mathbf{f} = -\boldsymbol{\tau}_n$. The resultant force $\boldsymbol{\tau}$ and torque $\boldsymbol{\tau}_t$ can be replaced by a wrench consisting of a force $\mathbf{f}(\phi, t) = \mathbf{u}f(\phi, t)$ along and a torque $\boldsymbol{\tau}_t(\phi, t) = \mathbf{u}\tau_t(\phi, t)$ about this axis (in a plane perpendicular to the force). For any force/torque system acting on a rigid body, this axis \mathbf{u} is unique and the corresponding construction leads to *Poinsot's theorem* [4], named for the French mathematician, Louis Poinsot (1777–1859):

A general system of forces and torques acting on a rigid body is equivalent to a wrench about a unique axis consisting of a force along this axis and a torque about this same axis. The axis is called the screw axis for the wrench. Moreover, any system of wrenches acting on a rigid body is equivalent to a single wrench acting on the screw axis.

In general, a *screw* is a line having a linear magnitude named *pitch* associated with it. In the case of differential displacements, Ball vectors define a screw that we call a *twist* that describes the velocity distribution of a rigid body. Similarly here, a *wrench* defines a screw that describes the force distribution on a rigid body. This indicates that Poinsot's theorem can be considered a dual of Chasles' theorem of differential kinematics or that a form of duality exists between the velocity distribution of a rigid body and the force distribution on a rigid body. A differential twist has an *amplitude* that is the differential rotation of the body and the pitch of the twist is the ratio of the differential translation to this amplitude, $\sigma = \delta d(\phi, t)/\delta\theta(\phi, t)$. Similarly, a screw defined by a wrench has an *intensity* that corresponds to the magnitude of the force and its pitch is defined as the ratio of the magnitude of the torque divided by this intensity, $h = \tau(\phi, t)/f(\phi, t)$. When the pitch is zero, a twist represents a rotation whereas a wrench represents a force. Similarly, an infinite pitch corresponds to a translation for a twist and a torque for a wrench.

In Chapter 6, we showed that a (4×4) matrix can be used to represent a screw corresponding to a given twist. Similarly, we can represent a screw associated with a given wrench in terms of a (4×4) matrix in the following manner. We first define a (3×3) skew-symmetric matrix $\tilde{\mathbf{u}}$ whose vector kernel is the unit vector \mathbf{u} ; observing that $\boldsymbol{\tau} = h\mathbf{f}$, we can then form a (4×4) matrix as follows:

$$f(\phi, t) \begin{bmatrix} 0 & -u^z & u^y & hu^x \\ u^z & 0 & -u^x & hu^y \\ -u^y & u^x & 0 & hu^z \\ 0 & 0 & 0 & 0 \end{bmatrix} = f(\phi, t) \begin{bmatrix} \tilde{\mathbf{u}} & h\mathbf{u} \\ 0 & 0 \end{bmatrix}. \quad (9.27).$$

In Eq. (9.27), the factor $f(\phi, t)$ corresponds to the intensity of the screw associated with the wrench and the (4×4) matrix is a matrix representation of the screw. The screw coordinates in the form of a (6×1) vector can be extracted from this (4×4) matrix as:

$$\hat{\mathbf{u}} = \begin{bmatrix} h\mathbf{u} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} hu^x \\ hu^y \\ hu^z \\ u^x \\ u^y \\ u^z \end{bmatrix}, \quad (9.28)$$

and the corresponding wrench is

$$\hat{f}(\phi, t) = f(\phi, t) \begin{bmatrix} hu^x \\ hu^y \\ hu^z \\ u^x \\ u^y \\ u^z \end{bmatrix}. \quad (9.29)$$

However, the specification of a wrench depends on the choice of the coordinate system. For example, the specification given in Eq. (9.27) is based on a coordinate system that has its origin on the axis of the wrench. We know from Chapter 3, however, that the (4×4) matrix representation of an axis can be written in terms of any other coordinate system using a similarity transformation. Let us consider, for example, shifting the origin of the coordinate system used for the specification of the wrench axis in Eq. (9.29) while keeping its orientation. Then, if \mathbf{r} represents a vector from the origin of the new coordinate system to any point on the wrench axis, the coordinate transformation between the two coordinate systems can be described by

$$T = \begin{bmatrix} I & \mathbf{r} \\ 0 & 1 \end{bmatrix},$$

and, using a similarity transformation, the specification of the wrench in this new coordinate system becomes:

$$T \begin{bmatrix} \tilde{u} & hu \\ 0 & 0 \end{bmatrix} T^{-1} = \begin{bmatrix} I & \mathbf{r} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{u} & hu \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & -\mathbf{r} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \tilde{u} & hu - \tilde{u}\mathbf{r} \\ 0 & 0 \end{bmatrix}.$$

Therefore, in this new coordinate system, the wrench is given by

$$f(\phi, t) \begin{bmatrix} \tilde{u} & hu - \tilde{u}\mathbf{r} \\ 0 & 0 \end{bmatrix}, \quad (9.30)$$

Or, in (6×1) format,

$$\hat{f}(\phi, t) = f(\phi, t) \begin{bmatrix} h\mathbf{u} + \mathbf{r} \times \mathbf{u} \\ \mathbf{u} \end{bmatrix}. \quad (9.31)$$

Now if the force distribution on body b of a multibody system is specified by a wrench as given in Eq. (9.29), the wrench can be written in terms of any other coordinate system of the multibody system by using the correct kinematic transformation matrix T in the form of a similarity transformation applied to Eq. (9.29).

Although wrenches provide an elegant representation of the force distribution in a multibody system, we will treat forces and torques and their contributions to the energy of the system separately in terms of the work performed by each. We

do this because treating forces and torques separately provides more convenience and flexibility for the user of a general purpose algorithm or computer program for complex multibody systems. In the remainder of this text, therefore, we treat these separately.

For the point of body b where an external force is applied, we can express a small displacement in terms of displacements of the generalized coordinates

$$\delta R_b = \sum_{j=1}^f W_{bj} R_b \delta \psi_j = \sum_{j=1}^f R'_{bj} \delta \psi_j.$$

The work done onto the system by an external force during a small displacement, then, is the vector dot product of the force and the displacement. Therefore,

$$\delta U_{bcd} = \delta R_b^t \mathbf{f}_{bcd} = \sum_{j=1}^f R'_{bj} u_{cd} f_{bcd}(\phi, t) \delta \psi_j.$$

The total work done by all such externally applied forces acting on the system during this small displacement is

$$\delta U = \sum_{j=1}^f \sum_{bcd} R'_{bj} u_{cd} f_{bcd}(\phi, t) \delta \psi_j. \quad (9.32)$$

For an arbitrary location on the body labeled c to which an external torque is applied, we can express a small displacement in terms of displacements of the generalized coordinates:

$$\delta R_c = \sum_{j=1}^f W_{cj} R_c \delta \psi_j.$$

After reviewing the material of section 6.4, however, and recalling that there is no particular point of application for a torque, we see that the small twist of the body labeled c is expressed by the form

$$\delta \theta_c = \sum_{j=1}^f W_{cj} \delta \psi_j = \sum_{j=1}^f \begin{bmatrix} 0 & -\frac{\delta \theta_c^{z_0}}{\delta \psi_j} & \frac{\delta \theta_c^{y_0}}{\delta \psi_j} & \frac{\delta R_{cO}^x}{\delta \psi_j} \\ \frac{\delta \theta_c^{z_0}}{\delta \psi_j} & 0 & -\frac{\delta \theta_c^{x_0}}{\delta \psi_j} & \frac{\delta R_{cO}^y}{\delta \psi_j} \\ -\frac{\delta \theta_c^{y_0}}{\delta \psi_j} & \frac{\delta \theta_c^{x_0}}{\delta \psi_j} & 0 & \frac{\delta R_{cO}^z}{\delta \psi_j} \\ 0 & 0 & 0 & 0 \end{bmatrix} \delta \psi_j.$$

If we rearrange these terms into Ball vector form as shown in Eq. (6.32), we find that a small angular displacement of the body labeled c is given by the second Ball

vector

$$\begin{aligned} \delta\widehat{\theta}_c &= \begin{Bmatrix} \delta\widehat{\theta}_c^{x_0} \\ \delta\widehat{\theta}_c^{y_0} \\ \delta\widehat{\theta}_c^{z_0} \end{Bmatrix} = \sum_{j=1}^f \begin{Bmatrix} \frac{\delta\theta_c^{x_0}}{\delta\psi_j} \\ \frac{\delta\theta_c^{y_0}}{\delta\psi_j} \\ \frac{\delta\theta_c^{z_0}}{\delta\psi_j} \end{Bmatrix} \delta\psi_j = \sum_{j=1}^f \begin{Bmatrix} W_{cj}(3, 2) \\ W_{cj}(1, 3) \\ W_{cj}(2, 1) \end{Bmatrix} \delta\psi_j \\ &= \sum_{j=1}^f \widehat{W}_{cj} \delta\psi_j, \quad c = 1, 2, \dots, \ell, \end{aligned} \quad (9.33)$$

where $W_{cj}(j, k)$ symbolizes the element from row j , column k of the (4×4) matrix W_{cj} , and where \widehat{W}_{cj} signifies that elements of the (4×4) W_{cj} matrix have been rearranged into a (3×1) column matrix in the order defined in Eq. (9.33) for the second Ball vector. Throughout all software, the W_{cj} matrices and also the Q_c , D_c , and A_{cjk} matrices can all be calculated and stored in Ball vector (screw coordinate) form, thus saving time and redundant storage.

The work done onto the system by an externally applied torque during a small displacement is the vector dot product of the torque and the small angular displacement expressed by the second Ball vector:

$$\delta U_{cde} = \delta\widehat{\theta}_c^t \boldsymbol{\tau}_{cde} = \sum_{j=1}^f \widehat{W}_{cj}^t u_{de} \tau_{cde}(\phi, t) \delta\psi_j, \quad c, d, e = 1, 2, \dots, \ell.$$

Therefore, the total work done by all such externally applied torques during this small twist is

$$\delta U = \sum_{cde} \delta\widehat{\theta}_c^t \boldsymbol{\tau}_{cde} = \sum_{j=1}^f \sum_{cde} \widehat{W}_{cj}^t u_{de} \tau_{cde}(\phi, t) \delta\psi_j. \quad (9.34)$$

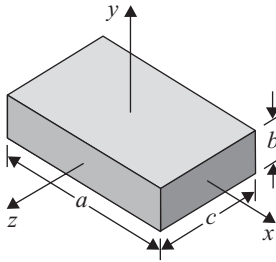
Again, no symbol has been defined for the number of such torques in the system model and no numbering convention or order has been chosen for their identification. In the actual software, the records for these torques are almost certainly implemented in a linked list or queue, and the summation is accomplished by incrementing through this list.

REFERENCES

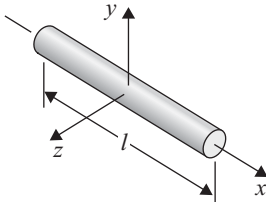
1. L. Euler, "Theoria motus corporum solidorum seu rigidorum (Treatise on the motion of solid or rigid bodies)," *Opera omnia II*, vol. 9, Rostock, 1765; also in "Formulae generales pro translatione quacumque corporum rigidorum, (General formulae for the motion of rigid bodies)," *Novi Comentarum Academiae Scientiarum Petropolitanae* (New memoirs of the imperial academy of sciences in St. Petersburg), vol. 20, 1776, pp. 189–207.
2. R. C. Hibbeler, *Engineering Mechanics*, 11th ed., Prentice-Hall Inc., 2007, Chapters 9 & 10.
3. D. J. Inman, *Engineering Vibration*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
4. L. Poinsot, Sur la composition des moments et la Composition des aires (On the composition of moments and areas), *Journal de l'Ecole Polytechnique*, Paris, 6, 182–205, 1806.
5. J. J. Uicker, Jr., G. R. Pennock, and J. E. Shigley, *Theory of Machines and Mechanisms*, 4th ed., Oxford University Press, 2011, section 14.10.

PROBLEMS

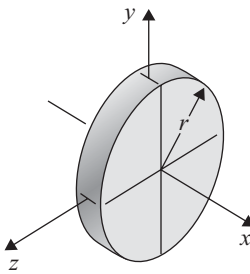
9.1 Form the inertia matrix entries for each of the geometric shapes shown:



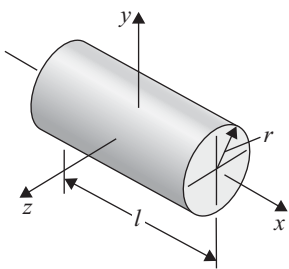
Rectangular prism



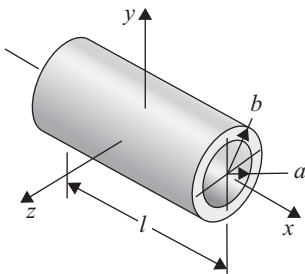
Thin rod



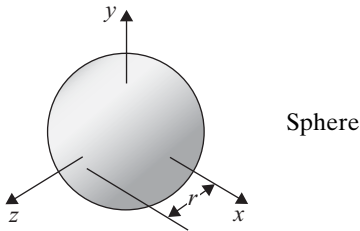
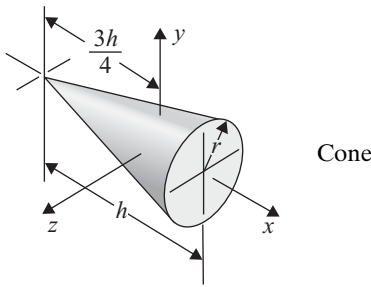
Circular disk



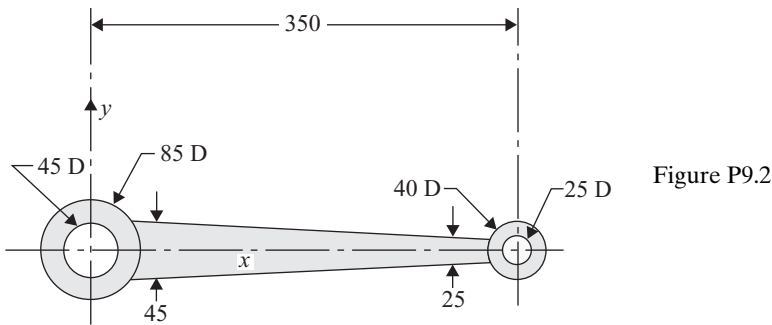
Solid Cylinder



Hollow cylinder



9.2 Figure P9.2 shows a connecting rod made of steel with density of $7\,830\text{ kg/m}^3$. The dimensions are shown in mm. The thicknesses of the two hubs are each 50 mm, and the shaft has constant thickness of 30 mm. Determine the inertia matrix for this connecting rod with the body coordinate axes shown.



9.3 Figure P4.5 shows an Oldham shaft coupling made of steel with density 2.8 lb/in^3 . The two shafts are each 6.0 in length with 0.5 in diameter. The circular hubs are each of 2 in diameter and 0.75 in thickness. The slots and rectangular ribs each have 0.50 in width and 0.375 in thickness. Determine the inertia matrix for each of the moving links with local body axes oriented as shown in Figure P4.5.

10 Dynamic Equations of Motion

10.1 Introduction

Throughout earlier chapters we have carefully formulated our equations in a very general, multi-degree of freedom form. In fact, our only two limiting assumptions so far have been: (1) that all bodies of our system are totally rigid, allowing no deformation or deflection, and (2) that all joints act precisely as described by their mathematical models shown in section 4.6, exhibiting no effects such as backlash or clearances. Indeed, our efforts have produced a kinematic model of our system that is extremely general and powerful. Even though its solution may be tedious for hand calculation, we recognize that evaluation is intended by digital computation and we hope to continue this generality and precision throughout our work in dynamics.

10.2 Lagrange's Equation

Although it may be possible to formulate the equations of motion for a general dynamic system by sketching free-body diagrams, assigning sign conventions and notation, and applying Newton's laws, such an approach is not used here because we are interested in complex and diversified three-dimensional mechanisms and multibody systems and our focus is on developing methods that can be coded for computation in a general setting. An approach based on energy and Lagrange's equation is adopted here, which results in a very general form and minimizes the potential for errors in formulation. Before we discuss the method, however, let us review a very brief history of energy methods in mechanics.

The concept of virtual work had been suggested by Aristotle (384–322 BC) in [2] and by Galileo Galilei (1564–1642) in [3], and then by Johannes Bernoulli (1667–1748), who stated the principle of virtual work in a letter, dated January 26, 1717, to French physicist Pierre Varignon (1654–1722), and it was later published by Varignon in [10]. However, it was Pierre Louis Moreau de Maupertuis (1698–1759) who first enunciated the principle of least action in a public session of the Academy of France on April 15, 1744, and published it in [8]. Then it was Joseph Louis Lagrange (1736–1813) in [7], and Sir William Rowan Hamilton (1805–65) in [4], who developed methods based on work and energy that are especially applicable to the class of dynamic systems that we now study.

Maupertius' *principle of least action* (sometimes called Hamilton's principle) states that nature is thrifty; that is, that nature finds a motion for a system between some beginning state and some ending state for which *action*, the difference between the system's kinetic energy H and potential energy V , is a stationary value. Through his ingenious first use of the *calculus of variations*, in [6] Lagrange developed this principle into what is now called *Lagrange's equation*. The particular form of this equation that fits our situation is as follows:

$$\frac{d}{dt} \left(\frac{\partial H}{\partial \dot{\psi}_i} \right) - \frac{\partial H}{\partial \psi_i} + \frac{\partial V}{\partial \psi_i} = F_i, \quad i = 1, 2, \dots, f, \quad (10.1)$$

where F_i are generalized forces applied at each of the f generalized coordinates ψ_i to account for effects that are not modeled by the kinetic and potential energy functions, H and V . Consistent with our energy approach, this form of Lagrange's equation (see, for example, [9]) requires that these generalized forces be found by modeling the work done during a small displacement of the system:

$$\delta U = \sum_{i=1}^f F_i \delta \psi_i. \quad (10.2)$$

Once the work is expressed in this form, the generalized forces F_i can be identified for use with Eq. (10.1).

10.3 Generalized Momentum

In applying Lagrange's equation, shown in Eq. (10.1), one of our first tasks is to find the partial derivatives of the system's kinetic energy with respect to its generalized velocities. These derivatives yield the components of the *generalized momentum* of the system. Because of this physical significance, we assign a new symbol p_i to these derivatives:

$$p_i = \frac{\partial H}{\partial \dot{\psi}_i}, \quad i = 1, 2, \dots, f. \quad (10.3)$$

Using the model in Eq. (9.7) for the kinetic energy of our system, we write

$$p_i = \frac{\partial}{\partial \dot{\psi}_i} \left[\frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (\omega_b T_{0b} J_b T_{0b}^t \omega_b^t) \right], \quad i = 1, 2, \dots, f. \quad (10.4)$$

Next, recognizing that ω_b are functions of the generalized velocities, we refer to Eq. (7.24) to recall that

$$\frac{\partial \omega_b}{\partial \dot{\psi}_i} = \frac{\partial}{\partial \dot{\psi}_i} \left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) = W_{bi}, \quad \begin{array}{l} b = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f. \end{array} \quad (10.5)$$

Using this in Eq. (10.4), and recognizing that all other factors are functions of position, but are independent of velocity, we find

$$p_i = \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t \omega_b^t) + \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (\omega_b T_{0b} J_b T_{0b}^t W_{bi}^t).$$

However, because the trace function produces a scalar result, which is identical to its own transpose, we can transpose the second term here to show that

$$p_i = \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t \omega_b^t) + \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b^t T_{0b}^t \omega_b^t),$$

and because the inertia matrix J_b is symmetric ($J_b^t = J_b$), we see that

$$p_i = \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t \omega_b^t), \quad i = 1, 2, \dots, f. \quad (10.6)$$

This equation gives the components of the generalized momentum of our system. These show the components of momentum (or angular momentum) of the system, as experienced at each of the generalized coordinates.

10.4 D'Alembert Inertia Forces

We can now continue with our development of the equations of motion of our system. If we take the overall effects of the moving masses, as described by their kinetic energy, these are the generalized d'Alembert inertia forces of the system. Equation (10.1) gives these as

$$G_i^{dyn} = \frac{d}{dt} (p_i) - \frac{\partial H}{\partial \psi_i}, \quad i = 1, 2, \dots, f,$$

and Eqs. (10.6) and (9.7) show these to be

$$G_i^{dyn} = \frac{d}{dt} \left[\sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t \omega_b^t) \right] - \frac{\partial}{\partial \psi_i} \left[\frac{1}{2} \sum_{b=1}^{\ell} \text{trace} (\omega_b T_{0b} J_b T_{0b}^t \omega_b^t) \right].$$

Next we use Eq. (7.24) to expand the forms for the angular velocity operator matrices (ω_b) so that the generalized velocities appear explicitly in the expressions

$$G_i^{dyn} = \frac{d}{dt} \left\{ \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right)^t \right] \right\} \\ - \frac{\partial}{\partial \psi_i} \left\{ \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\}.$$

We now perform the time derivative required in the first term by use of the chain rule

$$G_i^{dyn} = \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\ + \sum_{j=1}^f \frac{\partial}{\partial \psi_j} \left\{ \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \dot{\psi}_j \\ - \frac{\partial}{\partial \psi_i} \left\{ \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\},$$

and we use the derivative operator matrices of Eqs. (8.46) and (7.21) to perform the derivatives indicated

$$\begin{aligned}
G_i^{dyn} = & \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
& + \sum_{j=1}^f \left\{ \sum_{b=1}^{\ell} \text{trace} \left[A_{bij} T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \dot{\psi}_j \\
& + \sum_{j=1}^f \left\{ \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} W_{bj} T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \dot{\psi}_j \\
& + \sum_{j=1}^f \left\{ \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \dot{\psi}_j \\
& + \sum_{j=1}^f \left\{ \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f A_{bkj} \dot{\psi}_k \right)^t \right] \right\} \dot{\psi}_j \\
& - \left\{ \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f A_{bji} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \\
& - \left\{ \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \\
& - \left\{ \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t W_{bi}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right] \right\} \\
& - \left\{ \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left[\left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f A_{bki} \dot{\psi}_k \right)^t \right] \right\}.
\end{aligned}$$

By recognizing similar factors, we can regroup these terms as follows:

$$\begin{aligned}
G_i^{dyn} = & \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
& + \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{bij} + W_{bi} W_{bj}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\} \\
& + \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right]^t \right\} \\
& - \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{bji} + W_{bj} W_{bi}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\} \\
& - \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left\{ \left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right) T_{0b} J_b T_{0b}^t \left[\sum_{k=1}^f (A_{bki} + W_{bk} W_{bi}) \dot{\psi}_k \right]^t \right\}.
\end{aligned}$$

Next, we transpose the last term to give

$$\begin{aligned}
 G_i^{dyn} &= \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
 &+ \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{bij} + W_{bi} W_{bj}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\} \\
 &+ \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right]^t \right\} \\
 &- \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{aji} + W_{bj} W_{bi}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\} \\
 &- \frac{1}{2} \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{k=1}^f (A_{bki} + W_{bk} W_{bi}) \dot{\psi}_k \right] T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \dot{\psi}_j \right)^t \right\},
 \end{aligned}$$

and by interchanging the labels of the indices j and k in the last line, it combines directly with the fourth line to yield

$$\begin{aligned}
 G_i^{dyn} &= \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
 &+ \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{bij} + W_{bi} W_{bj}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\} \\
 &+ \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right]^t \right\} \\
 &- \sum_{b=1}^{\ell} \text{trace} \left\{ \left[\sum_{j=1}^f (A_{bji} + W_{bj} W_{bi}) \dot{\psi}_j \right] T_{0b} J_b T_{0b}^t \left(\sum_{k=1}^f W_{bk} \dot{\psi}_k \right)^t \right\}.
 \end{aligned}$$

However, according to the identity expressed in Eq. (8.49), we see that the second and fourth lines of the previous set of equations nullify each other. This leaves

$$\begin{aligned}
 G_i^{dyn} &= \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
 &+ \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right]^t \right\}, \quad i = 1, 2, \dots, f.
 \end{aligned} \tag{10.7}$$

Also, using Eqs. (8.53) and (7.24) these can be written in the form

$$G_i^{dyn} = \sum_{b=1}^{\ell} \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t (\alpha_b + \omega_b \omega_b)^t], \quad i = 1, 2, \dots, f. \quad (10.8)$$

10.5 Generalized Restoring Forces

The next contribution to our equations of motion is the set of *generalized restoring forces*, sometimes called *generalized static forces*. These are the effects that are derived from potential energy expressions. Using the symbol G_i^{st} , we see from Eq. (10.1) that

$$G_i^{st} = \frac{\partial V}{\partial \psi_i}, \quad i = 1, 2, \dots, f. \quad (10.9)$$

Reviewing the different sections of [Chapter 9](#), we expect to find restoring forces from the effects of gravity, joint stiffnesses, and point-to-point springs. Adding the contributions of Eqs. (9.12), (9.13), and (9.18), we find that the total potential energy from these three sources is

$$V = - \sum_{b=1}^{\ell} g^t T_{0b} J_b T_{0b}^t r_0 + \frac{1}{2} \sum_{h=1}^n k_h (\phi_h - \phi_{h0})^2 + \frac{1}{2} \sum_{cd} k_{cd} (\ell_{cd} - \ell_{cd0})^2. \quad (10.10)$$

Before taking the derivative of this expression, let us first recall Eq. (9.16),

$$\ell_{cd}^2 = R_{cd}^t R_{cd}, \quad c, d = 1, 2, \dots, \ell.$$

Defining another new symbol, $\ell'_{cdi} = \partial \ell_{cd} / \partial \psi_i$, we take the derivative of the previous equation with respect to the generalized coordinate ψ_i , which gives

$$2\ell_{cd} \ell'_{cdi} = R_{cdi}^t R_{cd} + R_{cd}^t R'_{cdi} = 2R_{cd}^t R'_{cdi}.$$

Dividing this by $2\ell_{cd}$, and using Eq. (9.17), we obtain

$$\ell'_{cdi} = \partial \ell_{cd} / \partial \psi_i = u_{cd}^t R'_{cdi} = R_{cdi}^t u_{cd}, \quad \begin{array}{l} c, d = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f. \end{array} \quad (10.11)$$

Finally, we are prepared to evaluate Eq. (10.9) by taking the derivative of Eq. (10.10):

$$\begin{aligned} G_i^{st} &= - \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 - \sum_{b=1}^{\ell} g^t T_{0b} J_b T_{0b}^t W_{bi}^t r_0 \\ &\quad + \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) + \sum_{cd} \ell'_{cdi} k_{cd} (\ell_{cd} - \ell_{cd0}). \end{aligned}$$

It is easily verified from the definition of r_0 that $W_{bi}^t r_0 = 0$; therefore, the second term of this equation is null. Using Eq. (10.11) to express the final term, we obtain

$$\begin{aligned} G_i^{st} &= - \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) \\ &\quad + \sum_{cd} R_{cdi}^t u_{cd} k_{cd} (\ell_{cd} - \ell_{cd0}) \quad i = 1, 2, \dots, f. \end{aligned} \quad (10.12)$$

This is an expression for the restoring force experienced at each generalized coordinate as the result of the combined effects of gravity, joint stiffnesses, and point-to-point springs.

10.6 Generalized Applied Forces

The final step in finding the complete equations of motion is to evaluate the *generalized applied forces*, F_i . As explained following Eq. (10.1) and in Eq. (10.2), we must first write an expression for the work done during a small displacement of the system. However, a quick review reminds us that we have already written these expressions in Chapter 9. Therefore, we can collect the expressions for the work done onto our system by joint damping, from Eq. (9.14), point-to-point dampers, from Eq. (9.23), forces applied at joint variables, from Eq. (9.24), forces applied at moving points, from Eq. (9.32), and torques applied on moving bodies, from Eq. (9.34). In a single expression, the overall work done on our system during a small displacement is as follows:

$$\begin{aligned} \delta U = & \sum_{i=1}^f \left[- \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj} \dot{\psi}_j) - \sum_{j=1}^f \sum_{bc} R'_{bci} u_{bc} c_{bc} u'_{bc} R'_{bcj} \dot{\psi}_j \right. \\ & \left. + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) + \sum_{bcd} R'_{bi} u_{cd} f_{bcd}(\phi, t) + \sum_{cde} \widehat{W}_{ci} u_{de} \tau_{cde}(\phi, t) \right] \delta \psi_i. \end{aligned}$$

According to Eq. (10.2), we can now identify from this work expression the generalized force F_i acting on each of the generalized coordinates during the small displacement $\delta \psi$:

$$\begin{aligned} F_i = & - \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} c_h \phi'_{hj} \dot{\psi}_j - \sum_{j=1}^f \sum_{bc} R'_{bci} u_{bc} c_{bc} u'_{bc} R'_{bcj} \dot{\psi}_j \\ & + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) + \sum_{bcd} R'_{bi} u_{cd} f_{bcd}(\phi, t) \\ & + \sum_{cde} \widehat{W}_{ci} u_{de} \tau_{cde}(\phi, t), \quad i = 1, 2, \dots, f. \end{aligned} \quad (10.13)$$

10.7 Complete Equations of Motion

Finally, we collect all the parts of Eq. (10.1) to form the full equations of motion of our multibody system. Collecting Eqs. (10.7), (10.12), and (10.13), we obtain

$$\begin{aligned} & \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\ & + \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_k \dot{\psi}_j \right]^t \right\} \end{aligned}$$

$$\begin{aligned}
 & - \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 \\
 & + \sum_{g=1}^n \phi'_{gi} k_g (\phi_g - \phi_{g0}) + \sum_{cd} R_{cdi}^t u_{cd} k_{cd} (\ell_{cd} - \ell_{cd0}) \\
 = & - \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} c_h \phi'_{hj} \dot{\psi}_j - \sum_{j=1}^f \sum_{bc} R_{bci}^t u_{bc} c_{bc} u_{bc}^t R'_{bcj} \dot{\psi}_j \\
 & + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) + \sum_{bcd} R_{bi}^t u_{cd} f_{bcd}(\phi, t) \\
 & + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t), \quad i = 1, 2, \dots, f.
 \end{aligned} \tag{10.14}$$

However, because it is usual practice in mechanical system dynamics to show the effects of gravity as applied forces and the effects of damping as restoring forces – in spite of how we have derived their expressions – we may wish to rearrange these equations into the following form:

$$\begin{aligned}
 & \sum_{b=1}^{\ell} \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^f W_{bj} \ddot{\psi}_j \right)^t \right] \\
 & + \sum_{b=1}^{\ell} \text{trace} \left\{ W_{bi} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right]^t \right\} \\
 & + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} c_h \phi'_{hj} \dot{\psi}_j + \sum_{j=1}^f \sum_{bc} R_{bci}^t u_{bc} c_{bc} u_{bc}^t R'_{bcj} \dot{\psi}_j \\
 & + \sum_{g=1}^n \phi'_{gi} k_g (\phi_g - \phi_{g0}) + \sum_{cd} R_{cdi}^t u_{cd} k_{cd} (\ell_{cd} - \ell_{cd0}) \\
 = & \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) + \sum_{bcd} R_{bi}^t u_{cd} f_{bcd}(\phi, t) \\
 & + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t), \quad i = 1, 2, \dots, f.
 \end{aligned} \tag{10.15}$$

These are the complete dynamic equations of motion of our system. They are a coupled set of second-order ($\ddot{\psi}_j$), highly nonlinear ($\dot{\psi}_j \dot{\psi}_k$) differential equations with variable coefficients. The solution of these equations is not a trivial task. This will require numeric integration. Worse yet, although it does not show explicitly, numeric iteration will be required repeatedly, throughout the motion, to insure that the kinematic loops remain properly closed. Yet the solution of these differential equations does describe how the motion of our system develops from some set of initial conditions, and it is from the solution of these differential equations that the values of the free generalized coordinates, and their velocities and accelerations, are determined.

The dynamic equations of motion can also be written in terms of acceleration terms that were discussed in [Chapter 8](#). For example, using Eq. (8.55), Eq. (10.15) can be rewritten as:

$$\begin{aligned}
& \sum_{b=1}^{\ell} \text{trace} [W_{bi} T_{0b} J_b \ddot{T}_{0b}^t] + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} c_h \phi'_h \dot{\psi}_j + \sum_{j=1}^f \sum_{bc} R'_{bci} u_{bc} c_{bc} u_{bc}^t R'_{bcj} \dot{\psi}_j \\
& + \sum_{g=1}^n \phi'_g k_g (\phi_g - \phi_{g0}) + \sum_{cd} R'_{cdi} u_{cd} k_{cd} (\ell_{cd} - \ell_{cd0}) \\
& = \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_h f_h(\phi, t) + \sum_{bcd} R'_{bi} u_{cd} f_{bcd}(\phi, t) \\
& + \sum_{cde} \widehat{W}_{ci} u_{de} \tau_{cde}(\phi, t), \tag{10.16}
\end{aligned}$$

where $\ddot{T}_{0b} = (\alpha_b + \omega_b \omega_b) T_{0b}$, which can be computed recursively from positions, velocities, and accelerations using:

$$T_{0b} = T_{0,b-1} T_{b-1,b},$$

$$\dot{T}_{0b} = \omega_b T_{0b},$$

$$\ddot{T}_{0b} = \alpha_b T_{0b} + \omega_b \dot{T}_{0b}.$$

Let us now consider an open-loop serial multibody system such as a robot manipulator with no springs or dissipative elements. If the system is composed of $\ell = n + 1$ links connected by n single-freedom joints, then, with the fixed link labeled 1 and the moving links labeled 2 through ℓ , the dynamic equations of motion further simplify to:

$$\begin{aligned}
\sum_{b=2}^{\ell} \text{trace} [W_{bi} T_{0b} J_b \ddot{T}_{0b}^t] &= \sum_{b=2}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_h f_h(\phi, t) + R'_{li} u_{cd} f_{lcd}(\phi, t) \\
&+ \widehat{W}_{li} u_{de} \tau_{lde}(\phi, t), \quad i = 1, 2, \dots, n, \tag{10.17}
\end{aligned}$$

where the final two terms on the right-hand side represent force and torque loads applied on the distal link of the chain, and $f_h(\phi, t)$ is the functional variation or the intensity of the joint torque/force in each joint h . If we assume no external force and torque on the distal link and substitute from Eq. (8.55) into Eq. (10.17), we obtain the so-called Uicker-Kahn formulation [5] of manipulator dynamics written in terms of the more general notation of this book:

$$\begin{aligned}
& \sum_{b=2}^{\ell} \left\{ \text{trace} \left[W_{bi} T_{0b} J_b T_{0b}^t \left(\sum_{j=1}^n W_{bj} \dot{\psi}_j + \sum_{j=1}^n \sum_{k=1}^n (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_j \dot{\psi}_k \right) \right]^t \right. \\
& \quad \left. - g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 \right\} \\
& = \sum_{h=1}^n \phi'_h f_h(\phi, t), \quad i = 1, 2, \dots, n. \tag{10.18}
\end{aligned}$$

REFERENCES

1. Aristotle, *The Works of Aristotle*, translated by W. D. Ross and J. A. Smith, Oxford: Clarendon Press, 1908–52.
2. J. d’Alembert, *Traité de Dymanique (Treatise on Dynamics)*, David ‘Ame, Paris, 1743, reprinted by Gauthier-Villars, Paris, 1921.
3. Galileo Galilei, *Della Scienza Meccanica (Science of Mechanics)*, 1594.
4. W. R. Hamilton, *Lectures on Quaternions*, Dublin, 1853.
5. J. M. Hollerbach, “A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-10, no. 11, Nov. 1980, pp. 730–736.
6. J. L. Lagrange, “Essai d’une nouvelle méthode pour déterminer les maxima et minima des formules indéfinies (Essay on a new method for the determination of maxima and minima of indefinite formulae),” *Miscellanea Taurinensia ou Mélanges de Turin (Turin Science Review)*, Turin Mathematical Society, vol. 1, 1759, pp. 3–20.
7. ———, *Méchanique Analytique (Analytical mechanics)*, Imprimeur-Libraire pour les Mathématiques (Imperial Library of Mathematics), Académie des Sciences, Paris, 1788, reprinted by Gauthier-Villars, Paris, 1888.
8. P. L. M. de Maupertius, *Essai de cosmologie (Essay on cosmology)*, Amsterdam, 1750.
9. S. W. McCusky, *An Introduction to Advanced Dynamics*, Addison-Wesley, 1959, Chap. 2.
10. P. Varignon, *Nouvelle mecanique (New mechanics)*, 1725 (posthumously), vol. 2, p. 174.

PROBLEMS

10.1 Continue from the results of problem 4.4 through 8.1 to write the nonlinear dynamic equations of motion of that nonlinear weighing system under the following assumptions:

- a) Only body 3 has mass; its mass is symbolized by m .
- b) There are no dampers.
- c) The only spring acts within joint D ; it has a stiffness of k with a free position of ϕ_{D0} .
- d) Gravity of magnitude g acts in the negative global Y direction.
- e) A vertical load of weight W is applied at point E at the free (left) end of body 2.

10.2 Verify that $W_{bi}^t r_0 = 0$.

10.3 Continue from the results of problem 4.5 through 9.3 to write the nonlinear dynamic equations of motion of the Oldham shaft coupling under the following assumptions:

- a) Bodies have the dimensions and mass distributions specified in problem 9.3.
- b) There are no dampers.
- c) There are no springs or joint stiffnesses.
- d) The effects of gravity are negligible in comparison with the applied loads.
- e) The shaft coupling is driven by a torque on the shaft of body 1 against a constant load torque M_D on the shaft of body 3.

10.4 Continue from the results of problem 10.3 using a value of $e = 0.25$ in for the shaft eccentricity. Make a plot of the steady state ($\dot{\psi} = 0$) variation with unit rotational velocity of the input shaft torque for one cycle of operation.

11 Linearized Equations of Motion

11.1 Introduction

The general equations of motion, developed in [Chapter 10](#), are, without question, the complete and proper model of our dynamic system. However, because of their nonlinear character, they are not directly amenable to the use of the many mathematical tools that are available for *linear* systems. For example, many vibration and automatic control techniques are directly valid only for linear systems. Indeed, even the electronic instrumentation that is available for measurement of the dynamics of multibody systems is often designed to operate in the frequency domain and, thus, inherently assumes that the system treated is linear.

If there is any hope for a general solution technique for these equations of motion, it is probably through numeric integration by digital computer. We will investigate such an approach in [Chapter 14](#). However, before looking at the general case, let us first study the dynamics of our system in the local vicinity of its current posture.

11.2 Linearization Assumptions

At its current posture, whatever posture this might be, we assert that the system in question exists in accordance with our general dynamic equations of motion. If it is not in equilibrium in the sense of being stationary or operating at constant velocity, then it is in *dynamic equilibrium*, meaning that its accelerations are consistent with these same equations of motion.

Also we expect that, during the next interval of time, our system will continue in its current state of motion, perhaps with acceleration if the applied forces are not in balance. Also, at least if the time interval is short, the system will not stray very far from its current geometry and velocity. Thus, for the current time t^* with the current system posture and velocity described by ψ^* and $\dot{\psi}^*$, the motion will proceed according to the equations

$$\begin{aligned}t &= t^* + \boldsymbol{\tau}, \\ \psi_i(t^* + \boldsymbol{\tau}) &= \psi_i^* + x_i(\boldsymbol{\tau}), \\ \dot{\psi}_i(t^* + \boldsymbol{\tau}) &= \dot{\psi}_i^* + v_i(\boldsymbol{\tau}), \quad i = 1, 2, \dots, f, \\ \ddot{\psi}_i(t^* + \boldsymbol{\tau}) &= \ddot{\psi}_i^* + \dot{v}_i(\boldsymbol{\tau}),\end{aligned}\tag{11.1}$$

where we have defined the new variable τ for the time increment commencing at this instant, and $x_i(\tau)$ and $v_i(\tau)$ to represent the changes in the position and velocity values of the generalized coordinates during the coming time increment. We should note that the treatment of position and velocity as independent of each other was an essential assumption in the derivation of the Lagrange equation by the calculus of variations. The requirement that the new generalized velocities be equal to the derivatives of the generalized positions is not included in their definitions, but will be included as additional differential equations:

$$\dot{x}_i(\tau) = \dot{\psi}_i^* + v_i(\tau), \quad i = 1, 2, \dots, f. \quad (11.2)$$

At $\tau = 0$ we will have $x_i(\tau) = v_i(\tau) = 0$ and, at least for a short time interval τ the motion will persist in such a way that all $x_i(\tau)$ and $v_i(\tau)$ are small. Indeed, we assert that there exists an upper bound on τ that ensures that all $x_i(\tau)$ and $v_i(\tau)$ are small for that interval. By small, it is meant that $x_i(\tau)^2 \ll x_i(\tau)$ and $v_i(\tau)^2 \ll v_i(\tau)$ so that all quadratic and higher degree forms of $x_i(\tau)$ and $v_i(\tau)$ can be ignored in comparison with $x_i(\tau)$ and $v_i(\tau)$.

11.3 Linearization

Therefore, for the general dynamic equations of motion, as found in Eqs. (10.15) and repeated here,

$$\begin{aligned} & \sum_{j=1}^f \sum_{b=1}^{\ell} \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t] \ddot{\psi}_j \\ & + \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk}^t)] \dot{\psi}_j \dot{\psi}_k \\ & + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} c_h \phi'_{hj} \dot{\psi}_j \\ & + \sum_{j=1}^f \sum_{bc} R'_{bci} u_{bc} c_{bc} u'_{bc} R'_{bcj} \dot{\psi}_j \\ & + \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) \\ & + \sum_{bc} R'_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) \\ & = \sum_{b=1}^{\ell} g^b W_{bi} T_{0b} J_b T_{0b}^t r_0 \\ & + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) \\ & + \sum_{bcd} R'_{bi} u_{cd} f_{bcd}(\phi, t) \\ & + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t), \quad i = 1, 2, \dots, f, \end{aligned} \quad (11.3)$$

we can make the substitutions indicated by Eqs. (11.1), expand all coefficients in Taylor series about their current positions and velocities, and discard all terms that are of quadratic or higher degree in the variables $x_i(\mathbf{t})$ and $v_i(\mathbf{t})$:

$$\begin{aligned}
& \sum_{j=1}^f \sum_{b=1}^{\ell} \text{trace}(W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t) | \dot{v}_j \\
& + \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t] \dot{\psi}_j \dot{\psi}_k |^* \\
& + \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} 2 \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t] \dot{\psi}_k |^* v_j \\
& + \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj}) \dot{\psi}_j |^* + \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj}) |^* v_j \\
& + \sum_{j=1}^f \sum_{bc} R''_{bci} u_{bc} c_{bc} u'_{bc} R'_{bcj} \dot{\psi}_j |^* + \sum_{j=1}^f \sum_{bc} R''_{bci} u_{bc} c_{bc} u'_{bc} R'_{bcj} |^* v_j \\
& + \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) |^* + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hj} k_h (\phi_h - \phi_{h0}) |^* x_j + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} |^* x_j \\
& + \sum_{bc} R''_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) |^* \\
& + \sum_{j=1}^f \sum_{bc} R''_{bcij} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) |^* x_j + \sum_{j=1}^f \sum_{bc} R''_{bci} u_{bc} k_{bc} R'_{bcj} |^* x_j \\
& + \sum_{j=1}^f \sum_{bc} R''_{bci} (I - u_{bc} u'_{bc}) R'_{bcj} k_{bc} (\ell_{bc} - \ell_{bc0}) / \ell_{bc} |^* x_j \\
& = \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 |^* + \sum_{j=1}^f \sum_{b=1}^{\ell} g^t (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T_{0b}^t r_0 |^* x_j \\
& + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) |^* + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hj} f_h(\phi, t) |^* x_j + \sum_{j=1}^f \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi, t) \phi'_{gj} |^* x_j \\
& + \sum_{bcd} R''_{bi} u_{cd} f_{bcd}(\phi, t) |^* + \sum_{j=1}^f \sum_{bcd} R''_{bij} u_{cd} f_{bcd}(\phi, t) |^* x_j \\
& + \sum_{j=1}^f \sum_{bcd} [R''_{bi} (I - u_{cd} u'_{cd}) R'_{cdj} f_{bcd}(\phi, t) / \ell_{cd}] |^* x_j \\
& + \sum_{j=1}^f \sum_{bcd} \sum_{g=1}^n R''_{bi} u_{cd} f'_{bcdg}(\phi, t) \phi'_{gj} |^* x_j \\
& + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t) |^* + \sum_{j=1}^f \sum_{cde} \widehat{A}_{cij}^t u_{de} \tau_{cde}(\phi, t) |^* x_j
\end{aligned}$$

$$\begin{aligned}
& + \sum_{j=1}^f \sum_{cde} [\widehat{W}_{ci}^t (I - u_{cd} u_{de}^t) R'_{dej} \tau_{cde}(\phi, t) / \ell_{de}]^* x_j \\
& + \sum_{j=1}^f \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci}^t u_{de} \tau'_{cdeg}(\phi, t) \phi'_{gj}^* x_j, \quad i = 1, 2, \dots, f,
\end{aligned} \tag{11.4}$$

where new symbols are defined as follows

$$\begin{aligned}
f'_{bg}(\phi, t) &= \partial f_b(\phi, t) / \partial \phi_g, \\
f'_{bcdg}(\phi, t) &= \partial f_{bcd}(\phi, t) / \partial \phi_g, \\
\tau'_{cdeg}(\phi, t) &= \partial \tau_{cde}(\phi, t) / \partial \phi_g.
\end{aligned} \tag{11.5}$$

The coefficients in Eqs. (11.4) are all evaluated numerically at time t^* and the only variables remaining in these equations are $\mathbf{z}, x_i(\mathbf{z}), v_i(\mathbf{z})$ and their time derivatives.

If we rearrange Eqs. (11.4) into standard form, in decreasing order of the derivatives, we see

$$\begin{aligned}
& \sum_{j=1}^f \sum_{b=1}^{\ell} \text{trace}(W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t) | \dot{v}_j \\
& + \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} 2 \text{trace}[W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t] \psi_k | v_j \\
& + \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj})^* v_j + \sum_{bc} R'_{bci} u_{bc} c_{bc} u_{bc}^t R'_{bcj} | v_j \\
& + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hij} k_h (\phi_h - \phi_{h0})^* x_j + \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} | x_j \\
& + \sum_{j=1}^f \sum_{bc} R'_{bcij} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0})^* x_j + \sum_{j=1}^f \sum_{bc} R'_{bci} u_{bc} k_{bc} u_{bc}^t R'_{bcj} | x_j \\
& + \sum_{j=1}^f \sum_{bc} R'_{bci} (I - u_{bc} u_{bc}^t) R'_{bcj} k_{bc} (\ell_{bc} - \ell_{bc0}) / \ell_{bc} | x_j \\
& - \sum_{j=1}^f \sum_{b=1}^{\ell} g^t (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T_{0b}^t r_0 | x_j \\
& - \sum_{j=1}^f \sum_{h=1}^n \phi'_{hij} f_h(\phi, t) | x_j - \sum_{j=1}^f \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi, t) \phi'_{gj} | x_j \\
& - \sum_{j=1}^f \sum_{bcd} R'_{bij} u_{cd} f_{bcd}(\phi, t) | x_j \\
& - \sum_{j=1}^f \sum_{bcd} R'_{bi} (I - u_{cd} u_{cd}^t) R'_{cdj} f_{bcd}(\phi, t) / \ell_{cd} | x_j
\end{aligned}$$

$$\begin{aligned}
& - \sum_{j=1}^f \sum_{bcd} \sum_{g=1}^n R_{ib}^n u_{cd} f'_{bcg}(\phi, t) \phi'_{gj} |^* x_j - \sum_{j=1}^f \sum_{cde} \widehat{A}_{cij}^t u_{de} \tau_{cde}(\phi, t) |^* x_j \\
& - \sum_{j=1}^f \sum_{cde} \widehat{W}_{bi}^t (I - u_{de} u_{de}^t) R'_{dej} \tau_{cde}(\phi, t) / \ell_{de} |^* x_j \\
& - \sum_{j=1}^f \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci}^t u_{de} \tau'_{cdeg}(\phi, t) \phi'_{gj} |^* x_j \\
= & - \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} \text{trace} [W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t] \psi_k \psi_j |^* \\
& - \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj}) \psi_j |^* - \sum_{bc} R_{bci}^n u_{bc} c_{bc} u_{bc}^t R'_{bcj} \psi_j |^* \\
& - \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) |^* - \sum_{bc} R_{bci}^n u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) |^* + \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 |^* \\
& + \sum_{h=1}^n \phi'_{hi} f_h(\phi, t) |^* + \sum_{bcd} R_{bi}^n u_{cd} f_{bcd}(\phi, t) |^* \\
& + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t) |^*, \quad i = 1, 2, \dots, f. \tag{11.6}
\end{aligned}$$

11.4 Linearized Equations of Motion

From the coefficients in Eqs. (11.6), we now define the following:

- the system mass matrix, $[M]$

$$M_{ij} = \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t)^*, \quad i = 1, 2, \dots, f, \quad j = 1, 2, \dots, f, \tag{11.7}$$

- the system damping matrix, $[D]$

$$\begin{aligned}
D_{ij} = & \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj}) + \sum_{bc} R_{bci}^n u_{bc} c_{bc} u_{bc}^t R'_{bcj} |^* \\
& + \sum_{k=1}^f \sum_{b=1}^{\ell} 2 \text{trace} [W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t] \psi_k^*, \quad i = 1, 2, \dots, f, \quad j = 1, 2, \dots, f, \tag{11.8}
\end{aligned}$$

- the system stiffness matrix, $[K]$

$$\begin{aligned}
K_{ij} = & \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) |^* + \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} |^* \\
& + \sum_{bc} R_{bci}^n u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) |^* + \sum_{bc} R_{bci}^n u_{bc} k_{bc} u_{bc}^t R'_{bcj} |^*
\end{aligned}$$

$$\begin{aligned}
 & + \sum_{bc} R''_{bci} (I - u_{bc} u_{bc}^t) R'_{bcj} k_b (\ell_{bc} - \ell_{bc0}) / \ell_{bc} |^* \\
 & - \sum_{b=1}^{\ell} g^t (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T_{0b}^t r_0 |^* \\
 & - \sum_{h=1}^n \phi'_{hij} f_h(\phi, t) |^* + \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi, t) \phi'_{gj} |^* \\
 & - \sum_{bcd} R''_{bij} u_{cd} f_{bcd}(\phi, t) |^* \\
 & - \sum_{bcd} R''_{bi} (I - u_{cd} u_{cd}^t) R'_{cdj} f_{bcd}(\phi, t) / \ell_{cd} |^* \\
 & - \sum_{bcd} \sum_{g=1}^n R''_{bi} u_{cd} f'_{bcdg}(\phi, t) \phi'_{gj} |^* \\
 & - \sum_{cde} \widehat{A}_{cij}^t u_{cd} \tau_{cde}(\phi, t) |^* \\
 & - \sum_{cde} \widehat{W}_{ci}^t (I - u_{de} u_{de}^t) R'_{dej} \tau_{cde}(\phi, t) / \ell_{de} |^* \\
 & - \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci}^t u_{de} \tau'_{cdeg}(\phi, t) \phi'_{gj} |^*, \quad \begin{matrix} i = 1, 2, \dots, f, \\ j = 1, 2, \dots, f, \end{matrix} \tag{11.9}
 \end{aligned}$$

- and finally, we define the column vector of *generalized applied forces*, F

$$\begin{aligned}
 F_i(\phi^*, \boldsymbol{\varkappa}) & = - \sum_{j=1}^f \sum_{k=1}^f \sum_{b=1}^{\ell} \text{trace} [W_{bi} T_{0b} J_b T_{0b}^t (A_{bjk} + W_{bj} W_{bk})^t]^* \psi_k^* \psi_j^* \\
 & - \sum_{j=1}^f \sum_{h=1}^n (\phi'_{hi} c_h \phi'_{hj})^* \psi_j^* - \sum_{bc} R''_{bci} u_{bc} c_{bc} u_{bc}^t R'_{bcj} |^* \psi_j^* \\
 & - \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) |^* - \sum_{bc} R''_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) |^* \\
 & + \sum_{b=1}^{\ell} g^t W_{bi} T_{0b} J_b T_{0b}^t r_0 |^* + \sum_{h=1}^n \phi'_{hi} f_h(\phi^*, \boldsymbol{\varkappa}) + \sum_{bcd} R''_{bi} u_{cd} |^* f_{bcd}(\phi^*, \boldsymbol{\varkappa}), \\
 & + \sum_{cde} \widehat{W}_{ci}^t u_{de} |^* \tau_{cde}(\phi^*, \boldsymbol{\varkappa}), \quad i = 1, 2, \dots, f, \tag{11.10}
 \end{aligned}$$

where the notations $f_h(\phi^*, \boldsymbol{\varkappa})$, $f_{bcd}(\phi^*, \boldsymbol{\varkappa})$, and $\tau_{cde}(\phi^*, \boldsymbol{\varkappa})$ signify that the applied forces and torques $f_h(\phi^*, \boldsymbol{\varkappa})$, $f_{bcd}(\phi^*, \boldsymbol{\varkappa})$, and $\tau_{cde}(\phi^*, \boldsymbol{\varkappa})$ are each evaluated for the current posture, but may still be variable functions of time.

With these definitions of the coefficient matrices in Eqs. (11.7), (11.8), and (11.9) and the column matrix of applied forces in Eq. (11.10), we are now able to express the linearized equations of motion, from Eq. (11.6), in the very compact form

$$M\dot{v} + Dv + Kx = F(\phi^*, \boldsymbol{\varkappa}), \tag{11.11}$$

where x is a column matrix of the changes $x_i(\boldsymbol{z})$ that are experienced by the f generalized coordinates ψ_i^* after time t^* , and $v(\boldsymbol{z})$ is a column matrix of the changes in velocity from their values of $\dot{\psi}_i^*$ after time t^* , as expressed in Eqs. (11.2). This set of differential equations must be solved simultaneously with Eqs. (11.2), subject to the initial conditions that, at time t^* , where $\boldsymbol{z} = 0$, we have $x(0) = v(0) = 0$. It must be emphasized again that the coefficient matrices and the column vector of generalized forces have all been evaluated numerically at time t^* and, for a short time interval \boldsymbol{z} , are treated in Eq. (11.11) as independent of changes in geometry. We should remember that the variations of $f_h(\phi^*, \boldsymbol{z})$, $f_{bcd}(\phi^*, \boldsymbol{z})$, and $\tau_{cde}(\phi^*, \boldsymbol{z})$ with changes of geometry are approximated to first order and included in the stiffness matrix through values of $f'_{hg}(\phi^*, \boldsymbol{z})$, $f'_{bcdg}(\phi^*, \boldsymbol{z})$, and $\tau'_{cdeg}(\phi^*, \boldsymbol{z}^*)$ as defined in Eq. (11.5).

By reviewing Eqs. (11.7), (11.8), and (11.9), we see that these coefficient values cannot change unless or until the system moves to a new posture, or the velocities $\dot{\psi}^*$ change. Experience also shows that, even for complex mechanisms and multibody systems, these values are usually weak functions of system geometry and do not change in surprising or dramatic fashion. Unless the system geometry changes – say, by changes in x of a 0.25 radian (or length unit) or more – it is not likely that these coefficient values will change noticeably, even if they are reevaluated. The same holds true for the geometric factors in Eq. (11.10). However, depending on the application to which the system is subjected, the applied forces and torques, $f_h(\phi^*, \boldsymbol{z})$, $f_{bcd}(\phi^*, \boldsymbol{z})$, and $\tau_{cde}(\phi^*, \boldsymbol{z})$ may be large and/or quickly changing, perhaps even impulsive, functions of time.

11.5 Dynamic Equations with Specified Input Motions

In section 6.6, we separated the f degrees of freedom and, therefore, the generalized coordinates into two groups. Those that have known input motions we defined as specified generalized coordinates (SGCs); there are NS of these. Those whose motions are not known we named free generalized coordinates (FGCs); there are NF of these. We now give these two sets the symbolism $\{\psi_S\}$ and $\{\psi_F\}$, respectively.

Next, we consider for which of these sets we have written the equations of motion. There is no question; because our energy and virtual work expressions were written to include all energy – regardless of cause or origin – we must agree that the equations found, both the nonlinear equations found in Chapter 10, Eqs. (10.15), and the linearized versions of Eqs. (11.11), are written for the combined sets of all degrees of freedom, including both FGCs and SGCs.

Recalling from section 6.6 that the FGCs precede the SGCs in the order of their identifying labels, we make the distinction between the two sets clear by subdividing the matrices of Eqs. (11.11) as follows:

$$\begin{bmatrix} M_{FF} & M_{FS} \\ M_{SF} & M_{SS} \end{bmatrix} \begin{Bmatrix} \dot{v}_F \\ \dot{v}_S \end{Bmatrix} + \begin{bmatrix} D_{FF} & D_{FS} \\ D_{SF} & D_{SS} \end{bmatrix} \begin{Bmatrix} v_F \\ v_S \end{Bmatrix} + \begin{bmatrix} K_{FF} & K_{FS} \\ K_{SF} & K_{SS} \end{bmatrix} \begin{Bmatrix} x_F \\ x_S \end{Bmatrix} = \begin{Bmatrix} F_F \\ F_S \end{Bmatrix}. \quad (11.12)$$

The upper submatrices describe the NF equations necessary to solve for the motions of the FGCs; the lower submatrices represent the NS equations for the

SGCs. If we expand the upper equations and rearrange them into the usual form, putting known information on the right of the equal sign, we have

$$M_{FF}\dot{v}_F + D_{FF}v_F + K_{FF}x_F = F_F - M_{FS}\dot{v}_S - D_{FS}v_S - K_{FS}x_S = G_F, \quad (11.13)$$

for the FGCs. Once these are solved for x_F and v_F , the remaining equations

$$F_S = M_{SF}\dot{v}_F + M_{SS}\dot{v}_S + D_{SF}v_F + D_{SS}v_S + K_{SF}x_F + K_{SS}x_S, \quad (11.14)$$

can be evaluated to find F_S . These give the driving forces and torques that must be applied at the SGC joint variables at each instant in order to actually achieve the SGC motions that are specified. Thus, all the equations are useful, but only those in Eqs. (11.13) need to be solved as differential equations. That is, only these need to be time integrated. Those in Eqs. (11.14) need only to be numerically evaluated. However, this last set is still very important in the sense of its engineering significance.

PROBLEMS

11.1 Continue from the results of problem 4.4 through 8.1, using the assumptions of problem 10.1 and the following numeric data: $l = 5$ in, $m = 9.6522$ lb, $k = 15$ lb/in, $\phi_{D0} = 20$ in. Find the linearized equation of motion near the posture where $\psi = -30^\circ$.

11.2 Continue from the results of problem 10.4 and determine the values of the mass matrix, damping matrix, stiffness matrix and generalized forcing function for the linearized equation of motion of the Oldham shaft coupling operating at $\dot{\psi}^* = 100$ rev/min.

12 Equilibrium Posture Analysis

12.1 Introduction

A challenge that arises occasionally in the analysis of mechanisms or multibody systems is that of determining a posture of static equilibrium for a mechanism or multibody system under a given set of applied loads. An example might be the question of how much a vehicle suspension will be displaced by putting a variety of known loads in its luggage compartment, or how much the rear axle will drop if the vehicle is hoisted through a known distance by the right-rear bumper. Can these be determined by analysis at the time the vehicle is being designed?

The description of such problems can be formulated from the equations of motion of the system Eq. (10.15). The requirement, however, is to determine values of the generalized coordinates ψ_F for which the system is in a posture of static equilibrium.

One way to find such a posture is to time integrate the equations of motion until all transients disappear through friction or damping. However, such a solution may be slow and inefficient. In this chapter, we hope to find a more direct technique for finding an equilibrium posture.

Once a system reaches a posture of static equilibrium and all transient effects subside, the generalized velocities and accelerations vanish. Let us assume that the applied loads are not time varying. However, they may still be functions of system geometry. Therefore, until the final posture is found, their final values may not be known.

At such an equilibrium posture, Eq. (10.15) takes the form:

$$\begin{aligned} & \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) + \sum_{bc} R''_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) \\ &= \sum_{b=1}^{\ell} g' W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_{hi} f_h(\phi) \\ &+ \sum_{bcd} R''_{bi} u_{cd} f_{bcd}(\phi) + \sum_{cde} \widehat{W}_{ci} u_{de} \tau_{cde}(\phi), \quad i = 1, 2, \dots, f. \end{aligned}$$

12.2 Seeking a Nearby Posture of Equilibrium

As we begin with a new system, we may not know the equilibrium posture, and we usually have no way to assure that the generalized coordinates are initially in a proper position for equilibrium. Because the final geometry is unknown, the previous equations may not be balanced initially; in fact, they are probably not balanced. Therefore, we transfer all terms to the right side of the equation, and give this total combination of forces the symbols F_i that, initially, represent the generalized unbalanced loads. From these equations we hope to seek a position for the generalized coordinates of our system for which these generalized forces F_i become zero:

$$\begin{aligned}
 F_i = & - \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) - \sum_{bc} R''_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) \\
 & + \sum_{b=1}^{\ell} g^j W_{bi} T_{0b} J_b T_{0b}^t r_0 + \sum_{h=1}^n \phi'_{hi} f_h(\phi) + \sum_{bcd} R''_{bi} u_{cd} f_{bcd}(\phi) \\
 & + \sum_{cde} \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi), \quad i = 1, 2, \dots, f.
 \end{aligned} \tag{12.1}$$

Let us now assert the hope that the true posture of static equilibrium is reasonably close to the current geometry. Said in another way, there may be multiple solutions and we will be satisfied with finding the equilibrium posture that is nearest to the current geometry. With this in mind, we expand Eqs. (12.1) in Taylor series to first order, dropping all quadratic and higher-order terms, just as we did in Chapter 11. This gives us a new formula for the generalized forces F_i at a new posture that, hopefully, is near the current posture. We then set the generalized forces at the new posture to zero with the hope that this new posture will more nearly describe an equilibrium posture:

$$\begin{aligned}
 F_i^* + \sum_{j=1}^f \left. \frac{\partial F_i}{\partial \psi_j} \right|_* \delta \psi_j & \approx 0 \\
 - \sum_{h=1}^n \phi'_{hi} k_h (\phi_h - \phi_{h0}) \Big|_* & - \sum_{j=1}^f \sum_{h=1}^n \phi''_{hij} k_h (\phi_h - \phi_{h0}) \Big|_* \delta \psi_j - \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} \Big|_* \delta \psi_j \\
 - \sum_{bc} R''_{bci} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) \Big|_* & - \sum_{j=1}^f \sum_{bc} R''_{bcij} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) \Big|_* \delta \psi_j \\
 - \sum_{j=1}^f \sum_{bc} R''_{bci} u_{bc} k_{bc} u'_{bc} R'_{bcj} \Big|_* & \delta \psi_j \\
 - \sum_{j=1}^f \sum_{bc} [R''_{bci} (I - u_{bc} u'_{bc}) R'_{bcj} k_{bc} & (\ell_{bc} - \ell_{bc0}) / \ell_{bc}] \Big|_* \delta \psi_j \\
 + \sum_{b=1}^{\ell} g^j W_{bi} T_{0b} J_b T_{0b}^t r_0 \Big|_* & + \sum_{j=1}^f \sum_{b=1}^{\ell} g^j (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T_{0b}^t r_0 \Big|_* \delta \psi_j
 \end{aligned}$$

$$\begin{aligned}
& + \sum_{h=1}^n \phi'_{hi} f_h(\phi) \Big|^* + \sum_{j=1}^f \sum_{h=1}^n \phi''_{hij} f_h(\phi) \Big|^* \delta\psi_j + \sum_{j=1}^f \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi) \phi'_{gj} \Big|^* \delta\psi_j \\
& + \sum_{bcd} R''_{bi} u_{cd} f_{bcd}(\phi) \Big|^* + \sum_{j=1}^f \sum_{bcd} R''_{bij} u_{cd} f_{bcd}(\phi) \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{bcd} [R''_{bi} (I - u_{cd} u_{cd}^t) R'_{cdj} f_{bcd}(\phi) / \ell_{cd}] \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{bcd} \sum_{g=1}^n R''_{bi} u_{cd} f'_{bcdg}(\phi) \phi'_{gj} \Big|^* \delta\psi_j \\
& + \sum_{cde} \widehat{W}_{ci} u_{de} \tau_{cde}(\phi) \Big|^* + \sum_{j=1}^f \sum_{cde} \widehat{A}_{cij} u_{de} \tau_{cde}(\phi) \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{cde} [\widehat{W}_{ci} (I - u_{de} u_{de}^t) R'_{dej} \tau_{cde}(\phi) / \ell_{cde}] \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci} u_{de} \tau'_{cdeg}(\phi) \phi'_{gj} \Big|^* \delta\psi_j \approx 0.
\end{aligned}$$

From this expansion, we now collect all zero-order terms. This gives an expression for F_i identical to Eqs. (12.1), but which is numerically evaluated at the current system posture and may not be zero.

Collecting the first-order terms, however, gives us

$$\begin{aligned}
& \sum_{j=1}^f \frac{\partial F_i}{\partial \psi_j} \Big|^* \delta\psi_j \\
& = - \sum_{j=1}^f \sum_{h=1}^n \phi''_{hi} k_h(\phi_h - \phi_{h0}) \Big|^* \delta\psi_j - \sum_{j=1}^f \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} \Big|^* \delta\psi_j \\
& - \sum_{j=1}^f \sum_{bc} R''_{bcij} u_{bc} k_{bc}(\ell_{bc} - \ell_{bc0}) \Big|^* \delta\psi_j - \sum_{j=1}^f \sum_{bc} R''_{bcij} u_{bc} k_{bc} u'_{bc} R'_{bcj} \Big|^* \delta\psi_j \\
& - \sum_{j=1}^f \sum_{bc} [R''_{bcij} (I - u_{bc} u_{bc}^t) R'_{bcj} k_{bc}(\ell_{bc} - \ell_{bc0}) / \ell_{bc}] \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{b=1}^{\ell} g^b (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T'_{0b} r_0 \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{h=1}^n \phi''_{hij} f_h(\phi) \Big|^* \delta\psi_j + \sum_{j=1}^f \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi) \phi'_{gj} \Big|^* \delta\psi_j \\
& + \sum_{j=1}^f \sum_{bcd} R''_{bcij} u_{cd} f_{bcd}(\phi) \Big|^* \delta\psi_j + \sum_{j=1}^f \sum_{bcd} [R''_{bi} (I - u_{cd} u_{cd}^t) R'_{cdj} f_{bcd}(\phi) / \ell_{cd}] \Big|^* \delta\psi_j
\end{aligned}$$

$$\begin{aligned}
 & + \sum_{j=1}^f \sum_{bcd} \sum_{g=1}^n R''_{bi} u_{cd} f'_{bcdg}(\phi) \phi'_{gj} |^* \delta \psi_j \\
 & + \sum_{j=1}^f \sum_{cde} \widehat{A}_{cij}^t u_{de} \tau_{cde}(\phi) |^* \delta \psi_j + \sum_{j=1}^f \sum_{cde} [\widehat{W}_{ci}^t (I - u_{cd} u_{cd}^t) R'_{dej} \tau_{cde}(\phi) / \ell_{de}] |^* \delta \psi_j \\
 & + \sum_{j=1}^f \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci}^t u_{de} \tau'_{cdeg}(\phi) \phi'_{gj} |^* \delta \psi_j.
 \end{aligned}$$

From these we identify the system stiffness matrix K as follows. Note the sign reversal; this resulted because, when we performed the Taylor series expansion, we brought all terms to the right side of the equation whereas K is usually on the left:

$$\begin{aligned}
 K_{ij} = & \sum_{h=1}^n \phi''_{hi} k_h(\phi_h - \phi_{h0}) + \sum_{h=1}^n \phi'_{hi} k_h \phi'_{hj} \\
 & + \sum_{bc} R''_{bcij} u_{bc} k_{bc}(\ell_{bc} - \ell_{bc0}) + \sum_{bc} R''_{bci} u_{bc} k_{bc} u_{bc}^t R'_{bcj} \\
 & + \sum_{bc} [R''_{bci} (I - u_{bc} u_{bc}^t) R'_{bcj} k_{bc}(\ell_{bc} - \ell_{bc0}) / \ell_{bc}] \\
 & - \sum_{b=1}^{\ell} g^j (A_{bij} + W_{bi} W_{bj}) T_{0b} J_b T_{0b}^t r_0 \\
 & - \sum_{h=1}^n \phi''_{hi} f_h(\phi) - \sum_{h=1}^n \sum_{g=1}^n \phi'_{hi} f'_{hg}(\phi) \phi'_{gj} \\
 & - \sum_{bcd} R''_{bcij} u_{cd} f_{bcd}(\phi) - \sum_{bcd} [R''_{bi} (I - u_{cd} u_{cd}^t) R'_{cdj} f_{bcd}(\phi) / \ell_{cd}] \\
 & - \sum_{bcd} \sum_{g=1}^n R''_{bi} u_{cd} f'_{bcdg}(\phi) \phi'_{gj} \\
 & - \sum_{cde} \widehat{A}_{cij}^t u_{de} \tau_{cde}(\phi) - \sum_{cde} [\widehat{W}_{ci}^t (I - u_{de} u_{de}^t) R'_{dej} \tau_{cde}(\phi) / \ell_{de}] \\
 & - \sum_{cde} \sum_{g=1}^n \widehat{W}_{ci}^t u_{de} \tau'_{cdeg}(\phi) \phi'_{gj}, \quad \begin{array}{l} i = 1, 2, \dots, f, \\ j = 1, 2, \dots, f, \end{array} \quad (12.2)
 \end{aligned}$$

and we note with satisfaction that this matches our previous definition in Eqs. (11.9).

Fitting the pieces back together, these definitions allow us to write our Taylor series expansion of Eqs. (12.1) in the form

$$K \delta \psi \approx F, \quad (12.3)$$

which, of course, we only expect to be approximately equal because we have dropped the higher-order terms of the Taylor series. We can, however, proceed to solve Eq. (12.3) as follows:

$$\delta \psi \approx K^{-1} F, \quad (12.4)$$

and then use these changes to adjust the generalized coordinate position values

$$\psi_i \approx \psi_i^* + \delta\psi_i, \quad i = 1, 2, \dots, f. \quad (12.5)$$

Once the generalized coordinate positions are modified, we must apply the numeric technique discussed in Chapter 6 to update the dependent joint variables, the transformation matrices, and all other geometric data. When this is done, the system should be at a new posture that is closer to equilibrium than the previous posture, but perhaps is still not exact.

How closely we have achieved equilibrium can be determined by reevaluating the unbalanced generalized forces of Eq. (12.1). From the magnitudes of these forces and the posture changes of Eq. (12.4), we can evaluate whether we have achieved an equilibrium posture within acceptable accuracy. If not, we can recursively apply Eqs. (12.2), (12.4), (12.5), and the posture solution of Chapter 6 until the desired accuracy is achieved.¹ We can then declare that an equilibrium posture has been found.

Of course, the process just described is a traditional Newton-Raphson numeric iteration process looking for a zero of the function F_i . As with all Newton-Raphson processes, it has an excellent (quadratic) convergence rate once in the vicinity of a solution. However, it also sometimes experiences difficulties, as will be discussed in the following section.

12.3 Seeking Equilibrium with Some Generalized Coordinates Specified

If we have a multiple degree of freedom system and some of the freedoms are SGCs, then we may wish to find an equilibrium position for only some of the generalized coordinates – the FGCs – under a given set of loads. An example of this might be finding the equilibrium posture of a vehicle with a known load in the luggage compartment, as previously described, but with the steering wheel set to a specified angle (as an SGC).

This situation presents only one additional complication. Before seeking equilibrium, any SGCs must be moved to their prescribed positions and the system geometry must be updated by the iteration process of Chapter 6. After this is done, the Newton-Raphson search for equilibrium can be performed in the manner described in section 12.2, with the exception that Eqs. (12.1), (12.2), (12.4), and (12.5) need only be formed for the FGCs rather than for all generalized coordinates, while holding the SGCs at their specified positions. Convergence is still excellent once in the vicinity of a solution. However, sometimes difficulties are still encountered. These are discussed next.

12.4 Large Increments of the Generalized Coordinates

One of the difficulties that can arise results when Eq. (12.4) predicts a very large increment for one or more of the generalized coordinates; in fact, so large that the iteration process of Chapter 6 does not converge, or converges to a different closure of the kinematic loop equations. Experience shows that this problem is usually

¹ The IMP software uses a default accuracy of $|\delta\psi_i| \leq 10^{-5}$. The user can override this limit if desired.

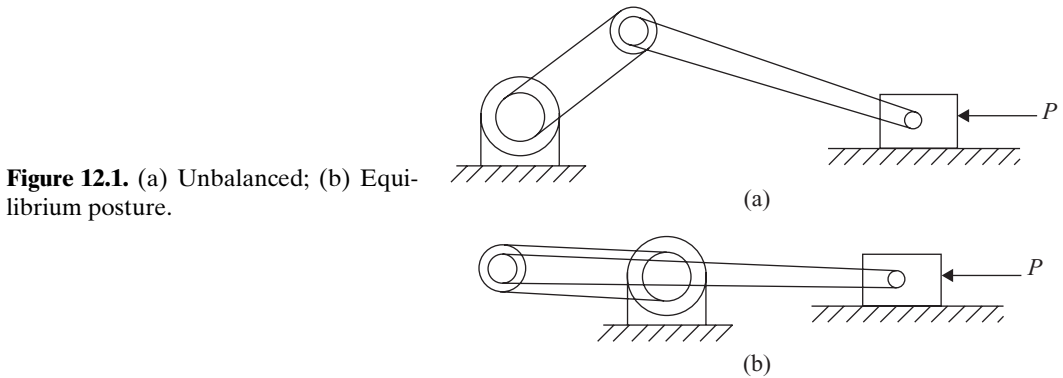


Figure 12.1. (a) Unbalanced; (b) Equilibrium posture.

the result of faulty data supplied by the user, such as with gravitational forces on masses with data mistakenly supplied in grams rather than in kilograms, or with stiffness matrix entries found from torsional spring rates erroneously supplied in inch-pounds per degree rather than per radian.

Whatever the cause, failure of the posture analysis equations can be quite frustrating to the user of the software. To help avoid such failures, after the calculation of the recommended increments for the generalized coordinates from Eq. (12.4) it is wise to test the magnitude of the total $\delta\psi$ vector and to scale its components proportionally smaller if the total vector is beyond a chosen limiting size.² This may slow convergence toward equilibrium, but it dramatically reduces the chance of failure of the iterative posture solution.

12.5 Stable versus Unstable Equilibrium

If the Newton-Raphson search for equilibrium is applied exactly as described in sections 12.2, and 12.3, it can lead to other unexpected situations. As an example, consider the slider crank system shown in Fig. 12.1a. With a large force P applied on the piston as shown and with gravity acting perpendicular to the plane of the page, we would probably expect to find an equilibrium posture with the piston translated to the far left and with the crank near the 180° position, as shown in Fig. 12.1b. Without modification, however, our software would actually move the piston to the right and would find equilibrium at a crank angle of very nearly zero!

How could this be? What would cause movement to the right when the force P appears to demand movement to the left? Well, if we carefully reconsider what the Newton-Raphson search procedure was asked to do, we see that it was only asked to find a nearby posture with F_i equal to zero. It was *not* told to move in the direction implied by the unbalanced forces. How can this be cured? First, if movement is in the “wrong direction,” how can this be detected?

We also know that, for a posture of stable equilibrium, potential energy should become a minimum. Does this result from our Newton-Raphson search procedure? Again, our procedure was not formulated to assure this.

² The IMP software uses a default limit of $|\delta\psi| \leq 1.0$ (radian or length unit) for this test. The user may override this limit if desired.

If we expand potential energy in Taylor series, we see

$$V = V^* + \sum_{i=1}^f \frac{\partial V}{\partial \psi_i} \delta \psi_i + \dots, \quad (12.6)$$

and, from the Lagrange equations, Eqs. (10.1), we know that, under static conditions,

$$\frac{\partial V}{\partial \psi_i} = F_i, \quad i = 1, 2, \dots, f. \quad (12.7)$$

Therefore, using Eq. (12.6), we can write

$$\delta V = V - V^* = F^t \delta \psi \leq 0. \quad (12.8)$$

This shows that the dot product of the vector of generalized unbalanced forces and the vector of generalized position increments found from Eq. (12.4) tell us whether the system's potential energy will increase or decrease during such a change of posture. Thus, we have found a very convenient test to tell whether our search is progressing toward or away from a posture of less potential energy. However, this test does not show how to cure the problem when the test shows that potential energy is increasing.

However, looking again at Eqs. (12.7), we see that the generalized unbalanced force vector F is equal to the *gradient* of the potential energy function, of which we hope to find a minimum. Also, an abundance of numerical methods exist in the literature that can generally be categorized as *steepest descent* methods; for example, see [1], section 10.6. All of these require that, from some starting point, the search for the minimum of a function should proceed in the direction of the negative gradient of the function. For our situation, this gives

$$\psi_i = \psi_i^* - \mu F_i, \quad i = 1, 2, \dots, f, \quad (12.9)$$

where μ controls the size of the vector increment of the generalized positions.

The increment size μ may be chosen in a variety of ways depending on which search method is chosen, and much has been written about the advantages and tradeoffs of the many methods available, and their rates of convergence toward a solution. However, in our specific application, we already know that once we find the neighborhood of a valid solution, the Newton-Raphson method of Eqs. (12.4) and (12.5) exhibits quadratic convergence and none of the steepest descent methods improve on this. Therefore, we only need Eq. (12.9) to avoid stepping toward a false solution; that is, toward a posture of unstable equilibrium, as indicated by failure of the test of inequality in Eq. (12.8).

Our newly modified algorithm starts each search increment by forming Eq. (12.4). Next, the inequality (12.8) is tested and, if this test is passed, the generalized coordinates are incremented according to Eq. (12.5). For those situations where the inequality test fails, however, Eq. (12.9) is applied instead with an appropriate value of μ . The software might use a default step size such as $\mu = 1.0/|F|$. This gives the system a noticeable change in posture (particularly because rotational generalized coordinates are measured in radians) and, hopefully, the search will then continue toward a posture of stable equilibrium by Eq. (12.4).

12.6 Postures of Neutral Equilibrium

Another problem that can be encountered in seeking equilibrium is that, as we apply the previous procedure, we may reach a posture where the stiffness matrix becomes singular or nearly singular and Eq. (12.4) cannot be applied. Let us discuss how this can occur and what can be done in such a situation.

Let us consider the portion of Eq. (12.3) that corresponds to the FGCs alone, as discussed in section 12.3. The equations start in the form

$$[K]\{\delta\psi\} = \{F\}.$$

Let us suppose that we employ a Gauss-Jordan row-reduction process with full pivoting for the solution. As we proceed, the augmented K matrix evolves at each step as follows:

$$\begin{bmatrix} K_{11}^{-1} & 0 \\ -K_{21}K_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} K_{11} & K_{12} & | & F_1 \\ K_{21} & K_{22} & | & F_2 \end{bmatrix} = \begin{bmatrix} I & K_{11}^{-1}K_{12} & | & K_{11}^{-1}F_1 \\ 0 & K_{22} - K_{21}K_{11}^{-1}K_{12} & | & F_2 - K_{21}K_{11}^{-1}F_1 \end{bmatrix}.$$

We seek out each new pivot element from the lower-right portion of the matrix, the portion that contains $[K_{22} - K_{21}K_{11}^{-1}K_{12}]$, and attempt to continue the reduction. If at some step we find that all elements of $[K_{22} - K_{21}K_{11}^{-1}K_{12}]$ are too small to consider,³ then the reduction process must cease. At such a step, the equations have been reduced to the form

$$\begin{bmatrix} I & K_{11}^{-1}K_{12} \\ 0 & \varepsilon \end{bmatrix} \begin{Bmatrix} \delta\psi_1 \\ \delta\psi_2 \end{Bmatrix} = \begin{Bmatrix} K_{11}^{-1}F_1 \\ F_2 - K_{21}K_{11}^{-1}F_1 \end{Bmatrix}.$$

That subset of the FGCs that, as the result of pivoting, has ended in the subgroup $\{\delta\psi_2\}$ can now be treated in either of two ways: we can set these $\{\delta\psi_2\}$ to zero in the hope that they may be corrected in the next search step, or we can use the corresponding unbalanced generalized forces $\{F_2 - K_{21}K_{11}^{-1}F_1\}$ and Eq. (12.9) for that subgroup and then use the solution of the upper equations; that is $\{\delta\psi_1\} = \{K_{11}^{-1}F_1 - K_{11}^{-1}K_{12}\delta\psi_2\}$ for the others.

Once the search for equilibrium concludes, any generalized coordinates that remain in the subgroup $\{\delta\psi_2\}$ of the FGCs are said to be in *neutral* equilibrium. If Eq. (12.9) is used as the basis for its increment, it is important that such steps be counted and an upper limit set. As an example, suppose that our vehicle with the load in its luggage compartment is postured on a smooth horizontal road and suppose that, as the result of numeric truncation error, a very small unbalanced load is calculated in the forward direction even though there is no fore and aft stiffness for the vehicle; then Eq. (12.9) will produce a small forward movement of the vehicle. Next, the same unbalanced load is found again, and another small movement results. When this happens repeatedly, no progress is made that can relieve this deadly embrace and an endless loop is the result. Thus, an increment counter with an upper limit is of critical importance.⁴

³ The IMP software uses $\varepsilon = 10^{-5}$ for this test. The user can override this limit if desired.

⁴ A default upper limit of 25 steps is used by the IMP software for this test. The user can override this limit if desired.

REFERENCE

1. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Fetterling, *Numerical Recipes, the Art of Scientific Computing*, Cambridge University Press, 1986.

PROBLEM

12.1 Continue problem 11.1 and find the equation relating W and ψ such that the scale is in equilibrium at each posture. Plot a graph of W versus ψ under these conditions for the range $-85^\circ \leq \psi \leq 0^\circ$.

13 Frequency Response of Mechanisms and Multibody Systems

13.1 Introduction

When a mechanism or multibody system such as a rotating machine powered by an electric motor or an internal combustion engine is driven cyclically, it may operate with periodic motion, but not usually with harmonic motion. Such a system is not characterized by the linearized equations of [Chapter 11](#) because its kinematic parameters and its dynamic equation coefficients change throughout the cycle of operation. For these or other situations where the linearized equations of motion do not pertain, the entire field of frequency response is not applicable and the methods of this chapter should not be used.

However, there are situations where a mechanism or multibody system operates over a limited amplitude range and is described quite well by linearized equations of motion with approximately constant dynamic coefficients. Examples include the vibratory motion of vehicle suspensions, belt-tensioning idler pulleys, or many vibration isolation systems. These and other situations, where the linearized equations of motion do pertain, are the focus of this chapter.

13.2 Homogeneous First-order Equations of Motion

As early as 1750, Daniel Bernoulli (1700–82) [1] – the son of Johann Bernoulli (1667–1748), Leonhard Euler (1707–83) [3], and Joseph Louis Lagrange (1736–1813) [6], had been using trigonometric series to represent periodic functions, but it was Marquis Pierre Simon Laplace (1740–1827) [7] and Baron Jean Baptiste Joseph Fourier (1768–1830) [4] who showed how these can be used in the solution of linear differential equations.

Let us start by considering the linearized equations of motion of our mechanical system. Equation (11.13) for the free generalized coordinates is

$$M_{FF}\dot{v}_F + D_{FF}v_F + K_{FF}x_F = G_F, \quad (13.1)$$

and these must be solved along with Eq. (11.2) to ensure that

$$v_F = \dot{x}_F - \dot{\psi}_F^*. \quad (13.2)$$

Here we consider only the equations dealing with the $N\mathcal{F}$ free generalized coordinates (FGCs) because the motions of the specified generalized coordinates (SGCs) are known a priori.

Because we believe that our system exhibits periodic motion for its transient response to a disturbance, we attempt to use a harmonic form for a trial solution

$$x_F = \{\xi\} e^{\lambda t},$$

and, from Eq. (13.2),

$$v_F = \lambda \{\xi\} e^{\lambda t} - \dot{\psi}_F^*.$$

Substituting these into Eq. (13.1), taking the homogeneous part, and dividing by the common factor of $e^{\lambda t}$ leaves

$$[\lambda^2 M_{FF} + \lambda D_{FF} + K_{FF}] \xi = 0. \quad (13.3)$$

However, for this equation to have a nontrivial solution, the coefficient matrix must be singular:

$$\det[\lambda^2 M_{FF} + \lambda D_{FF} + K_{FF}] = 0.$$

Because this determinant is of size $(N\mathcal{F} \times N\mathcal{F})$ with elements that are each quadratic functions of λ , it produces a polynomial of degree $2N\mathcal{F}$ in λ . Assumedly, the $2N\mathcal{F}$ complex values of λ that satisfy as roots of this equation could be found by a numerical search. This equation could have been studied in depth but, historically, this was not to be. Instead, the following parallel linear form shown in Eq. (13.9) has received much more attention.

Instead, we return to Eq. (11.13), including the applied forces. Augmenting these with Eq. (13.2), in the form of the identity $M_{FF}\dot{x}_F - M_{FF}v_F = M_{FF}\dot{\psi}_F^*$, this set of differential equations can be written in the form

$$\begin{bmatrix} 0 & M_{FF} \\ M_{FF} & 0 \end{bmatrix} \begin{Bmatrix} \dot{v}_F \\ \dot{x}_F \end{Bmatrix} - \begin{bmatrix} M_{FF} & 0 \\ -D_{FF} & -K_{FF} \end{bmatrix} \begin{Bmatrix} v_F \\ x_F \end{Bmatrix} = \begin{Bmatrix} M_{FF}\dot{\psi}_F^* \\ G_F \end{Bmatrix}.$$

However, by adding $D_{FF}\dot{x}_F - D_{FF}v_F = D_{FF}\dot{\psi}_F^*$, which is another variation of Eq. (13.2), to the lower set of equations, they become

$$\begin{bmatrix} 0 & M_{FF} \\ M_{FF} & D_{FF} \end{bmatrix} \begin{Bmatrix} \dot{v}_F \\ \dot{x}_F \end{Bmatrix} - \begin{bmatrix} M_{FF} & 0 \\ 0 & -K_{FF} \end{bmatrix} \begin{Bmatrix} v_F \\ x_F \end{Bmatrix} = \begin{Bmatrix} M_{FF}\dot{\psi}_F^* \\ D_{FF}\dot{\psi}_F^* + G_F \end{Bmatrix}. \quad (13.4)$$

This modification is important because it makes the coefficient matrices symmetric, and still follows the linearization assumptions of Chapter 11. Notice also that, even though \dot{v}_F and \dot{x}_F may be large, v_F and x_F are both small.

Defining new coefficient matrices,

$$A = \begin{bmatrix} 0 & M_{FF} \\ M_{FF} & D_{FF} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} M_{FF} & 0 \\ 0 & -K_{FF} \end{bmatrix}, \quad (13.5)$$

and a new vector of unknowns,

$$y = \begin{Bmatrix} v_F \\ x_F \end{Bmatrix}, \quad (13.6)$$

of size $2NF$, our system of Eqs. (13.4) is now described by a first-order set of differential equations

$$A\dot{y} - By = \begin{Bmatrix} M_{FF}\dot{\psi}_F^* \\ D_{FF}\dot{\psi}_F^* + G_F \end{Bmatrix}, \quad (13.7)$$

with the initial conditions that when $t = 0$, we must have $y = 0$. The vector space for this form, which treats v_F and x_F as independent, is referred to as “*state space*.”

We take notice that because the M_{FF} , D_{FF} , and K_{FF} coefficient matrices of the original equations are real and symmetric, the form of the definitions of Eqs. (13.5) show that the new A and B matrices are also real and symmetric. This becomes important following Eq. (13.11).

As we did previously, we attempt a harmonic form for a trial solution

$$y = \{\eta\} e^{\lambda t}. \quad (13.8)$$

Substituting this and its derivative into the homogeneous part of Eq. (13.7) and dividing by the common factor of $e^{\lambda t}$, leaves

$$[\lambda A - B]\eta = 0. \quad (13.9)$$

In the form shown, where the coefficients are linear expressions in λ , Eq. (13.9) is called the *generalized eigenvalue problem*. Alternatively, if pre-multiplied by A^{-1} , Eq. (13.9) can be put into the form

$$[\lambda I - (A^{-1}B)]\{\eta\} = \left[\lambda I - \begin{pmatrix} -M_{FF}^{-1}D_{FF} & -M_{FF}^{-1}K_{FF} \\ I & 0 \end{pmatrix} \right] \{\eta\} = 0, \quad (13.10)$$

which is called simply the *eigenvalue problem*. This equation has been studied extensively over many years, and software for its solution exists in almost all numerical software libraries [8]. As one example of such software, EISPACK is a public domain collection of FORTRAN subroutines for computing the eigenvalues and/or eigenvectors of various types of matrices. It includes software for both Eq. (13.9) and Eq. (13.10). EISPACK was developed with the support of the National Science Foundation (NSF) in the mid-1960s and was one of the first completely systematized collections of linear algebra software. Since that time, however, much has changed in computer architectures and mathematical algorithms, and a newer linear algebra package, LAPACK, has been developed to supercede EISPACK. For further information, see <http://www.netlib.org/lapack/>.

For either form of the equation to have a nontrivial solution, the coefficient matrix must be singular. That is, starting from Eq. (13.9),

$$\det[\lambda A - B] = 0. \quad (13.11)$$

The expansion of this determinant yields a polynomial of degree $2NF$ in λ that is called the *characteristic* or *secular equation* of the system. This equation can be

solved for $2NF$ values of λ , which are called *characteristic values* or *eigenvalues* of the system. These values of λ , in general, are complex. However, because the A and B matrices are real and symmetric in this case, it has been proven [5] that the eigenvalues must either be real or must occur in complex conjugate pairs.

Assuming, as we now do, that all eigenvalues are distinct, they can be arranged to form a diagonal matrix:

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_{2NF} \end{bmatrix}.$$

Extension to the case of repeated eigenvalues is presented in section 14.3.

Each of the eigenvalues λ_i , in turn, can be back-substituted into either Eq. (13.9) or Eq. (13.10), as appropriate, and solved for a complex column vector η_i called an *eigenvector*. However, recalling Eq. (13.11), we know that the coefficient matrix of Eq. (13.9) or Eq. (13.10) is singular; therefore, each eigenvector can only be determined to the nearest arbitrary multiplying constant. However, once found, these eigenvectors can be arranged in the same order as the diagonal elements of A to form columns of another ($2NF \times 2NF$) complex matrix,

$$\eta = [\eta_1 \quad \cdots \quad \eta_i \quad \cdots \quad \eta_{2NF}], \quad (13.12)$$

which is called the *modal* matrix.

When we reconstruct the general form of Eq. (13.8), with all eigenvalues and eigenvectors included, we get the full solution to the homogeneous linearized equations of motion

$$y = \eta \exp(A\boldsymbol{\tau})C \quad (13.13)$$

where $\exp(A\boldsymbol{\tau})$ is a diagonal matrix with values of $e^{\lambda_i \boldsymbol{\tau}}$ as diagonal entries, $\boldsymbol{\tau}$ is the time interval following the instant of linearization, and C is a column of $2NF$ yet unknown complex constants that depend on the initial conditions. Note that these constants compensate for the ability to determine each eigenvector to only an arbitrary multiplying constant.

Even though we have manipulated Eq. (13.9) into the form called the generalized eigenvalue problem so that we can take advantage of available software for solution, we should understand that the eigenvalues λ_i that are found must be the same as if we had continued with the solution of Eq. (13.3) because the same physical system, with the same characteristic frequencies and damping rates, is described in each case.

13.3 Modal Coordinates

We will return to the completion of the solution of Eq. (13.13) in Chapter 14. However, in this chapter, as the title specifies, we wish to study the frequency response of our mechanism or multibody system to a harmonic external disturbance. In order to do this we have need for the eigenvalues and eigenvectors of the system. That has been our reason for their introduction in the preceding section.

Before completing our solution in the time domain in the next chapter, let us consider further the implications of the form of Eq. (13.13). Suppose that we define a new column vector of coordinates z , of size $2NF$, called *modal coordinates*, with the following definition

$$y = \eta z. \quad (13.14)$$

Although this might seem like a reverse manner for making a definition, recalling Eq. (13.13) shows that this form is advantageous because it gives

$$z = \exp(\Lambda t) C.$$

This means that, if only one of the constants – C_i – is nonzero, then only that single modal coordinate z_i becomes active and the entire system responds with frequency and damping defined by the i th eigenvalue λ_i alone, whereas all other modal coordinates remain motionless (because eigenvalues occur in complex conjugate pairs, the ideas here must be understood as referring to a single real eigenvalue or a single complex conjugate pair). However, even more, if we now return to Eq. (13.14), we see that each of the y coordinates – the NF FGC displacements and velocities – respond at the same characteristic normal mode frequency and damping rate, but with different amplitudes defined by the elements of η_i , the i th eigenvector. When this happens, the system is said to be operating in the i th of its *principal* or *normal modes*.

Next let us consider two copies of Eq. (13.9): one corresponding to the j^{th} eigenvalue and the other corresponding to the k^{th} eigenvalue:

$$\begin{aligned} [\lambda_j A - B] \eta_j &= 0, \\ [\lambda_k A - B] \eta_k &= 0. \end{aligned}$$

Let us pre-multiply the first of these by η_k^t and the second by η_j^t :

$$\begin{aligned} \eta_k^t [\lambda_j A - B] \eta_j &= 0, \\ \eta_j^t [\lambda_k A - B] \eta_k &= 0. \end{aligned}$$

If we now subtract the second of these equations from the transpose of the first, remembering that A and B are symmetric, we obtain

$$(\lambda_j - \lambda_k) \eta_j^t A \eta_k = 0. \quad (13.15)$$

This equation says that, for $j \neq k$, because we have assumed that $\lambda_j \neq \lambda_k$, we must have $\eta_j^t A \eta_k = 0$ (extension to the case of repeated eigenvalues is presented in section 14.3). Furthermore, let us consider the form

$$\eta^t [\Lambda A - B] \eta = 0. \quad (13.16)$$

If we define two new matrices,

$$\mathcal{A} = \eta^t A \eta \quad \text{and} \quad \mathcal{B} = \eta^t B \eta,$$

then Eq. (13.15) shows that \mathcal{A} must be a diagonal matrix because it has zeroes for all off-diagonal elements where $j \neq k$. Also, from Eq. (13.16), \mathcal{B} must be diagonal as well. Therefore, Eq. (13.16) is referred to as a set of orthogonality conditions. We

see that the modal matrix η is not orthogonal in the usual sense because η^{-1} may not equal η^t , but η is said to be *orthogonal with respect to both A and B*. As shown by Eqs. (13.15) and (13.16) it *diagonalizes* both A and B .

Recalling that each of the eigenvectors is only determined to the nearest multiplying constant, each can now be scaled by an arbitrary constant. Therefore, we are free to divide each eigenvector η_i by $\sqrt{\eta_i^t A \eta_i}$, thus scaling them so that \mathcal{A} becomes equal to the identity matrix. Then Eq. (13.16) shows that $\mathcal{B} = \Lambda$. If we consider Eq. (13.5), we see that the only way in which this scale factor can be zero is if mass has been ignored for enough bodies that $[M_{FF}]$ is singular; that is, such that it is possible to find a set of nonzero velocities for the *FGCs* for which the system has no kinetic energy. If this occurs, it is considered a modeling error; the user should be notified and requested to modify the model with appropriate mass values.

If we also pre-multiply Eq. (13.7) by η^t and recall Eq. (13.14), we find that the equations of motion, in modal coordinates, become

$$\dot{z} - \Lambda z = \eta^t \left\{ \begin{array}{c} M_{FF} \dot{\psi}_F^* \\ D_{FF} \dot{\psi}_F^* + G_F \end{array} \right\}. \quad (13.17)$$

Here we see that we have a set of $2N_F$ first-order differential equations. However, because Λ is diagonal, they are now *uncoupled* and can be considered either together, as in Eq. (13.17), or separately, as

$$\dot{z}_i - \lambda_i z_i = \eta_i^t \left\{ \begin{array}{c} M_{FF} \dot{\psi}_F^* \\ D_{FF} \dot{\psi}_F^* + G_F \end{array} \right\}, \quad i = 1, 2, \dots, 2N_F.$$

13.4 Laplace Transformed Equations of Motion

Instead of completing our solution in the time domain, let us now switch to the frequency domain by taking the Laplace transform of our equations of motion. Remembering the definition of G_F from Eq. (11.13), the Laplace transform of Eq. (13.17) is

$$[sz(s) - z_0] - \Lambda z(s) = \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + F_F(s) - M_{FS} [s^2 x_S(s) - s x_{S0} - \dot{x}_{S0}] - D_{FS} [s x_S(s) - x_{S0}] - K_{FS} x_S(s) \end{array} \right\},$$

where s represents the Laplace transform variable, x_{S0} and \dot{x}_{S0} are the initial positions and velocities of the *SGCs*, z_0 are the initial values of the modal coordinates, and $z(s)$, $F_F(s)$, and $x_S(s)$ are the Laplace transforms of the modal coordinates, the generalized applied forces, and the *SGC* displacements, respectively.

Remembering that the initial conditions require that $x_{S0} = 0$ and $\dot{x}_{S0} = \dot{\psi}_S^*$, this equation can be rearranged to read

$$[sI - \Lambda] z(s) = z_0 + \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + M_{FS} \dot{\psi}_S^* + F_F(s) - [s^2 M_{FS} + s D_{FS} + K_{FS}] x_S(s) \end{array} \right\}.$$

However, for what is usually called frequency response, we are only interested in the sustained response of our system after the transients have been damped to zero. Therefore, we eliminate the z_0 initial condition terms of the previous equation:

$$[sI - \Lambda] z(s) = \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + M_{FS} \dot{\psi}_S^* + F_F(s) - [s^2 M_{FS} + sD_{FS} + K_{FS}] x_S(s) \end{array} \right\}.$$

The solution to this equation can be written

$$z(s) = [sI - \Lambda]^{-1} \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + M_{FS} \dot{\psi}_S^* + F_F(s) - [s^2 M_{FS} + sD_{FS} + K_{FS}] x_S(s) \end{array} \right\},$$

and, from the definition of z in Eq. (13.14), this becomes

$$y(s) = \eta [sI - \Lambda]^{-1} \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + M_{FS} \dot{\psi}_S^* + F_F(s) - [s^2 M_{FS} + sD_{FS} + K_{FS}] x_S(s) \end{array} \right\},$$

which, along with Eqs (13.6), gives

$$\begin{aligned} & \left\{ \begin{array}{c} v_F(s) \\ x_F(s) \end{array} \right\} \\ &= \eta [sI - \Lambda]^{-1} \eta^t \left\{ \begin{array}{c} \frac{1}{s} M_{FF} \dot{\psi}_F^* \\ \frac{1}{s} D_{FF} \dot{\psi}_F^* + F_F(s) + M_{FS} \dot{\psi}_S^* - [s^2 M_{FS} + sD_{FS} + K_{FS}] x_S(s) \end{array} \right\}. \end{aligned} \quad (13.18)$$

Remembering that as long as the eigenvalues are distinct, the matrix $[sI - \Lambda]$ is diagonal and can be inverted term by term, we find that, in index notation, Eq. (13.18) becomes

$$\begin{aligned} x_i(s) = & \sum_{k=1}^{2NF} \frac{\eta_{NF+i,k}}{(s - \lambda_k)} \sum_{j=1}^{NF} \left\{ \frac{\eta_{jk}}{s} \left[\sum_{h=1}^{NF} M_{jh} \dot{\psi}_h^* \right] + \frac{\eta_{NF+j,k}}{s} \left[\sum_{h=1}^{NF} D_{jh} \dot{\psi}_h^* \right] \right. \\ & \left. + \eta_{NF+j,k} \left[F_j(s) + \sum_{h=NF+1}^f M_{jh} \dot{\psi}_h^* - \sum_{h=NF+1}^f (s^2 M_{jh} + sD_{jh} + K_{jh}) x_h(s) \right] \right\}, \\ & i = 1, 2, \dots, NF. \end{aligned} \quad (13.19)$$

13.5 Frequency Response

Equations (13.18) and (13.19) show the Laplace transform of the response of the FGCs to all generalized applied forces $F_F(s)$ and SGC motions $x_S(s)$ acting on the system simultaneously. If we choose a particular one of these as the input of interest, we can express the Laplace transform of the response of our system to excitation from that particular source.

For example, if we assume that the only input is an excitation coming from a particular generalized force F_p alone, then the Laplace transform of the response of FGC x_i to force F_p is

$$x_{i,p}(s) = \sum_{k=1}^{2NF} \frac{\eta_{NF+i,k}}{(s - \lambda_k)} \left\{ \sum_{j=1}^{NF} \left[\frac{\eta_{jk}}{s} \sum_{h=1}^{NF} M_{jh} \psi_h^* + \frac{\eta_{NF+j,k}}{s} \sum_{h=1}^{NF} D_{jh} \psi_h^* + \eta_{NF+j,k} \sum_{h=NF+1}^f M_{jh} \psi_h^* \right] + \eta_{NF+p,k} F_p(s) \right\}, \quad i, p = 1, 2, \dots, NF. \quad (13.20)$$

However, if the excitation comes from the motion of SGC x_p alone, then the Laplace transform of the response of FGC x_i is

$$x_{i,p}(s) = \sum_{k=1}^{2NF} \frac{\eta_{NF+i,k}}{(s - \lambda_k)} \sum_{j=1}^{NF} \left\{ \left[\frac{\eta_{jk}}{s} \sum_{h=1}^{NF} M_{jh} \psi_h^* + \frac{\eta_{NF+j,k}}{s} \sum_{h=1}^{NF} D_{jh} \psi_h^* + \eta_{NF+j,k} \sum_{h=NF+1}^f M_{jh} \psi_h^* \right] - \eta_{NF+j,k} [s^2 M_{jp} + s D_{jp} + K_{jp}] x_p(s) \right\}, \quad \begin{array}{l} i = 1, 2, \dots, NF \\ p = NF + 1, \dots, f \end{array} \quad (13.21)$$

Both of these sets of equations give the Laplace transforms of the motions of the FGCs to specified excitations. However, it is much easier for the analyst to judge their physical meaning if they are shown as functions of frequency, rather than as Laplace transforms. The format called frequency response of either of these equations can be determined by setting the Laplace operator s equal to $j\omega$ [2] where $j = \sqrt{-1}$. This results in a complex function of frequency ω for the chosen equation. The magnitude and phase angle of the function can then be plotted versus frequency in graphs called *Bode plots*. These plots are quite useful to a controls engineer in judging the response, and in determining how to control it.

REFERENCES

1. D. Bernoulli, "Réflexions et éclaircissement sur les nouvelles vibrations des corde, (Reflection and clarification on natural vibration of strings)," *Memories de l'Academie Royale des Sciences et Belles Letters de Berlin (Memoirs of the Royal Academy of Sciences and Letters at Berlin)*, vol. 9, Berlin, 1753, pp. 147–72.
2. J. G. Bollinger and N. A. Duffie, *Computer Control of Machines and Processes*, Addison-Wesley Publishing Co., 1988.
3. L. Euler, *Nova Acta Academiae Scientiarum Petropropolitanae, (New acts of the Academy of Sciences in St. Petersburg)*, vol. 5, 1754–55, pp. 164–204.
4. J. B. J. Fourier, *Théorie analytique de la chaleur, (The analytical theory of heat)*, vol. 1, Firmin Didot, Paris, 1822.
5. F. B. Hildebrand, *Methods of Applied Mathematics*, Prentice-Hall, Inc., 2nd ed., 1965.
6. J. L. Lagrange, *Miscellanea Taurinensia ou Mélanges de Turin (Turin Science Review)*, vol. 1, Turin Mathematical Society, 1759.
7. P. S. Laplace, *Traité de Mécanique Céleste, (Treatise on Celestial Mechanics)*, Paris: Crapelet for Duprat, 1799.
8. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965.

PROBLEMS

13.1 Continue from the results of problem 11.1 and find the eigenvalues and eigenvectors of the spring scale under study when in equilibrium under a constant load W in the posture of $\psi = -30^\circ$.

13.2 Continue from the results of problem 13.1 and make a Bode plot for small oscillations around this posture.

14 Time Response of Mechanisms and Multibody Systems

14.1 Inverse Laplace Transform

The overall purpose of this chapter is to complete the solutions to the equations of motion of our dynamic system for the unknown motions of the FGCs as functions of time. With these and the methods of the previous chapters, the overall motion of the entire system becomes known.

There are two sets of equations for which we have yet to find final solutions: 1) the modal amplitude equations have not yet been integrated to provide explicit functions of time, and 2) the Laplace transform solutions are still in the frequency domain and have not yet been transformed back to the time domain. In this chapter, we hope to finish both of these solutions.

Chapter 13 shows the Laplace transform of the response of the FGCs of our system to any generalized applied forces $F_F(s)$ and SGC motions $x_S(s)$ that may be acting on the system. Remembering Eq. (13.19), we have

$$x_i(s) = \sum_{h=1}^{2NF} \frac{\eta_{NF+i,h}}{(s - \lambda_h)} \sum_{j=1}^{NF} \left\{ \frac{\eta_{jh}}{s} \left[\sum_{k=1}^{NF} M_{jk} \dot{\psi}_k^* \right] + \frac{\eta_{NF+j,h}}{s} \left[\sum_{k=1}^{NF} D_{jk} \dot{\psi}_k^* \right] \right. \\ \left. + \eta_{NF+j,h} \left[F_j(s) + \sum_{k=NF+1}^f M_{jk} \dot{\psi}_k^* - \sum_{k=NF+1}^f (s^2 M_{jk} + sD_{jk} + K_{jk}) x_k(s) \right] \right\}, \\ i = 1, 2, \dots, NF. \quad (14.1)$$

We must now consider how we will invert this Laplace transform and thereby return our solution to the time domain. By definition, the inverse of any Laplace transform can be written as follows:

$$x_i(t) = \lim_{\sigma \rightarrow 0} \left[\frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} x_i(s) e^{st} ds \right], \quad i = 1, 2, \dots, NF, \quad (14.2)$$

which is called the *Bromwich integral* after Thomas John l'Anson Bromwich (1875–1929), Saint John's College, Cambridge, England, or, sometimes, the *Fourier-Mellin integral* after Robert Hjalmar Mellin (1854–1933), first professor of mathematics, Technical University of Finland.

For a function such as $x_i(s)$ that is defined over the complex s plane, a location where the denominator becomes zero produces an infinite value and is called a *pole*

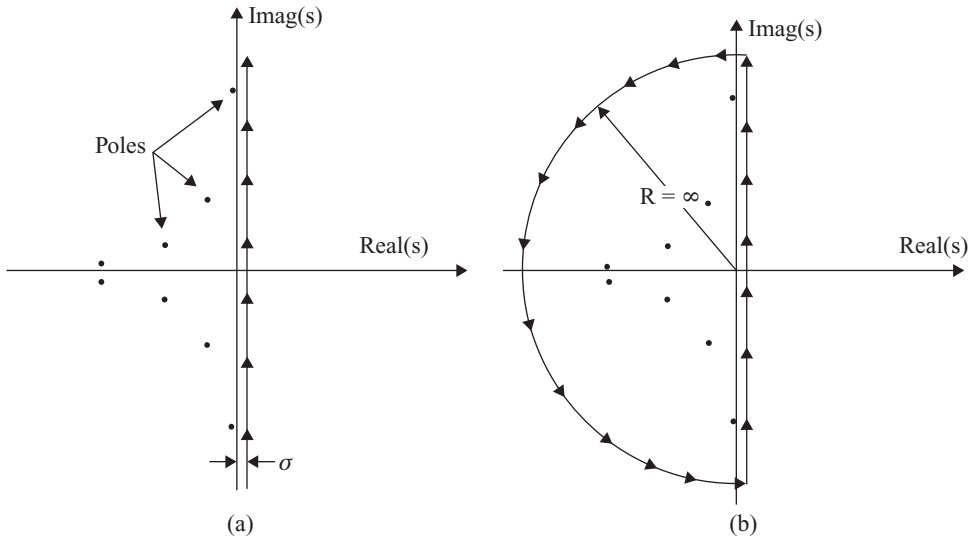


Figure 14.1. (a) Integration path for Bromwich integral; (b) Jordan curve.

of the function. As shown in Figure 14.1a, Eq. (14.2) is a line integral in the complex s plane taken along a line parallel to the imaginary axis and taken to the right of all poles of the integrand.

It is possible to evaluate an approximate value for this integral by numeric integration; however, there is a better way that computes more quickly and avoids approximation. The line integral of Eq. (14.2) can be extended along a semi-circular arc at infinity surrounding the left half-plane as shown in Figure 14.1b, thus forming a simple-closed curve called a *Jordan curve* [5], named after French mathematician, Marie Ennemond Camille Jordan (1838–1922), which encloses all of the poles.

It has been shown [2] that the value of the portion of the integral of Eq. (14.2) along the semi-circular arc at infinity is zero and does not change the value of the total integral. Therefore, the integral sought is equal to the integral around the closed curve shown in Figure 14.1b, where σ is chosen large enough that the curve surrounds all poles of $x_i(s)$

$$x_i(\mathbf{t}) = \frac{1}{2\pi j} \oint x_i(s)e^{s\mathbf{t}} ds, \quad i = 1, 2, \dots, NF.$$

Remembering the definition of G_F from Eq. (11.13), and substituting Eq. (14.1), this integral becomes

$$x_i(\mathbf{t}) = \frac{1}{2\pi j} \oint \sum_{h=1}^{2NF} \frac{\eta_{NF+j,h}}{(s - \lambda_h)} \sum_{j=1}^{NF} \left\{ \frac{\eta_{jh}}{s} \left[\sum_{k=1}^{NF} M_{jk} \psi_k^* \right] + \frac{\eta_{NF+jh}}{s} \left[\sum_{k=1}^{NF} D_{jk} \psi_k^* \right] \right. \\ \left. + \eta_{NF+j,h} \left[\sum_{k=NF+1}^f M_{jk} \psi_k^* \right] + \eta_{NF+j,h} G_j(s) \right\} e^{s\mathbf{t}} ds, \quad i = 1, 2, \dots, NF. \quad (14.3)$$

14.2 Cauchy's Residue Theorem

The integrand of our equation is an analytic function of the complex operator s throughout the region of the complex plane enclosed by the Jordan curve of Figure 14.1b, except at isolated singular points where the denominator becomes zero, such as the origin and the $2NF$ points where $s = \lambda_h$. Additional singularities may occur where there are zeroes in the denominators of the forcing functions, $G_j(s)$. Let us require that the forcing functions be replaced by partial fraction expansions [7] and, thereby, expressed in the form

$$G_j(s) = \sum_{k=1}^{NP} \frac{g_{jk}(s)}{(s - \lambda_{2NF+k})}, \quad j = 1, 2, \dots, NF, \quad (14.4)$$

where NP represents the number of additional values of λ_h , which are identified as zeroes of the denominators of Eq. (14.4) and, through these, are poles of Eq. (14.3). Let us also require that these additional poles be numbered consecutively to follow the numbering of the eigenvalues. In keeping with our current assumption that the eigenvalues of our system are distinct, we assume, temporarily, that the additional poles of the forcing functions are distinct from each other, from the origin, and from the eigenvalues. Thus, all $(2NF + NP + 1)$ poles of the response function, we assume for now, are *simple poles*. The extension to the case of multiple poles is shown in section 14.3.

When a function $f(s)$ has an isolated singular point (a pole) at the location in the complex plane where $s = \lambda_m$, then the value defined by

$$\text{Res}_f(\lambda_m) = [(s - \lambda_m)f(s)]|_{s=\lambda_m}, \quad m = 1, 2, \dots, 2NF + NP + 1, \quad (14.5)$$

is called the *residue* of the function at that location. With this definition, the Cauchy residue theorem, named after the French mathematician, Baron Augustin Louis Cauchy (1789–1857) [1], states that the integral of a complex function around a closed curve is equal to $2\pi j$ (where $j = \sqrt{-1}$) times the sum of the residues of the function at all poles surrounded by the curve.¹ Where there are only simple poles, this theorem (see [2, section 67]) says that

$$\oint f(s)ds = 2\pi j \sum_m \text{Res}_f(\lambda_m) = 2\pi j \sum_m \{(s - \lambda_m)f(s)\}|_{s=\lambda_m}. \quad (14.6)$$

For our situation, the residue theorem can be used to replace the integral of Eq. (14.3) with a summation

$$\begin{aligned} x_i(\boldsymbol{\tau}) = & \frac{2\pi j}{2\pi j} \sum_{h=1}^{2NF} \sum_{j=1}^{NF} \sum_{k=1}^{NF} \frac{\eta_{NF+i,h}}{-\lambda_h} (\eta_{j,h} M_{jk} + \eta_{NF+j,h} D_{jk}) \dot{\psi}_k^* \\ & + \frac{2\pi j}{2\pi j} \sum_{m=1}^{2NF+NP} \sum_{h=1}^{2NF} \sum_{j=1}^{NF} \left[\frac{(s - \lambda_m)\eta_{NF+i,h}}{(s - \lambda_h)s} \sum_{k=1}^{NF} (\eta_{j,h} M_{jk} + \eta_{NF+j,h} D_{jk}) \dot{\psi}_k^* \right] \end{aligned}$$

¹ There is a (bad) joke about a mathematician who named his dog Cauchy because it left a residue at every pole.

$$\begin{aligned}
 & + \frac{(s - \lambda_m)\eta_{NF+i,h}\eta_{NF+j,h}}{(s - \lambda_h)} \sum_{k=NF+1}^f M_{jk}\psi_k^* \\
 & + \left. \frac{(s - \lambda_m)\eta_{NF+i,h}\eta_{NF+j,h}}{(s - \lambda_h)} \sum_{k=1}^{NP} \frac{g_{jk}(s)}{(s - \lambda_{2NF+k})} \right] e^{st} \Big|_{s=\lambda_m}, \quad i = 1, 2, \dots, NF,
 \end{aligned}
 \tag{14.7}$$

where the first line results from the pole at the origin.

Careful evaluation shows that when the λ_m value is taken for s in the terms of the second and third lines, the numerator produces zero for the $(s - \lambda_m)$ factor for every term of the summation over m except when $m = h$ or when $m = 2NF + k$. For those particular terms, the $(s - \lambda_m)$ factor in the numerator, which evaluates to zero for most terms, is balanced by one of the factors in the denominator. Also, the first line combines with the first set of terms of the second line, leaving

$$\begin{aligned}
 x_i(\mathbf{z}) &= \sum_{h=1}^{2NF} \frac{\eta_{NF+i,h}}{\lambda_h} \sum_{j=1}^{NF} \sum_{k=1}^{NF} (\eta_{j,h}M_{jk} + \eta_{NF+j,h}D_{jk})\psi_k^*(e^{\lambda_h\mathbf{z}} - 1) \\
 & + \sum_{h=1}^{2NF} \eta_{NF+i,h} \sum_{j=1}^{NF} \eta_{NF+j,h} \sum_{k=NF+1}^f M_{jk}\psi_k^* e^{\lambda_h\mathbf{z}} \\
 & + \sum_{h=1}^{2NF} \eta_{NF+i,h} \sum_{j=1}^{NF} \eta_{NF+j,h} \sum_{k=1}^{NP} \frac{g_k(\lambda_h)}{(\lambda_h - \lambda_{2NF+k})} e^{\lambda_h\mathbf{z}} \\
 & + \sum_{h=1}^{2NF} \eta_{NF+i,h} \sum_{j=1}^{NF} \eta_{NF+j,h} \sum_{k=1}^{NP} \frac{g_k(\lambda_{2NF+k})}{(\lambda_{2NF+k} - \lambda_h)} e^{\lambda_{2NF+k}\mathbf{z}}, \quad i = 1, 2, \dots, NF,
 \end{aligned}$$

which can be rearranged to read

$$\begin{aligned}
 x_i(\mathbf{z}) &= \sum_{h=1}^{2NF} (e^{\lambda_h\mathbf{z}} - 1) \frac{\eta_{NF+i,h}}{\lambda_h} \sum_{j=1}^{NF} \sum_{k=1}^{NF} (\eta_{j,h}M_{jk} + \eta_{NF+j,h}D_{jk})\psi_k^* \\
 & + \sum_{h=1}^{2NF} \eta_{NF+i,h} \sum_{j=1}^{NF} \eta_{NF+j,h} \left[\sum_{k=NF+1}^f M_{jk}\psi_k^* e^{\lambda_h\mathbf{z}} + \sum_{k=1}^{NP} \frac{g_k(\lambda_h)e^{\lambda_h\mathbf{z}} - g_k(\lambda_{2NF+k})e^{\lambda_{2NF+k}\mathbf{z}}}{(\lambda_h - \lambda_{2NF+k})} \right], \\
 & \quad i = 1, 2, \dots, NF. \tag{14.8}
 \end{aligned}$$

A little reflection shows what powerful advantages we have achieved by use of the residue theorem for performing the integral of Eq. (14.3). We see that the integration required for the inversion of the Laplace transform has become a summation over a discrete and fairly small set of terms, and no approximation has been required. Although finding eigenvalues is thought by some to be a time-consuming calculation, it must be remembered that even a complex mechanism or multibody system usually has mobility (f) of less than ten. Therefore, for this application, the computation is actually quite rapid.

Although the formula may look a little foreboding, it is really quite straightforward. Typically, the several terms of the nested summations combine numerically

into damped sinusoidal oscillations at $N\mathcal{F}$ natural frequencies of the system, defined by the complex conjugate pairs of eigenvalues in the first line of the previous equation. In the second line, we see a set of response terms with damped frequencies coming from forces caused by the SGC motions in the first sum over h , and in the final sum over h , we see $N\mathcal{P}$ more response terms with damped frequencies caused by applied forces and torques.

14.3 Systems with Repeated Eigenvalues

In the preceding sections, we found it convenient to assume that the eigenvalues of our system are distinct. Although this is usually the case, we now wish to find a solution to our linearized equations of motion that is not subject to this restriction.

If, while using the residue theorem to invert the Laplace transform of Eq. (14.1), we encounter a repeated eigenvalue or a multiple pole coming from Eq. (14.4) with value of λ_k , – say, with multiplicity m_k – then in the inverse transform of Eq. (14.7), there will be m_k terms, all of which are of the form

$$2\pi j(s - \lambda_k) f(s) e^{st} \Big|_{s=\lambda_k}. \quad (14.9)$$

If m_k is more than unity, these must be replaced with a sum of m_k terms of the form

$$2\pi j \sum_{h=0}^{m_k-1} \frac{d^h}{ds^h} \left\{ (s - \lambda_k)^{m_k} f(s) \right\} \frac{t^{(m_k-1-h)}}{h!} e^{st} \Big|_{s=\lambda_k}. \quad (14.10)$$

However, further expansion of this case is left to the reader as an exercise. Instead, we will extend the modal solutions of Chapter 13 to include the case of repeated eigenvalues.

For the class of systems considered in this text, we found in Eq. (11.13), that the linearized equations of motion for the free-generalized coordinates are

$$M_{FF} \dot{v}_F + D_{FF} v_F + K_{FF} x_F = F_F - M_{FS} \dot{v}_S - D_{FS} \dot{x}_S - K_{FS} x_S = G_F,$$

and, in Eq. (13.4), we wrote these in first-order form as

$$\begin{bmatrix} 0 & M_{FF} \\ M_{FF} & D_{FF} \end{bmatrix} \begin{Bmatrix} \dot{v}_F \\ \dot{x}_F \end{Bmatrix} - \begin{bmatrix} M_{FF} & 0 \\ 0 & -K_{FF} \end{bmatrix} \begin{Bmatrix} v_F \\ x_F \end{Bmatrix} = \begin{Bmatrix} M_{FF} \dot{\psi}_F^* \\ D_{FF} \dot{\psi}_F^* + G_F \end{Bmatrix}.$$

Using the definition of Eq. (13.6)

$$y = \begin{Bmatrix} v_F \\ x_F \end{Bmatrix},$$

and multiplying these equations by the inverse of the lead matrix, we can put them into the form

$$\dot{y} - \begin{bmatrix} -M_{FF}^{-1} D_{FF} & -M_{FF}^{-1} K_{FF} \\ I & 0 \end{bmatrix} y = \begin{Bmatrix} M_{FF}^{-1} G_F \\ \dot{\psi}_F^* \end{Bmatrix}. \quad (14.11)$$

From the values at the linearization set point, the initial conditions for this system are that when $t = 0$, we have $v_F = x_F = y = 0$.

As pointed out in Eq. (13.10), the homogeneous form of Eq. (14.11) gives an eigenvalue problem, and has the same eigenvalues as previously found from

Eq. (13.9). However, in this section, we are no longer willing to assume that the eigenvalues are distinct. Nevertheless, we still define a modal matrix η , as we did in Eq. (13.12), and define a set of modal coordinates z :

$$y = \eta z. \quad (14.12)$$

In terms of these modal coordinates, Eq. (14.11) becomes

$$\eta \dot{z} - \begin{bmatrix} -M_{FF}^{-1} D_{FF} & -M_{FF}^{-1} K_{FF} \\ I & 0 \end{bmatrix} \eta z = \begin{Bmatrix} M_{FF}^{-1} G_F \\ \dot{\psi}_F^* \end{Bmatrix},$$

which can be pre-multiplied by η^{-1} to show that

$$\dot{z} - \eta^{-1} \begin{bmatrix} -M_{FF}^{-1} D_{FF} & -M_{FF}^{-1} K_{FF} \\ I & 0 \end{bmatrix} \eta z = \eta^{-1} \begin{Bmatrix} M_{FF}^{-1} G_F \\ \dot{\psi}_F^* \end{Bmatrix}$$

with the initial conditions that at $\tau = 0$, $z = 0$.

Let us now define two new symbols,

$$\Lambda = \eta^{-1} \begin{bmatrix} -M_{FF}^{-1} D_{FF} & -M_{FF}^{-1} K_{FF} \\ I & 0 \end{bmatrix} \eta \quad \text{and} \quad \zeta = \eta^{-1} \begin{Bmatrix} M_{FF}^{-1} G_F \\ \dot{\psi}_F^* \end{Bmatrix}, \quad (14.13)$$

which reduce our modal differential equations of motion to the form

$$\dot{z} - \Lambda z = \zeta. \quad (14.14)$$

In previous sections, where we assumed all eigenvalues to be distinct, the Λ matrix formed in Eq. (14.13) was of size $(2NF \times 2NF)$ and became diagonalized with values equal to the eigenvalues on its diagonal. Now, however, where we consider that some of the eigenvalues may be repeated, this diagonalization may not be possible. In such a case, the system is said to be *defective*. However, even with a defective system, it is always possible to find a matrix η that reduces the Λ matrix to what is called *Jordan normal form* [4, 6, 8, 9] (this is the same Jordan as for the Jordan curve of section 14.1). Such a form is block diagonal

$$\Lambda = \eta^{-1} \begin{bmatrix} -M_{FF}^{-1} D_{FF} & -M_{FF}^{-1} K_{FF} \\ I & 0 \end{bmatrix} \eta = \begin{bmatrix} \Lambda_1 & 0 & \cdots & 0 \\ 0 & \Lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Lambda_m \end{bmatrix}, \quad (14.15)$$

where each Λ_k block is called a *Jordan block*.

Each Jordan block Λ_k is of size $(m_k \times m_k)$ to match the number of times that each distinct eigenvalue λ_k is repeated, and $m_1 + m_2 + \cdots + m_m = 2NF$. Each Jordan block has m_k equal values of the eigenvalue λ_k on its diagonal and, for blocks where m_k is greater than one, the block has values of unity on its first super-diagonal, whereas all other entries of the block are zeroes. Therefore, each is of the form

$$\Lambda_k = \begin{bmatrix} \lambda_k & 1 & \cdots & 0 \\ 0 & \lambda_k & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & \lambda_k \end{bmatrix}, \quad k = 1, 2, \dots, m. \quad (14.16)$$

Many of the Jordan blocks Λ_k are of size m_k equal to one. In this, the usual case, the eigenvalue λ_k is not repeated; η_k becomes an eigenvector and Λ_k contains one distinct eigenvalue on the diagonal, as dealt with previously. However, with a defective system, the Jordan normal form is the simplest form possible for the corresponding equations of motion of Eq. (14.14).

There are now m_k equations of motion corresponding to each Jordan block, and each set is of the form

$$\dot{z}_k - \Lambda_k z_k = \zeta_k, \quad k = 1, 2, \dots, m. \quad (14.17)$$

There are two possible cases: the value of λ_k for a particular Jordan block may be zero or it may be nonzero. When the value of λ_k for a particular Jordan block is nonzero, the homogeneous equations are

$$\{\dot{z}_k\}_H - [\Lambda_k]\{z_k\}_H = 0,$$

and because λ_k may be repeated in a particular Jordan block, the form of the homogeneous solution is

$$\{z_k\}_H = [\dot{\mathcal{T}}_k(\boldsymbol{\tau})]C_k e^{\lambda_k \boldsymbol{\tau}},$$

where $[\dot{\mathcal{T}}_k(\boldsymbol{\tau})]$ is an $(m_k \times m_k)$ matrix of the form

$$[\dot{\mathcal{T}}_k(\boldsymbol{\tau})] = \begin{bmatrix} 1 & \boldsymbol{\tau} & \cdots & \frac{\boldsymbol{\tau}^{(m_k-1)}}{(m_k-1)!} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{\tau} \\ 0 & \cdots & 0 & 1 \end{bmatrix}, \quad k = 1, 2, \dots, m, \quad (14.18)$$

and C_k is a column of complex constants that depend on the initial conditions.

By assuming that the form of the particular solution is $\{z_k\}_P$ is constant, we find from Eq. (14.17) that

$$\{z_k\}_P = -[\Lambda_k]^{-1}\zeta_k,$$

with

$$[\Lambda_k]^{-1} = \begin{bmatrix} 1 & -1 & \cdots & \frac{(-1)^{m_k-1}}{\lambda_k^{m_k}} \\ \frac{1}{\lambda_k} & \frac{-1}{\lambda_k^2} & \cdots & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{1}{\lambda_k} & \frac{-1}{\lambda_k^2} \\ 0 & \cdots & 0 & \frac{1}{\lambda_k} \end{bmatrix}, \quad k = 1, 2, \dots, m. \quad (14.19)$$

Therefore, combining the homogeneous and particular solutions, the complete solution for a Jordan block with nonzero λ_k is

$$z_k = [\dot{\mathcal{T}}_k(\boldsymbol{\tau})]C_k e^{\lambda_k \boldsymbol{\tau}} - [\Lambda_k]^{-1}\zeta_k.$$

With the initial conditions that at $\tau = 0$, $z_k = 0$, Eq. (14.18) shows that $[\dot{\mathcal{T}}_k(\tau)] = I$ and the previous equation gives $C_k = [\Lambda_k]^{-1}\zeta_k$. Therefore, the complete solution for the response of a Jordan block with nonzero λ_k is

$$z_k = [\dot{\mathcal{T}}_k(\tau)e^{\lambda_k\tau} - I][\Lambda_k]^{-1}\zeta_k, \quad k = 1, 2, \dots, m. \quad (14.20)$$

The remaining case is when the value of λ_k for a particular Jordan block is $\lambda_k = 0$. In this case, the Jordan block of Eq. (14.16) takes the form

$$\Lambda_k = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Equations (14.17) for such a block have a solution of the form

$$z_k = [\dot{\mathcal{T}}_k(\tau)]C_k e^{\lambda_k\tau} + [\mathcal{T}_k(\tau)]\zeta_k, \quad k = 1, 2, \dots, m, \quad (14.21)$$

where

$$[\mathcal{T}_k(\tau)] = \begin{bmatrix} \tau & \frac{\tau^2}{2} & \cdots & \frac{\tau^{m_k}}{m_k!} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \tau & \frac{\tau^2}{2} \\ 0 & \cdots & 0 & \tau \end{bmatrix}, \quad k = 1, 2, \dots, m. \quad (14.22)$$

With the initial conditions that at $\tau = 0$, $z_k = 0$, Eq. (14.18) shows that $[\dot{\mathcal{T}}_k(0)] = I$ and Eq. (14.22) gives $[\mathcal{T}_k(0)] = 0$. Therefore, Eq. (14.21) shows that $C_k = 0$, and the complete solution for the response of a Jordan block with λ_k equal to zero is

$$z_k = [\mathcal{T}_k(\tau)]\zeta_k, \quad k = 1, 2, \dots, m. \quad (14.23)$$

To summarize the total solution process, we must first reduce our total set of first-order equations of motion for the system FGCs to Jordan normal form as shown by Eqs. (14.14), (14.15), and (14.16). Once the Jordan blocks are identified, we can form the overall modal response matrix

$$[Z(\tau)] = \begin{bmatrix} Z_1(\tau) & 0 & \cdots & 0 \\ 0 & Z_2(\tau) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_m(\tau) \end{bmatrix}, \quad (14.24)$$

where the $[Z_k(\tau)]$ blocks are of sizes $(m_k \times m_k)$ as found from the Jordan blocks, and have values of

$$[Z_k(\tau)] = \begin{cases} [\dot{\mathcal{T}}_k(\tau)e^{\lambda_k\tau} - I][\Lambda_k]^{-1} & \text{for } \lambda_k \neq 0, \\ [\mathcal{T}_k(\tau)] & \text{for } \lambda_k = 0, \end{cases} \quad k = 1, 2, \dots, m, \quad (14.25)$$

and $[A_k]^{-1}$, $[\dot{\mathcal{T}}_k]$, and $[\mathcal{T}_k]$ are given by Eqs. (14.19), (14.18), and (14.22), respectively. Finally, from these, we can find the modal amplitude response vector

$$z = [Z(\boldsymbol{z})]\zeta \quad (14.26)$$

for the total system. From Eq. (14.12), z gives the state space response vector y that includes values for both the changes in position x and the changes in velocity v from those at the linearization set point:

$$\begin{bmatrix} v \\ x \end{bmatrix} = y = \eta z = \eta [Z(\boldsymbol{z})]\zeta. \quad (14.27)$$

For purposes that will become clear in section 14.5, we also have need to find the time derivative of these, in order to find the accelerations \dot{v} of the system FGCs:

$$\begin{bmatrix} \dot{v} \\ \dot{x} \end{bmatrix} = \dot{y} = \eta \dot{z} = \eta [\dot{Z}(\boldsymbol{z})]\zeta, \quad (14.28)$$

where

$$[\dot{Z}(\boldsymbol{z})] = \begin{bmatrix} \dot{Z}_1(\boldsymbol{z}) & 0 & \cdots & 0 \\ 0 & \dot{Z}_2(\boldsymbol{z}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{Z}_m(\boldsymbol{z}) \end{bmatrix}, \quad (14.29)$$

$$[\dot{Z}_k(\boldsymbol{z})] = \begin{cases} [\ddot{\mathcal{T}}_k(\boldsymbol{z}) + \dot{\mathcal{T}}_k(\boldsymbol{z})\lambda_k]e^{\lambda_k t}[A_k]^{-1} & \text{for } \lambda_k \neq 0, \\ [\dot{\mathcal{T}}_k(\boldsymbol{z})] & \text{for } \lambda_k = 0, \end{cases} \quad k = 1, 2, \dots, m, \quad (14.30)$$

and

$$[\ddot{\mathcal{T}}_k(\boldsymbol{z})] = \begin{bmatrix} 0 & 1 & \boldsymbol{z} & \cdots & \frac{\boldsymbol{z}^{(m_k-2)}}{(m_k-2)!} \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{z} \\ 0 & \ddots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad k = 1, 2, \dots, m. \quad (14.31)$$

14.4 Time Integration Algorithm

As we have progressed through the methods of [Chapter 13](#) and then the preceding material of this chapter, our real intent has been to find the form of the solution for the nonlinear equations of motion, Eqs. (10.15), in order to compute the time history of the FGCs of our mechanism or multibody system. Because the time histories of the SGCs are known a priori, this completes our full knowledge of the motions of the generalized coordinates and, therefore, of the entire system. From these and the methods of Chapters 6, 7, and 8, the full kinematics of all parts of our system become known.

Well, that has been our intention. However, as usual, things are never quite so simple. Instead of this original goal, we found it necessary to linearize the equations around some set point, which we did in [Chapter 11](#), and then to solve this linearized set to find Eqs. (14.27). The intention has been that this set of solutions could be applied recursively over short time intervals in an incremental time integration algorithm. However, a major question remains. How large should our time increment be? If the increment is too large, errors coming from the linearization approximation will accumulate and the solution will be inaccurate. If the time increment is too small, the algorithm will operate unacceptably slowly.

The solutions that we found in Eqs. (14.27) are only valid as long as our linearization assumptions hold. The primary assumption has been that the M , D , K , and G matrices remain essentially constant during each time increment. Well, experience shows that this assumption remains suitably valid as long as (a) the changes in system geometry remain reasonably small, and (b) the time variation of the forces contained in G are reasonably approximated. However, as the time increment increases, both assumptions fall more and more into question. Ultimately, there is a limit on the time increment for which they hold. The solution to this dilemma, of course, is to keep the time increment small enough to assure compliance with these assumptions.

Let us declare that we are starting the simulation of our dynamic multibody system at a known value of time $t = t_0$, from a known set of FGC positions $\psi_F(t_0)$, and with a known set of FGC velocities $\dot{\psi}_F(t_0)$. We can also choose a value for the desired time interval Δt between printing or display of results, and an estimated value for a time interval $\tau \leq \Delta t$ to be used for the calculation time increment. Then we can apply the following recursive procedure:

1. Evaluate current values of the SGC positions $\psi_S(t)$, velocities $\dot{\psi}_S(t)$, and accelerations $\ddot{\psi}_S(t)$.
2. Apply the numeric methods of Chapter 6 to ensure that all dependent position information for the system is known precisely at the current time.
3. Evaluate the M , D , and K coefficient matrices of the system equations of motion using Eqs. (11.7), (11.8), and (11.9), and the applied forces F using Eq. (11.10), and G_F using Eq. (11.13).
4. Form the coefficient matrices $M_{FF}^{-1}D_{FF}$, $M_{FF}^{-1}K_{FF}$, and $M_{FF}^{-1}G_F$ of Eq. (14.11).
5. Find the Jordan normal form Λ and the generalized eigenvectors η of Eq. (14.15).
6. Form the forcing functions ζ of Eq. (14.13).
7. Find a predicted set of FGC position increments $x(\tau)$ from Eq. (14.27). If any are large, reduce the time increment $\tau \rightarrow \tau/2$ and repeat this step.
8. Evaluate predicted values of the SGC positions $\psi_S(t + \tau)$, velocities $\dot{\psi}_S(t + \tau)$, and accelerations $\ddot{\psi}_S(t + \tau)$ from their definitions.
9. Predict new FGC positions $\psi_F(t + \tau) = \psi_F(t) + x(\tau)$ from Eq. (11.1).
10. Attempt the numeric posture solution procedure of Chapter 6 to find the dependent ϕ positions for time $t + \tau$. If this procedure fails, reduce the time increment $\tau \rightarrow \tau/2$ and repeat from step 7.
11. Find new FGC velocity increments $v(\tau)$ from Eq. (14.27).
12. Predict new FGC velocities $\dot{\psi}_F(t + \tau) = \dot{\psi}_F(t) + v(\tau)$ from Eq. (11.1). Find all dependent velocities for time $t + \tau$.
13. Set time to $t \rightarrow t + \tau$.

14. Evaluate the M , D , and K coefficient matrices of the system equations of motion using Eqs. (11.7), (11.8), and (11.9), the applied forces F from Eq. (11.10), and G_F from Eq. (11.13).
15. Find the FGC accelerations from Eq. (11.13), $\ddot{\psi}_F(t) = \dot{v}_F(t) = M_{FF}^{-1} (G_F - D_{FF}v_F - K_{FF}x_F)$. Find all dependent accelerations.
16. Test whether time t has reached the value for the next printing time and/or output display time; if so, process and output data as necessary.
17. Test whether time t has reached the proper value for ending the simulation; if not, return to step 4 to continue.

Before finishing this section, it is very important that we review the class of problems being considered in this text and contrast the time integration method presented here with others that might be considered. Let us first remember that the number of degrees of freedom being considered is, perhaps, between ten and twenty in a very complex system. Therefore, it is quite feasible to consider solving the iterative loop-closure computation of Chapter 6 at every time step, particularly with the extremely quick convergence reported in that chapter. For the same reason, solving for the Jordan normal form, as in step 5, or for eigenvalues and eigenvectors is not unreasonable at every time step.

However, the very fact that the numeric algorithm of Chapter 6 is being used does imply that this algorithm makes a new decision at each time step as to which and even how many FGC variables exist for each new time step, and there is no assurance that these will be the same variables from one time step to the next. In many problems, the decision of which joint variables are chosen as FGCs does change as the system geometry changes. Also, if the system being simulated passes through a singular (for example, a dead-center) posture, the number of FGCs increases at such a posture and then decreases again after leaving that posture. These considerations do not totally prevent use of some of the better-known numeric integration algorithms, but they do provide additional complications not present in other applications, such as in predictor/corrector techniques, and not addressed in numerical analysis texts.

It must be understood that the software being considered here is for simulating multibody dynamic systems. This is an extremely different class of problems from the class addressed in finite element analysis (FEA). In that class:

- (a) The bodies being simulated may be of high stiffness, but they are not rigid. Indeed, in that class, the strain (change of geometry) variables are among the fundamental unknowns being sought.
- (b) The number of degrees of freedom is often in the hundreds or even in the thousands.
- (c) The solution of FEA problems is not expected to operate quickly enough for real-time animation during a dynamic simulation. Much slower solution methods are usually acceptable, with animation done only by post-processing.

Traditional wisdom in numerical analysis texts advises that the time step in the numeric solution of an initial value problem should not be greater than about one-tenth of one cycle of the highest frequency of the problem being simulated. The natural frequencies of typical problems being considered here often range from a

single Herz or less for the low frequency, to tens of MegaHerz for the high frequency. Therefore, this advice suggests a typical time-step size of $\tau \approx (10)^{-8}$ seconds, or 100 million integration steps per second simulated. It must be understood that this advice is based on the recognition that almost all well-known methods of numeric integration are based on *power* series in the independent variable τ . Here we have used a *harmonic* series – recall $e^{\lambda_k \tau}$ in Eqs. (14.8) and (14.30) – and we have found the complete and exact theoretical solution to the linearized differential equations during the time increment being integrated. Our approximation has been in assuming that the linearized equations are a good approximation throughout the time step. This assumption is limited only by the size of the time step and the changes in geometry and applied forces, not by the natural frequencies involved.

In relatively recent times, numeric methods have been developed for integrating the class of problems known as *stiff differential equations* [3]. This is the class in which the spectrum of resonant frequencies range over several orders of magnitude and, therefore, traditional methods have called for thousands or even millions of time steps per cycle of progress in the overall simulation. This is precisely the class usually represented in multibody dynamic systems. However, as just explained, the method developed *is* such a method. Particularly with the improvements of the next section, it can and often will pass over hundreds, thousands, or even millions of cycles of high frequencies during a single time step, particularly when their amplitudes within the actual response are small because of the energy required and because those modes are not excited.

14.5 Adaptive Time-step Control

Sometimes there are difficulties with the time integration algorithm presented in the previous section. As shown in steps 7 and 10, for example, there may be situations where the algorithm predicts egregiously large changes in the FGCs positions and it becomes necessary to reduce the size of the time increment, if only to keep the numeric loop-closure process under control.

However, even if extremes are avoided successfully, this does not ensure that the algorithm produces reliable accuracy. How can we tell if the time increment might still be too large? Well, one way to test this might be to perform the simulation again with a smaller time increment, and to compare results to see if they agree within acceptable limits. However, this does not seem reasonable as a continuing requirement. Is there some way to test as the time integration process evolves? Yes, there is.

Immediately after step 12 in our time integration process explained in the previous section, we can predict for time $t + \tau$ the expected new values of the accelerations \dot{v} of the FGCs by using Eq. (14.28). These are the accelerations that occur according to the prediction of the linearized equations of motion. They should be of acceptable accuracy if the time step is within the range for which the linearized equations are valid.

However, in step 13, we set the time to $t + \tau$ and, in step 15, we recalculate these same accelerations with new data that are updated for the modified geometry. These values are accurate in the sense that they fit the nonlinear Lagrange equations at the

new posture. By subtracting the earlier predictions from this later calculation, we can determine the differences in the FGC accelerations that result from the inaccuracy of using the linearized equations for our nonlinear system.

If this difference in accelerations is found to be larger than some chosen upper limit, then the size of the time increment should be reduced, for example, to $\tau \rightarrow \tau/2$. If the error is well beyond this upper limit, then it may be considered necessary to repeat the current step with the smaller time increment.

Of course, it is to be expected that there will always be some difference in accelerations; this is true for any numeric solution of nonlinear differential equations, no matter what method is used. In fact, if this difference in accelerations is smaller than some chosen lower bound, then the time step being used is probably too small and integration is progressing more slowly than necessary. Under these circumstances, the time increment may be increased, say to $\tau \rightarrow \min(2\tau, \Delta t)$. This will ensure that the time integration proceeds more quickly, but also that the time step does not exceed the interval between desired output time steps.

Before finishing this section, we should recognize that step 10 of our integration algorithm is continuously ensuring that all geometric constraints of our system are enforced to good accuracy. Therefore, even though forces in various parts of our system may include errors, the geometry of the system being simulated is always valid. That is, it always represents a possible posture that can realistically be experienced by the modeled system. We should also recognize that error in acceleration is a very sensitive test. It is a convenient technique for sensing linearization error because we predict with the linearized equations and then recalculate with the nonlinear equations. Moreover, numeric errors should be expected to be larger when calculating derivatives. Therefore, controlling error in acceleration should control errors in velocity and position even more accurately.

Another improvement has been found in helping the user to estimate or evaluate the limits to be used in controlling error. The calculation technique can proceed as previously explained. However, by redefining error so that it includes pre-multiplication by the mass matrix – that is, $\varepsilon = M[\ddot{\psi} - \dot{v}]$ – we can recast error into units of error in force. Most users find it more intuitive to choose an accuracy limit that they wish to attain in terms of accuracy in force rather than accuracy in acceleration.

REFERENCES

1. A. L. Cauchy, *Sur un nouveau genre de calcul analogue au calcul infinitésimal (On a new application of analog calculus to the calculation of infinitesimals)*, Paris, 1826.
2. R. V. Churchill, *Complex Variables and Applications*, McGraw-Hill Book Company, Inc., New York, 1960.
3. W. C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
4. M. E. C. Jordan, *Traité des substitutions et des équations algébriques, (Treatise on substitutions and algebraic equations)*, Paris, 1870, pp. 114–25.
5. ———, *Cours d'Analyse de l'École Polytechnique (Analysis course of the École Polytechnic)*, Paris, 1887.
6. B. Kågström and A. Ruhe, "An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Transactions on Mathematical Software*, vol. 6, no. 3, New York, September 1980.

7. G. D. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Book Company, Inc., New York, 1961, section 1.7–4, pp. 21–22.
8. B. Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969, section 11.6.
9. B. Sridhar and D. Jordan, “An Algorithm for Calculation of the Jordan Canonical Form of a Matrix,” *Computers and Electrical Engineering*, vol. 1, no. 2, 1973, pp. 239–54.

PROBLEM

14.1 Continue from Eq. (14.9), using Eqs. (14.6) and (14.10), to apply the residue theorem to the inversion of the Laplace transform where there is a single pair of equal eigenvalues; write the solution out explicitly for the case of $f = 2$, $NF = 2$, $NP = 1$, $g_{j,1}(s) = 1$, and $g_{j,2}(s) = 0$, with $\lambda_3 = \lambda_4$.

15 Collision Detection

15.1 Introduction

Through simulation of multibody systems as explained in the preceding chapters, we can solve a variety of useful problems with no further enhancement. However, with the methods explained so far, we still lack the capability to simulate collisions, either between moving bodies or between a single moving body and its fixed surroundings. Simulation software developed strictly with the formulae presented so far assumes that a moving body may simply pass through others with no interference or impact. Clearly, this can benefit from enhancement.

Collision or contact between bodies cannot be detected unless it is through computations relating the geometries of the bodies' surfaces. Therefore, we must have accurate geometric shapes for all bodies for which collisions are to be considered, and in as much detail and accuracy as we wish to monitor their possible contact. We need data for vertices, edges, and surfaces, and we need to distinguish the material from the exterior sides of such surfaces. Therefore, we need *solid* models of the bodies to be considered. Either *constructive solid geometry* (CSG) or *boundary representation* (B-Rep) or hybrid combinations may be considered, but wire-frame data are not sufficient.¹

On review of the material of [Chapter 3](#), we see that each body has a body coordinate system, and that the posture of that body (with label b) is found by determining its transformation matrix T_{0b} as explained in Chapters 4 and 6. Each body has such a transformation, even the stationary body. Therefore, each body can be given geometric shape by attaching one or more solid models with data measured with respect to that body's coordinate system. That is, for every geometric feature of the shape, the local coordinate data r_b are known.

The only difference between a stationary body and a moving body is whether its transformation matrix with respect to the fixed body changes or remains constant. Also, each geometric feature of each shape takes on the motion of the body to which it is attached. Thus, each shape has a velocity matrix ω_b and an acceleration matrix α_b as found in Chapters 7 and 8, and these motions include both translation and rotation data.

¹ The IMP software uses a half-edge polyhedral B-Rep data structure similar to that explained in [3].

The simulation of collisions consists of two main tasks: collision detection and impact analysis. Solution of the first of these tasks is the intent of this chapter; the second is covered in [Chapter 16](#). Collision detection involves determining when (the value of time) and where (the point r_b of each body at which) contact occurs between bodies of the system simulated.

Because testing for collision may be required during every interval of time in a simulation, and possibly between many combinations of objects, it is extremely important that the algorithm for collision detection be very efficient. For this reason, we limit ourselves to consideration of only two shapes coming into contact at a time. If more than two bodies make simultaneous contact, the software can find the contacts in consecutive pairs if necessary.

Conceptually, because we require that solid models of body shapes be available, it is possible to perform a complete intersection calculation by the algorithms of solid modeling between each pair of bodies considered at every moment in time. Without further improvement, however, this approach is quite inefficient. We also find that those algorithms known to date that offer efficiency advantages, do so partly by restricting their consideration to pairs of convex polyhedra. The assumption of convexity is not an unreasonable constraint because more than one convex subshape may be simultaneously attached to the same body coordinate system. The assumption of polyhedral geometry implies that body shapes are bounded solely by flat surfaces. Curved surfaces may be approximated by flat facets, but as the number of facets becomes large, the efficiency again decreases; a realistic compromise between accuracy and efficiency must be sought.

In order to avoid monitoring an unnecessarily large number of body pairs – thus causing slow performance – we can require that the user explicitly identify those pairs of shapes (or convex subshapes) to be monitored. However, it is advised that two shapes of the same body, or two that are directly connected by a joint (except for an open joint), not be acceptable for collision monitoring because they would be in continual contact at their elemental surfaces.

15.2 Vertex-Face Contact

At the moment that a vertex of one body labeled b actually makes contact with a flat facet of a surface of another body labeled c , such a contact may be represented by an equation of the form

$$A_c X_b + B_c Y_b + C_c Z_b + D_c = 0,$$

where capital letters recognize that this equation is written in terms of the global coordinate system. The plane coefficients are normalized so that the first three coefficients $[A_c \ B_c \ C_c]$ form a unit normal vector directed outward from the shape. If we choose the symbol P_c to represent the column vector of the data of this planar facet, then, in global homogeneous coordinates, the equation has the form

$$[A_c \ B_c \ C_c \ D_c] \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = P_c^t R_b = 0. \quad (15.1)$$

Remembering Eq. (3.5), we know how the local coordinates of the vertex r_b relate to their global values R_b , and thus, we can also find the relation that must hold for the local coordinate data p_c of the planar facet:

$$p_c^t T_{0c}^t T_{0b} r_b = 0. \quad (15.2)$$

Now, assuming that the contact does not begin until a short time increment τ after the current time t ,

$$\begin{aligned} p_c^t T_{0c}^t (t + \tau) T_{0b} (t + \tau) r_b &= 0, \\ p_c^t T_{0c}^t [I + \omega_c^t \tau] [I + \omega_b \tau] T_{0b} r_b &= 0, \\ P_c^t \omega_c^t \omega_b R_b \tau^2 + P_c^t (\omega_c^t + \omega_b) R_b \tau + P_c^t R_b &= 0. \end{aligned}$$

If we assume that τ is small enough to ignore the very small τ^2 term, we can solve for the time increment at which contact begins:

$$\tau = \frac{-P_c^t R_b}{P_c^t (\omega_c^t + \omega_b) R_b} = \frac{-p_c^t [T_{0c}^t T_{0b}] r_b}{p_c^t [T_{0c}^t (\omega_c^t + \omega_b) T_{0b}] r_b}. \quad (15.3)$$

Notice that the factors in square brackets in both the numerator and the denominator of this equation can be computed at each time interval during our time integration as soon as bodies b and c are identified. This is usually worthwhile because many vertices and faces from these bodies may require testing.

If the value of τ found from this equation is negative, then the time of contact is either fictitious or has already passed. However, finding a positive time increment does not assure that contact actually occurs; it only signifies that the vertex r_b reaches the plane of the facet p_c at a future time. We must also find the implied point of contact on the planar surface of body c at the indicated time

$$\begin{aligned} (I + \omega_c \tau) T_{0c} r_c &= (I + \omega_b \tau) T_{0b} r_b, \\ r_c &= [T_{0c}^{-1} (I + \omega_c \tau)^{-1} (I + \omega_b \tau) T_{0b}] r_b, \end{aligned} \quad (15.4)$$

and we must verify that this point r_c falls within the boundary of the actual face of body c contained in the plane p_c . If this is true, then – and only then – we have verified that τ represents a time increment at which contact may actually occur between the vertex r_b and the flat facet p_c .

15.3 Edge-Edge Contact

Next, let us consider the situation when two edges come into contact. Let us say, for example, that an edge of a body labeled b is defined by the intersection of planar faces 1 and 2, whereas an edge of a body labeled c lies at the intersection of planar faces 3 and 4. Then, at the time τ at which these two edges come into contact, a new point of contact with global coordinates R is defined by the four equations

$$\begin{bmatrix} p_1^t T_{0b}^t (I + \omega_b^t \tau) \\ p_2^t T_{0b}^t (I + \omega_b^t \tau) \\ p_3^t T_{0c}^t (I + \omega_c^t \tau) \\ p_4^t T_{0c}^t (I + \omega_c^t \tau) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = ER = 0, \quad (15.5)$$

where the matrix of coefficients E is of size (4×4) and every element is a linear expression in the unknown time increment τ .

However, for this set of homogeneous equations to have a nontrivial solution, the matrix of coefficients must have a zero determinant, $\det(E) = 0$, from which we get a quartic equation in τ . We are interested in finding the smallest nonnegative value of $\tau \leq \mathfrak{z}$ that satisfies this equation, if such a root exists. If such a value of τ is found, then any three of the four Eqs. (15.5) become solvable for R . This solution, however, must still be tested to ensure that it falls between the two limiting vertices (r' and r'') on each of the two edges. This may be done by solving any two components of the conditions

$$\begin{aligned} R &= \mu_b(I + \omega_b\tau)T_{0b}r'_b + (1 - \mu_b)(I + \omega_b\tau)T_{0b}r''_b \\ &= \mu_c(I + \omega_c\tau)T_{0c}r'_c + (1 - \mu_c)(I + \omega_c\tau)T_{0c}r''_c \end{aligned} \quad (15.6)$$

for the unknown parameters μ_b and μ_c , and verifying that each lies within the interval $0 \leq \mu \leq 1$. Once this is verified, then τ refers to the time increment until a possible contact between the edges tested. If not, then such an edge-edge contact does not occur within the coming time interval.

15.4 Finding the Time Increment until Contact

As explained in section 14.4, the overall simulation process takes place under the control of a time integration process for the solution of the equations of motion for the system FGCs. As this time integration process progresses, it is necessary to continually monitor each of the possible contact pairs chosen by the user and, as a contact approaches, to anticipate and meet the precise moment of contact at a transition between time steps in the integration process.

Consider the effect of error in finding the precise time of contact. If we suppose that the onset of contact is not met exactly, then two bodies may already have penetrated each other as the end of a time step is reached. If there is a stiffness associated with that contact, then that stiffness may have already become deflected by the end of the time step and, far worse, no energy would have been required to produce such a state of strain. With the stiffnesses associated with typical mechanical parts, this may represent a significant energy increase in the system. Even worse, once this energy enters the system – even through numeric error – it remains there, causing errors in future velocities or other motion parameters. For this reason, it is extremely important that the moment of contact be met precisely so that high stiffnesses do not produce strain or errors in system energy.

As the simulation develops, the size of the time increment \mathfrak{z} of the integration process is controlled as explained in sections 14.4 and 14.5. In addition, as described there, the user may have requested that possible contact be monitored between particular pairs of bodies. For each possible contact pair, we know that we are to monitor the motion of body c with respect to body b . What we wish to do next is to predict the time increment from the current moment t to the onset of that contact. What we wish to know is whether we anticipate the initiation of a contact within the coming time interval \mathfrak{z} . If so, by how much is it necessary to reduce this time step to exactly match the time of the initial contact.

In section 15.2, we discussed the contact of some vertex r_b of body b with some flat facet p_c of body c and in section 15.3, between two edges of these bodies. The choice of the two bodies was identified by the user, but we have not yet spoken of how this particular vertex and this particular facet or how this pair of edges will be identified. The question of collision detection has seen much research over the past several years. The algorithm presented here is a variation of one of the faster and better-known algorithms, called the GJK minimum-distance algorithm from the initials of its authors [2], with subsequent extensions by Cameron [1].

As a start, a single face is chosen arbitrarily from each body, b and c . Let us identify these as p_b and p_c . We also choose an arbitrary vertex r_c on face p_c to start the following recursive procedure:

1. Test each vertex r_b of face p_b using Eq. (15.3) to find the minimum nonnegative time increment τ for contact of r_b with face p_c . If none is found, choose a different face p_c and repeat.
2. Test each vertex r'_b connected by an edge to vertex r_b , using Eq. (15.3), seeking a vertex with smaller nonnegative time increment τ' . If found, change r_b and τ to r'_b and τ' and, if necessary, change p_b to a new face that includes the new vertex r_b .
3. If r_b was changed by step 1 or step 2, repeat from step 1 until no further change is found.
4. Test each face p_b containing vertex r_b for the contact time with vertex r_c , using Eq. (15.3), seeking a smaller nonnegative time increment τ' , and updating the face p_b and time increment τ to this new minimum.
5. If p_b was changed from that of step 1, repeat from step 1.
6. Test each vertex r_c of face p_c using Eq. (15.3) to find the minimum nonnegative time increment τ for contact of r_c with face p_b .
7. Test each vertex r'_c connected by an edge to vertex r_c , using Eq. (15.3), seeking a vertex with smaller nonnegative time increment τ' . If found, change r_c and τ to r'_c and τ' and, if necessary, change p_b to a new face that includes the new r_c .
8. If r_c was changed by step 6 or step 7, repeat from step 6 until no further change is found.
9. Test each face p_c connected to vertex r_c for the contact time with vertex r_b , using Eq. (15.3), seeking a smaller nonnegative time increment τ' , and updating the face p_c and time increment τ' to this new minimum.
10. If p_c was changed from that of step 6, repeat from step 6.
11. If any of these data has changed from that of step 1, repeat from step 1.
12. After iterating in this manner and finding the minimum time increment τ and locations r_b and r_c until no further improvement is found; then, using Eq. (15.5), test all edges connected to vertex r_b for contact with each edge connected to vertex r_c to see if an even smaller initial contact time τ is found.

Once this iteration process has finished, we have found the time increment τ until the initial contact between bodies b and c . We must then repeat the process for the next contact pair that was chosen by the user until all requested pairs have been tested. When finished with all, we know the smallest time increment τ before the first contact. We have also found the body numbers b and c for this contact, the type of that contact (vertex-face or edge-edge), and the corresponding vertex, face, and/or

edge data at the location of impending contact. All of this information becomes very important in [Chapter 16](#) where we analyze how this collision affects the dynamics of the rest of our system.

Before doing this, however, we should look back to section 14.4 and understand that this entire process is to be inserted after step 16 of the time integration algorithm explained there. As the previous time step is completed, the search for possible collision is performed before completion of the next time step takes place. Upon completion of the impending contact algorithm explained here, the time step Δt for time integration must be set equal to τ , if it is smaller, before recursing in step 17 to continue the time integration. This ensures meeting the precise time of contact at the completion of the next integration step. At that time, the impact analysis of [Chapter 16](#) is applied, and the time step is set back to Δt before continuing.

REFERENCES

1. S. Cameron, "Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra," *International Conference on Robotics and Automation*, April, 1997.
2. E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, April, 1988, pp. 193–203.
3. M. Mäntylä, *An Introduction to Solid Modeling*, Computer Science Press, Rockville, MD, 1988.

16 Impact Analysis

16.1 Applied Impulsive Loads

Before continuing the analysis of the impact resulting from a collision between moving bodies and how these can be incorporated into our time integration of the dynamic equations of motion, let us consider the effects of applied impulsive loads in general.

It should be noted that the material of this entire chapter is based on the research of Dr. W. Wisutmethangoon [4], and is presented here with his permission.

As we saw in section 10.2, the general Lagrange equations of motion for our dynamic system are given by Eq. (10.1):

$$\frac{d}{dt} \left(\frac{\partial H}{\partial \dot{\psi}_i} \right) - \frac{\partial H}{\partial \psi_i} + \frac{\partial V}{\partial \psi_i} = F_i, \quad i = 1, 2, \dots, NF,$$

where H and V are the kinetic and potential energies of the system and F_i is the generalized force acting at the free generalized coordinate ψ_i .

If we integrate this equation over a short interval of time from t to $t+\delta t$, we obtain

$$\int_t^{t+\delta t} \frac{d}{dt} \left(\frac{\partial H}{\partial \dot{\psi}_i} \right) dt - \int_t^{t+\delta t} \frac{\partial H}{\partial \psi_i} dt + \int_t^{t+\delta t} \frac{\partial V}{\partial \psi_i} dt = \int_t^{t+\delta t} F_i dt, \quad i = 1, 2, \dots, NF.$$

Because the time interval for an impact is very short, there is insufficient time for changes in geometry, even though velocities can change under impulsive accelerations. Therefore, the second and third terms of this equation are very small and the equations reduce to

$$\left(\frac{\partial H}{\partial \dot{\psi}_i} \right) \Big|_{t+\delta t} - \left(\frac{\partial H}{\partial \dot{\psi}_i} \right) \Big|_t = \Delta \left(\frac{\partial H}{\partial \dot{\psi}_i} \right) = \int_t^{t+\delta t} F_i dt, \quad i = 1, 2, \dots, NF. \quad (16.1)$$

From Eq. (10.3), we recognize that this set of equations shows the changes in the components of the generalized system momentum and, from Eq. (10.6), we have

$$p_i = \frac{\partial H}{\partial \dot{\psi}_i} = \sum_{b=1}^{\ell} \text{trace} (W_{bi} T_{0b} J_b T_{0b}^t \omega_b^t), \quad i = 1, 2, \dots, NF,$$

and, using Eq. (7.24), this becomes

$$p_i = \frac{\partial H}{\partial \dot{\psi}_i} = \sum_{j=1}^f \sum_{b=1}^{\ell} \text{trace}(W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t) \dot{\psi}_j, \quad i = 1, 2, \dots, NF.$$

Again recognizing that the time interval is too short for changes in system geometry, but that system velocities can change, Eq. (16.1) becomes

$$\Delta p_i = \sum_{j=1}^{NF} \sum_{b=1}^{\ell} \text{trace}(W_{bi} T_{0b} J_b T_{0b}^t W_{bj}^t) \Delta \dot{\psi}_j = \int_t^{t+\delta t} F_i dt, \quad i = 1, 2, \dots, NF,$$

and from Eq. (11.7), we recognize here the elements of the system mass matrix. Therefore, this equation reduces to

$$\sum_{j=1}^{NF} M_{i,j} \Delta \dot{\psi}_j = \int_t^{t+\delta t} F_i dt, \quad i = 1, 2, \dots, NF. \quad (16.2)$$

In section 9.10, we defined a model for a force $f_h(\phi, t)$ acting within a joint with identifying joint label h . This force was modeled as a function of time t and of system geometry ϕ . Here, we take the same approach with the exception that now we take the time variation of the force to be an impulsive load acting within joint h , and we give this impulse the symbol $\epsilon_h(\phi, t)$:

$$\epsilon_h(\phi, t) = \int_t^{t+\delta t} f_h(\phi, t) dt, \quad h = 1, 2, \dots, n.$$

From Eqs. (9.23) and (10.13), the generalized force resulting at generalized coordinate ψ_i from a force applied within joint h is

$$F_i = \phi'_{hi} f_h(\phi, t), \quad \begin{array}{l} h = 1, 2, \dots, n, \\ i = 1, 2, \dots, f, \end{array}$$

and when such a force is impulsive in nature, the generalized impulse becomes

$$\int_t^{t+\delta t} F_i dt = \phi'_{hi} \epsilon_h(\phi, t), \quad \begin{array}{l} h = 1, 2, \dots, n, \\ i = 1, 2, \dots, f. \end{array} \quad (16.3)$$

Similarly, in section 9.11, we defined a model for a force $f_{bcd}(\phi, t)$ applied at a point on body b with direction defined by points on bodies c and d . Here, we define a similar impulsive load $\epsilon_{bcd}(\phi, t)$ acting at a point on body b with direction defined by points on bodies c and d :

$$\epsilon_{bcd}(\phi, t) = \int_t^{t+\delta t} f_{bcd}(\phi, t) dt, \quad b, c, d = 1, 2, \dots, \ell.$$

In Eqs. (9.25) and (10.13), we found the generalized force resulting at generalized coordinate ψ_i from such an applied force to be

$$F_i = R_{bi}^n u_{cd} f_{bcd}(\phi, t), \quad \begin{array}{l} b, c, d = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f, \end{array}$$

and when such an applied force is impulsive, its generalized impulse is

$$\int_t^{t+\delta t} F_i dt = R_{bi}^t u_{cd} \dot{\epsilon}_{bcd}(\phi, t), \quad \begin{array}{l} b, c, d = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f. \end{array} \quad (16.4)$$

Then, in section 9.12, we defined a model for a torque $\tau_{cde}(\phi, t)$ applied on body c with direction defined by a line through points on bodies d and e . Here, we define a similar impulsive torque $\mathfrak{t}_{cde}(\phi, t)$ applied on body c with direction defined by a directed line through points on bodies d and e :

$$\mathfrak{t}_{cde}(\phi, t) = \int_t^{t+\delta t} \tau_{cde}(\phi, t) dt, \quad c, d, e = 1, 2, \dots, \ell.$$

In Eqs. (9.29) and (10.13), we found the generalized force at generalized coordinate ψ_i from such an applied torque to be

$$F_i = \widehat{W}_{ci}^t u_{de} \tau_{cde}(\phi, t), \quad \begin{array}{l} c, d, e = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f, \end{array}$$

where \widehat{W}_{bi} is defined in Eq. (9.28). When the applied torque is impulsive in nature, its generalized impulse is

$$\int_t^{t+\delta t} F_i dt = \widehat{W}_{ci}^t u_{de} \mathfrak{t}_{cde}(\phi, t), \quad \begin{array}{l} c, d, e = 1, 2, \dots, \ell, \\ i = 1, 2, \dots, f. \end{array} \quad (16.5)$$

Finally, when we allow multiple impulses and collect the elements from Eqs. (16.3), (16.4), and (16.5), then Eq. (16.2) expands to

$$\sum_{j=1}^{NF} M_{ij} \Delta \dot{\psi}_j = \sum_{h=1}^n \phi'_{hi} \dot{\epsilon}_h(\phi, t) + \sum_{bcd} R_{bi}^t u_{cd} \dot{\epsilon}_{bcd}(\phi, t) + \sum_{cde} \widehat{W}_{ci}^t u_{de} \mathfrak{t}_{cde}(\phi, t) = \mathcal{I}_i(\phi, t), \quad i = 1, 2, \dots, NF. \quad (16.6)$$

As the time integration process of section 14.4 advances, it must be continually monitored to ensure that it results in incrementing a time step at the precise moment of occurrence of an impulse. At that instant, the right-hand side of Eq. (16.6) is evaluated and yields a vector of length NF of applied impulses $\mathcal{I}_F(\phi, t)$. However, because we assume that the SGC coordinates do not change velocities under impulsive loading, the set of Eqs. (16.6) referring to the NF velocities of the FGCs is of the form

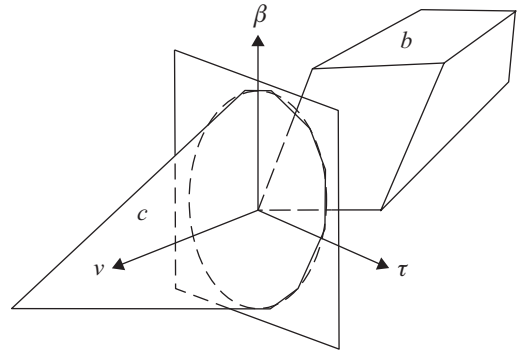
$$M_{FF} \Delta \dot{\psi}_F = \mathcal{I}_F.$$

Because M_{FF} is positive definite, it cannot be singular and these equations can be solved for the changes in the FGC velocities at that time:

$$\Delta \dot{\psi}_F = M_{FF}^{-1} \mathcal{I}_F. \quad (16.7)$$

These changes are added to the FGC velocities before continuing with the next step in the time integration process.

Figure 16.1. Coordinate axes at a point of contact.



16.2 Location and Type of Contact

The other way in which impulsive loading can be induced within our mechanism or multibody system is through collisions between moving bodies. This is the reason for our detailed discussion of detecting such collisions in [Chapter 15](#). Through the algorithms presented there we can detect the precise instant of contact between colliding bodies.

At the moment detected for a collision, the time integration process is interrupted and we know: (1) the identification of the contact pair h being monitored and of the two bodies, b and c , which are making contact; (2) the global coordinates $R_h = T_{0b}r_b = T_{0c}r_c$ of the point of contact, and the velocities of the two contacting points, $\dot{R}_b = \omega_b R_h$ and $\dot{R}_c = \omega_c R_h$; and (3) the type of contact, either vertex-face or edge-edge.

Let us define ν_h to be the unit vector normal to the plane of contact pointing in the direction from body b toward body c as shown in [Figure 16.1](#). For a vertex-face contact, we find ν_h from the data for the face of body b or of body c , whichever is the face in contact. For an edge-edge contact, we find ν_h from the cross product of vectors along the two contacting edges. Next, we take τ_h to be the unit vector in the plane of contact directed parallel to the relative velocity between the two points of contact. This is found by taking and normalizing the vector difference $\tau_h = \dot{R}_b - \dot{R}_c$; if the magnitude of this vector difference is less than some tolerance, then τ_h is chosen arbitrarily in the plane of contact. Next, we find and normalize the bi-normal vector β_h to be a unit vector orthogonal to both ν_h and τ_h such that $\beta_h = \nu_h \times \tau_h$. Finally, we recalculate and normalize the tangent vector τ_h to be the unit vector orthogonal to both β_h and ν_h such that $\tau_h = \beta_h \times \nu_h$. The global coordinates of these unit vectors can be determined from the geometry of the contacting shapes, their locations T_{0b} and T_{0c} , and velocities ω_b and ω_c at the instant of contact.

16.3 Simple Impact Model

As a beginning, let us assume that, at the point of contact, the only reaction between the two bodies is an impulse normal to the plane of contact. Let the normal impulse acting from body b onto body c be symbolized by $e_h^v \nu_h$, where e_h^v is an unknown scalar signifying the amplitude of the impulse. There is an equal and opposite normal impulse $-e_h^v \nu_h$ acting as a reaction from body c back onto body b . Then, from

Eq. (16.4) we can write the generalized impulse caused by this contact as

$$\begin{aligned} \mathcal{I}_h^v(\phi, t) &= R_{bi}^t v_h^v e_h^v - R_{ci}^t v_h^v i_h^v, \\ &= v_h^t (R_{bi}^t - R_{ci}^t) e_h^v, \quad i = 1, 2, \dots, NF, \\ &= v_h^t (W_{bi} - W_{ci}) R_h e_h^v, \end{aligned}$$

and from this, the Lagrange impulse equations for this contact can be written as shown in Eq. (16.6):

$$\sum_{j=1}^{NF} M_{ij} \Delta \dot{\psi}_j = v_h^t (W_{bi} - W_{ci}) R_h e_h^v, \quad i = 1, 2, \dots, NF. \quad (16.8)$$

However, because the magnitude of the impulse e_h^v is still unknown, this is a set of NF equations with $(NF + 1)$ unknowns. Another equation is needed to make the set solvable. This additional equation can be obtained from the definition of the *normal coefficient of restitution*, e_h^v , which is the ratio of the relative normal velocity between the contacting points after the impact to that before it. That is,

$$e_h^v = -\frac{v_h^t (\dot{R}_b^+ - \dot{R}_c^+)}{v_h^t (\dot{R}_b - \dot{R}_c)},$$

where \dot{R}_b^+ and \dot{R}_c^+ (or similar symbols) indicate values after the impact and the negative sign shows the reversal in sense of the relative velocity at the points of contact. The coefficient of restitution is a value representative of the materials making contact and varies from unity for completely *elastic* impact to zero for *plastic* impact. The value of this coefficient can be determined experimentally and must be supplied by the user as data for this model of impact. A good discussion of coefficient of restitution and test procedures for measuring it is given in [2].

Once this coefficient is supplied, the equation expressing its definition can be rearranged to read

$$\begin{aligned} v_h^t (\dot{R}_b^+ - \dot{R}_c^+) &= -e_h^v v_h^t (\dot{R}_b - \dot{R}_c), \\ v_h^t (\omega_b^+ - \omega_c^+) R_h &= -e_h^v v_h^t (\omega_b - \omega_c) R_h, \\ \sum_{j=1}^f v_h^t (W_{bj} - W_{cj}) R_h \dot{\psi}_j^+ &= -e_h^v \sum_{j=1}^f v_h^t (W_{bj} - W_{cj}) R_h \dot{\psi}_j. \end{aligned} \quad (16.9)$$

The set of Eqs. (16.8) and (16.9) can now be brought together into a single set, and written in the form

$$\begin{bmatrix} M_{ij} & v_h^t (W_{bi} - W_{ci}) R_h \\ v_h^t (W_{bj} - W_{cj}) R_h & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi}_j^+ \\ i_h^v \end{Bmatrix} = \begin{bmatrix} M_{ij} \\ -e_h^v v_h^t (W_{bj} - W_{cj}) R_h \end{bmatrix} \{\dot{\psi}_j\}.$$

These equations are solvable for the modified velocities of the free generalized coordinates that result after the impact and for the magnitude of the impulse that results. Typically, this model suits well for simulating the collision of objects with smooth, hard surfaces.

16.4 Impact Model with Tangential Impulse

In the case of collision between objects with rough or soft surfaces, when a relative tangential velocity exists between them, it is likely that a tangential impulse may also be transmitted by the impact. Unless we assume that this tangential impulse is zero, another unknown is present in the system. Therefore, yet another equation is needed for solution.

Two approaches have been proposed [1] for formulating another equation. The first is to define a coefficient μ_h reflecting the ratio of the tangential and normal impulse components, very similar to a coefficient of friction:

$$e_h^\tau = \mu_h e_h^v.$$

The system of equations for this impact model is

$$\begin{aligned} & \begin{bmatrix} M_{ij} & v_h^t(W_{bi} - W_{ci})R_h & \tau_h^t(W_{bi} - W_{ci})R_h \\ v_h^t(W_{bj} - W_{cj})R_h & 0 & 0 \\ 0 & -\mu_h & 1 \end{bmatrix} \begin{Bmatrix} \dot{\psi}_j^+ \\ e_h^v \\ e_h^\tau \end{Bmatrix} \\ & = \begin{bmatrix} M_{ij} \\ -e_h^v v_h^t(W_{bj} - W_{cj})R_h \\ 0 \end{bmatrix} \{\dot{\psi}_j\}. \end{aligned} \quad (16.10)$$

This model, with an appropriate value of μ_h , fits quite well for collisions with high relative tangential velocity.

In another model, the definition of a *tangential coefficient of restitution* is introduced. The definition is similar to that in the normal direction; that is,

$$e_h^\tau = -\frac{\tau_h^t(\dot{R}_b^+ - \dot{R}_c^+)}{\tau_h^t(\dot{R}_b - \dot{R}_c)}.$$

The system of equations using this model is

$$\begin{aligned} & \begin{bmatrix} M_{ij} & v_h^t(W_{bi} - W_{ci})R_h & \tau_h^t(W_{bi} - W_{ci})R_h \\ v_h^t(W_{bj} - W_{cj})R_h & 0 & 0 \\ \tau_h^t(W_{bj} - W_{cj})R_h & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi}_j^+ \\ e_h^v \\ e_h^\tau \end{Bmatrix} \\ & = \begin{bmatrix} M_{ij} \\ -e_h^v v_h^t(W_{bj} - W_{cj})R_h \\ -e_h^\tau \tau_h^t(W_{bj} - W_{cj})R_h \end{bmatrix} \{\dot{\psi}_j\}. \end{aligned} \quad (16.11)$$

This model is more appropriate with small relative tangential velocity at the contact points or for objects with high, shear elastic surfaces.

Figure 16.2 shows a typical graph of experimental data obtained from impact of a steel specimen with a stationary steel block [3]. A plot of relative tangential velocity before and after impact typically follows the solid lines. This graph shows that for low relative tangential velocity, the tangential coefficient of restitution model better fits experimental evidence, whereas the frictional tangential impulse model is more suitable for higher relative tangential velocity.

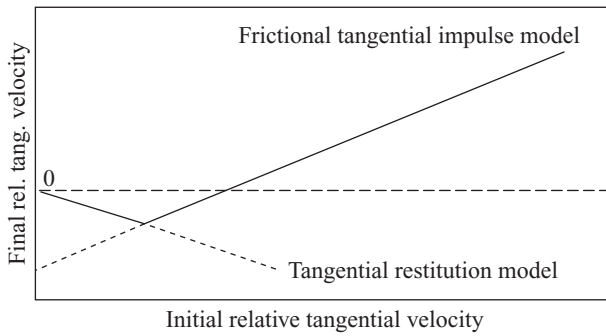


Figure 16.2. Typical experimental data for tangential impulses.

Also suggested by Brach [1] is a bilinear model, which is a combination of the two previously described models, with appropriate values of μ_h and e_h^t to fit experimental data. The way this model is used is to solve both Eqs. (16.10) and (16.11), and then to accept the solution that results in the higher relative tangential velocity after impact.

16.5 Impact Model with Normal Torsional Impulse

Next, we consider the case where the impact is not isolated to a single point, but is distributed over an area on the contacting surfaces. The normal impulse generated during impact in such a case may be accompanied by a torsional impulse in the plane of contact if there is a relative angular velocity about the surface normal at the time of impact. Again, because of the added unknown in this case, an additional equation is required. Similar to the normal coefficient of restitution, a *torsional coefficient of restitution* can be defined as the negative ratio of the relative angular velocity after the impact to that before,

$$e_h^v = -\frac{v_h^t(\dot{\Theta}_b^+ - \dot{\Theta}_c^+)}{v_h^t(\dot{\Theta}_b - \dot{\Theta}_c)},$$

where $\dot{\Theta}_b$, for example, is the angular velocity vector of body b that is expressed by the second Ball vector of its velocity,

$$\dot{\Theta}_b = \hat{\omega}_b = \sum_{j=1}^f \left\{ \begin{array}{l} W_{bj}(3, 2) \\ W_{bj}(1, 3) \\ W_{bj}(2, 1) \end{array} \right\} \dot{\psi}_j = \sum_{j=1}^f \hat{W}_{bj} \dot{\psi}_j.$$

This additional equation can be used for the case of small relative angular velocity about the surface normal axis. However, another model based on the coefficient of friction, which is more suitable with higher relative angular velocity, is

$$\mathfrak{r}_h^v = \text{sgn}[v_h^t(\dot{\Theta}_b - \dot{\Theta}_c)]\mu_h r_p e_h^v.$$

Here, the use of the sgn function accounts for the direction of the torsional impulse caused by friction, which must be in the direction opposite to the relative angular velocity about the normal to the surface. The symbol r_p denotes a characteristic distance called the *pitch radius*, which relates the torsional impulse to the normal force impulse. This pitch radius is assumed constant and must be given as data by

the user. It may be evaluated from different theories. For example, with a uniform pressure distribution over a circular contact region, $r_p = \frac{2}{3}r$, where r is the radius of the contact area.

As with the tangential impulse case, the two models can be combined and used as a bilinear model.

16.6 Impact Model with Moment Impulse

The normal impulse generated during impact may be accompanied by an impulsive moment about an axis in the plane of contact. If this moment impulse is considered to have components in both the tangential and the bi-tangential directions, this introduces two more unknowns into the system. In order to formulate two additional equations, a definition of *moment coefficient of restitution* [2] is introduced. For the tangential direction, it is defined in a manner similar to the previous case:

$$e_h^\tau = -\frac{\tau_h^t(\dot{\hat{\Theta}}_b^+ - \dot{\hat{\Theta}}_c^+)}{\tau_h^t(\dot{\hat{\Theta}}_b - \dot{\hat{\Theta}}_c)} = -\frac{\tau^t(\widehat{W}_{bj} - \widehat{W}_{cj})\dot{\psi}_j^+}{\tau^t(\widehat{W}_{bj} - \widehat{W}_{cj})\dot{\psi}_j},$$

and a similar equation can be written for the bi-tangential direction

$$e_h^\beta = -\frac{\beta_h^t \text{big}(\dot{\hat{\Theta}}_b^+ - \dot{\hat{\Theta}}_c^+)}{\beta_h^t(\dot{\hat{\Theta}}_b - \dot{\hat{\Theta}}_c)} = -\frac{\beta^t(\widehat{W}_{bj} - \widehat{W}_{cj})\dot{\psi}_j^+}{\beta^t(\widehat{W}_{bj} - \widehat{W}_{cj})\dot{\psi}_j}.$$

16.7 Integrated Model of Impact

The general set of equations for an impact, using all of the aforementioned models, can be written as

$$\begin{bmatrix} M_{ij} & v_h^t(W_{bi} - W_{ci})R_h & \tau_h^t(W_{bi} - W_{ci})R_h & v_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) & \tau_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) & \beta_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) \\ v_h^t(W_{bj} - W_{cj})R_h & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t(W_{bj} - W_{cj})R_h & 0 & 0 & 0 & 0 & 0 \\ v_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) & 0 & 0 & 0 & 0 & 0 \\ \beta_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi}_j^+ \\ e_h^v \\ e_h^\tau \\ \dot{\psi}_h^v \\ \dot{\psi}_h^\tau \\ \dot{\psi}_h^\beta \end{Bmatrix} = \begin{bmatrix} M_{ij} \\ -e_h^v v_h^t(W_{bj} - W_{cj})R_h \\ -e_h^\tau \tau_h^t(W_{bj} - W_{cj})R_h \\ -e_h^v v_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) \\ -e_h^\tau \tau_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) \\ -e_h^\beta \beta_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) \end{bmatrix} \{\dot{\psi}_j\}, \tag{16.12}$$

OR

$$\begin{aligned}
 & \begin{bmatrix} M_{ij} & v_h^t(W_{bi} - W_{ci})R_h & \tau_h^t(W_{bi} - W_{ci})R_h & v_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) & \tau_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) & \beta_h^t(\widehat{W}_{bi} - \widehat{W}_{ci}) \\ v_h^t(W_{bj} - W_{cj})R_h & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu_h & 1 & 0 & 0 & 0 \\ 0 & -\text{sgn}[v_h^t(\widehat{\Theta}_b - \widehat{\Theta}_c)]\mu_h r_p & 0 & 1 & 0 & 0 \\ \tau_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) & 0 & 0 & 0 & 0 & 0 \\ \beta_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \psi_j^+ \\ \epsilon_h^v \\ \epsilon_h^t \\ \tau_h^v \\ \tau_h^t \\ \tau_h^\beta \end{Bmatrix} \\
 & = \begin{bmatrix} M_{ij} \\ -e_h^v v_h^t(W_{bj} - W_{cj})R_h \\ 0 \\ 0 \\ -e_h^t \tau_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) \\ -e_h^\beta \beta_h^t(\widehat{W}_{bj} - \widehat{W}_{cj}) \end{bmatrix} \{\psi_j\}. \tag{16.13}
 \end{aligned}$$

Equations (16.12) use tangential and torsional coefficients of restitution for the tangential and torsional impulses, respectively. Equations (16.13), however, implement tangential and torsional coefficient of friction models. The bilinear combination model can be used in place of either or both if the user prefers.

The impact equations, as presented in this and the previous sections, show only the effects of a single collision of a single contact pair of moving bodies. If more than one collision is to happen in a simulation, they are treated in succession. Only one collision happens at a particular instant. This collision causes changes in the generalized coordinate velocities. The time integration is then continued with the modified velocities, and another collision may then take place. In fact, several collisions may take place before another printing or output display time comes to pass.

16.8 Impact Analysis with SGCs

The impact equations previously presented have been developed with the point of view that all generalized coordinates are free to respond to impacts and that none are driven by power sources able to resist changes in velocity; that is, that all are FGCs. In some cases of dynamic simulation, various of the independent generalized coordinates are SGCs and the number of unknowns in the dynamic impact equations can be reduced to only those of the FGCs.

For example, consider the case of Eqs. (16.12) when the system includes both FGC and SGC independent generalized coordinates. Then, these equations are of the form

$$\begin{bmatrix} M_{FF} & M_{FS} & v_h^t(W_{bF} - W_{cF})R_h & \tau_h^t(W_{bF} - W_{cF})R_h & v_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) & \tau_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) & \beta_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) \\ M_{SF} & M_{SS} & v_h^t(W_{bS} - W_{cS})R_h & \tau_h^t(W_{bS} - W_{cS})R_h & v_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) & \tau_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) & \beta_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) \\ v_h^t(W_{bF} - W_{cF})R_h & v_h^t(W_{bS} - W_{cS})R_h & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t(W_{bF} - W_{cF})R_h & \tau_h^t(W_{bS} - W_{cS})R_h & 0 & 0 & 0 & 0 & 0 \\ v_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) & v_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) & \tau_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) & 0 & 0 & 0 & 0 & 0 \\ \beta_h^t(\widehat{W}_{bF} - \widehat{W}_{cF}) & \beta_h^t(\widehat{W}_{bS} - \widehat{W}_{cS}) & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \psi_F^+ \\ \psi_S^+ \\ \epsilon_h^v \\ \epsilon_h^t \\ \tau_h^v \\ \tau_h^t \\ \tau_h^\beta \end{Bmatrix}$$

$$= \begin{bmatrix} M_{FF} & M_{FS} \\ M_{SF} & M_{SS} \\ -e_h^v v_h^t (W_{bF} - W_{cF}) R_h & -e_h^v v_h^t (W_{bS} - W_{cS}) R_h \\ -e_h^\tau \tau_h^t (W_{bF} - W_{cF}) R_h & -e_h^\tau \tau_h^t (W_{bS} - W_{cS}) R_h \\ -e_h^v v_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -e_h^v v_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \\ -e_h^\tau \tau_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -e_h^\tau \tau_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \\ -e_h^\beta \beta_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -e_h^\beta \beta_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \end{bmatrix} \begin{Bmatrix} \dot{\psi}_F \\ \dot{\psi}_S \end{Bmatrix}$$

where the subscripts F and S refer to the subsets of free and specified generalized coordinates, respectively. If we assume that the velocities of the SGCs are not modified by the impact – that is, that $\dot{\psi}_S^+ = \dot{\psi}_S^-$ – as is required for the motions specified to actually be achieved, then these equations reduce to

$$\begin{bmatrix} M_{FF} & v_h^t (W_{bF} - W_{cF}) R_h & \tau_h^t (W_{bF} - W_{cF}) R_h & v_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & \tau_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & \beta_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) \\ v_h^t (W_{bF} - W_{cF}) R_h & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t (W_{bF} - W_{cF}) R_h & 0 & 0 & 0 & 0 & 0 \\ v_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & 0 & 0 & 0 & 0 & 0 \\ \tau_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & 0 & 0 & 0 & 0 & 0 \\ \beta_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi}_F^+ \\ \dot{\psi}_h^v \\ \dot{\psi}_h^\tau \\ \dot{\psi}_h^\beta \end{Bmatrix} = \begin{bmatrix} M_{FF} & 0 \\ -e_h^v v_h^t (W_{bF} - W_{cF}) R_h & -(e_h^v + 1) v_h^t (W_{bS} - W_{cS}) R_h \\ -e_h^\tau \tau_h^t (W_{bF} - W_{cF}) R_h & -(e_h^\tau + 1) \tau_h^t (W_{bS} - W_{cS}) R_h \\ -e_h^v v_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -(e_h^v + 1) v_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \\ -e_h^\tau \tau_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -(e_h^\tau + 1) \tau_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \\ -e_h^\beta \beta_h^t (\widehat{W}_{bF} - \widehat{W}_{cF}) & -(e_h^\beta + 1) \beta_h^t (\widehat{W}_{bS} - \widehat{W}_{cS}) \end{bmatrix} \begin{Bmatrix} \dot{\psi}_F \\ \dot{\psi}_S \end{Bmatrix}. \tag{16.14}$$

REFERENCES

1. R. M. Brach, *Mechanical Impact Dynamics*, John Wiley & Sons, New York, 1991.
2. D. T. Greenwood, *Principles of Dynamics*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988, pp. 157–60.
3. N. Maw, “The Role of Elastic Tangential Compliance in Oblique Impact,” *ASME Transactions*, vol. 103, Jan. 1981, pp. 74–80.
4. W. Wisutmethangoon, “Collision Detection and Dynamic Impact Simulation of Mechanisms,” PhD Dissertation, University of Wisconsin, Madison, WI, 1998.

PROBLEM

16.1 Continuing problem 11.1, let us assume that the weight W that resulted in equilibrium for the scale at the posture where $\psi = -30^\circ$ was originally put in place with a downward impulse as shown in Eq. (16.4) of $\epsilon_{bcd} = 0.10 \text{ l} \cdot \text{bs}$. Find the FGC velocity that results, and that must be attenuated before equilibrium is achieved.

17 Constraint Force Analysis

17.1 Introduction

From the very beginning of this text, the primary purpose has been to provide background for the development of a digital simulation system as a computational aid to a designer in the creation or modification of a complex mechanism or multibody system. Although such simulation can assure that the designed system moves through its desired trajectory with proper timing, this can usually be done by other means. By far, the biggest advantage of simulation comes from helping the designer to predict the forces transmitted between connected bodies so that the component parts, and the joints between them, can be designed to withstand the imposed loads without overloading and resultant failure, and without undue factors of safety or overdesign.

As pointed out in Chapter 9, however, computers gain no guidance about the solutions of problems from scanning free-body diagrams. Therefore, techniques based on the application of Newton's laws are probably not the best approach for simulation. Here again, as in our formulation of the equations of motion, we very much prefer methods based on work and energy.

As an example, suppose that we wish to find the torque T delivered to the crank by the force P of expanding gas pressure on the piston of the slider-crank linkage shown in [Figure 17.1](#).

If we suppose a small displacement of the system from its current posture, then the crank moves by a small angle $\delta\phi_A$ while the piston slides through a small increment $\delta\phi_D$. However, we know that the total work done during this displacement must be zero. Therefore,

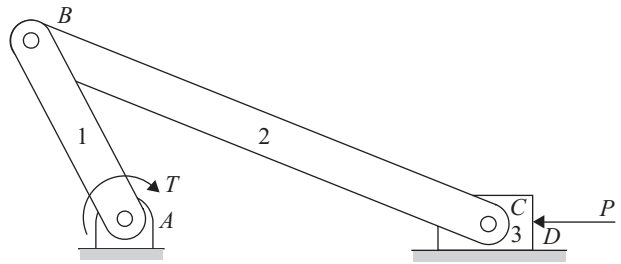
$$\delta W = P\delta\phi_D - T\delta\phi_A = 0,$$

where the negative sign indicates energy leaving the system.

Also, we know that these increments are related by the closure constraint of the kinematic loop. Therefore, because this is a single degree of freedom system, we know that, although the independent variable moves by $\delta\psi_1$, the other joints move by

$$\delta\phi_D = \phi'_{D1}\delta\psi_1 \quad \text{and} \quad \delta\phi_A = \phi'_{A1}\delta\psi_1.$$

Figure 17.1. Slider-crank linkage.



Substituting these into the work equation, we can solve for the torque T :

$$P\phi'_{D1}\delta\psi_1 - T\phi'_{A1}\delta\psi_1 = 0,$$

$$T = (\phi'_{D1}/\phi'_{A1})P.$$

This is the style of solution that we hope to generalize to find the constraint forces of any mechanism or multibody system fitting the conditions of the preceding chapters.

17.2 Fictitious Displacements

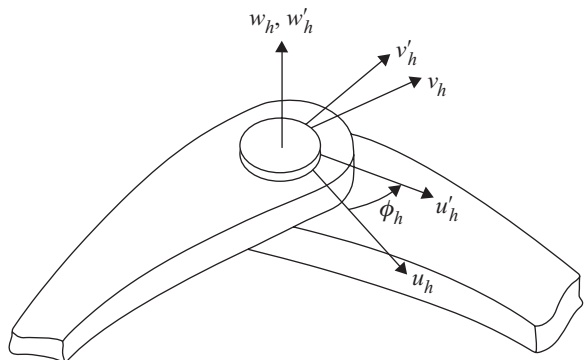
In order to generalize this work and energy approach, let us first consider the displacements that will be needed. In the previous example, we are very fortunate that the torque that we wish to find happens to coincide with the axis of motion of a joint. Therefore, the displacement in this case is possible as a real displacement. However, how will we make a displacement that allows us to find a force or torque component that does not align with the motion of a joint variable [1]?

Let us recall the uvw coordinate system preceding a joint. Each joint, no matter what type, has such a coordinate system. In Chapter 4, we modeled each type of joint so that the transformation from the xyz coordinate system of the body preceding the joint, to the uvw coordinate system preceding the joint, then to the $u'v'w'$ coordinate system following the joint, and then to the xyz coordinate system of the body following the joint, is represented by Eq. (5.3) that reads

$$r_{h-} = S_{h-,h}\Phi_h S_{h+,h}^{-1}r_{h+}, \quad h = 1, 2, \dots, n.$$

Suppose, as an example, that we wish to find the component in the u_h direction of the force transmitted through the pin joint shown in Figure 17.2. Then we

Figure 17.2. Joint axes associated with a typical joint.



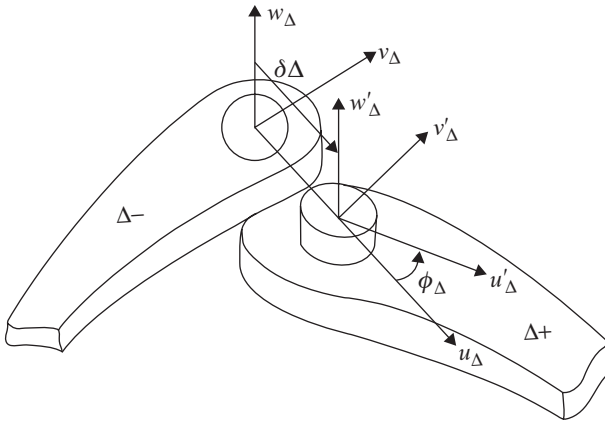


Figure 17.3. Fictitious displacement $\delta\Delta$ along the u_Δ axis direction of joint Δ .

choose to imagine a *fictitious* displacement $\delta\Delta$ in that chosen direction as shown in [Figure 17.3](#). It is true that this displacement cannot happen as a physical possibility, but that is why it is called a fictitious displacement. Let us note carefully, however, that, even though this one constraint (the one for which we seek the force) is violated by the fictitious displacement, all other constraints, such as the (other) loop-closure conditions, continue without violation.

Now that we see in the figure the fictitious displacement that we wish to use, we must consider how this will be modeled mathematically. After some consideration, we see that the previous equation must be modified to the form

$$r_{\Delta-} = S_{\Delta-, \Delta} (I + Q_\Delta \delta\Delta) \Phi_\Delta S_{\Delta+, \Delta}^{-1} r_{\Delta+}, \tag{17.1}$$

where the symbol Δ is used in several ways. First, Δ refers to the label of the joint at which the displacement is made, and where the force is sought. Second, $\Delta-$ and $\Delta+$ refer to the labels for the bodies before and after joint Δ . Third, Δ implies the particular component of the force or torque sought and, therefore, the axis along or about which the displacement is made. Fourth, the symbol $\delta\Delta$ after the Q_Δ matrix denotes the magnitude of the fictitious displacement.

For a displacement in the u_Δ direction, as is our first case, we choose the Q_Δ matrix to read

$$Q_\Delta^u = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for } \delta\Delta \text{ along } u_\Delta. \tag{17.2}$$

If, on the other hand, we wish to find the component of force in the v_Δ direction, then Eq. (17.1) still holds, but we set the Q_Δ matrix to

$$Q_\Delta^v = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for } \delta\Delta \text{ along } v_\Delta, \tag{17.3}$$

and to find the component of force in the w_Δ direction, we set the Q_Δ matrix to

$$Q_\Delta^w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for } \delta\Delta \text{ along } w_\Delta. \quad (17.4)$$

If we seek the component of torque exerted through this joint about the u_Δ axis, then we need a fictitious twist of magnitude $\delta\Delta$ about the u_Δ axis. We can still represent such a fictitious displacement using Eq. (17.1) by setting the Q_Δ matrix to

$$Q_\Delta^{\theta_u} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for a twist of } \delta\Delta \text{ about } u_\Delta. \quad (17.5)$$

For the component of torque about the v_Δ axis, the fictitious twist is defined by setting the Q_Δ matrix to

$$Q_\Delta^{\theta_v} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for a twist of } \delta\Delta \text{ about } v_\Delta, \quad (17.6)$$

and, finally, for the component of torque about the w_Δ axis, we define the fictitious twist by setting the Q_Δ matrix to

$$Q_\Delta^{\theta_w} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for a twist of } \delta\Delta \text{ about } w_\Delta. \quad (17.7)$$

This covers all possible cases, all six components of force and torque that can be transmitted through the chosen joint Δ . Once these forces and torques are found, we can then switch to a different joint and use these same six fictitious displacements again, but with a different choice of joint Δ .

17.3 Fictitious Derivatives

Our next task is to find how other parts of our system move under the action of the fictitious displacement chosen. To determine this, we take the derivative of the loop-closure equations with respect to the fictitious displacement. Let us first define the symbol

$$\phi'_{h,\Delta} = \frac{\partial \phi_h}{\partial \Delta}, \quad h = 1, 2, \dots, n. \quad (17.8)$$

Of course, we must recognize that, while traversing a certain one of the kinematic loops, we may or may not come across the fictitious displacement itself, depending on whether joint Δ is included in the particular loop being traced. That is, the loop-closure conditions, including the fictitious displacement, are of the form

$$T_{0\Delta-} S_{\Delta-, \Delta} [I + L(i, \Delta) Q_\Delta \delta\Delta] \Phi_\Delta S_{\Delta+, \Delta} T_{\Delta+, 0} = I, \quad i = 1, 2, \dots, NL.$$

Now, if we define

$$D_{\Delta} = (T_{0,\Delta} S_{\Delta-\Delta}) Q_{\Delta} (T_{0,\Delta} S_{\Delta-\Delta})^{-1}, \quad (17.9)$$

then the loop-closure conditions above become

$$[I + L(i, \Delta) D_{\Delta} \delta \Delta] T_{00} = I, \quad i = 1, 2, \dots, NL, \quad (17.10)$$

which we see become equal to the original loop-closure conditions when $\delta \Delta = 0$.

If we differentiate the loop-closure equations starting from Eqs. (17.10) with respect to the fictitious displacement $\delta \Delta$, then pass to the limit for which $\delta \Delta = 0$ and $T_{00} = I$, we can write

$$L(i, \Delta) D_{\Delta} + \sum_{h=1}^n L(i, h) D_h \phi'_{h,\Delta} = 0, \quad i = 1, 2, \dots, NL.$$

Putting this result into equivalent screw-coordinate form, it can be rearranged to read

$$\sum_{h=1}^n L(i, h) \hat{D}_h \phi'_{h,\Delta} = -L(i, \Delta) \hat{D}_{\Delta} = \hat{C}'_{i,\Delta}, \quad i = 1, 2, \dots, NL, \quad (17.11)$$

where we define

$$\hat{C}'_{i,\Delta} = -L(i, \Delta) \begin{bmatrix} D_{\Delta}(1, 4) \\ D_{\Delta}(2, 4) \\ D_{\Delta}(3, 4) \\ D_{\Delta}(3, 2) \\ D_{\Delta}(1, 3) \\ D_{\Delta}(2, 1) \end{bmatrix}, \quad i = 1, 2, \dots, NL, \quad (17.12)$$

with an additional constraint equation of the form

$$2\phi_h^1 \phi'_{h,\Delta} + 2\phi_h^2 \phi'^2_{h,\Delta} + 2\phi_h^3 \phi'^3_{h,\Delta} + 2\phi_h^4 \phi'^4_{h,\Delta} = 0, \quad (17.13)$$

for each spheric joint in the system, and another

$$2\phi_h^4 \phi'^4_{h,\Delta} + 2\phi_h^5 \phi'^5_{h,\Delta} + 2\phi_h^6 \phi'^6_{h,\Delta} + 2\phi_h^7 \phi'^7_{h,\Delta} = 0 \quad (17.14)$$

for each open joint in the system; a total of NC constraint equations for joints that are modeled with Euler-Rodrigues parameters.

Therefore, we recognize that Eqs. (17.11), (17.13), and (17.14) finally reduce to the form

$$\mathcal{J} \phi'_{\Delta} = \hat{C}'_{\Delta}, \quad (17.15)$$

where \mathcal{J} is the same $[(6NL + NC) \times n]$ Jacobian matrix that resulted from the numeric iteration process of section 6.5 and, for a given choice of Δ , ϕ'_{Δ} is an $(n \times 1)$ vector of fictitious derivatives that are yet to be determined. The $[(6NL + NC) \times 1]$ column \hat{C}'_{Δ} is composed of the NL vectors from the right-hand sides of Eq. (17.11) for each loop, augmented by the NC right-hand sides of zeroes for Eq. (17.13) for each spheric joint and Eq. (17.14) for each open joint.

Reviewing the numeric iteration process of section 6.5, we recall that the pivoting scheme used in the modified Gauss-Jordan process reordered the joint variables so that all dependent joint variables are numbered first, followed next by the $N\mathcal{F}$ joint variables of the FGCs and finally by the $N\mathcal{S}$ joint variables of the SGCs. Recognizing that both the FGC and the SGC joint variables are accepted as generalized coordinates ψ , we see that, *by definition*, they are independent of each other and of Δ . Therefore, for a particular choice of Δ , the ϕ'_Δ matrix must be of the form

$$\phi'_\Delta = \begin{bmatrix} \phi'_{1,\Delta} \\ \phi'_{2,\Delta} \\ \vdots \\ \phi'_{n-f,\Delta} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \phi'_{dep,\Delta} \\ 0 \end{bmatrix}. \quad (17.16)$$

However, remembering Eq. (6.43), we see that this is the same subdivision that resulted from the Gauss-Jordan elimination process. Reverting to the notation of Eq. (6.47), we have

$$\left[\begin{array}{c|cc} J_{11} & J_{12} & J_{13} \\ \hline J_{21} & J_{22} & J_{23} \end{array} \right] \begin{bmatrix} \phi'_{dep,\Delta} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (\hat{C}'_\Delta)_1 \\ -(\hat{C}'_\Delta)_2 \end{bmatrix},$$

and, after the Gauss-Jordan process is complete, these equations are reduced to the form of Eq. (6.48)

$$\left[\begin{array}{c|cc} I & J_{11}^{-1}J_{12} & J_{11}^{-1}J_{13} \\ \hline 0 & 0 & 0 \end{array} \right] \begin{bmatrix} \phi'_{dep,\Delta} \\ 0 \\ 0 \end{bmatrix} = \left[\begin{array}{c|c} J_{11}^{-1} & 0 \\ \hline -J_{21}J_{11}^{-1} & I \end{array} \right] \begin{bmatrix} (\hat{C}'_\Delta)_1 \\ -(\hat{C}'_\Delta)_2 \end{bmatrix},$$

which has for a solution

$$\phi'_{dep,\Delta} = J_{11}^{-1}(\hat{C}'_\Delta)_1, \quad (17.17)$$

with the additional condition that

$$(\hat{C}'_\Delta)_2 - J_{21}J_{11}^{-1}(\hat{C}'_\Delta)_1 = 0. \quad (17.18)$$

Finally, in Eq. (17.17), we have the solution we have sought for the fictitious derivatives with respect to our chosen fictitious displacement, and we find that they are extremely easy to compute. All that is required, after choosing the desired fictitious displacement, is to form Q_Δ and, from it, the D_Δ operator of Eq. (17.9). From this we form \hat{C}'_Δ according to Eq. (17.12) and, finally, because the Gauss-Jordan process of Chapter 6 has been completed, Eq. (17.17) requires only a matrix multiplication.

However, it is critically important that we not forget to verify Eq. (17.18). Unlike previous chapters, where this additional condition was satisfied automatically, that

is not always true with these fictitious derivatives. On the contrary, when Eq. (17.18) is satisfied, it shows that the result of Eq. (17.17) is valid. However, when Eq. (17.18) is not satisfied, then the result of Eq. (17.17) is *not* valid. How can this happen? This happens when it is not possible to make the chosen fictitious displacement without violating another constraint of the system geometry, and this is what happens when we try to evaluate a force component in an *indeterminate* direction. The failure of the condition of Eq. (17.18) is an immediate signal that the fictitious displacement chosen corresponds to a *statically indeterminate component of force or torque*. No further computation is required and none will be useful because statically indeterminate forces cannot be determined from a rigid body model. Note that the precision of this test should not be more stringent than that used in testing loop-closure, in the Gauss-Jordan method of Chapter 6.¹

It should be noted that if the force computations had been formulated according to Newton's laws, we would have formulated a large set of simultaneous equations for all unknown force components. Failure to solve for one or more because of static indeterminacy would mean failure to solve for *any*. In such problems, we would find no force results because some are indeterminate. With the work and energy approach taken here, each component is found independently and only unsolvable component(s) fail to yield a result. This is another key advantage of our transformation matrix approach.

17.4 Lagrange Equation for Constraint Force

Now that we have found the fictitious derivatives corresponding to a chosen fictitious displacement, how will we proceed from these to find a constraint force? Well, if we think carefully, we see that what we have done is to imagine our real system, which, in physical reality, has f degrees of freedom, and imagined it to have one additional degree of freedom, namely our fictitious displacement Δ . However, if we accept this point of view, then the Lagrange equation of motion for the additional degree of freedom is

$$\frac{d}{dt} \left(\frac{\partial H}{\partial \dot{\Delta}} \right) - \frac{\partial H}{\partial \Delta} + \frac{\partial V}{\partial \Delta} = F_{\Delta} \quad (17.19)$$

where F_{Δ} is the component of force we seek. Of course, in order to correspond to the physical system, we must evaluate this equation under the conditions that $\Delta = \dot{\Delta} = \ddot{\Delta} = 0$; that is, under the conditions that the fictitious displacement is not moving.

Now, in order to carefully derive the terms of Eq. (17.19), it is necessary to formulate kinetic and potential energy formulae that are functions of all f degrees of freedom and of Δ and $\dot{\Delta}$, and to take the derivatives specified, carefully evaluating each term as indicated. The authors have done this in full detail. The extension of these methods to include dynamic forces was first published in [2]. However, rather than to show each step here, it should not surprise the reader to see that the final result reads exactly parallel to the equations of motion of Eq. (10.15).

¹ The IMP software tests that this condition is satisfied within a default tolerance of 0.0001.

This is

$$\begin{aligned}
 F_{\Delta} = & \sum_{a=1}^{\ell} \text{trace} \left[W_{a\Delta} T_{0a} J_a T_{0a}^t \left(\sum_{j=1}^f W_{aj} \ddot{\psi}_j \right)^t \right] \\
 & + \sum_{b=1}^{\ell} \text{trace} \left\{ W_{b\Delta} T_{0b} J_b T_{0b}^t \left[\sum_{j=1}^f \sum_{k=1}^f (A_{bjk} + W_{bj} W_{bk}) \dot{\psi}_k \dot{\psi}_j \right]^t \right\} \\
 & + \sum_{j=1}^f \sum_{h=1}^n (\phi'_{h\Delta} c_h \phi'_{hj}) \dot{\psi}_j + \sum_{j=1}^f \sum_{bc} R'_{bc\Delta} u_{bc} c_{bc} u'_{bc} R'_{bcj} \dot{\psi}_j \\
 & + \sum_{h=1}^n \phi'_{h\Delta} k_h (\phi_h - \phi_{h0}) + \sum_{bc} R'_{bc\Delta} u_{bc} k_{bc} (\ell_{bc} - \ell_{bc0}) - \sum_{b=1}^{\ell} g^t W_{b\Delta} T_{0b} J_b T_{0b}^t r_0 \\
 & - \sum_{h=1}^n \phi'_{h\Delta} f_h(\phi, t) - \sum_{bcd} R'_{b\Delta} u_{cd} f_{bcd}(\phi, t) - \sum_{cde} \widehat{W}_{c\Delta} u_{de} \tau_{cde}(\phi, t), \quad (17.20)
 \end{aligned}$$

where the following additional notation using the fictitious derivatives has been defined

$$W_{b,\Delta} = \sum_{h=1}^n P(b, h) D_h \phi'_{h,\Delta}, \quad b = 1, 2, \dots, \ell, \quad (17.21)$$

$$R'_{b,\Delta} = \frac{\partial R_b}{\partial \Delta} = W_{b,\Delta} T_{0b} r_b = W_{b,\Delta} R_b, \quad b = 1, 2, \dots, \ell, \quad (17.22)$$

$$R'_{bc\Delta} = R'_{b\Delta} - R'_{c\Delta} = W_{b\Delta} R_b - W_{c\Delta} R_c, \quad b, c = 1, 2, \dots, \ell. \quad (17.23)$$

In order to properly use, Eq. (17.20), it is necessary to understand the sign conventions chosen in the derivation. The force F_{Δ} is the single component of force or torque acting from body $\Delta-$ onto body $\Delta+$ along or about one of the u_{Δ} , v_{Δ} , or w_{Δ} axes consistent with the choice of Q_{Δ} . The component F_{Δ} is positive when it acts onto body $\Delta+$ in the positive direction of the corresponding displacement axis. The sign conventions for $f_h(\phi, t)$, $f_{bcd}(\phi, t)$, and $\tau_{bcd}(\phi, t)$ are those explained in sections 9.10, 9.11, and 9.12, respectively.

EXAMPLE 17.1 To illustrate the previous process, let us continue the analysis of the Cardan/Hooke universal joint started in example 5.1 and continued in examples 7.1 and 8.1. Suppose we now wish to find the six components of force and torque in the input joint D for a given static load torque P_A at output joint A . Because we have no mass distribution data, we seek only the static forces. From these previous examples we have already found

$$T_{04} S_{4D} = \begin{bmatrix} 0 & -\sin \beta & \cos \beta & d_2 \cos \beta \\ 1 & 0 & 0 & h \\ 0 & \cos \beta & \sin \beta & d_2 \sin \beta \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathcal{J} = \left[\begin{array}{ccc|c} 0 & \frac{-h \cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -h \cos \beta \sin \psi & h \sin \beta \\ 0 & 0 & 0 & 0 \\ -h & 0 & -h \sin \beta \sin \psi & -h \cos \beta \\ \hline 1 & 0 & \sin \beta \sin \psi & \cos \beta \\ 0 & \frac{-\cos \beta \sin \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & \cos \psi & 0 \\ 0 & \frac{-\cos \psi}{\sqrt{1 - \sin^2 \beta \sin^2 \psi}} & -\cos \beta \sin \psi & \sin \beta \end{array} \right],$$

and we have already noted that the first three rows of \mathcal{J} are either trivial or $\pm h$ multiples of other rows. That is, these three rows form $[\mathcal{J}_{21} \ \mathcal{J}_{22}]$; the final three rows form $[\mathcal{J}_{11} \ \mathcal{J}_{12}]$.

First, using Eq. (4.13), we find

$$(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & 1 & 0 & -h \\ -\sin \beta & 0 & \cos \beta & 0 \\ \cos \beta & 0 & \sin \beta & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and, from Eq. (17.10), using Eqs. (17.2) through (17.7), the six D_{Δ} matrices for joint D are

$$D_D^u = (T_{04}S_{4D})Q_{\Delta}^u(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D^v = (T_{04}S_{4D})Q_{\Delta}^v(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & 0 & 0 & -\sin \beta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D^w = (T_{04}S_{4D})Q_{\Delta}^w(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & 0 & 0 & \cos \beta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D^{\theta_u} = (T_{04}S_{4D})Q_{\Delta}^{\theta_u}(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & 0 & 1 & -d_2 \sin \beta \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & d_2 \cos \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D^{\theta_v} = (T_{04}S_{4D})Q_{\Delta}^{\theta_v}(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & -\cos \beta & 0 & h \cos \beta \\ \cos \beta & 0 & \sin \beta & -d_2 \\ 0 & -\sin \beta & 0 & h \sin \beta \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D_D^{\theta_w} = (T_{04}S_{4D})Q_{\Delta}^{\theta_w}(T_{04}S_{4D})^{-1} = \begin{bmatrix} 0 & -\sin\beta & 0 & h\sin\beta \\ \sin\beta & 0 & -\cos\beta & 0 \\ 0 & \cos\beta & 0 & -h\cos\beta \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Now, extracting the negative Ball vectors from each of these, according to Eq. (17.12),

$$C'_{\Delta} = \begin{bmatrix} 0 & \sin\beta & -\cos\beta & d_2\sin\beta & -h\cos\beta & -h\sin\beta \\ -1 & 0 & 0 & 0 & d_2 & 0 \\ 0 & -\cos\beta & -\sin\beta & -d_2\cos\beta & -h\sin\beta & h\cos\beta \\ 0 & 0 & 0 & 0 & \sin\beta & -\cos\beta \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\cos\beta & -\sin\beta \end{bmatrix}.$$

However, as we have noted before, for compatibility with the Jacobian matrix, the first row of each of these should be an h multiple of the sixth row, the second row should be zero, and the third row should be the negative h multiple of the fourth row. However, these conditions are only met by the sixth column; that is, by the $\delta\theta_w$ fictitious displacement; all other fictitious displacements give contradictory equations. This implies that only the $\delta\theta_w$ fictitious displacement can be made without violating another of the constraints and, consequently, only the torque about the w_D axis can be found; the other five components of force and torque in joint D are each statically indeterminate.

The equations for the fictitious derivatives for this one valid fictitious displacement are

$$\begin{bmatrix} 1 & 0 & \sin\beta\sin\psi & \cos\beta \\ 0 & \frac{-\cos\beta\sin\psi}{\sqrt{1-\sin^2\beta\sin^2\psi}} & \cos\psi & 0 \\ 0 & \frac{-\cos\psi}{\sqrt{1-\sin^2\beta\sin^2\psi}} & -\cos\beta\sin\psi & \sin\beta \end{bmatrix} \begin{bmatrix} \phi_A^{\theta_w} \\ \phi_B^{\theta_w} \\ \phi_C^{\theta_w} \\ \phi_D^{\theta_w} \end{bmatrix} = \begin{bmatrix} -\cos\beta \\ 0 \\ -\sin\beta \end{bmatrix},$$

and this set of equations has the solution

$$\begin{bmatrix} \phi_{A\Delta}^{\theta_w} \\ \phi_{B\Delta}^{\theta_w} \\ \phi_{C\Delta}^{\theta_w} \\ \phi_{D\Delta}^{\theta_w} \end{bmatrix} = \begin{bmatrix} \frac{-\cos\beta}{1-\sin^2\beta\sin^2\psi} \\ \frac{\sin\beta\cos\psi}{\sqrt{1-\sin^2\beta\sin^2\psi}} \\ \frac{\sin\beta\cos\beta\sin\psi}{1-\sin^2\beta\sin^2\psi} \\ 0 \end{bmatrix}.$$

It is no surprise that these particular fictitious derivatives are identical to the first geometric derivatives found in example 7.1 because this fictitious displacement is aligned directly in series with the real generalized coordinate of joint

D. Thus, this particular displacement duplicates the ideas shown in section 17.1; however, that section gave no indication that other components are statically indeterminate. This is now proven.

Finally, with the given conditions, because positive P_A represents a torque that is counterclockwise about the positive w_A axis, Eq. (17.20) reduces to

$$F_D^{\theta_w} = -\phi_{A\Delta}^{\theta_w} P_A,$$

$$F_D^{\theta_w} = \frac{\cos \beta}{1 - \sin^2 \beta \sin^2 \psi} P_A. \quad \underline{Ans.}$$

This says that, for a counterclockwise load torque P_A , the driving torque is also positive and therefore, counterclockwise. Isn't this contradictory? No. This driving torque is counterclockwise about the w_D axis. Careful study of Figure 5.4 shows that, for small angles of β , this is directed almost opposite to the w_A axis.

REFERENCES

1. J. Denavit, R. S. Hartenberg, R. Razi, and J. J. Uicker, Jr., "Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages," *Journal of Applied Mechanics, ASME Transactions*, 1965, vol. 87, pp. 903–10.
2. J. J. Uicker, Jr., "Dynamic Force Analysis of Spatial Linkages," *Journal of Applied Mechanics, ASME Transactions*, vol. 89, June 1967.

PROBLEMS

17.1 Continue problem 11.1 and find all components of force and torque from link 4 onto link 3 through joint *D* of the spring scale when in equilibrium under a constant load *W* in the posture of $\psi = -30^\circ$.

17.2 Continue problem 10.4 and find all components of force and torque from link 4 onto link 1 through joint *A* of the Oldham shaft coupling under a constant output shaft load M_D . Plot the input shaft torque $F_A^{\theta_w}$ versus ψ for one revolution.

Index

- Absolute
 - coordinate system, 85
 - position, 118–119
- Acceleration, 197–223
 - body, 197, 220–222
 - definition, 197
 - generalized, 198
 - joint variable, 197–198, 214–216
 - point, 197, 223
 - system, 198
- Action, 245
- Adaptive time-step control, 291–292
- Affine geometry, 44
- American Gear Manufacturer’s Association (AGMA), 135n
- Ampere, Andre Marie, 3, 20
- Amplitude of twist, 238
- Analysis, 1, 2, 3
 - acceleration, 197–198, 207–223
 - constraint force, 310–320
 - kinematic architecture, 21–36
 - position and posture (analytic), 119–144
 - position and posture (numeric), 164–181
 - structural, 23
 - topological, 21–36
 - velocity, 183–196
- Angeles, Jorge, 176, 182
- Angular orientation in 3-D, 47–51
- Angular momentum, 246
- Arc, 25
- Architecture, kinematic, 21–36
- Aristotle, 244, 253
- Assembly, 24, 27
 - connected, 27
 - oriented, 25
- Auxilliary coordinate system, 81–82
- Attitude, 47

- Ball, Sir Robert Stawell, 161, 182
- Ball vectors, 159–163, 240–241, 306
- Barycentric coordinates, 43
- Base, 12, 85

- Bernoulli, Daniel, 271, 279
- Bernoulli, Johannes, 244, 271
- Betti, Enrico, 30
- Bevel gear joint, straight-tooth, 102–104, 158, 204
- Bilinear impulse model, 306
- Biomechanical model, 4
- Bode plots, 277
- Body
 - acceleration, 197, 220–222
 - degree of, 26
 - coordinate system, 57, 80–81
 - first geometric derivative, 191–193
 - mechanical, 3, 10–11, 24
 - posture, 81
 - second geometric derivatives, 216–220
 - velocity, 194–195
- Bollinger, John G., 279
- Boundary representation (BRep), 294
- Brach, R. M., 309
- Bromwich, Thomas John l’Anson, 280
- Bromwich integral, 280
- Bryan, George Hartley, 68

- Calculus of variations, 245
- Cameron, S., 298, 299
- Cardano, Gerolamo, 68, 78, 122
- Cardan angles, 68, 94, 97
- Cardan/Hooke universal shaft coupling, 8, 9, 122–126, 188–189, 191–193, 195, 211–214, 216, 217–220, 222, 317–320
- Cartesian coordinates of a point, 42, 51, 55
- Cauchy, Baron Augustin Louis, 282, 292
- Cauchy’s residue theorem, 282–284
- Ceccarelli, Marco, 60, 78
- Cederbaum, I., 41
- Center of mass, 228
- Chain, mechanical, 10–11
- Characteristic equation, 273
- Characteristic values, 273
- Chasles, Michel, 60, 78
- Chasles’s theorem, 60

- Cheng, H., 79
 Chord, 30
 Cholesky's method, 171
 Churchill, Ruel V., 292
 Closed-form solution for joint variable positions, 121–144
 Coefficient of restitution, 304–307
 Moment, 307
 normal, 304
 tangential, 305
 torsional, 306
 Collision detection, 294–299
 Complete pivoting, 171
 Computation speed, 181
 Connectivity of a graph, 27, 30
 Constraint, 3, 14, 20
 Constraint force analysis, 310–320
 Constructive solid geometry (CSG), 294
 Convergence, 177–181
 Contact, location and type, 303
 Convergence, 177–181
 Coolidge, Julian Lowell, 78
 Coordinates, 55
 generalized, 111
 Coordinate transformation, 51–54
 Coordinate systems
 absolute, 85, 118
 auxilliary, 81–82, 85
 body, 80–81, 85
 specifying data for, 82–84, 86
 joint, 81–82, 85
 measurement, 82–84, 86
 Craig, John J., 182
 Crout's method, 171
 Cut-set, 30
 Cyclomatic number of a graph, 30
 Cylindric coordinates of a point, 43, 55
 Cylindric joint, 15, 17, 18, 92–93, 155, 200
- D'Alembert, Jean le Rond, 253
 D'Alembert inertia forces, 246–249
 Dampers, 3, 232–233, 234–235
 Damping matrix, system, 258
 Defective system, 285
 Degree of freedom (see mobility)
 Denavit-Hartenberg transformation, 116–118
 Denavit-Hartenberg parameters, 116–118, 179
 Denavit, Jacques, 21, 147, 182, 196, 224, 320
 Derivative operator matrices, 153–158, 192, 194, 216
 derivatives of, 198–205, 205–207
 Derivatives, fictitious, 313–316
 Descartes, René, 43, 78
 Design, 1
 Differential kinematics, 148–181
 Direct kinematics, 130–133
 Direction cosines, 48–51
 Disk cam and follower, 126–130
 Displacement, 8, 56–57, 183
 body, 56–60, 62
 differential, 159–163
 fictitious, 311–315
 line, 74
 point, 56
 Duality, geometric, 44–45
 Duffie, Neil A., 279
 Dynamics, 2, 225
 Dynamic equations of motion, 244–252
 Dynamic equilibrium, 254
 Dynamic modeling, 225–241
- Edge, 25
 Edge-edge contact, 296–297, 303
 Eigenvalue problem, 273
 Eigenvalues, 273
 Eigenvectors, 274
 EISPACK, 273
 Elastic impact, 304
 Equations of motion, 244–252
 uncoupled, 276
 Equilibrium posture analysis, 262–269
 Equivalence, kinematic, 19–20
 Euler, Leonhard, 3, 21, 49, 78, 225, 243, 271, 279
 Euler angles, 50–51, 94, 97
 Euler-Rodrigues parameters, 69–74, 94, 97, 166, 201, 202, 314
 Euler's theorem, 61, 66
- False solutions to loop-closure equations, 168–169
 Ferguson, James, 134, 147
 Ferguson's paradox, 134–140
 Fetterling, W. T., 270
 Feuerbach, Karl Wilhelm, 43, 78
 Feuerbach coordinates of a point, 43
 Fictitious displacements, 311–315
 Fictitious derivatives, 313–316
 Finite element analysis (FEA), 94, 290
 First Betti number of a graph, 30
 First geometric derivatives
 body positions, 191–193
 joint positions, 186–189
 point positions, 195
 Flat joint, 15, 17, 18, 94–95, 156, 201
 Flannery, B. P., 270
 Flexibility, 10
 Force analysis, 310–320
 Force applied at a point, 236–241
 Force or torque applied with joint variable, 235–236
 Forward kinematics, 130–133
 Fourier, Baron, Jean Baptiste Joseph, 271, 279
 Fourier-Mellin integral, 280
 Frame, 12
 Free generalized coordinates (FGC), 174, 260, 266, 271
 Frequency response of machinery, 271–278
 Freudenstein, Ferdinand, 140, 147
 Friberg, O., 70, 78
 Frictional tangential impulse model, 305

- Galileo, Galilei, 244, 253
 Gauss, Carl Friedrich, 171
 Gauss-Jordan method, 171–173, 269, 315
 Gear, William C., 293
 Generalized applied forces, 250
 Generalized applied forces, linearized system, 259
 Generalized coordinates, 111, 173–174, 183
 free (FGC), 174
 identification of, 173–174
 number, 173–174
 specified (SGC), 171–174
 Generalized eigenvalue problem, 273
 Generalized forces, 245
 Generalized impulse, 302
 Generalized momentum, 245–246
 Generalized position vector, 111
 Generalized restoring forces, 249–250
 Generalized static forces, 249
 Geometric derivatives
 first, 186–189, 191–193, 195
 second, 207–214, 216–220
 Geometric duality, 44–45
 Gergonne, Joseph Diaz, 45, 78
 Gilbert, E. G., 299
 Gimbal lock singularity, 69, 94, 97
 Global coordinate system, 85
 Globular joint, 15, 17, 18, 93–94, 156, 200
 Goldstein, Herbert, 50, 78
 Gough/Stewart platform, 5, 37
 Graph theory, 24
 Graph, mechanical, 24
 Gravitation effects, 230–232
 potential energy of, 231–232
 Greenwood, D. T., 309
 Ground, 12
 Gupta, K. C., 79
- Hamilton, Sir William Rowan, 74, 78, 244, 253
 Hamilton's principle, 245
 Harary, Frank, 41
 Hartenberg, Richard S., 3, 21, 147, 182, 196, 224, 320
 Helical gears, 98
 Helical joint, 15–17, 88–90, 155, 199
 Helical motion, 60
 Helix angle, 98, 100
 Helix pitch, 16, 60–61
 Herringbone gear teeth, 98, 100
 Hibbeler, Russell Charles, 243
 High-friction disks, 102
 Hildebrand, F. B., 279
 Hollerbach, J. M., 253
 Homogeneous coordinates, 42–44, 53
 Homogeneous first-order equations of motion, 271–274
 Homologous points, 60, 61
 Homotopy methods, 140
 Hong, Dennis, 4
 Hooke, Robert, 122
 Howell, Larry L., 21
- Impact analysis, 295, 300–309
 Impulsive loads, 300–303
 Incidence, 24
 Incidence matrix, 25
 Incidence table, 24
 Indeterminate components of force, 316
 Inertia matrix, 227–229
 Inman, Daniel J., 243
 Instantaneous screw axis, 161
 Integrated Mechanisms Program (IMP), xv, 71n, 104n, 108n, 172n, 211n, 236n, 266n, 267n, 269n, 294n, 316n
 Integrated model of impact, 307–308
 Intensity of wrench, 238
 Inverse kinematics, 133–134
 Inverse Laplace transform, 280–281
 Inverse transformation matrix, 121
 Inverse Denavit-Hartenberg transformation, 121
 Inversion, kinematic, 11, 12
 Involute rack-and-pinion joint, 100–102, 157, 204
 Involute tooth profile, 98, 100
- Jacobi, Carl Gustav Jacob, 167, 182
 Jacobian, 167, 182, 208
 James, M. L., 182
 Johnson, D. W., 299
 Joint, 3, 12–14, 24
 accelerations, 197, 214–216
 bevel-gear, straight-tooth, 102–104, 158, 204
 coordinate system, 81–82, 87–108
 cylindric, 15, 17, 18, 92–93, 155, 200
 damping, 232–233
 elements, 12–14
 hollow, 13
 solid, 13
 flat, 15, 17, 18, 94–95, 156, 201
 globular, 15, 17, 18, 93–94, 156, 200
 helical, 15–17, 88–90, 155, 199
 involute rack-and-pinion, 100–102, 157, 204
 line tangent to a planar-curve, 106–108, 128, 158, 205
 open, 19, 96–98, 120, 156, 190, 201, 209, 215
 orientation, 25
 parallel-axis gear, 98–100, 157, 202
 planar, 15, 17, 18, 94–95, 156, 201
 point on a planar-curve, 104–106, 158, 205
 prismatic, 15, 17, 18, 91–92, 155, 199
 rack-and-pinion, involute, 100–102, 157, 204
 revolute, 15, 17, 90–91, 155, 199
 rigid, 19, 96, 156, 201
 screw, 15–17, 88–90, 155, 199
 spheric, 15, 17, 18, 93–94, 156, 190, 200, 209, 215
 stiffness, 232
 straight-tooth bevel-gear, 102–104, 158, 204
 transformation matrix, 87–88, 88–108
 variables, 14, 88, 88–108, 186–189, 207–214, 214–216
 velocities, 189–191
- Jordan block, 285
 Jordan curve, 281

- Jordan, D., 293
 Jordan, Marie Ennemond Camille, 281, 293
 Jordan normal form, 285
 Jordan, Wilhelm, 171
- Kågström, B., 293
 Kaiser, K. S., 182
 Keerthi, S. S., 299
 Kennedy, Alexander Blackie William, 7, 21
 Kinematics, 2, 3, 225
 Kinematic architecture, 21–36
 Kinematic equivalence, 19
 Kinematic inversion, 11–12
 Kinematic loop, 27–32
 Kinematic path, 32–36
 Kinematic tree, 29–30
 Kinetics, 2, 3, 225
 Kinetic energy, 226–227, 245
 Korn, Granino A., 182, 293
 Korn, Theresa M., 182, 293
- Lagrange, Joseph Louis, 244, 253, 271, 279
 Lagrange's equation, 244–245, 268, 300, 316–320
 LAPACK, 273
 Laplace, Marquis Pierre Simon, 271, 279
 Laplace transform, 276–277
 Lead of screw (see pitch)
 Line coordinates, 45–47
 Line displacement, 74
 Line of nodes, 50–51
 Line tangent to a planar-curve joint, 106–108, 128, 158, 205
 Linearization assumptions, 254–255
 Linearization procedure, 255–258
 Linearized equations of motion, 254–261
 Linkage, 18
 Loop, kinematic, 27–32
 closure equation, 111, 119–121
 Oriented loop matrix, 28
- Machine, 7
 Mäntylä, Martti, 299
 Mass matrix, system, 258
 Mass units, 230
 Maupertius, Pierre Louis Moreau de, 244, 245, 253
 Maw, N., 309
 Maxwell, E. A., 78
 McCormick, John, 4
 McCuskey, S. W., 253
 Mechanical graph, 24
 oriented, 25
 Mechanics, 2
 Mechanism, 3, 5
 planar, 7–8
 spatial, 8
 spherical, 8–9
 Mellin, Robert Hjalmar, 280
 Minimal representation of orientation, 49–51
 Mobility, 173–174
 Möbius, August Ferdinand, 43, 78
- Modal coordinates, 274–276
 Modal matrix, 274
 Moment coefficient of restitution, 307
 Moment impulse model, 307
 Momentum, 246
 Morgan, A. P., 140, 147
 Motion, 87, 183
 Mozzi, Giulio, 60, 78
 Multibody system, 3–6
- Nautical angles, 68
 Neutral equilibrium postures, 269
 Newton, Sir Isaac, 177, 182
 Newton-Raphson iteration, 177, 266
 Noble, B., 293
 Normal coefficient of restitution, 304
 Normal modes, 275
 Normal torsional impulse, 306–307
 Nullity of a graph, 30
 Numeric solution of posture equations, 148–181
 Nutation angle, 66, 67
- Olsen, William J., 179–182
 Open joint, 19, 96–98, 120, 156, 186, 190, 201, 209, 215, 314
 Orientation, joint, 25
 Orientation, angular in 3-D, 47–51
 Orrery, 134
 Orthogonality conditions, 49
- Pair, kinematic, 13
 higher, 14–20
 lower, 14–19
 Parallel-axis gear joint, 98–100, 157, 202
 Part, mechanical, 3, 24
 Partial fraction expansion, 282
 Path, kinematic, 28, 32–36
 Oriented path matrix, 32
 Paul, R. P., 147
 Pennock, Gordon R., 21, 147, 243
 Pieper, D., 147
 Pitch angle, 68
 Pitch-circle radii, 98, 100
 Pitch-cone half-angles, 102
 Pitch of helix/screw, 16, 63, 173, 238
 Pitch radius, 306
 Planar joint, 15, 17, 18, 94–95, 156, 201
 Plastic impact, 304
 Plücker coordinates, 45
 Plücker identity, 46
 Plücker, Julius, 45, 78
 Plücker vectors, 45–47, 74
 Poincot, Louis, 238, 243
 Poincot's theorem, 238
 Point
 acceleration, 197, 223
 barycentric coordinates, 43
 Cartesian coordinates, 42
 cylindric coordinates, 43
 Feuerbach coordinates, 43

- first geometric derivative, 195
- homogeneous coordinates, 43–45
- homologous, 60
- position, 55, 183
- second geometric derivative, 220
- spheric coordinates, 43
- velocity, 196
- Point at infinity, 44, 53–54
- Point on a planar-curve joint, 104–106, 158, 205
- Poles of a function, 280
- Pose (see posture), 56
- Position, 55
- Posture, 56
 - assembly, 56
 - coordinate system, 55–60
 - equilibrium, 262–269
 - mechanism, 56, 183
 - reference, 59
 - rigid body, 55–60, 191–193
 - system, 56, 183
- Potential energy, 231–232, 245, 268
- Precession angle, 66, 67
- Press, William H., 270
- Pressure angle, 176
- Principle of least action, 245
- Principal modes, 275
- Prismatic joint, 15, 17, 18, 91–92, 155, 199
- Projective geometry, 44

- Quality index, 176–177, 189, 214
- Quaternion, 74–75

- Rack-and-pinion joint, 100–102, 157, 204
- Raghaven, M., 140, 147
- Raphson, Joseph, 177, 182
- Ravani, Bahram, xvii
- Razi, Rafael, 196, 224, 320
- Reed, Myril B., 41
- Repeated eigenvalues, 284–288
- Residue, 282
- Residue theorem, 282–284
- Restraint, 3, 20
- Reuleaux, Franz, xiii, xv, 7, 14, 15, 21
- Revolute joint, 15, 17, 90–91, 155, 199
- Rigid body, 11
- Rigid joint, 19, 96, 156, 201
- Robot, 5, 130–134
- Robustness, 177–181
- Rodrigues, Benjamin Olinde, 61, 78
- Roll angle, 68
- Rotation axis, 60
- Rotation matrix, 48
- Roth, Bernard, 140, 147
- Ruhe, A., 293

- SCARA robot, 130–134
 - forward kinematics, 130–133
 - inverse kinematics, 133–134
- Scaling length units, 174–175

- Screw
 - axis, 60, 159–163
 - coordinates, 45, 162, 166, 186, 190, 208, 215, 241, 314
 - displacement, 60–61, 65
 - joint, 15–17, 88–90, 155, 199
 - parameters, 61–62, 159
 - pitch, 16, 60, 160, 238
 - Song of, vi
- Second geometric derivatives
 - body postures, 216–220
 - joint variables, 207–214
 - point position, 220
- Secular equation, 273
- Self-loop, 26
- Seshu, S., 41
- Shabana, Ahmed A., 21
- Shape matrix, 86
- Sheth, Pradip N., xvii, xviii, 147, 182
- Shigley, Joseph E., 21, 147, 243
- Simple impact model, 303–304
- Simple poles, 282
- Smith, G. M., 182
- Solid models, 87, 294
- Sommese, A. J., 140, 147
- Song of the Screw, vi
- Specified generalized coordinates (SGC), 174, 260, 266
- Spheric coordinates of a point, 43, 55
- Spheric joint, 15, 17, 18, 93–94, 156, 186, 190, 200, 209, 215, 314
- Spin angle, 66, 67
- Springs, 3, 233–234
- Sridhar, B., 293
- Stable vs. unstable equilibrium, 267–268
- Stanford manipulator, 141–144
- State space, 273
- Statically indeterminate force components, 99, 101, 106, 107, 316
- Statics, 2, 225
- Stereotomy, 3
- Steepest descent methods, 268
- Stiff differential equations, 291
- Stiffness matrix, system, 258–259, 265
- Straight-tooth bevel-gear joint, 102–104, 158, 204
- Structure, 7
- Structural analysis, 21
- Stuelpnagel, J., 78
- Sylvester, James Joseph, 32
- Sylvester's law of nullity, 32
- Synthesis, 1
 - dimensional, 2
 - number, 1
 - type, 1

- Tait, Peter Guthrie, 68
- Tait-Bryan angles, 68
- Tangential coefficient of restitution, 305
- Tangential impulse model, 305–306
- Teukolsky, S. A., 270

- Thelen, Darryl, 4
- Time, 184
- Time increment until contact, 297–299
- Time integration algorithm, 288–291
- Time response, 280–292
- Time-step control, adaptive, 291–292
- Topology, 21–36
- Torsional coefficient of restitution, 306
- Torsional impulse model, 306–307
- Torque applied on a body, 236–241
- Torque applied with a joint variable, 235–236
- Transformation of coordinates, 51–54
- Transformation matrix, 53–54
- Transverse pressure angle, 98, 101
- Tree, kinematic, 29–30
- Tsai, L. W., 140, 147
- Twist, 162, 238, 313

- Uicker, John J., xvii, 21, 147, 182, 196, 224, 230, 243, 320
- Uicker-Kahn formulation of manipulator dynamics, 252
- Unimodular property, 36
- Universal joint, 4, 122–126, 188–189, 191–193, 211–214, 216, 217–220, 222

- Units, systems of, 230
- Varignon, Pierre, 244, 253
- Velocity, 183–196
 - body, 184, 194–195
 - definition, 184–185
 - generalized, 185
 - joint variable, 184–185, 189–191
 - point, 184, 196
 - system, 185
- Vertex, 25
- Vertex-face contact, 295–296, 303
- Virtual displacements, 232–233, 234–235, 240–241
- Virtual twist, 240–241
- Virtual work, 232–233, 240–241
- Viscous damping, 232–233, 234–235

- Wampler, C. W., 140, 147
- Wilkinson, James H., 279
- Wire-frame data, 294
- Wisutmethangoon, Warawut, 300, 309
- Wolford, James C., 182
- World coordinate system, 85
- Wrench, 237, 239

- Yaw angle, 68